AXISYMMETRIC INERTIAL GRAVITY OSCILLATIONS OF A COMPRESSIBLE STRATIFIED FLUID IN A RIGID SPHERE: AN APPROXIMATION TO THE EARTH'S OUTER CORE

SUSAN JANE WEBB, B.Sc.

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur SRC 1970, c. C-30, et ses amendements subséquents

AXISYMMETRIC INERTIAL GRAVITY OSCILLATIONS OF A COMPRESSIBLE STRATIFIED FLUID IN A RIGID SPHERE: AN APPROXIMATION TO THE EARTH'S OUTER CORE

BY

© Susan Jane Webb, B.Sc.

A thesis submitted to the School of Graduate

Studies in partial fulfilment of the

requirements for the degree of

Master of Science

Department of Earth Sciences (Geophysics)

Memorial University of Newfoundland

February 1992

St. John's Newfoundland

ISBN 0-315-73288-1

Canada

Abstract. This thesis investigates several theoretical methods for solving the sub-seismic wave equation (SSWE) in an incompressible or slightly stratified liquid bounded by a rigid fixed spherical boundary as an approximation to the Earth's liquid core. The solution of the SSWE yields eigenvalues for the inertia/gravity wave spectrum. The first method considered uses a variational principle based on a more general functional than has been previously used to evaluate the SSWE. The resulting eigenvalues are compared with previous work done by Aldridge and Toomre (1969). The work is then extended to include a density profile and a shift is noted in the eigenspectrum. A perturbation approach was used to include stably stratified regions characterised by negative values of the stability parameter ß; unfortunately this method only works for such small values of ß as to be uninteresting. Finally a Galerkin approach was adopted that allows for the evaluation of negative ß values. This results in a further change of the inertial eigenspectrum and the appearance of gravity modes.

Acknowledgments

I would like to sincerely thank and acknowledge the following:

Dr. M. G. Rochester for much help, guidance and financial support for this thesis project and most especially for his patience.

Memorial University for awarding me a fellowship.

The physics department at Memorial for awarding me several teaching assistantships.

Gold Fields of South Africa Ltd. for the use of word processing equipment.

Many friends and associates at Memorial, especially Mike Reed, Tony Dearness, Phil Kirby and Jennifer Rendell who helped me learn about computers and programming. Lew Ashwal for never letting me give up, and my parents for encouraging me to try new things.

Acknowledgement is hereby made of the use of the algebraic manipulation program ©MACSYMA.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Symbols

Symbols are listed in the order in which they appear with similar symbols being grouped together. All symbols are in italics with vectors and dyadics in boldface.

Symbol	Significance	Equation Where Initially Used
g_o	Gravity	1.1
V_o, V_1	Gravitational Potential	1.1, 1.5
ρ_o, ρ_1, ρ_o'	Density	1.2, 1.5, 1.22
v	Velocity	1.2
t	Time	1.2
$\mathbf{\Omega}$	Angular Velocity	1.3
P_1, P, P	Pressure, Linear Momentum, Summation Limit	1.3, 6 2, 2.25
α	Compressional Wave Speed	1.4
G, G_{pq}	Gravitational Constant, Matrix	1.5, 2.21
u	Displacement	1.6
ω	Frequency	1.7
r	Position, Spherical Coordinate, Summation Index	1.7, 2.25, 2.34

χ, χ_o	Scaler Field Variable	1.2, 4.6
β	Stability or buoyancy parameter	1.15
N	Brunt-Väisälä frequency, Summation Limit	1.16, 2.25
W_o	Gravitational Potential	1.23
Γ, Γ_p, Γ_2	Dyadics	1.26, 1.27, 2.12
$\mathbf{1}$	Unit Dyadic	1.27
e_3, n	Unit Vectors	1.27, 1.32
C, C_o	Vector Definitions	1.28, 4.7
B, $\boldsymbol{B}$	Scaler, Matrix	1.29, 3.11
σ, σ_2	Dimensionless Frequency	1.30, 4.2
F	Functional, Definition	2.1, 5.9
$\mathcal{L}$	Linear Operator	2.1
Ψ	Arbitrary Function	2.1
dv, ds	Volume, Surface Integration	2.1
ϕ	Azimuthal Coordinate, Trial Function	2.16, 2.19
m	Azimuthal Number, Summation Index	2.16, 3.1
R	Cylindrical Radial Coordinate	2.17
c_p, c_{ij}, c	Constants, Function	2.19, 2.28, 2.32
p, q	Summation Index	2.19
r, θ, z	Spherical Coordinates	2.25
a	Radius of Core Mantle Boundary	2.25
Z	Cylindrical Height Coordinate	2.26
i, j, k, l	Summation Index	2.28, 2.32
f	Function, Definition	2.33, 5.29

d, d_{ij}	Function, Constants	3.5, 6.1
A, A	Square Matrix, Constant	3.10, 4.2
λ	Eigenvalue	3.10
x	Eigenvector	3.10
q	Summation Index	3.13
Q	Summation Limit	3.13
D_{pq}, D	Matrix, Definition	3.16, 5.8
s	Summation Index	4.28
cu_m	Expressions	4.30
E	Definition	5.10
H_{qp}, H	Matrix, Total Angular Momentum	5.27, 6.4
γ	Definition	5.28
$I(i,j)$, I	Definition, Inertia Tensor	5.30, 6.7
A_{qp}, B_{qp}, ...	Matrices	5.31 - 5.44
t, v, w	Summation Index	5.36, 5.41, 5.36
h	Flow	6.9

CHAPTER 1

1.1 Introduction and Motivation

Unlike the Earth's surface, the core of the Earth is not directly accessible for study, consequently much is still unknown about its behaviour. Attempts have been made to characterize this region using theoretical and observational studies of ray seismology, short-period free oscillations, tides and wobble/nutation. These studies have generated a great deal of information about the core.

Specifically, studies of ray seismology, tides and wobble/nutation have established the fluidity of the core. Observation and analysis of seismic wave propagation have allowed determination of the velocity structure within the liquid core, and established the existence of the inner core. Studies of short-period free oscillations have helped to determine the solidity of the inner core. This work has been summarized in an historical sense by Brush (1980). Currently, seismic tomography is being used to map the topography of the core mantle boundary (Anderson & Dziewonski 1984).

These short-period free oscillations are what seismologists simply call free oscillations and have periods on the order of one hour. They are termed short-period here, as a way of distinguishing them from the much longer period free oscillations that will be discussed later in this work.

Unfortunately, several key properties of the core cannot be firmly established from these studies. One of the most important is the value and sign of ß, a dimensionless parameter that represents the fractional departure of the density gradient from a strictly neutral, or adiabatic, value. This

parameter is defined in section 1.3.2, equation (1.15) and was originally developed by Pekeris & Accad (1972). The parameter ß gives an indication of the ability of a region of the core to convect and shall be referred to as the stability parameter. If the value is positive then that region will be unstable, if it is negative then it will be stably stratified and if the value is zero then the region will be neutrally stratified. A study by Masters (1979) using ray seismology estimates an upper limit of the absolute value of ß to be less than 0.03 - 0.05. The value of ß is related to another parameter, N, called the Brünt-Väisälä frequency. This value is defined in terms of ß in equation (1.16) in section 1.3.2. This frequency refers to the oscillations of a particle about its equilibrium position in a stably stratified fluid. Thus the oscillations will only occur if ß is negative. If the value of ß is of the size estimated above and the sign is negative, then the period of this free oscillation in the Earth's core would be greater than several hours. This is called a long-period free oscillation as the period is significantly longer than that for short-period free oscillations.

There are actually two distinct types of long-period free oscillations that could occur in the Earth's core. The type mentioned above are called gravity waves or core undertones and depend on negative buoyancy for a restoring force. The presence of negative buoyancy implies stable stratification at some location in the Earth's core. Thus, some part of the Earth's core would have to be stably stratified, indicated by a negative ß value, in order for gravity waves to exist. Their periods are on the order of $2\pi/N_{max}$ and longer, where N_{max} is the limiting Brünt-Väisälä frequency. As noted earlier, this can be several hours depending on the magnitude of ß.

The other type of long-period free oscillation that could exist in the core is termed an inertial wave. This type of oscillation depends upon the Coriolis effect as the restoring force. Thus

inertial waves require rotation for their very existence. Their periods are on the order of half a day and longer.

If the stratification present in the core is such that the periods of the gravity waves are comparable with the periods of the inertial waves, then the presence of negative buoyancy will modify the inertial waves, and the gravity waves will be altered by rotation. This leads to the terminology of "inertia/gravity waves" when referring to both types of long-period free oscillations when their spectra overlap. This will occur in a rotating stably stratified liquid for which the magnitude of the rotation period and $2\pi/N_{max}$ are comparable. This may well be the case in some regions in the Earth's core.

The range of frequency response in the core covered by the short-period free oscillation data is relatively small, having periods on the order of one hour. This means that only a very small portion of the Earth's free oscillation spectrum has actually been well studied. The dominant restoring force for short-period free oscillations is elasticity, as opposed to the Coriolis effect and negative buoyancy for the inertia/gravity wave spectrum.

Theoretical studies of short-period free oscillations have generally used spherical harmonic expansions as a normal mode can be described by a single spherical harmonic. Unfortunately this is not the case for inertia/gravity wave studies. The cylindrical symmetry that is imposed by rotation on the problem means that a normal mode can no longer be described by a single spherical harmonic. A whole chain of spherical harmonics must be used instead. In order to solve the problem this chain must, at some stage, be truncated. Unfortunately, this truncation makes the solutions inaccurate and other methods of solution need to be investigated.

In an effort to simplify the mathematics of the inertia/gravity wave problem, Smylie & Rochester (1981) introduced the subseismic approximation (SSA). It was hoped that this simplification would suggest an alternative method of solution that avoided or simplified the spherical harmonic expressions. The equation resulting from this simplification, known as the subseismic wave equation (SSWE), can be solved using a novel separation of variables under very specific conditions (Smylie & Rochester 1986a). However, solving this equation using spherical harmonics still results in a truncation problem. Other methods of solution of this equation have also been summarized by Rochester (1989).

Despite the progress made on the theoretical front, recognition of a signal from inertia/gravity waves will be difficult. The detection of a long period gravity signal interpretable as evidence of core waves could provide additional constraints on core properties, in particular ß, as was first pointed out by Smylie (1974). However, such waves produce very small changes in density at great distances from the instrument and consequently a very small gravity signal is produced. The extreme sensitivity of superconducting gravimeters suggests they are the instrument of choice for observing these signals, and close to a dozen are now deployed worldwide with this as one of the principal objectives. While Melchior & Ducarme (1986) reported observations suggesting the detection of gravity waves, and Aldridge & Lumb (1987) argued that these same observations could be interpreted as due to inertial waves, both of these explanations have been contested by Zurn *et al.* (1987). At present all that can be said with certainty is that more data from globally distributed instruments are needed before unambiguous interpretations become possible.

Setting aside these observational difficulties, the mathematical description of the inertia/gravity wave spectrum of a rotating liquid-filled spherical shell still presents challenges which must be

met before the observations can be reliably interpreted. This thesis is concerned with one aspect of this theory, namely the use of a variational principle to describe the axisymmetric modes of a compressible neutral or stably stratified rotating liquid filling a rigid spherical container, as a first approximation to representing the Earth's liquid core.

1.2 Review of Previous Work

This section summarizes previous work on: (1) the state of the liquid core, (2) inertial wave theory, and (3) inertia/gravity wave theory applied to the Earth's core.

1.2.1 State of the Liquid Core

The most readily available source of Earth-penetrating energy is that supplied by earthquakes. This energy was used by Jeffreys (1926), who established the fluidity of the Earth's core. He showed that the presence of a large zero-rigidity core surrounded by a mantle with a rigidity determined from ray seismology led to an Earth with a mean rigidity corresponding to that inferred earlier from solid Earth tides and the Chandler wobble period. The idea that the fluid core is convecting has been argued on several different points, including considerations of heat, summarized by Stacey (1977). While the arguments for thermal convection given by Elsasser (1950) as an explanation of the geodynamo were convincing at the time, evidence now suggests that convection in the core may be compositional. This would be due to gravitational segregation of light and heavy fractions of the inhomogeneous core material. In fact it may be possible for

dynamo action to occur in stably stratified regions of the core (Singer & Olson 1984).

Other observations of Earth behaviour have provided information about the core. Benioff *et al.* (1954) devised a long period strain meter in an attempt to record natural periods of whole-Earth oscillations. These were predicted theoretically by Love (1911). While the Kamchatka earthquake of 1952 provided initial results that were considered to be short-period free oscillations, an unambiguous observation of this type of oscillation was made immediately following the 1960 Chilean earthquake.

These early observations were used by Alterman *et al.* (1959) to test several theoretical models (Bullen 1950, Bullard 1957) which were based on data from higher frequency ray seismology. Their inference that free oscillations correlated better with the Bullen B model, with an inner core, was really only confirmed by the data from the Chilean earthquake just mentioned.

An earthquake in Alaska in 1964 further confirmed the existence of whole Earth short-period free oscillations, and provided a wealth of new data. Dziewonski & Gilbert (1972) used this data to refine the current model of the Earth's interior and to establish the solidity of the inner core. While the density distribution is fairly well known from short-period free oscillation data, the details of its departure from a strictly neutral gradient are not well constrained by these data. Consequently, the sign and size of the stability parameter in the core remain unknown in spite of the information provided by short-period free oscillation data.

A compilation of seismic, free oscillation and nutation data enabled Dziewonski & Anderson (1981) to establish the preliminary reference Earth model or PREM. This model provides a basic

reference state for many Earth parameters, including density, for consistent studies of the Earth. In this widely accepted model of the Earth, the liquid core is mostly neutrally stratified. An argument for stable regions in the core has also been given by Gubbins *et al.* (1982), who inferred properties of the core's outer layers from the geomagnetic secular variation. Thus the state of the Earth's liquid core may allow the existence of long-period free oscillations, as inertia/gravity waves. While initial results have been interpreted as both core undertones (Melchior & Ducarme 1986) and as inertial waves (Aldridge & Lumb 1987) more data needs to be collected before these results can be fully understood.

1.2.2 Theory of Inertial Waves

The theory of inertial waves was established by Poincaré (1885), who developed the equation for oscillations in an homogeneous, incompressible, rotating, and inviscid fluid. This equation, (later called the Poincaré equation), describes pure inertial waves and can be solved analytically. A method of solution was developed by Bryan (1889), using a double transformation to obtain an "oblate spheroidal" coordinate system, which allowed separable solutions to the Poincaré problem and were represented as products of Legendre functions. Thus, the theoretical foundation for inertial waves in a rigid, spherical container was established.

Kudlick (1966) greatly expanded the theoretical work on this problem by investigating several effects. He was able to expand the linear theory to include the effects of viscosity by using a superposition of the natural oscillatory modes of the inviscid problem that had been corrected to first order for the effects of viscosity. He also extended the theory to include containers of

arbitrary shape and investigated the effects of an oscillatory body force applied to the container walls. Of particular interest to studies of the Earth, he found a first order inviscid solution for a precessing, fluid filled, rotating spheroid.

Greenspan (1968) provided an excellent summary of rotating fluid dynamic theory. This work considered rotating fluids exclusively, and included a comprehensive investigation of viscosity effects.

A simplified form of Bryan's (1889) solution to the Poincaré problem was found by Aldridge & Toomre (1969). The simplified solutions, which can only be obtained for the case of an incompressible liquid inside a rigid ellipsoidal boundary, were presented as polynomials in cylindrical coordinates for the spherical and spheroidal modes. They also conducted experiments on fluids contained in a rotating sphere for the case when viscosity is considered. The observed spectrum of oscillations obtained from these trials matched well with the values they predicted theoretically, where the analytical solution is possible only for the case of an incompressible liquid inside a rigid ellipsoidal boundary. This experimental set up can be considered as a crude model of the Earth, and the presence of these waves gave impetus to further investigation of more realistic Earth models.

The actual configuration of the Earth's core, a thick shell, presents mathematical difficulties in the theory of inertial waves, as the presence of an inner core renders the problem ill-posed (Stewartson & Rickard 1969). An attempt was made by Aldridge (1972) to circumvent the ill-posed nature of the inertial wave problem in a thick shell by using a variational formulation of the Poincaré equation. Although no analytical solution was found, his experimental results

indicated that inertial waves do exist in the thick shell configuration.

1.2.3 Inertia/Gravity Wave Theory for the Earth's Core: the Subseismic Wave Equation (SSWE)

A study of gravity waves by Pekeris & Accad (1972) in a nonrotating Earth model was done using asymptotic theory to the solution of the short-period free oscillation formulation. However, as the model considered was nonrotating, the Coriolis force was missing in the treatment of this problem; this was shown later to have a very large effect on the solutions (Smylie 1974, Johnson & Smylie 1977). By including the effects of rotation the mathematics of the problem were considerably complicated. When traditional vector spherical harmonics were used to solve this formulation of the problem, the rotational coupling of formerly independent displacement fields of different degree resulted in each normal mode displacement being represented by an infinite chain of spheroidal and toroidal fields. Numerical calculations of course require severe truncation of such chains, which can make the results obtained inconclusive.

Smylie & Rochester (1981) sought to reformulate the problem of inertia/gravity waves in the Earth's core by means of a scaling argument applied to the basic equations. They were able to make the 'subseismic' approximation (SSA) which neglects the effects of flow pressure on density at sub-acoustic frequencies. This in turn led to the subseismic wave equation (SSWE), which governs large scale rotating core dynamics in the sub-acoustic frequency range. It should be noted that the SSWE reduces to the familiar Poincaré equation when the core is treated as homogeneous and incompressible. They also considered the possible regions of stability in the core that can support these oscillations. Further work by Crossley (1984) investigated the

possible frequency regimes, excitation and damping mechanism for these waves. See also the summary by Smylie *et al.* (1984).

A qualitative investigation by Friedlander (1985) determined several regions in the Earth's core where inertia/gravity waves could exist. The structure and location of these regions is very sensitive to the stability structure within the core, confirming the suitability of these waves for use in studying the stability parameter. This work was extended in Friedlander (1987) where asymptotic solutions were extended from limited known solutions. Further investigations in Friedlander (1988) also considered very long period oscillations affected by a magnetic field.

A solution of the SSWE using the separation of variables technique was attempted by Smylie & Rochester (1986a). These solutions were restricted to the case where the Brünt-Väisälä frequency N, and the local compressibility number were uniform throughout the core. This work does demonstrate the relationship between the solutions and physical conditions in the core. A variational principle developed by Smylie & Rochester (1986b) offered an alternative and more powerful method of solution. Unfortunate mistakes in this formulation were discovered and later corrected by Rochester (1989) in a summary of work done on the SSWE. In this summary, several different solution techniques for the SSWE were explored, including separable solutions, asymptotic solutions and the variational principle. A discussion of the formulation of a variational principle for the SSWE alone (i.e. decoupled from the Poisson equation) is given by Wu & Rochester (1990), where they note that it can be developed for a stratified core with deformable boundaries, but only if the stratification is neutral. Recent work by Crossley & Rochester (1991) has shown that for long-period oscillations the subseismic approximation is valid in the interior o the liquid core but becomes unusable at the boundaries.

1.3 The Governing Equations of Core Dynamics

In order to study motion within the Earth's core, an equilibrium reference state must first be chosen. The departures from this state will then define the motions of interest. The reference state chosen here, as an approximation to real core conditions, is of an inviscid fluid in hydrostatic equilibrium in a uniformly rotating coordinate system. The equilibrium gravity, g_o, is given by:

$$g_o = -\nabla V_o \tag{1.1}$$

where V_o is the equilibrium gravitational potential. The Lorentz force is negligible for the time scale of the Earth's rotation and is ignored (Crossley & Smylie, 1975). Starting with these considerations the SSWE can be derived from the basic equations of fluid dynamics as found in Greenspan (1968, pp. 11-12). Notation has been changed here and the formulation includes the Poisson equation to account for the effects of self-gravitation.

The linearized equations of motion, that describe the Eulerian departure of the density (ρ_1), pressure (P_1), and gravitational potential (V_1) from their equilibrium values (ρ_o, P_o, V_o), are the conservation of mass, momentum, entropy and gravitational flux for a self-gravitating system:

$$\frac{\partial \rho_1}{\partial t} = -(v \cdot \nabla \rho_0 + \rho_0 \nabla \cdot v) \tag{1.2}$$

$$\frac{\partial v}{\partial t} + 2\Omega \times v = \nabla V_1 - \frac{1}{\rho_0} \nabla P_1 - \frac{\rho_1}{\rho_0} \nabla V_0 \tag{1.3}$$

$$\frac{\partial P_1}{\partial t} = -\rho_0 (\alpha^2 \nabla \cdot v - v \cdot \nabla V_0) \tag{1.4}$$

$$\nabla^2 V_1 = -4\pi G \rho_1 \tag{1.5}$$

where v is velocity, t is time, α is the compressional wave speed in the core, G is the gravitational constant and Ω is the angular velocity of the steadily rotating reference frame. As the liquid is treated as inviscid, dissipative effects are missing from equations (1.3) and (1.4). The system of equations, (1.2-1.5), must be solved to describe the motions in the core.

In order to solve these equations the Lagrangian displacement from an equilibrium location of a liquid mass element is defined as u. As small departures from equilibrium are being considered, this displacement u can be expressed as:

$$v = \frac{\partial u}{\partial t}. \tag{1.6}$$

The frequency of these small oscillations will be given by ω. These oscillations can then be represented by:

$$u(r,t) = Re\,[u(r)\, e^{i\omega t}] \tag{1.7}$$

where $u(r)$ is generally a complex expression. This will replace equations involving real quantities with equations that are real parts of complex equations. Similar expressions can also be written for ρ_1, V_1, and P_1. Substituting (1.6) and (1.7) into the governing equations (1.2 - 1.5) results in:

$$\rho_1 = -\nabla\cdot(\rho_o u) \tag{1.8}$$

$$-\omega^2 u + 2i\omega\,\Omega\times u = \nabla V_1 - \frac{1}{\rho_o}\nabla P_1 - \frac{\rho_1}{\rho_o}\nabla V_o \tag{1.9}$$

$$P_1 = -\rho_o(\alpha^2\,\nabla\cdot u - u\cdot\nabla V_o) \tag{1.10}$$

$$\nabla^2 V_1 = -4\pi G\rho_1 \tag{1.11}$$

where the equations now relate complex field quantities but involve only spatial differentiation.

For the sake of future simplification, the scalar field variable χ can be defined as:

$$\chi = \frac{P_1}{\rho_0} - V_1. \tag{1.12}$$

This can be substituted into equation (1.9) to give:

$$-\omega^2 u + 2i\omega\Omega \times u = -\nabla\chi + P_1 \nabla\left(\frac{1}{\rho_0}\right) - \frac{\rho_1}{\rho_0}\nabla V_0 \tag{1.13}$$

for the momentum equation.

1.3.1 Density Gradient

In order to consider departures of the equilibrium density gradient ($\nabla\rho_0$) from the adiabatic stratification assumed by the Adams-Williamson equation (Melchior 1986), Pekeris & Accad (1972) introduced ß, the stability parameter. This parameter changes the Adams-Williamson equation:

$$\nabla\rho_0 = \frac{\rho_0 g_0}{\alpha^2} \tag{1.14}$$

that describes an adiabatically or neutrally stratified fluid, to:

$$\nabla \rho_0 = (1-\beta)\frac{\rho_0 g_0}{\alpha^2} \tag{1.15}$$

which can also describe a fluid that is stably or unstably stratified. If ß = 0, the fluid is neutrally stratified and the equation reduces to the Adams-Williamson equation. Otherwise as ß < 0 or ß > 0, the fluid is locally stably or unstably stratified, respectively. In general ß is expected to be radially dependent, but for simplicity core models with a constant ß will be considered.

The stability parameter, ß, is related to the Brünt-Väisälä frequency (Melchior 1986), N, which describes the frequency of oscillation of a particle about its equilibrium point in a stably stratified density profile. This parameter is more commonly used in oceanography and is related to the stability parameter by:

$$N^2 = -\frac{\beta g_0^2}{\alpha^2}. \tag{1.16}$$

The equations which govern dynamics in the core can now be written with the inclusion of the stability parameter as:

$$-\omega^2 u + 2i\omega\Omega\times u = -\nabla\chi + \beta g_0 \nabla\cdot u \tag{1.17}$$

$$\nabla^2 V_1 = 4\pi G \left[\rho_o \nabla\cdot u + \frac{(1+\beta)}{\alpha^2}\rho_o u\cdot g_o\right] \qquad (1.18)$$

where both the conservation of mass equation (1.8) and the entropy conservation equation (1.10) have been decoupled from the equations by substituting for ρ_1 and P_1 respectively. Even this reduced system of equations in general cannot be solved exactly without encountering the truncation problem already mentioned.

1.3.2 Sub-Seismic Approximation

In previous studies of long-period free oscillations of the core (e.g. Greenspan 1968), the solenoidal flow approximation,

$$\nabla\cdot u = 0 \qquad (1.19)$$

has been made in the entropy conservation equation. The incompressibility approximation is made in an attempt to simplify the mathematics of obtaining solutions in the presence of rotation. While this approximation is applicable in laboratory conditions, it should not be used in the core where compression is a factor. Therefore, a different approximation from the solenoidal flow approximation needs to be made in the governing equations. This approximation needs to simplify the mathematics yet preserve the physics of a compressible core.

This approximation was formulated by Smylie & Rochester (1981) who used a scaling argument to eliminate a term in the entropy conservation equation (1.10). This was done by calculating the magnitude of individual terms in the entropy conservation equation after choosing realistic values for particular Earth properties. Equation (1.10) can be rewritten as:

$$\nabla \cdot \boldsymbol{u} = - \frac{p_1}{\rho_0 \alpha^2} - \frac{\boldsymbol{u} \cdot \boldsymbol{g}_0}{\alpha^2}. \qquad (1.20)$$

These terms relate the local compression of the LHS, to the effect of compression due to flow pressure, and compression due to transport through the equilibrium pressure gradient, the first and second terms of the RHS. By writing the governing equations in dimensionless form, using characteristic values for Earth properties, and considering a longer time scale than that used in conventional seismology, Smylie & Rochester (1981) demonstrated that the first term of the right hand side can be eliminated from the system of equations. This is because it is at least 3 orders of magnitude smaller than other terms, and contrasts to the situation in the acoustic frequency range where this term is substantial. This neglects the effect of compression due to flow pressure in the governing equations and is termed the subseismic approximation. It leads to a simplified expression for the entropy conservation equation:

$$\nabla \cdot \boldsymbol{u} = - \frac{\boldsymbol{u} \cdot \boldsymbol{g}_0}{\alpha^2}. \qquad (1.21)$$

This expression can also be rewritten as:

$$\nabla\cdot(\rho_o^1\, u) = 0 \tag{1.22}$$

where

$$\rho_o^1 = \rho_o\, e^{-\int \frac{\beta}{\alpha^2}\, dW_o} \tag{1.23}$$

and where W_o is the gravitational potential that includes rotation effects. The simplified form of the entropy conservation equation (1.22) will be used later.

Making the substitution of (1.21) into the governing equations (1.17 and 1.18) we arrive at:

$$-\omega^2 u + 2i\omega\ \Omega\times u = -\nabla\chi + \frac{\beta}{\alpha^2}\, g_o\, u\cdot g_o \tag{1.24}$$

$$\nabla^2 V_1 = -4\pi G\frac{\beta\,\rho_o}{\alpha^2}\, u\cdot g_o. \tag{1.25}$$

Equation (1.12) is used to obtain P_1 from χ and V_1 after these potentials have been obtained from equations (1.24) and (1.25).

At this point no further approximations are needed and the subseismic wave equation (SSWE) can be derived directly from equation (1.24). To do this, both the dot and cross products of (1.24)

with e_3, the unit vector parallel to Ω, must be formed. These results are substituted back into (1.24). This expression is then solved for u by forming various other products of the original expression. When this is finished the result is a simple expression:

$$u = \Gamma \cdot \nabla\chi \tag{1.26}$$

where:

$$\Gamma = \frac{1}{\omega^2(\sigma^2-1)}\left\{\sigma^2 1 - e_3 e_3 - \frac{C^* C}{B} + i\sigma\, e_3 \times 1\right\} \tag{1.27}$$

$$C = -\sigma^2 g_0 + (e_3 \cdot g_0) e_3 + i\sigma\, e_3 \times g_0 \tag{1.28}$$

$$B = \frac{\omega^2 \alpha^2 (\sigma^2-1)}{\beta} + \sigma^2 g_0^{\,2} - (e_3 \cdot g_0)^2 \tag{1.29}$$

and σ is defined as:

$$\sigma = \frac{\omega}{2\Omega} \tag{1.30}$$

where C^* is the complex conjugate of C, and 1 is the unit dyadic. The SSWE can now be given, by combining (1.22) and (1.26) as:

$$\nabla\cdot(\rho_o^t\,\Gamma\cdot\nabla\chi) = 0. \tag{1.31}$$

The momentum equation (1.26) and the subseismic wave equation (1.31) were first given in this form by Rochester (1989).

1.3.3 Boundary Conditions

The final consideration is that of the boundary conditions. Three types of boundaries seem plausible for problems of the kind we are considering: rigid fixed, deformable, and free boundaries. The boundary condition explored here is a rigid fixed boundary, an approximation to real Earth conditions at the core mantle boundary (CMB). This means there is no mechanical energy exchange between the core and the rest of the Earth. The free boundary condition would be used in the study of stars and the deformable condition could be used in a more detailed study of the core than that presented here. The rigid fixed boundary condition is formed by noting that the normal component of the displacement vector must be zero at the boundary, or:

$$\boldsymbol{u}\cdot\boldsymbol{n} = 0 \tag{1.32}$$

where $\boldsymbol{n}$ is the unit normal vector. By noting that the normal vector can be written as:

$$-n = \frac{g_0}{g_0} \tag{1.33}$$

because the equilibrium gravity, g_0 is everywhere normal to the core mantle boundary, an alternative expression for the boundary condition can be given as:

$$C \cdot \nabla\chi = 0. \tag{1.34}$$

When this is expanded as:

$$-\sigma^2 g_0 n \cdot \nabla\chi + (e_3 \cdot g_0)(e_3 \cdot \nabla\chi) + i\sigma (e_3 \times g_0) \cdot \nabla\chi \tag{1.35}$$

it is seen to be a very complicated boundary condition as it involves several different components of $\nabla\chi$ and also contains σ, the dimensionless frequency. Fortunately, in chapter 2, (1.34) will be shown to be a 'natural' boundary condition when solving (1.31) by means of a variational principle, and thus need not be invoked independently to constrain the solutions. The solution to the problem of long period free oscillations in the core is then found by solving (1.31) for χ and subsequently solving for u, ρ_1, V_1 and P_1 in equations (1.26), (1.8), (1.25) and (1.12) respectively.

1.4 Solving the Subseismic Wave Equation: Prospectus

There are several methods that can be used to solve the subseismic wave equation. Many of these have been examined by Rochester (1989): separable solutions, asymptotic solutions, and the variational principle. In this work, the variational principle, asymptotic solutions and a Galerkin approach are explored.

This thesis investigates several different methods of solution of the subseismic wave equation in an incompressible or slightly stably stratified liquid bounded by a rigid fixed spherical surface, as a preliminary to an attack on the inertial/gravity wave spectrum for more realistic Earth models. In the next chapter, the variational principle is applied to the Poincaré equation (as a limiting form of the SSWE) to reproduce the results obtained by Aldridge & Toomre (1969). This will demonstrate the utility of the variational principle for the SSWE, which retains the ability to examine more of the problem than the formulation provided by Aldridge & Toomre (1969). Such is done in chapter 3 where the effect of compressibility is studied using a radial density profile from the Preliminary Reference Earth Model (PRFM) of Dziewonski & Anderson (1981). In the fourth chapter the perturbation approach is employed to establish the response of the system when the stability parameter, β, is nonzero. Finally, in chapter 5, the Galerkin approach is used with both a neutrally and stably stratified density profile. Several variations of this approach are considered to delineate changes in inertial frequencies and detect new gravity modes. The results are then compared with the previous methods. The final chapter summarizes the work done and briefly explores some of the possibilities for further research.

CHAPTER 2

2.1 Introduction

The purpose of this chapter is first to review the demonstration that a variational principle can be used as a method of solution for the SSWE. The resulting functional, which is more general than that derived by Aldridge (1972) for axisymmetric solutions of the Poincaré equation, is then used to reproduce the eigenvalue results of Aldridge and Toomre (1969), hereafter referred to as A&T.

2.2 Variational Principle

The variational principle will be applied to a functional where the SSA has been made throughout the fluid volume and the outer boundary is considered to be rigid and fixed. The functional chosen for this work is:

$$F = \int_v \chi^* \mathcal{L} \chi \, dv + \int_s \psi^* n \cdot u \, ds \qquad (2.1)$$

where χ and ψ are arbitrary functions we are free to choose and $\mathcal{L}$ is defined as the linear operator for the SSWE:

$$\mathcal{L}\chi = \nabla\cdot(\rho_o^t \Gamma\cdot\nabla\chi). \tag{2.2}$$

We need to ensure that this functional is suitable for the implementation of a variational principle with trial functions χ which do not satisfy the rigid fixed boundary condition:

$$n\cdot u = 0. \tag{2.3}$$

Implementing a variation in the functional, $\delta\chi$ and $\delta\psi$, we obtain:

$$\begin{aligned} \delta F = &\int_v \{\delta\chi^*\mathcal{L}\chi + \delta\chi(\mathcal{L}\chi)^*\}dv \\ &+ \int_s \{\delta\psi^* n\cdot u + \psi^* n\cdot\delta u\}ds \\ &+ \omega^2(\sigma^2-1)\int_s \{\rho_o^t\chi^* n\cdot\delta u - \rho_o^t\delta\chi\, n\cdot u^*\}ds \end{aligned} \tag{2.4}$$

where $\delta\chi$ and $\delta\psi$ are arbitrary in the volume and on the boundary. Rearranging terms we can write:

$$\begin{aligned} &\delta F - \int_v \{\delta\chi^*\mathcal{L}\chi + \delta\chi(\mathcal{L}\chi)^*\}dv \\ &- \int_s \delta\chi^* n\cdot u\, ds + \omega^2(\sigma^2-1)\int_s \rho_o^t\delta\chi\, n\cdot u^* ds \\ &= \int_s \{\psi^* + \omega^2(\sigma^2-1)\rho_o^t\chi^*\} n\cdot\delta u\, ds \\ &= 0 \end{aligned} \tag{2.5}$$

for a choice of:

$$\psi = -\omega^2(\sigma^2 - 1)\rho_0^t \chi \tag{2.6}$$

at the boundary without restricting the choice of $\delta\chi$ or δu. Therefore with ψ chosen as above we have demonstrated that:

$$\delta F = 0 \rightarrow \begin{array}{ll} \mathcal{L}\chi = 0 & \textit{in } v \\ \textit{and} & \\ n \cdot u = 0 & \textit{on the boundary.} \end{array} \tag{2.7}$$

since $\delta F = 0$ only if both of these statements are true in equation (2.4) above. Thus for the functional F so chosen, we have demonstrated that a variational principle can be implemented with trial functions χ that do not need to independently satisfy the boundary condition. This boundary condition is then called a natural boundary condition.

The functional F in equation (2.1) can be rewritten, first by substituting for ψ, and $\mathcal{L}$ to get:

$$F = \int_v \chi^* \nabla \cdot (\rho_0^t \Gamma \cdot \nabla \chi)\, dv - \omega^2(\sigma^2 - 1) \int_s \rho_0^t \chi^* n \cdot u\, ds. \tag{2.8}$$

Next substituting for u and using a vector theorem yields:

$$F = \int_v [\nabla\cdot(\chi^* \rho_o^t \Gamma\cdot\nabla\chi) - \nabla\chi^* \rho_o^t \Gamma\cdot\nabla\chi]\, dv - \int_s \rho_o^t \chi^* n\cdot\Gamma\cdot\nabla\chi\, ds. \qquad (2.9)$$

Then using the divergence theorem this becomes:

$$F = \int_s \chi^* \rho_o^t n\cdot\Gamma\cdot\nabla\chi\, ds - \int_v \nabla\chi^* \rho_o^t \Gamma\cdot\nabla\chi\, dv - \int_s \rho_o^t \chi^* n\cdot\Gamma\cdot\nabla\chi\, ds. \qquad (2.10)$$

And we are left with:

$$F = -\int_v \rho_o^t \nabla\chi^*\cdot\Gamma\cdot\nabla\chi\, dv. \qquad (2.11)$$

This is only true for the rigid fixed boundary condition; if free or deformable boundaries were being considered, then other terms would need to be added to the functional in equation (2.1) to account for these conditions.

2.3 Approximations to Produce the Poincaré Equation

Attention can now be given to the specific problem that A&T considered, which consisted of axisymmetric, inertial oscillations in a homogeneous, incompressible fluid-filled sphere. Several approximations can now be made to the governing subseismic wave equation. The homogeneous condition means that density is independent of location and consequently ρ_o^t is a constant. The incompressible condition implies that the acoustic wave speed is infinite, with $\alpha = \infty$. In reality

α is observed to be 8-10 km/s. This assumption makes the value of B = ∞, thus eliminating the third term in (1.27), and reduces the expression for Γ to Γ_p:

$$\Gamma_p = \{\sigma^2 \mathbf{1} - \mathbf{e}_3\mathbf{e}_3 + i\sigma\, \mathbf{e}_3\times\mathbf{1}\} \tag{2.12}$$

and since ρ_o' is now a constant, the SSWE (1.31) reduces to:

$$\nabla\cdot(\Gamma_p\cdot\nabla\chi) = 0 \tag{2.13}$$

which when expanded yields:

$$\sigma^2\nabla^2\chi - (\mathbf{e}_3\cdot\nabla)^2\chi = 0 \tag{2.14}$$

the well known Poincaré equation. The rigid fixed boundary condition ($\mathbf{u}\cdot\mathbf{n}=0$) is still natural in the reduced expression and the functional can simply be written as:

$$F(\chi) = \int_v \nabla\chi^*\cdot\Gamma_p\cdot\nabla\chi\, dv \tag{2.15}$$

and the variational principle can be applied.

2.4 General Form of the Trial Function

Up to now, no consideration has been given to the construction of the trial function χ. The azimuthal dependence ϕ, can be separated out by writing:

$$\chi(\psi_1, \psi_2, m, \sigma)\, e^{im\phi} \tag{2.16}$$

with ψ_1 and ψ_2 any independent curvilinear coordinates orthogonal to ϕ: cylindrical coordinates (R, Z) have been adopted in this thesis. This form for χ is now substituted in the simplified functional, (2.15), and can be expanded as:

$$F = \sigma^2 \int_v |\nabla\chi|^2\, dv - \int_v |e_3 \cdot \nabla\chi|^2\, dv + \int_s |\chi|^2 \{\frac{m\sigma (e_3 \times n)\cdot\phi}{R}\}\, ds \tag{2.17}$$

where R is the radial coordinate of cylindrical coordinates. This formulation explicitly demonstrates the azimuthal dependence of the third term which results from the application of the divergence theorem. It is also apparent that the separation of azimuthal dependence from χ has left the integrands in (2.17) in a form which clearly indicates that $\chi(R, Z, m, \sigma)$ can be treated as entirely real as all terms are squared. However, in order to consider the axisymmetric case, as A&T did using a stream function, m, the azimuthal number must be set equal to zero. The option to explore nonaxisymmetric cases demonstrates one advantage of the use of χ, over that of a stream function. The nonaxisymmetric case would involve the introduction of new oscillation modes and result in the modification of previously discovered modes due to the

presence of σ in the final term of the functional. Considering the axisymmetric case, $m = 0$, the functional (2.17) reduces to:

$$F = \int_v \{\sigma^2 |\nabla\chi|^2 - |e_3 \cdot \nabla\chi|^2\}\, dv. \tag{2.18}$$

The application of the variational principle can now be considered in more detail.

2.5 General Solution Method

In order to proceed with the variational principle, the trial function χ is represented as a linear combination of basis functions as:

$$\chi = \sum_p c_p \phi_p \tag{2.19}$$

where the c_p's are constants to be determined, and the ϕ_p's are the basis functions that must be selected. By substituting this general form of χ in the axisymmetric functional (2.18), the result is:

$$F = \sum_p \sum_q c_p c_q \int_v \{\sigma^2 (\nabla\phi_p)\cdot(\nabla\phi_q) - (e_3 \cdot \nabla\phi_p)(e_3 \cdot \nabla\phi_q)\}\, dv. \tag{2.20}$$

This can be simplified by defining the matrix G_{pq} in the functional as:

$$F = \sum_{p} \sum_{q} c_p c_q G_{pq}. \tag{2.21}$$

Now the variational principle can be applied by taking the variation of the functional with respect to constant, c_p to give:

$$\frac{\partial F}{\partial c_p} = 0, \text{ then } \sum_{p} c_p G_{pq} + \sum_{q} c_q G_{qp} = 0 \tag{2.22}$$

where the variation has now been transferred to the c_j constants. The matrix G is obviously symmetric, since all the terms involve squares of quantities. The problem thus reduces to:

$$\sum_{p} c_p G_{pq} = 0 \quad or \quad Gc = 0 \tag{2.23}$$

an eigenvalue/eigenvector problem where:

$$\det G = 0 \tag{2.24}$$

gives the eigenvalues, σ, and the constants, c_p, give the associated eigenvectors.

2.6 Specific Formulation of χ and Results

At first it would seem that any linearly independent set of functions could be used to construct the trial functions. For example, in spherical coordinates, (r and Θ), χ might be constructed from:

$$\chi = \sum_{p=0}^{P} \sum_{n=0}^{N} c_{p,n} \sin \frac{p \pi r}{a} \; P_n^m(\cos\theta) \quad \textit{where} \quad n \geq |m| \tag{2.25}$$

with a as the radius to the core-mantle boundary. The different values of p explore the radial dependence and the various values of n explore the angular dependence. However, the problem as a whole must be carefully considered to ensure that an appropriate trial function is chosen.

When constructing a suitable trial function, it is helpful to compare the procedure used by A&T with our work. Since they considered the axisymmetric case for an incompressible fluid ($\alpha = \infty$), they were able to formulate the problem in terms of a stream function. A comparison can be made between this stream function ψ of A&T and the current formulation for the trial function χ by noting that displacement u is proportional to the velocity v used in A&T. The two are related since the displacement, u leads to a velocity of: $v = i\,\omega\,u$, which when equated with the velocity components from the stream function of A&T results in:

$$\sigma^2 R \frac{\partial \chi}{\partial R} = \frac{\partial \psi}{\partial Z} \quad \textit{and} \quad R(\sigma^2 - 1) \frac{\partial \chi}{\partial Z} = -\frac{\partial \psi}{\partial R} \tag{2.26}$$

when expressed in cylindrical coordinates (R,Z). Thus a possible construction for the trial function χ is suggested by the formulation of ψ used by Aldridge (1972, p. 34). In order to compare results with those obtained by A&T, attention is confined to those axisymmetric modes compatible with the excitation method used in the experiments performed by A&T, namely:

$$\frac{\partial \chi}{\partial R} = 0 \quad on \quad R = 0 \quad and \quad \frac{\partial \chi}{\partial Z} = 0 \quad on \quad Z = 0. \tag{2.27}$$

These restrictions mean that there is no flow across the rotation axis and no flow across the equatorial plane respectively. Thus χ should have the form (in cylindrical coordinates) of:

$$\chi = \sum_{i=0}^{N} \sum_{j=\delta_{i0}}^{N-i} c_{ij} R^{2i} Z^{2j} \tag{2.28}$$

where δ_{i0} is the Kronecker delta symbol and N is sufficiently large to ensure that all the eigenfrequencies of interest are found. That this should be a successful form for χ to take is also apparent from the similarity to the trial function used by Aldridge (1967), which provides an exact solution to the problem in the homogeneous, incompressible case, the same as that being considered here.

It is important to note that cylindrical coordinates have been chosen here. While this may seem like an odd choice when working with a spherical boundary, a comparison of the expanded functional in the cylindrical:

$$F = \int_v \left[\sigma^2 \left(\frac{\partial \chi}{\partial R}\right)^2 + (\sigma^2 - 1)\left(\frac{\partial \chi}{\partial Z}\right)^2 \right] dv \qquad (2.29)$$

and spherical coordinate systems:

$$F = \sigma^2 \int_v \left\{ \left(\frac{\partial \chi}{\partial r}\right)^2 + \frac{(1-z^2)}{r^2}\left(\frac{\partial \chi}{\partial z}\right)^2 - \left[z\frac{\partial \chi}{\partial r} + \frac{(1-z^2)}{r}\frac{\partial \chi}{\partial z} \right] \right\} dv \qquad (2.30)$$

suggests that the equations are much more straightforward in cylindrical coordinates. This is due to the cylindrical symmetry imposed on the problem by rotation, which is more important than the spherical shape of the boundary. The actual integration of resulting expressions can be carried out in cylindrical or spherical coordinates, depending on which is simpler.

Since in this formulation the boundary condition is natural, no attempt is made to force the trial function to satisfy it. The trial function (2.28) is then substituted into the functional (2.29), resulting in:

$$F = \sigma^2 c - (1-\sigma^2) f \qquad (2.31)$$

where

$$c = \sum_{i=1}^{N} \sum_{j=0}^{N-i} \sum_{k=1}^{N} \sum_{l=0}^{N-k} c_{ij}\, c_{kl}\, i\, k \int_0^1 \int_0^{\sqrt{1-z^2}} R^{2(i+k-1)}\, Z^{2(j+l)}\, R\, dZ\, dR \qquad (2.32)$$

and

$$f = \sum_{i=0}^{N-1} \sum_{j=1}^{N-i} \sum_{k=0}^{N-1} \sum_{l=1}^{N-k} c_{ij}\, c_{kl}\, j\, l \int_0^1 \int_0^{\sqrt{1-Z^2}} R^{2(i+k)}\, Z^{2(j+l-1)}\, R\, dZ\, dR \tag{2.33}$$

and a constant factor of 32π has been dropped, the integration limits result from normalizing the radius of the spherical boundary to unity. These terms can then be integrated in cylindrical coordinates to yield:

$$c = \sum_{i=1}^{N} \sum_{j=0}^{N-i} \sum_{k=1}^{N} \sum_{l=0}^{N-k} c_{ij}\, c_{kl}\, \frac{i\,k}{i+k} \sum_{r=0}^{i+k} \frac{(i+k)!}{r!(i+k-r)!}\, \frac{(-1)^r}{[2(j+l+r)+1]} \tag{2.34}$$

and

$$f = \sum_{i=0}^{N-1} \sum_{j=1}^{N-i} \sum_{k=0}^{N-1} \sum_{l=1}^{N-k} c_{ij}\, c_{kl}\, \frac{2\,j\,l}{i+k+1} \sum_{r=0}^{i+k} \frac{(i+k)!}{r!(i+k-r)!}\, \frac{(-1)^r}{[4(j+l+r)^2-1]}. \tag{2.35}$$

The integration over the Z coordinate was performed using binomial coefficients since:

$$\begin{aligned} \sum_p \sum_q \int_0^1 z^{2p}(1-z^2)^q\, dz &= \sum_p \sum_q \sum_{r=0}^{q} \frac{q!(-1)^r}{r!(q-r)!} \int_0^1 z^{2r+2p}\, dz \\ &= \sum_p \sum_q \sum_{r=0}^{q} \frac{q!(-1)^r}{r!(q-r)!}\, \frac{1}{2(r+p)+1}. \end{aligned} \tag{2.36}$$

This integration can be performed in other equivalent ways that are detailed in Appendix A. Taking the variation of (2.31) with respect to the c_{ij}'s results in the eigenvalue problem that needs to be solved for the dimensionless frequency σ. By noting that c and f are symmetric with respect to the pairs of indices (i,j) and (k,l), the resulting eigenvalue/eigenvector equations can be written as:

$$\sigma^2 \sum_{k=1}^{N} \sum_{l=0}^{N-k} \left\{ \frac{ik}{i+k} \sum_{r=0}^{i+k} \frac{(i+k)!}{r!(i+k-r)!} \frac{(-1)^r}{[2(j+l+r)+1]} \right\} c_{lk}$$
$$-(1-\sigma^2) \sum_{k=0}^{N-1} \sum_{l=1}^{N-k} \left\{ \frac{2jl}{i+k+1} \sum_{r=0}^{i+k} \frac{(i+k)!}{r!(i+k-r)!} \frac{(-1)^r}{[4(j+l+r)^2-1]} \right\} c_{lk} = 0. \tag{2.37}$$

The summation limits have been changed to reflect only those contributions that are non-zero. The non-zero contributions given by i and j correspond exactly to these limits. The restructuring of the limits allows for simplification of the calculations because values of $\sigma^2 = 0$ and 1 can be immediately eliminated, as the eigenfrequencies for this boundary value problem lie in the range $0 < \sigma < 1$. This is done by noting three distinct contributions to the eigenvalue/eigenvector equations where now the range of i and j do not exactly correspond to the range of k and l:

1. For $i = 0; j = 1$ to N

   the contribution is:

$$\sum_{k=0}^{N-1} \sum_{l=1}^{N-k} \left\{ \frac{2l}{k+1} \sum_{r=0}^{k} \frac{k!}{r!(k-r)!} \frac{(-1)^r}{[4(j+l+r)^2-1]} \right\} c_{kl} = 0. \tag{2.38}$$

2. For $i = 1$ to N; $j = 0$

the contribution is:

$$\sum_{k=1}^{N}\sum_{l=0}^{N-k}\left\{\frac{k}{i+k}\sum_{r=0}^{i+k}\frac{(i+k)!}{r!(i+k-r)!}\frac{(-1)^r}{[2(l+r)+1]}\right\}c_{kl} = 0. \tag{2.39}$$

3. For $i = 1$; $j = 1$ to N-i

the contribution is:

$$\begin{aligned}&\sigma^2 i\sum_{k=1}^{N}\sum_{l=0}^{N-k}\left\{\frac{k}{l+k}\sum_{r=0}^{i+k}\frac{(i+k)!}{r!(i+k-r)!}\frac{(-1)^r}{[2(j+l+r)+1]}\right\}c_{kl}\\ &-(1-\sigma^2)j\sum_{k=0}^{N-1}\sum_{l=1}^{N-k}\left\{\frac{2l}{i+k+1}\sum_{r=0}^{i+k}\frac{(i+k)!}{r!(i+k-r)!}\frac{(-1)^m}{[4(j+l+r)^2-1]}\right\}c_{kl} = 0.\end{aligned} \tag{2.40}$$

In order to express these in matrix form, as $G\,c = 0$, it is necessary to transform from the double index (i,j) to a single index (say p). This is accomplished with:

$$\begin{aligned}p &= j+\frac{i(2N+3-i)}{2}\\ q &= l+\frac{k(2N+3-k)}{2}\end{aligned} \tag{2.41}$$

and the problem can be expressed as:

$$\sum_q G_{pq} c_q = 0. \tag{2.42}$$

The eigenvalues are found from:

$$\det G = 0. \tag{2.43}$$

This is now an algebraic problem that can be solved using a computer program. The details of the program sp5.f are found in Appendix B. The frequency range searched was from zero to one. The program uses the IMSL subroutine DZREAL to search for the eigenvalues from the determinant. The value used for N was initially $N = 3$, then $N = 4$ and finally $N = 5$. The value $N = 5$ reproduced the results of A & T. Larger values of N had no effect on these eigenfrequencies, but of course permitted new ones to be found.

The results obtained from the program exactly match those given by A&T (Table 1). This demonstrates the validity of the application of the variational principle as a method of solution for the Poincaré equation, when implemented with trial functions which take advantage of the 'natural' property of the rigid fixed boundary condition.

The next chapter will introduce a density profile into the equations as a better approximation of a realistic Earth model.

Table 1. Comparison of Dimensionless Eigenfrequencies Obtained by Aldridge and Toomre (1969) and from the Application of the Variational Principle to the Poincaré Equation.

Aldridge and Toomre (1969)	Program with N = 3	Program with N = 4	Program with N = 5
0.6547	0.6547	0.6547	0.6547
0.4688	0.4688	0.4688	0.4688
0.8302	0.8302	0.8302	0.8302
0.3631		0.3631	0.3631
0.6772		0.6772	0.6772
0.8998		0.8998	0.8998
0.2958			0.2958
0.5652			0.5652
0.7845			0.7845
0.9340			0.9340

CHAPTER 3

3.1 Introduction

In this chapter a density profile is included in the model of a neutrally stratified, rotating, self gravitating liquid sphere enclosed by a rigid fixed boundary to determine its effect on the inertial wave eigenfrequencies. The eigenvectors for these eigenfrequencies are also obtained as they will be used in the next chapter.

3.2 Density Profile (PREM)

In order to make the working model approach conditions in the Earth's core, the density profile chosen is that given by Dziewonski and Anderson (1981) in their Preliminary Reference Earth Model (PREM), for the outer core. The density expression is a third degree polynomial that is extended to the centre of the working model since the inner core is currently not being considered. There are three problems with this that must be addressed. The first is that the chosen density profile is not actually valid at the centre of a sphere because the second term in the polynomial does not ensure that gravity is zero there. However, this profile is valid everywhere else in the model core so its use is justified as a first approximation to actual core conditions. The second problem is that the stability parameter, ß, resulting from this density profile is nonzero near the inner core boundary and near the outer core boundary. However, it is neutral in a major portion of the liquid core, so using ß = 0 in the formulation of the problem

is justified for this initial approximation. Figure 1 demonstrates the range of values of ß(r) throughout the liquid core of the PREM model. The nonzero portions of ß(r) are most likely due to the choice by Dziewonski and Anderson (1981), of third degree polynomials for representing Earth parameters. Later, in Chapter 5, a density profile will be developed which is essentially neutral throughout the core. The third difficulty is the ill-posed nature of the problem when an inner core boundary is included (Stewartson & Rickard 1969). This is avoided in the present work by using a fluid sphere.

3.3 Variational Formulation of the Eigenvalue Problem (PREM Core)

The actual density profile used is:

$$\rho_o(u) = \sum_{m=1}^{4} \rho_m u^{m-1} \qquad (3.1)$$

where the coefficients of the polynomial expression from PREM have been renormalized to an average outer core radius of 3480 km and are given by:

ρ_1 = 12.5815

ρ_2 = -0.6903191

ρ_3 = -1.0868125

ρ_4 = -0.900929

where the units are gms/cm^3. The dimensionless radius u is defined as: $u = r/b$ where b is the radius of the sphere representing the outer core boundary and r is the radius in the outer core which has been extended to the centre. The symbol m has been used as a summation index here

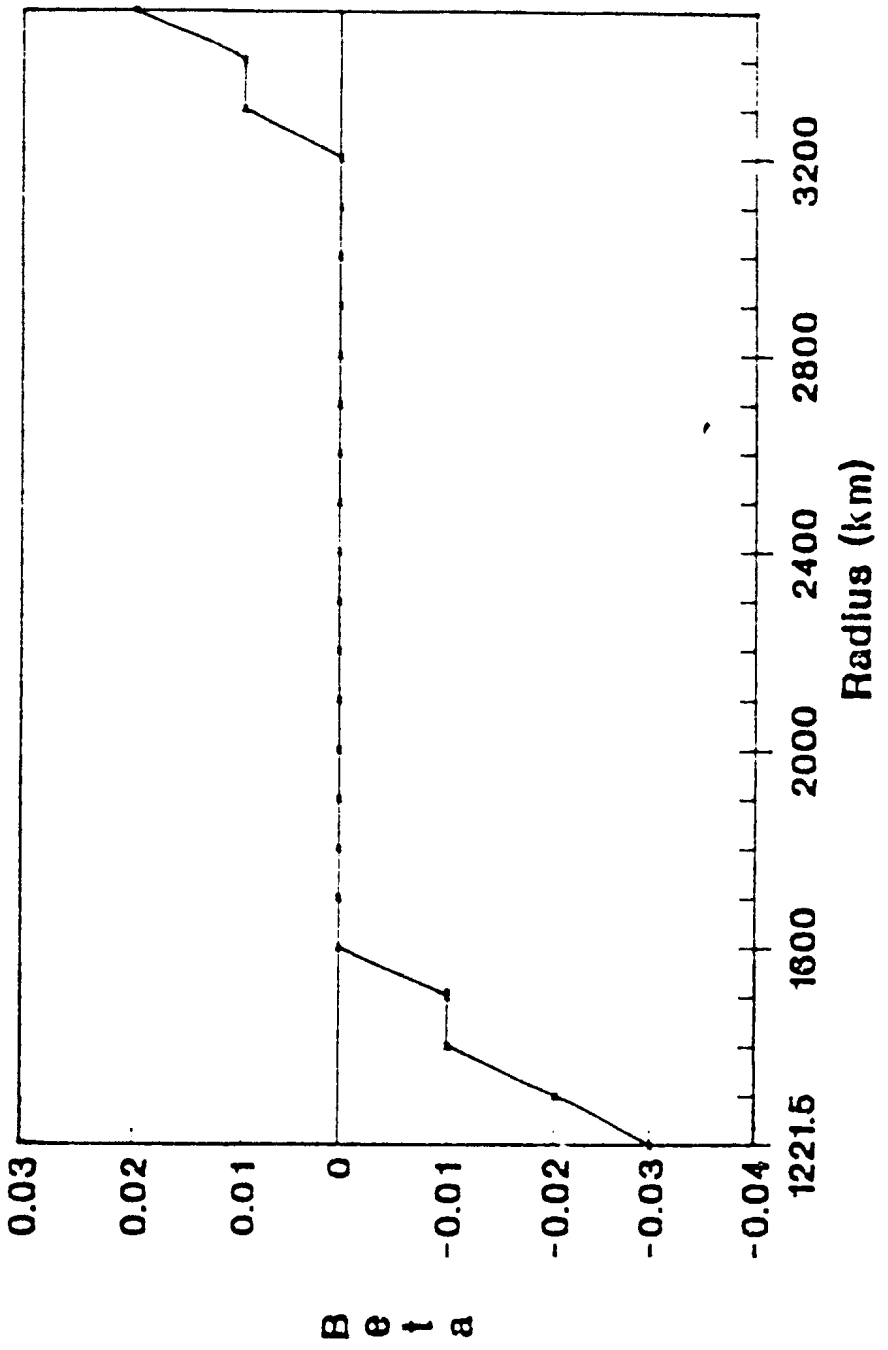

Figure 1. Variation of beta with radius in the liquid core. Values are from PREM.

and henceforth as it is no longer needed to stand for an azimuthal quantum number as our attention has been restricted to axisymmetric oscillations.

The addition of a density profile does not significantly change the mathematics of the problem. The functional now becomes:

$$F = \sigma^2 \int_v \rho_o \, |\nabla\chi|^2 \, dv - \int_v \rho_o \, | e_3 \cdot \nabla\chi |^2 \, dv \qquad (3.2)$$

where ρ_o is the polynomial density expression given above and renormalized from PREM. This is analogous to equation (2.18) in Chapter 2. The boundary condition is still natural and does not need to be considered further. This formulation will give a quick check on the effect, upon the inertial wave eigenfrequencies calculated for a homogeneous liquid, of introducing compressibility via a neutral density profile.

The functional (3.2) above, can be expanded to:

$$F = \int_v [\sigma^2 \rho_o (\frac{\partial\chi}{\partial R})^2 + (\sigma^2 - 1) \rho_o (\frac{\partial\chi}{\partial Z})^2] \, dv \qquad (3.3)$$

where the terms of ρ_o are given in spherical coordinates and the derivatives are in cylindrical coordinates. By transforming the derivative expressions into spherical coordinates just for the integration of the expression, the mathematics of the problem are significantly simplified. Care must be taken to ensure that the density coefficients are properly normalized. The integration

results in:

$$F = \sigma^2 \sum_{i=1}^{N} \sum_{j=0}^{N-i} \sum_{k=1}^{N} \sum_{l=0}^{N-k} d(i,j;\, k,l)\, c_{ij}\, c_{kl} + (\sigma^2 - 1) \sum_{i=0}^{N-1} \sum_{j=1}^{N-i} \sum_{k=0}^{N-1} \sum_{l=1}^{N-k} f(i,j;\, k,l)\, c_{ij}\, c_{kl} \tag{3.4}$$

where

$$d(i,j;\, k,l) = ik \sum_{m=1}^{4} \frac{\rho_m}{m+2(i+j+k+l)} \sum_{r=0}^{i+k-1} \frac{(i+k-1)!\,(-1)^r}{r!\,(i+k-1-r)!\,2(j+l+r)+1} \tag{3.5}$$

and

$$f(i,j;\, k,l) = jl \sum_{m=1}^{4} \frac{\rho_m}{m+2(i+j+k+l)} \sum_{r=0}^{i+k} \frac{(i+k)!\,(-1)^r}{r!\,(i+k-r)!\,2(j+l+r)-1} \tag{3.6}$$

and ρ_m are the coefficients of the PREM polynomial for density in the outer core as given earlier. The summation limits indicate the nonzero contributions. Again applying the variational principle results in an eigenvalue system similar to equation (2.37) in Chapter 2. The equations are now:

$$\sigma^2 d(i,j;k,l)\, c_{kl} - (1-\sigma^2) f(i,j;k,l)\, c_{kl} = 0 \qquad (3.7)$$

where d and f are defined above and the summations have been dropped for clarity. At no point in the summation do the denominators of these terms go to zero, so all terms can be calculated. These expressions can now be easily programmed by the addition of a loop in the computer program sp5.f to account for the density summation. The revised program spden5.f is found in Appendix C. The eigenfrequencies obtained from this program are shown in Table 2 and are compared with those obtained for the homogeneous model considered in Chapter 2. The frequency range that was searched by the program was between 0 and 1. The program uses the IMSL subroutine DZREAL to search for the eigenvalues from the determinate. The value of N used was $N = 5$, as the results from the previous chapter showed that all eigenvalues of interest were found using this value and we are presently interested in the modification of these eigenvalues.

Table 2. Comparison of Dimensionless Eigenfrequencies in an Homogeneous, Neutrally Stratified Sphere and in a Neutrally Stratified Sphere with a Density Profile of the Outer Core of PREM.

Homogeneous	PREM
.2958	.2976
.3631	.3653
.4688	.4714
.5652	.5660
.6547	.6572
.6772	.6779
.7845	.7847
.8302	.8306
.8998	.8998
.9340	.9340

These results indicate that the addition of a density profile actually increases the eigenfrequencies, more significantly at the lower end of the spectrum than at the higher end where there is little or no effect. This contradicts Friedlander (1987), who predicted a decrease in the values of the eigenspectrum with the addition of a density profile. This may be due to her choice of a reference state of an homogeneous, incompressible sphere which was perturbed to a non-neutral, compressible state. In fact one would expect that replacing a homogeneous incompressible liquid by a neutrally stratified compressible one would increase the eigenfrequencies, since the compressibility provides a small restoring force supplementing the dominant one due to the Coriolis effect. This thesis, in Chapter 5, will show that the addition of a non-neutral density profile is not a small perturbation and thus treating both compressibility and non-neutral stratification as perturbations is not an acceptable approach. The development followed here has as an initial model an homogeneous, incompressible sphere, as used in Chapter 2. This is then altered to the reference state, a neutrally stratified, compressible model as is developed in the present chapter. In the succeeding chapter a perturbation method will be applied to this reference state to determine the effect of non-neutral stratification.

3.4 Eigenvectors

At this point it is worthwhile to consider the eigenvectors associated with each eigenvalue for the case of a neutrally stratified, compressible model as these will be used in the next chapter. The trial function used:

$$\chi = \sum_{i=1}^{N} \sum_{j=\delta_{ij}}^{N-i} c_{ij} R^{2i} Z^{2j} \tag{3.8}$$

when $N = 5$, results in 20 coefficients c_{ij} for the eigenvector. These are ordered using the double to single index given previously as equation (2.41). Thus,

$$c_p = c_{i,\frac{(2N-3-i)}{2}} \tag{3.9}$$

In a conventional eigensystem the eigenvalue\vector equations can be expressed as:

$$A x = \lambda x \tag{3.10}$$

where A is a square matrix, x is the eigenvector, and λ is the eigenvalue. While the formulation being considered here is not a conventional eigensystem it can be written in a form analogous to equation (3.10) by rewriting equation (3.7) as:

$$A c = \frac{1}{\sigma^2} B c \tag{3.11}$$

or in component form:

$$A_{pq} c_p = \frac{1}{\sigma^2} B_{pq} c_p \tag{3.12}$$

where A is formed from $d(i,j;k,l) + f(i,j;k,l)$, B from $f(i,j;k,l)$ and c is the eigenvector with p^{th} row c_p. Again equation (2.41) has been used to obtain the transformation between double and single indices. Unfortunately, the eigenvectors cannot be found using this formulation because the matrix A is very nearly singular. However, the eigenvectors can be successfully found by reformulating the problem.

The eigenequations can be written as:

$$\sum_{q=1}^{Q} G_{pq} c_q = 0 \tag{3.13}$$

where G_{pq} is obtained from equation (3.4) in analogy with equation (2.42). Note that because the eigenvalues have already been obtained there are only Q-1 parameters that can be still be obtained from the equations (3.13). Thus one of the coefficients of the eigenvectors must be found some other way. This is done by noting that these are free oscillations, i.e. of arbitrary amplitude, so there is a degree of freedom that must be constrained by choosing one of the coefficients of the eigenvectors. By making the first coefficient 1, the rest of the coefficients can then be found and they will be scaled to this initial choice. The eigenequations can then be rewritten as:

$$\sum_{q=2}^{Q} G_{pq}\, c_q = -G_{p1} \tag{3.14}$$

where $q = 2, 3, \ldots Q$ and letting $c_1 = 1$. This is now a linear system of equations that can be solved for the remaining eigenvector coefficients c_{q-1} after the appropriate eigenvalue has been substituted into the matrix G_{pq}. The system of equations is now one degree smaller than it previously was; the first row and column have been removed from the matrix. By letting:

$$p \rightarrow p + 1 \quad \textit{and} \quad q \rightarrow q + 1 \tag{3.15}$$

where the arrows indicate that a value is replaced by another, then defining:

$$D_{pq} = G_{p+1,\,q+1} \quad \textit{and} \quad d_q = c_{q+1} \tag{3.16}$$

the equations can be written as:

$$\sum_{q=1}^{Q-1} D_{pq}\, d_q = -G_{p+1,\,1} \tag{3.17}$$

and the eigenvectors d_q solved for using a linear systems solution package. The program spden5.f was modified to include the IMSL subroutine DLSASF to solve the system and is given in Appendix D as wspdlinsys5.f. The resulting eigenvectors with their associated eigenfrequency

are shown in Table 3, where the first coefficient has been assigned a value of 1. As before, the eigenvalue range searched was between 0 and 1. Also the value of N is again N = 5. The program is similar to spden5.f described earlier, except that now the eigenvalues are used in the subroutine DLSASF to determine the eigenvectors. These eigenvectors will be used in the next chapter in the formulation of a perturbation solution.

Table 3. Eigenvectors Associated With Eigenfrequencies for a Neutrally Stratified Sphere With a Density Profile of the Outer Core From PREM.

Eigenvalue	Coefficient Number	Eigenvector Coefficients
.2976	1	1.0000
	2	-2.6500
	3	8.9165
	4	-10.7797
	5	4.397
	6	0.3537
	7	13.8265
	8	-83.6591
	9	98.7765
	10	-42.1415
	11	-0.1354
	12	-28.2156
	13	156.6337
	14	-81.6015
	15	-2.4620×10^{-2}
	16	15.0030
	17	-86.7444
	18	-5.5746×10^{-3}
	19	-0.1975
	20	-2.6719×10^{-2}

Table 3. Continued.

Eigenvalue	Coefficient Number	Eigenvector Coefficients
.3653	1	1.0000
	2	-2.6002
	3	8.8842
	4	-10.7879
	5	4.4066
	6	0.3455
	7	18.0207
	8	-107.9571
	9	123.2600
	10	-49.5360
	11	-0.1268
	12	-42.0971
	13	236.6772
	14	-124.5127
	15	-1.9056×10^{-2}
	16	25.1843
	17	-144.6092
	18	-3.5114×10^{-3}
	19	-0.1846
	20	-3.2102×10^{-2}
.4714	1	1.0000
	2	-2.1625
	3	7.3522
	4	-8.9393
	5	3.6524
	6	0.3198
	7	6.0140
	8	-40.7797
	9	61.7661
	10	-33.8769
	11	-0.1416
	12	4.1845
	13	-31.1128
	14	13.8618
	15	-1.7779×10^{-2}
	16	-13.0839
	17	72.6477
	18	6.1467×10^{-3}
	19	-0.2506
	20	-7.5419×10^{-3}

Table 3. Continued.

Eigenvalue	Coefficient Number	Eigenvector Coefficients
.5660	1	1.0000
	2	-1.7370
	3	6.5811
	4	-8.2377
	5	3.3769
	6	0.1793
	7	20.3612
	8	-127.5279
	9	159.0353
	10	-67.2105
	11	-0.1031
	12	-31.7695
	13	175.4742
	14	-107.0989
	15	2.2310×10^{-2}
	16	5.5569
	17	-33.1810
	18	3.3977×10^{-2}
	19	-0.1821
	20	-4.7135×10^{-4}
.6572	1	1.0000
	2	-2.2776
	3	6.9005
	4	-8.0848
	5	3.3154
	6	0.7054
	7	-12.2665
	8	75.8173
	9	-85.8010
	10	22.2029
	11	-0.1848
	12	30.9130
	13	-185.3521
	14	129.8382
	15	-8.5156×10^{-2}
	16	-10.3549
	17	56.0833
	18	-5.0501×10^{-2}
	19	-0.4764
	20	-8.4625×10^{-2}

Table 3. Continued.

Eigenvalue	Coefficient Number	Eigenvector Coefficients
.6779	1	1.0000
	2	-2.0021
	3	6.3320
	4	-7.5182
	5	3.0836
	6	0.6069
	7	-5.6797
	8	35.0166
	9	-37.5676
	10	4.7975
	11	-0.1634
	12	17.7919
	13	-110.4150
	14	82.3259
	15	-5.9867×10^{-2}
	16	-6.5822
	17	34.7164
	18	-3.1142×10^{-2}
	19	-0.4394
	20	-7.4840×10^{-2}
.7847	1	1.0000
	2	-0.9445
	3	3.2661
	4	-3.8223
	5	1.5024
	6	0.5064
	7	-0.8593
	8	10.5255
	9	-24.8698
	10	6.0642
	11	-0.1009
	12	8.6148
	13	-69.2478
	14	78.5027
	15	-6.3849×10^{-3}
	16	-5.6108
	17	28.4398
	18	-7.6611×10^{-3}
	19	-0.5220
	20	-0.1074

Table 3. Continued.

Eigenvalue	Coefficient Number	Eigenvector Coefficients
.8306	1	1.0000
	2	-0.4435
	3	2.1245
	4	-2.8513
	5	1.2077
	6	0.4549
	7	13.9078
	8	-102.6534
	9	167.3023
	10	-85.0624
	11	-5.8474×10^{-2}
	12	-25.8536
	13	165.1072
	14	-151.1843
	15	3.7774×10^{-2}
	16	10.6117
	17	-64.3893
	18	4.0071×10^{-2}
	19	-0.5644
	20	0.1422
.8998	1	1.0000
	2	2.4922
	3	-8.5904
	4	11.5842
	5	-5.3751
	6	0.3849
	7	7.8422
	8	-57.7494
	9	81.2868
	10	-39.1405
	11	0.3668
	12	-18.2271
	13	101.1813
	14	-83.3434
	15	0.2750
	16	3.0354
	17	-28.0091
	18	0.2005
	19	-1.6236
	20	-0.5825

Table 3. Continued.

Eigenvalue	Coefficient Number	Eigenvector Coefficients
.9340	1	1.0000
	2	-23.6395
	3	93.4954
	4	-130.7021
	5	61.2706
	6	2.1455
	7	24.2786
	8	-139.7413
	9	267.0663
	10	-183.7994
	11	-5.6188
	12	31.2635
	13	-161.8030
	14	230.8040
	15	-2.8322
	16	22.1556
	17	-50.5829
	18	-1.9131
	19	13.0792
	20	5.2476

CHAPTER 4

4.1 Introduction

A perturbation of the reference state developed in Chapter 3 to include non-neutral stratification is made here to determine the effect on the eigenfrequencies. A corrected version of equation (82) from Rochester (1989) is developed and used.

4.2 Perturbation Theory for the Eigenvalue Problem (Slightly Stable Core)

In Chapter 3 a solution to the problem:

$$\nabla\cdot(\rho_o \mathbf{\Gamma}\cdot\nabla\chi) = 0 \tag{4.1}$$

was found using the variational principle. This is the zeroth order problem given by Rochester (1989) where ρ_o has been substituted for $\rho_o^\dagger$. This problem can now be perturbed to determine the effect of a small non-zero stability parameter ß. The value for the perturbed eigenfrequency will be found from:

$$\sigma = \sigma_0 + A\sigma_1 + A^2\sigma_2 \tag{4.2}$$

where σ_o is the eigenvalue found in Chapter 3, σ_1 and σ_2 are the eigenvalues from the first and second order perturbations, respectively. The value of A is defined by:

$$A = \frac{N_{max}}{2\Omega}. \tag{4.3}$$

Since

$$\overline{N} = \frac{N}{N_{max}} \tag{4.4}$$

then

$$\overline{N}^2 A^2 = \frac{-\beta g_o^2}{4\Omega^2 \alpha^2} \tag{4.5}$$

where here N is the Brunt-Väisälä frequency. The solution of the first order problem results in $\sigma_1 = 0$ as noted by both Friedlander (1987) and Rochester (1989). The second order perturbation is found directly from Rochester (1989) equation (81), with $\rho_o^\dagger$ replaced by ρ_o:

$$\int_v \rho_o \nabla\chi_o^* \cdot \Gamma_2 \cdot \nabla\chi_o \, dv = 0 \tag{4.6}$$

where

$$\Gamma_2 = 2\sigma_2\sigma_0 1 + \frac{\bar{N}^2}{\sigma_0^2(\sigma_0^2-1)} C_0^* C_0 + i\sigma_2 e_3 \times 1. \tag{4.7}$$

Substituting (4.7) into (4.6) results in 3 terms. The first of these is given by:

$$2\sigma_0\sigma_2 \int_v \rho_0 |\nabla\chi_0|^2 \, dv \tag{4.8}$$

and the second term gives

$$\frac{\bar{N}^2}{\sigma_0^2(\sigma_0^2-1)} \int_v \rho_0 |C_0 \cdot \nabla\chi_0|^2 \, dv. \tag{4.9}$$

The third term initially gives

$$i\sigma_2 \int_v \rho_0 \nabla\chi_0^* \cdot (e_3 \times \nabla\chi_0) \, dv \tag{4.10}$$

which can be expanded as:

$$i\sigma_2 \int_v \rho_0 \nabla \cdot (\nabla\chi_0^* \times \chi_0 e_3) \, dv \tag{4.11}$$

since

$$\nabla \times \nabla \chi_o^* = 0. \tag{4.12}$$

This can be rewritten as:

$$i\sigma_2 \int_v (\nabla \cdot (\rho_o \nabla \chi_o^* \times \chi_o e_3) - \nabla \rho_o \cdot (\nabla \chi_o^* \times \chi_o e_3)) \, dv \tag{4.13}$$

and the divergence theorem is applied to yield:

$$i\sigma_2 \int_s \rho_o \chi_o n \cdot (\nabla \chi_o^* \times e_3) \, ds - i\sigma_2 \int_v \chi_o \nabla \rho_o \cdot (\nabla \chi_o^* \times e_3) \, dv \tag{4.14}$$

which, when rearranged is:

$$i\sigma_2 \int_s \rho_o \chi_o \nabla \chi_o^* \cdot (e_3 \times n) \, ds - i\sigma_2 \int_v \chi_o \nabla \chi_o^* \cdot (e_3 \times \nabla \rho_o) \, dv. \tag{4.15}$$

The cross products in these two terms yield components of ϕ only. The dot product is then taken with the ϕ component of $\nabla \chi_o$ and results in:

$$i\sigma_2 \int_s \rho_0 \chi_0 \left[\frac{-im\chi_0^*}{R}\right] \nabla \cdot (e_3 \times n)\, ds$$
$$- i\sigma_2 \int_v \chi_0 \left[\frac{-im\chi_0^*}{R}\right] \nabla \cdot \left(e_3 \times \frac{\rho_0 g_0}{\alpha^2}\right) dv. \qquad (4.16)$$

In this instance however, $m = 0$, thus all the elements of the third term can be dropped from the formulation and the equation for the second order perturbation consisting solely of the first two terms is:

$$2\sigma_0\sigma_2 \int_v \rho_0 |\nabla\chi_0|^2\, dv + \frac{1}{\sigma_0^2(\sigma_0^2 - 1)} \int_v \bar{N}^2 \rho_0 |C_0 \cdot \nabla\chi_0|^2\, dv = 0 \qquad (4.17)$$

which when rearranged gives this expression for σ_2:

$$\sigma_2 = \frac{-\frac{1}{\sigma_0^2(\sigma_0^2-1)} \int_v \frac{-\beta g_0^2}{4\Omega^2 a^2} \rho_0 |C_0 \cdot \nabla\chi_0|^2\, dv}{2\sigma_0 \int_v \rho_0 |\nabla\chi_0|^2\, dv} \qquad (4.18)$$

where ß is a constant. When expression (4.16) is included in the denominator, (4.18) becomes the corrected equation (82) in Rochester (1989).

In order to evaluate equation (4.18) the denominator is considered first as it is a simpler expression than the numerator. The denominator integral can be expanded as:

$$2\sigma_o \int_v \rho_o(r) \left[\left(\frac{\partial \chi}{\partial R}\right)^2 + \left(\frac{\partial \chi}{\partial Z}\right)^2\right] dv \qquad (4.19)$$

where χ is still given by equation (2.28) in Chapter 2 with $N = 5$. The appropriate expressions for derivatives of χ are substituted into the integral above and then the coordinates are changed to spherical coordinates to be compatible with the density expression. The density profile is that given by Dziewonski and Anderson (1981) for the outer core, as used in Chapter 3 equation (3.1) and is already in spherical coordinates. The double to single index as given in the first element of equation (2.41) is still used.

With the whole denominator from equation (4.18) in spherical coordinates, the integral can be evaluated. Then equation (4.19) becomes:

$$4\sigma_o \sum_{i=0}^{N} \sum_{j=\delta_{i0}}^{N-i} \sum_{k=0}^{N} \sum_{l=\delta_{i0}}^{N-k} c_{ij} c_{kl} \left[\sum_{m=1}^{4} \frac{\rho_m}{2(i+j+k+l)+m}\right] \cdot \left[\frac{ik\,\Gamma(j+l+\frac{1}{2})\,\Gamma(i+k) + jl\,\Gamma(j+l-\frac{1}{2})\,\Gamma(i+k+1)}{\Gamma(i+j+k+l+\frac{1}{2})}\right] \qquad (4.20)$$

where $\Gamma(x)$ is the gamma function resulting from the evaluation of the z integrals. This expression for the denominator can now be easily programmed.

The more complicated numerator will now be considered in several segments. The first expression to be considered is:

$$|\, C_o \cdot \nabla \chi_o \,|^2 \tag{4.21}$$

where

$$C_o = \frac{[-\sigma_o^2 g_o + (e_3 \cdot g_o)\, e_3 + i\sigma_o\, e_3 \times g_o]}{g_o} \tag{4.22}$$

and χ is defined as above. Expanded in a mix of cylindrical and spherical coordinates, $C_o \cdot \nabla \chi_o$ gives:

$$C_o \cdot \nabla \chi_o = -\sigma_o^2 \sin\theta \frac{\partial \chi}{\partial R} + (1 - \sigma_o^2) \cos\theta \frac{\partial \chi}{\partial Z}. \tag{4.23}$$

Substituting in for χ, converting to only spherical coordinates, squaring the expression, and evaluating the z integral yields:

$$\begin{aligned} \int_v |C_o \cdot \nabla \chi_o|^2 \, dz = 4 \sum_{i=0}^{N} \sum_{j=\delta_{i0}}^{N-i} \sum_{k=0}^{N} \sum_{l=\delta_{k0}}^{N-k} c_{ij} c_{kl} \\ \cdot \{\sigma_o^4 ik - \sigma_o^2 (1-\sigma_o^2)(il + jk) + (1-\sigma_o^2)^2 jl\} \\ \cdot \{\mu^{2(i+j+k+l-1)}\} \cdot \{\frac{\Gamma(j+l+\frac{1}{2})\,\Gamma(i+k+1)}{2\,\Gamma(i+j+k+l+\frac{3}{2})}\} \end{aligned} \tag{4.24}$$

where the z integral has been evaluated as this is the only expression containing z terms. The remaining u integral for the numerator can now be written as:

$$\int \frac{\rho_0 g_0}{\alpha^2} u^{2(i+j+k+l-1)} u^2 \, du. \tag{4.25}$$

The expression for ρ_0 is that given earlier as equation (3.1). In the spirit of performing analytic integrations as far as possible, I shall not evaluate equation (4.25) numerically but instead develop polynomial expressions for g_0 and $1/\alpha^2$, and will rely on MACSYMA to evaluate complicated expressions. First an expression for g_0^2 will be developed using the definition for g_0:

$$g_0 = \frac{4\pi G b}{u^2} \int_0^u \rho_0(u) u^2 \, du \tag{4.26}$$

where b is the core-mantle boundary radius making u dimensionless. Although (4.26) ignores the rotational contribution to gravity, the error involved is of the order 0.4%. Substituting the polynomial expression for ρ_0 in equation (4.26) and integrating yields:

$$g_0 = 4\pi G b \sum_{m=1}^{4} \frac{\rho_m u^m}{m+2} \tag{4.27}$$

which can then be squared.

To develop a simple polynomial expression for $1/\alpha^2$, the expression from Dziewonski and Anderson (1981) for α^2 in the outer core is used to find values of $1/\alpha^2$, where the lower range of the radius has been extended to 0. These values are then used to develop the polynomial expression for $1/\alpha^2$ by using the IMSL subroutine DRCURV. This routine fits a polynomial curve to supplied data using the least squares method. A seventh degree polynomial was developed, which can be expressed as:

$$\frac{1}{\alpha^2} = \frac{1}{[\alpha(0)]^2} \sum_{s=1}^{8} a_s u^{s-1} \qquad (4.28)$$

and that matches the original values to four significant figures of precision. The coefficients of this polynomial are given in Table 4.

Table 4. Coefficients of a Seventh Degree Polynomial that Approximates the Value of $1/\sigma^2$.

Coefficient	Value
α_1	8.1914×10^{-3}
α_2	3.2968×10^{-3}
α_3	-1.6245×10^{-3}
α_4	5.5783×10^{-3}
α_5	-9.3306×10^{-3}
α_6	1.9870×10^{-2}
α_7	-1.8170×10^{-2}
α_8	7.5631×10^{-3}

The u integral, equation (4.25) is now a very extensive expression given by:

$$(4\pi Gb)^2 \int \{\sum_{m=1}^{4} \rho_m u^{m-1}\}\{\sum_{r=1}^{4} \frac{\rho_m u^m}{m+2}\}\{\frac{1}{[\alpha(0)]^2}\sum_{s=1}^{8} \alpha_s u^{s-1}\} \{u^{2[i+j+k+l-1]}\} r^2 dr. \tag{4.29}$$

In order to reduce the risk of algebraic error, this expression was expanded and integrated using MACSYMA, an algebraic manipulation program. The resulting coefficients of u from the integration were exported from MACSYMA in FORTRAN format to be incorporated into the perturbation program.

The final expansion for the numerator is then:

$$\frac{8\beta\pi^2 G^2 b^2}{\Omega^2 \sigma_o^2(\sigma_o^2-1)} \sum_{i=0}^{N} \sum_{j=\delta_{i0}}^{N-i} \sum_{k=0}^{N} \sum_{l=\delta_{k0}}^{N-k} c_{ij} c_{kl} \{\sigma_o^4 ik - \sigma_o^2(1-\sigma_o^2)(il+jk) + (1-\sigma_o^2)^2 jl\}$$
$$\{\frac{\Gamma(j+l+\frac{1}{2})\Gamma(i+j+1)}{\Gamma(i+j+k+l+\frac{3}{2})}\}\{\sum_{m=2}^{18} \frac{cu_m}{2(i+j+k+l)+m+1}\} \tag{4.30}$$

where the cu_m's are complicated expressions involving α_s and ρ_m resulting from the u integration that was evaluated using MACSYMA. These expressions can be found in Appendix E in the program Perturbation.f. By combining expressions (4.20) for the denominator and (4.30) for the numerator, a final expression for the perturbation, σ_2, is obtained. These values are then added to the reference state eigenvalues to give the perturbed eigenfrequencies. This can now be easily

programmed using the eigenvalues, σ_o, and associated eigenvector elements, c_{ij}, from the reference state developed in Chapter 3. The program used, Perturbation.f, is given in Appendix E. The results from this program for various values of ß are given in Table 5 below.

4.3 Discussion

The perturbation method clearly fails for the larger values of ß. In fact, it is questionable even at low frequencies where ß = -0.0005. For even smaller values of ß, the spectrum varies only marginally from that considered previously, unfortunately values of ß this small are uninteresting. For the perturbation treatment to be valid A, as given in equation (4.2), must be $<< 1$. The larger values of ß do not ensure this (A is then ~ .15, when ß = -0.0005), so the perturbation method is inappropriate for them. However, the eigenvalues obtained for ß = -0.0001 are valid, as they ensure $A << 1$. In order to determine how good the first order perturbation calculations could be, a second order perturbation analysis would be necessary. However, a comparison of these perturbation results for ß = -0.0001 and results from Chapter 5 using a Galerkin method of solution demonstrate that the perturbation results are valid for this small value of the stability parameter.

Table 5. Comparison of Reference State Eigenvalues, σ_o, and Eigenvalues, σ, Found From the Perturbation Method.

ß	σ_o	σ
-0.00001	0.2976	0.2988
	0.3653	0.3653
	0.4714	0.4714
	0.5660	0.5661
	0.6572	0.6572
	0.6779	0.6779
	0.7847	0.7848
	0.8306	0.8307
	0.8998	0.9002
	0.9304	0.9346
-0.0001	0.2976	0.3090
	0.3653	0.3697
	0.4714	0.4738
	0.5660	0.5671
	0.6572	0.6580
	0.6779	0.6786
	0.7847	0.7856
	0.8306	0.8316
	0.8998	0.9043
	0.9340	0.9403

Table 5. Continued.

ß	σ_o	σ
-0.0005	0.2976	0.3546
	0.3653	0.3871
	0.4714	0.4833
	0.5660	0.5716
	0.6572	0.6611
	0.6779	0.6817
	0.7847	0.7891
	0.8306	0.8359
	0.8998	0.9225
	0.9340	0.9655
-0.001	0.2976	0.4116
	0.3653	0.4088
	0.4714	0.4952
	0.5660	0.5772
	0.6572	0.6650
	0.6779	0.6855
	0.7847	0.7934
	0.8306	0.8413
	0.8998	0.9451
	0.9340	0.9970

Table 5. Continued.

ß	σ_o	σ
-0.003	0.2976	0.6396
	0.3653	0.4958
	0.4714	0.5427
	0.5660	0.5997
	0.6572	0.6806
	0.6779	0.7008
	0.7847	0.8109
	0.8306	0.8627
	0.8998	1.0358
	0.9340	1.1230

CHAPTER 5

5.1 Introduction

In this chapter the Galerkin method is used to solve the SSWE. A density profile is developed that ensures a neutral stability parameter throughout the range of the fluid core. This density profile is utilized in the Galerkin solution to the SSWE. The Galerkin method is used to reproduce the results of Chapters 3 and 4. The method is then used to produce results for long-period axisymmetric oscillations when ß is nonzero, i.e. small and negative.

5.2 Galerkin Method

The Galerkin method of solution is an approximate method for solving differential equations. We seek a solution of the partial differential equation:

$$\mathcal{L}\chi = 0 \tag{5.1}$$

where $\mathcal{L}$ is a linear operator. The trial function χ can still be written as equation (2.28) found in Chapter 2 as was used for the variational principle. The ϕ_p's must be members of a linearly independent set and have the orthogonality properties described below. Recognizing that the trial solution does not satisfy equation (5.1) exactly, the Galerkin procedure is to achieve this as nearly

as possible by requiring that:

$$\mathcal{L}\left(\sum_p c_p \phi_p\right) \tag{5.2}$$

be orthogonal to each ϕ_q over the domain of validity of equation (5.1). If the trial functions do not directly satisfy the boundary conditions they can be included using Lagrange multipliers; then the orthogonality requirement, in this case, can be written as:

$$\sum_p c_p \int_v \phi_q^* \mathcal{L} \phi_p \, dv + \sum_p c_p \int_s \psi_q^* n \cdot u \, ds = 0 \tag{5.3}$$

where the ψ_q's are the Lagrange multipliers.

This leads to the following set of equations, for determining the corresponding c_q:

$$\sum_p G_{qp} c_p = 0 \tag{5.4}$$

where

$$G_{qp} = \int_v \phi_q^* \mathcal{L} \phi_p \, dv \tag{5.5}$$

The eigenvalues, which are dispersed throughout the operator $\mathcal{L}$, can be solved for if the determinant of G_{β} equals zero. Unfortunately, this matrix will not in general be symmetric. This is the theory for the Galerkin method which will be applied to the SSWE.

5.3 Application to the SSWE

When the functional for the SSWE is written in its full form with no approximations as

$$F = \int_{v} \rho_0^t \nabla\chi^* \cdot \Gamma \cdot \nabla\chi \, dv \tag{5.6}$$

where Γ and its components are defined by equations (1.27 - 1.30) in Chapter 1, there is a potential problem with solving the integral. When $B = 0$, the integral becomes improper and the SSWE,

$$\nabla \cdot (\rho_0^t \Gamma \cdot \nabla\chi) = 0 \tag{5.7}$$

becomes singular. As we are only concerned with negative β values this restriction applies to the following frequency ranges:

i) in a weakly stably stratified region ($0 < N \leq 1$) when $0 < \sigma^2 \leq (N^2)_{max}$

ii) in a strongly stably stratified region ($N > 1$) when $0 < \sigma^2 < 1$ (Wu & Rochester 1990).

Previously this has not been a problem, as the term involving B has disappeared for one reason

or another. In Chapter 2, the density is a constant and the compressional wave speed is taken as infinite, thus eliminating the term containing B from consideration. This formulation produces solutions for pure inertial waves. The inclusion of a neutral radial density profile in Chapter 3 does not require the consideration of the B term, nor does the perturbation performed in Chapter 4. The inclusion of the neutral density profile did not introduce new frequencies, it only shifted those found in Chapter 2. The perturbation method similarly only caused a shift of the results of Chapter 3. However, because both B and C contain σ, the inclusion of this term will introduce new frequencies in addition to altering those already found.

In order to alleviate the difficulties caused by the possibility of the B term equalling zero, the SSWE is rewritten and the Galerkin method is used to solve it. The following definitions are made to facilitate rewriting the SSWE:

$$D = \frac{\beta B}{\alpha^2} = \omega^2(\sigma^2 - 1) + \frac{\beta}{\alpha^2} E \tag{5.8}$$

$$F = -\sigma^2 g_0 + (e_3 \cdot g_0) e_3 \tag{5.9}$$

and

$$E = \sigma^2 g_0^2 - (e_3 \cdot g_0)^2 \tag{5.10}$$

Making these substitutions and multiplying through by D^2, the SSWE (5.7) can be written as:

$$D^2\,\nabla\cdot(\rho_o^{\prime}\,\Gamma_\rho\cdot\nabla\chi) - \frac{\beta D}{\alpha^2}\,\nabla\cdot(\rho_o^{\prime}\,C^{*}C\cdot\nabla\chi)$$
$$= \omega^2(\sigma^2-1)\,\rho_o^{\prime}\,C\cdot\nabla\chi\;\;F\cdot\nabla\left(\frac{\beta}{\alpha^2}\right) - \left(\frac{\beta}{\alpha^2}\right)^2\rho_o^{\prime}\,C\cdot\nabla\chi\;\;F\cdot\nabla E \qquad (5.11)$$

where Γ_ρ as defined in equation (2.12) has been used. As was done in Chapter 4, this formulation of the SSWE can be perturbed. However, in this case the value of B can equal zero and there is no constraint on the frequency range where the perturbation method is valid. The perturbation of order β/α^2 on the rewritten SSWE, equation (5.11) above, results in:

$$\sigma_2 = \frac{\frac{1}{4\Omega^2 e_o^2(e_o^2-1)}\int_v \rho_o^{\prime}\,g_o^{\,2}\,\frac{\beta}{\alpha^2}\,|C\cdot\nabla\chi_o|^2\,dv}{2\,\sigma_o\int_v \rho_o^{\prime}\,|\nabla\chi_o|^2\,dv + i\int_v \rho_o^{\prime}\,\nabla\chi_o^{*}\cdot(e_3\cdot\nabla\chi_o)\,dv} \qquad (5.12)$$

which is the same result as found in Chapter 4. Thus the values found in Chapter 4 do not need to be altered to consider the case when $B = 0$.

Attention can now be focused on the expression for the modified density profile, $\rho_o^{\prime}$. This was introduced by Friedlander (1988) and expressed by Rochester (1989) as:

$$\rho_o^{\prime} = \rho_o\,e^{-\int\frac{b}{a^2}\,dW_o}$$
$$W_o = V_o - \frac{1}{2}\,|\Omega\times r|^2 \qquad (5.13)$$

where W_o is the gravity potential. This can be used to rewrite equation (5.11) since:

$$\nabla \rho_o^t = \frac{g_o}{a^2} \rho_o^t \tag{5.14}$$

Then the SSWE becomes:

$$\begin{aligned} & D^2 \nabla \cdot (\Gamma_p \cdot \nabla \chi) + D^2 \frac{g_o}{a^2} \cdot \Gamma_p \cdot \nabla \chi - \frac{\beta D}{a^2} \nabla \cdot (C^* C \cdot \nabla \chi) \\ & - \frac{\beta D}{a^2} \frac{g_o \cdot C^*}{a^2} C \cdot \nabla \chi = D C \cdot \nabla \chi \; F \cdot \nabla (\frac{\beta}{a^2}) - \frac{\beta}{a^2} C \cdot \nabla \chi \; F \cdot \nabla (\frac{\beta g}{a^2}) \end{aligned} \tag{5.15}$$

The Galerkin method, using the orthogonality requirement expressed in equation (5.3) can now be applied to this formulation of the SSWE resulting in:

$$\begin{aligned} & \sum_p c_p \int_v \rho_o [D \nabla \phi_q^* \cdot (D \Gamma_p - \frac{\beta}{a^2} C^* C) \cdot \nabla \phi_p + 2 \frac{\beta}{a^2} \phi_q^* \nabla E \cdot (D \Gamma_p - \frac{\beta}{a^2} C^* C) \cdot \nabla \phi_p \\ & + 2 \omega^2 (\sigma^2 - 1) \frac{\beta}{a^2} \nabla \cdot (\phi_q^* \frac{g}{g_o^2} C \cdot \nabla \phi_p g_o) + \omega^2 (\sigma^2 - 1) \frac{\beta}{a^2} (D + 2(1 - \beta) \frac{g}{a^2}) \phi_q^* C \cdot \nabla \phi_p] \, dv \\ & - \omega^2 (\sigma^2 - 1) \int_s (\psi_q + [\omega^2 (\sigma^2 - 1) - \frac{\beta g}{a^2}] \phi_q) \frac{C \cdot \nabla \phi_p}{g_o} \, ds = 0 \end{aligned} \tag{5.16}$$

where u in the boundary condition,

$$n \cdot u = 0 \qquad on \; S \tag{5.17}$$

has been rewritten using the definitions in equations (5.8 - 5.9) and ψ_q are Lagrange multipliers used to include the boundary condition. The other terms in the surface integral come from the rearrangement of the expression using the divergence theorem. This surface integral can be eliminated by choosing the ψ_q such that:

$$\psi_q + [\omega^2(\sigma^2 - 1) - \frac{\beta E}{\alpha^2}]\phi_q = 0 \qquad (5.18)$$

Thus, in a sense $C \cdot \nabla\chi = 0$ is a natural boundary condition for the Galerkin procedure, in the sense that the trial functions used to implement the latter need not themselves be made to satisfy the boundary condition a priori.

When $\beta = 0$, the Galerkin equations reduce to a slightly rearranged Poincaré problem as was considered earlier in Chapter 2:

$$\sum_p c_p \int_v \rho_0 [D\nabla\phi_q^* \cdot (D\Gamma_p \cdot \nabla\phi_p)]\, dv = 0 \qquad (5.19)$$

where, when $\beta = 0$, $D = \omega^2(\sigma^2 - 1)$ is just a constant. The additional terms in equation (5.16) that occur when β is slightly negative account for additional frequencies (gravity waves) and the modification of existing ones due to the presence of negative buoyancy ($\beta < 0$). To solve for the eigenvalues, equation (5.16) is written as:

$$\sum_p G_{qp} c_p = 0 \qquad (5.20)$$

and the eigenvalues are found from:

$$\det G = 0. \tag{5.21}$$

Unfortunately, G is not a symmetric matrix. At this point the expressions in G_{ij} can be expanded and programmed for solution by the computer. But first a closer look is needed at the density profile to be used.

5.4 Density Profile

As was discussed in Chapter 3, the density profile previously used from PREM does not lead to a core exactly neutrally stratified throughout. This discrepancy will now be greatly reduced by the development of a density profile that ensures that the stability parameter, β, is less than 1 part in 10^{-5} throughout the entire spherical model core range. Two ways of accomplishing this will be briefly discussed here: a least squares approach and an orthogonality relationship.

5.4.1 Least Squares

The least squares approach requires that the difference between ß, as defined by the density gradient formula (1.15), and the prescribed value $ß_p$ is minimized over the core in a least squares

sense, i.e. we minimize the expression:

$$F = \int_0^1 [1 - \beta_p + (\frac{\alpha^2}{\rho_o r_o b}) \frac{d\rho_o}{du}]^2 u^2 \, du \qquad (5.22)$$

where the normalized radius defined $u = r/b$ has been used.

The expression for α^2 from the PREM model can be used if it is assumed valid when extended to the centre of the model as it is a smooth function in the inner core range. The density is considered to be a polynomial that can be expanded as:

$$\rho_o = \sum_{m=1}^{M} \rho_m u^{m-1} \qquad (5.23)$$

and the value of ρ_1 will be constrained by the mass enclosed within the sphere being considered. The fact that gravity equals zero at the centre of the sphere requires that the value of ρ_2 equal zero. In order for F to be a minimum,

$$\frac{\partial F}{\partial \rho_i} = 0 \qquad for\ i = 3, N. \qquad (5.24)$$

These equations then form the basis for an iterative process. The starting values for the polynomial of ρ_o are taken from PREM with $\rho_2 = 0$. At each stage of the iteration the values of the coefficients ρ_m are recalculated using values of ρ_o g_o obtained from the ρ_m of the previous stage. If β becomes small enough or values of the ρ_m's no longer change, the iterative process is halted. The coefficient values for a prescribed value $\beta_p = 0$ using the least squares method are found in Table 6, and the values of β for various locations in the model sphere are in Table 7.

Table 6. Coefficients ρ_m of Neutral Density Profile Fitted by (a) Least Squares and (b) Orthogonality Relation.

Coefficient index m	(a)	(b)
1	12.365415	12.365415
2	0.0	0.0
3	-2.1355473	-2.1357751
4	-0.5724153	-0.5687766
5	0.4755506	0.4511777
6	-0.2017774	-0.1185908
7	0.0276528	-0.1303701
8	-0.2573055	-0.0878988
9	0.2394251	0.1435077
10	-0.0905760	-0.0682689

Table 7. Comparison of Actual Values of ß for Neutral Density Profile Fitted by (a) Least Squares and (b) Orthogonality Relation.

Normalized Radius	(a)	(b)
0.0	0.10418×10^{-3}	-0.17078×10^{-5}
0.05	0.14402×10^{-4}	-0.94609×10^{-5}
0.1	-0.64605×10^{-5}	-0.72470×10^{-5}
0.15	-0.41660×10^{-5}	-0.48525×10^{-5}
0.2	0.10167×10^{-5}	-0.45842×10^{-5}
0.25	0.28317×10^{-5}	-0.55451×10^{-5}
0.3	0.15468×10^{-5}	-0.61821×10^{-5}
0.35	-0.59195×10^{-6}	-0.55616×10^{-5}
0.4	-0.17503×10^{-5}	-0.37353×10^{-5}
0.45	-0.13720×10^{-5}	-0.15112×10^{-5}
0.5	-0.53384×10^{-7}	0.85017×10^{-7}
0.55	0.11062×10^{-5}	0.39176×10^{-6}
0.6	0.12656×10^{-5}	-0.49905×10^{-6}
0.65	0.36728×10^{-6}	-0.17233×10^{-5}
0.7	-0.81458×10^{-6}	-0.20767×10^{-5}
0.75	-0.11879×10^{-5}	-0.81707×10^{-6}
0.8	-0.25509×10^{-6}	0.14805×10^{-5}
0.85	0.10813×10^{-5}	0.26582×10^{-5}
0.9	0.78620×10^{-6}	0.37067×10^{-6}
0.95	-0.14095×10^{-5}	-0.32568×10^{-5}
1.0	0.37869×10^{-5}	0.85257×10^{-5}

5.4.2 Orthogonality Relationship

The development of an orthogonality relationship is similar to the least squares method. Now, however, we require that the difference between ß, as defined by the density gradient formula (1.15), and the prescribed value β_p be orthogonal to the lowest M-2 powers of the normalized radius, i.e. that

$$\int_0^1 [1 - \beta_p + (\frac{e^2}{\rho_o \varepsilon_o b}) \frac{d\rho_o}{du}] \, u^{m-1} \, du = 0 \qquad (5.25)$$

for $m = 1,\ldots M$-2. Again ρ_1 is known from the total mass constraint and $\rho_2 = 0$. The iterative process is again used and the results for this method are shown in Tables 6 and 7. The coefficient values for the orthogonality relationship give a density profile that is neutral to within 1 part in 10^5, which is essentially neutral throughout the entire core range. Thus this is the density profile to be used in the Galerkin method when equation (5.16) is invoked, in the case of a neutral ρ_o. This density profile shall be referred to as ρ_o = NEUT.

5.5 Expansion of Equation (5.16)

Equation (5.16) can now be expanded and the elements of the matrix G_φ determined. At this point only those terms without azimuthal (ϕ) dependence are considered, as has been done in the previous Chapters. After a substantial amount of algebra G_φ can be rewritten as:

$$G_{qp} = 4\sigma^2(\sigma^2 - 1)\, H_{qp} \tag{5.26}$$

where

$$\begin{aligned} H_{qp} = {} & \sigma^6 (A_{qp} + \gamma\beta f B_{qp}) + \sigma^4 [C_{qp} + \gamma\beta f D_{qp} + \beta f E_{qp} + \gamma\beta f^2 (2-\beta) G_{qp}] \\ & + \sigma^2 [P_{qp} + \gamma\beta f Q_{qp} + \beta f R_{qp} + \gamma\beta f^2 (2-\beta) T_{qp} + \beta^2 f^2 U_{qp}] \\ & + [\beta f V_{qp} + \gamma\beta f^2 (2-\beta) X_{qp} + \beta^2 f^2 Y_{qp}] \end{aligned} \tag{5.27}$$

and where

$$\gamma = \frac{\Omega^2}{2\pi G \rho_o(0)} \tag{5.28}$$

and

$$f = \frac{4\pi^2 G^2 [\rho_o(0)]^2 b^2}{\Omega^2 [\alpha(0)]^2} \tag{5.29}$$

G is the gravitational constant, $1/\alpha^2$ is defined by equation (4.28), $\alpha(0)$ is the compressional p-wave speed evaluated at $u = 0$, similarly $\rho(0)$ is the density evaluated at $u = 0$. For brevity in defining the matrix elements A_{qp} ... Y_{qp} we introduce $I(i,j)$:

$$I(i,j) = \int_0^1 z^{2i}(1-z^2)^j\,dz = \frac{\Gamma(i+\frac{1}{2})\,\Gamma(j+1)}{2\,\Gamma(i+j+\frac{3}{2})} \tag{5.30}$$

where p, q are related to i, j, k, and l by (2.41). Then the individual matrices comprising H_{qp} are as follows:

$$A_{qp} = -\sum_{m=1}^{M} \frac{\rho_m}{2(i+j+k+l)+m}\,[ik\,I(i+k-1,j+l) + jl\,I(i+k,j+l-1)] \tag{5.31}$$

$$B_{qp} = -\sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{s=1}^{8} \frac{\rho_m \rho_n \alpha_s\ (i+j)}{(n+2)[2(i+j+k+l)+m+n+s]}\,I(i+k,j+l) \tag{5.32}$$

$$C_{qp} = \sum_{m=1}^{M} \frac{\rho_m}{2(i+j+k+l)+m}\,[ik\,I(i+k-1,j+l) + 2jl\,I(i+k,j+l-1)] \tag{5.33}$$

$$D_{qp} = \sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{s=1}^{8} \frac{\rho_m \rho_n \alpha_s\ (i+2j)}{(n+2)[2(i+j+k+l)+m+n+s]}\,I(i+k,j+l) \tag{5.34}$$

$$E_{qp} = \sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{8} \frac{\rho_m \rho_n \rho_r \alpha_s}{(n+2)(r+2)[2(i+j+k+l)+m+n+r+s-1]}$$
$$\{-i(2i+3k-1)\,I(i+k-1,j+l+1) + i(2i+k-1)\,I(i+k-1,j+l)$$
$$+[3l(i+j)+j(4i+2j+3k-1)+2i+2j]\,I(i+k,j+l) - 2jl\,I(i+k,j+l-1)\} \tag{5.35}$$

$$G_{qp} = -\sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{8}\sum_{t=1}^{M}\sum_{w=1}^{8} \frac{\rho_m \rho_n \rho_r \alpha_s \rho_t \alpha_w}{(n+2)(r+2)(t+2)} \left\{ \frac{(i+j)\, I(i+k,j+l)}{2(i+j+k+l)+m+n+r+s+t+w-1} \right\} \tag{5.36}$$

$$P_{qp} = -\sum_{m=1}^{M} \frac{\rho_m}{2(i+j+k+l)+m} [jl\, I(i+k,j+l-1)] \tag{5.37}$$

$$Q_{qp} = -\sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{s=1}^{8} \frac{\rho_m \rho_n \alpha_s\ j}{(n+2)[2(i+j+k+l)+m+n+s]}\, I(i+k,j+l) \tag{5.38}$$

$$\begin{aligned} R_{qp} = \sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{8} & \frac{\rho_m \rho_n \rho_r \alpha_s}{(n+2)(r+2)[2(i+j+k+l)+m+n+r+s-1]} \\ & \{i(2i+2k-1)\, I(i+k-1,j+l+2) - i(2i-1)\, I(i+k-1,j+l-1) \\ & -[j(2j+2k+2l-1)+2i(2j+l)+4(i+j)]\, I(i+k,j+l+1) \\ & -[l(i+2j)+j(2i+2j+3k-1)]\, I(i+k,j+l)+2jl\, I(i+k,j+l-1)\} \end{aligned} \tag{5.39}$$

$$T_{qp} = \sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{8}\sum_{t=1}^{M}\sum_{w=1}^{8} \frac{\rho_m \rho_n \rho_r \alpha_s \rho_t \alpha_w}{(n+2)(r+2)(t+2)} \left\{ \frac{(i+j)\, I(i+k,j+l+1) + j\, I(i+k,j+l)}{2(i+j+k+l)+m+n+r+s+t+w-1} \right\} \tag{5.40}$$

$$U_{qp} = \sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{8}\sum_{t=1}^{M}\sum_{v=1}^{M}\sum_{w=1}^{8} \frac{\rho_m \rho_n \rho_r \alpha_s \rho_t \rho_v \alpha_w}{(n+2)(r+2)(t+2)(v+2)}$$
$$\left\{ \frac{-ik\, I(i+k-1,j+l+1) + (jk+il+jl)\, I(i+k,j+l) - jl\, I(i+k,j+l-1)}{2(i+j+k+l)+m+n+r+s+t+v+w-2} \right\} \quad (5.41)$$

$$V_{qp} = \sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{8} \frac{\rho_m \rho_n \rho_r \alpha_s}{(n+2)(r+2)[2(i+j+k+l)+m+n+r+s-1]}$$
$$\{[j(2i+2j+2k+2l-1)+4j]\, I(i+k,j+l+1) - (jl+2j)\, I(i+k,j+l)\} \quad (5.42)$$

$$Y_{qp} = -\sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{8}\sum_{t=1}^{M}\sum_{w=1}^{8} \frac{\rho_m \rho_n \rho_r \alpha_s \rho_t \alpha_w}{(n+2)(r+2)(t+2)}$$
$$\left\{ \frac{j\, I(i+k,j+l+1)}{2(i+j+k+l)+m+n+r+s+t+w-1} \right\} \quad (5.43)$$

$$Y_{qp} = \sum_{m=1}^{M}\sum_{n=1}^{M}\sum_{r=1}^{M}\sum_{s=1}^{8}\sum_{t=1}^{M}\sum_{v=1}^{M}\sum_{w=1}^{8} \frac{\rho_m \rho_n \rho_r \alpha_s \rho_t \rho_v \alpha_w}{(n+2)(r+2)(t+2)(v+2)}$$
$$\left\{ \frac{ik\, I(i+k-1,j+l+2) - [jl+il+jk+2(i+j)]\, I(i+k,j+l+1) + (jl+2j)\, I(i+k,j+l)}{2(i+j+k+l)+m+n+r+s+t+v+w-2} \right\} \quad (5.44)$$

The summation values deserve some attention. The sums over m, n, r, t, v all go from 1 to M, where M is either 4 or 10, depending on which density profile is selected, either ρ_e = PREM with values on page 40, or ρ_e = NEUT with values from Table 6. The sums over s and w all go from 1-8, as this is the polynomial approximation for $1/\alpha^2$, and is given by equation (4.28). The function $I(p,q)$ results from integration over the z terms, such as are detailed in Appendix

A. A value of $N=5$ has been used in the trial function (2.28) to produce the results presented here.

The expression for H_{φ} can now be programmed. To facilitate changing parameters, two programs were actually used. The first, biggerbetas.f, calculates the various elements of the matrices inside H_{φ}, and is found in Appendix F. The second, called readbb.f also in Appendix F, evaluates H_{φ} at an interval spacing of approximately .008 and determines where det H changes sign. These sign changes are then used to zero in on the eigenfrequencies. These particular parameters were chosen as the results of the previous chapters are accurately reproduced without taking exorbitant amounts (> 150 hours) of computer time. The program does take a substantial amount of time (> 6 hours) to run even with the use of a computer supporting the vector capabilities of FORTRAN, which is recommended (e.g. CONVEX C-1).

5.6 Results

Several different values of β were used, and the density polynomials, ρ_o = PREM and ρ_o = NEUT were used in programming the Galerkin solution. The various configurations are considered individually with their results.

5.6.1 $\beta = 0$, ρ_o = PREM, and ρ_o = NEUT

This comparison was made to see if the results using the Galerkin method would match the results

found in Chapter 3 for the different density profiles. The results are listed in Table 8. A comparison of values shows only one minor difference with previous results when ρ_o = NEUT. The values obtained when ρ_o = NEUT are preferable because ρ_o = PREM is not valid in a sphere, as it does not ensure that gravity is zero at the sphere's centre. However, since the values match so closely it is apparent that the approximation of using ρ_o = PREM was acceptable to use. Thus as a first approximation, the variational principle can be used with ρ_o = PREM to obtain the inertial eigenfrequencies in a fluid filled sphere. These results also demonstrate that the ρ_o = NEUT density profile does not significantly alter the response of the model core.

Table 8. Comparison of Dimensionless Eigenfrequencies Calculated Using the Variational Principle (VP) and the Galerkin Method.

σ from Variational Principle $\beta = 0$, ρ_0 = PREM	σ from Galerkin Method $\beta = 0$, ρ_0 = PREM	σ from Galerkin Method $\beta = 0$, ρ_0 = NEUT
0.2976	0.2976	0.2976
0.3653	0.3653	0.3653
0.4714	0.4714	0.4714
0.5660	0.5660	0.5660
0.6572	0.6572	0.6572
0.6779	0.6779	0.6778
0.7847	0.7847	0.7847
0.8306	0.8306	0.8306
0.8998	0.8998	0.8998
0.9340	0.9340	0.9340

5.6.2 $\beta < 0$, $|\beta| \leq 0.0001$, ρ_o = PREM

This selection of parameters was made to compare the results of the Galerkin method with those of the perturbation method. Since the Galerkin method used here will give a direct calculation of the result, an exact match of the eigenvalues from the perturbation method is not expected. The results are shown in Table 9. It should be noted that due to the presence of gravity modes, new eigenfrequencies could appear in the Galerkin calculation since β is non-zero. Presumably the gravity modes associated with these values of β have periods so long as to lie outside the frequency range examined for zeros of det H. However, as seen in Table 9, for ρ_o = PREM this is not the case. The comparison has been made only between the smallest values of β as the perturbation method is known to fail for the larger values. It should be noted that for the extremely small values of $|-\beta| = -.00001$ the resulting eigenvalues are all very close to the results for $\beta = 0$. Thus an extremely small stability parameter will be indistinguishable from $\beta = 0$ for this model of the core. It must be noted here that the density profile ρ_o = PREM is not actually valid for nonzero β, as it implies that $\beta = 0$ throughout most of the core. This density profile was used as a to provide a comparison with the previously calculated perturbation results.

Table 9. Comparison of Dimensionless Eigenfrequencies Calculated using the Perturbation and Galerkin Methods, with ρ_o = PREM.

ß = -0.00001		ß = -0.0001	
Perturbation	Galerkin	Perturbation	Galerkin
0.2988	0.2978	0.3090	0.2992
0.3658	0.3655	0.3697	0.3667
0.4717	0.4715	0.4738	0.4725
0.5661	0.5661	0.5671	0.5670
0.6573	0.6573	0.6580	0.6581
0.6779	0.6779	0.6786	0.6787
0.7848	0.7848	0.7856	0.7855
0.8307	0.8306	0.8316	0.8313
0.9002	0.8999	0.9043	0.9005
0.9346	0.9340	0.9403	0.9346

5.6.3 $\beta < 0$, $|\beta| \geq 0.0001$, ρ_o = NEUT

The first thing to notice about the results in Table 10 is the appearance of new frequencies. Previous results have all yielded only 10 eigenfrequency values within the frequency range searched for zeroes of det H. The first new frequency to appear occurs when $\beta = -0.0001$, however, the period of this extra mode is close to 12 hours indicating that it is not a newly appearing gravity mode. Most likely this is another inertial mode that was missed in the calculation of det H. This is possible due to the high density of inertial modes in the frequency domain as the dimensionless frequency approaches the value 1 from below. Thus it is possible as the modes close to the dimensionless frequency 1 are shifted due to different β values they will start to appear as distinct frequencies. Another possibility is that ρ_o = NEUT is a better representation of density by virtue of its larger number of coefficients.

The periods of the expected newly appearing gravity waves can be quickly estimated by ignoring rotation and noting that the period will be given by:

$$\frac{2\pi}{N_{max}} \tag{5.45}$$

where for uniform values of β

$$N_{max} = (-\beta)^{\frac{1}{2}} \left(\frac{g_o}{a}\right)_{max} \tag{5.46}$$

thus for a ß = -0.0001 the minimum period would be near 130 hours and would yield a σ value of .09231. But the actual minimum value should be lower than this due to the presence of rotation adding the Coriolis effect as an additional restoring force to negative buoyancy.

Another point to notice is that the negative buoyancy will add a restoring force to the inertial modes, their eigenfrequencies should increase as ß becomes more negative. Thus, in Table 10 an attempt has been made to correlate these frequencies as they increase with increasing ß. At the larger values of the eigenfrequencies this becomes difficult due to different eigenfrequencies appearing that may have been missed by previous calculation due to the density of the frequencies as we approach 1.

At some point as -ß increases the negative buoyancy and the Coriolis effect become comparable restoring forces and it becomes difficult to distinguish between the inertial and gravity modes.

It must be noted that this calculation is only a first approximation to stable core conditions as ideally a separate density profile should be developed for each β value being considered. While the density profile of the core is expected to be nearly neutral, as in PREM, the results in Table 10 indicate that small changes in the stratification lead to large changes in the eigenspectrum. This indicates how useful these calculations will be in determining the stability structure of the Earth's core.

Table 10. Tentative Comparison of Dimensionless Eigenfrequencies Calculated Using the Galerkin Method, with ρ_o = NEUT.

ß = -0.0001	ß = -0.001	ß = -0.002	ß = -0.003	ß = -0.004
		0.06376	0.07623	0.08578
	0.1734	0.1669	0.2026	0.2315
	0.2760	0.2496	0.3131	0.4480
0.2993	0.3020	0.3090	0.3470	
0.3668	0.3786		0.4227	
			0.4453	
0.4726	0.4824	0.4930	0.5019	
0.5670	0.5758	0.5843	0.5909	0.5952
0.6581	0.6654	0.6736	0.6816	0.6894
0.6787	0.6867	0.6952	0.7032	
		0.7918	0.7445	
0.7855	0.7926	0.8117	0.8165	
0.8313	0.8377	0.8449	0.8520	0.8583
			0.8782	0.8869
	0.8872	0.9126	0.9152	
0.9004	0.9061	0.9226	0.9292	
0.9346	0.9443			
	0.9611	0.9650	0.9756	0.9968
	0.9698			
0.9876	0.9832	0.9864	0.9982	0.9999
	0.9937			

CHAPTER 6

6.1 Introduction

In this thesis a variety of theoretical methods have been used to investigate the axisymmetric inertial wave spectrum, and its modification by small negative buoyancy, for a model approximately representing conditions in the Earth's liquid core. These methods, the variational principle, the perturbation, and the Galerkin methods have all proven useful in calculating results of an approximate Earth model progressing to a more complex and realistic one. However, several important aspects have been neglected; these will be detailed below, and could be the subject of further research.

6.2.1 Inner Core

The most obvious difference between the model presented here and the actual Earth is the lack of a solid inner core. This could be accounted for by using a trial function similar to that already used but including extra terms, such as:

$$\chi = \sum_{i=0}^{N} \sum_{j=\delta_m}^{N-i} c_{ij} R^{2i} Z^{2j} + \sum_{i=1}^{M} \sum_{j=1}^{M} d_{ij} \frac{Z^{2j}}{(R^2 + Z^2)^{i/2}} \qquad (6.1)$$

The second term in this expression can only be used when an inner core is present, as it will degenerate in a sphere, as R and Z both approach zero. This second term substantially increases the amount of work to be done, as the d_{ij}'s must be found independently of the c_{ij}'s. It is best to use cylindrical coordinates, as the effect of rotation appears to impose its configuration on the problem more so than the physical spherical shape of the boundaries. The functionals for the variational principle and the Galerkin method will remain the same when an inner core is included, except for the integration limits. The inner core boundary condition can also be shown to be a natural boundary condition. The presence of the inner core will significantly affect the eigenfrequencies as has been shown in experimental work by Aldridge (1967) and Aldridge (1972). Unfortunately, the inclusion of the inner core makes the problem an ill-posed one, unless viscosity is introduced as in section 6.2.6.

6.2.2 Normal Modes Carrying Linear or Angular Momentum

Both linear (translational) and angular momentum were ignored in this model. Linear momentum, expressed as:

$$\boldsymbol{P} = i\omega \int_{v} \rho_{o}\, \boldsymbol{u}\, dv \tag{6.2}$$

needs to be considered in order to filter out those frequencies due to simple translation, however it should not be a problem with the simple model presented here as the rigid fixed boundary condition prevents any transfer of linear momentum to the fluid outside, or vice versa.

Consideration of linear momentum is especially important when the inner core is included due to the possibility of Slichter modes where the solid inner core "sloshes" within the fluid outer core. These modes can be filtered out using Lagrange multipliers associated with the various components of u, i.e. by adding to the functional

$$\sum_i \lambda_i \int_v \rho_o u_i \, dv. \tag{6.3}$$

This filtering only needs to be done for the cases when $m = 0, \pm 1$, where the ϕ dependence has been expressed in the trial function as $e^{im\phi}$.

The angular momentum, which was also disregarded, is somewhat more complicated. The angular momentum can be changed in several ways: altering the flow, changing the relative angular momentum and changing the moment of inertia. This is of most concern when the boundaries are deformable and not spherical. These effects must be filtered out with Lagrange multipliers. The total angular momentum can be expressed as:

$$\Delta H = \int_m (r + u) \times \frac{d}{dt}(r + u) \, dm - \int_m r \times \frac{dr}{dt} \, dm. \tag{6.4}$$

This can be expressed as a complex expression with no time dependence as:

$$\Delta H = \int_m [u \times (\Omega \times r) + r \times (\Omega \times u) + i\omega \, r \times u] \, dm, \tag{6.5}$$

where the first two terms account for changes in the moment of inertia and the last term for changes in the flow. Alternatively, this can be written as:

$$\Delta H = h + \Omega \cdot \Delta I, \tag{6.6}$$

where the change in the inertia tensor I is given by:

$$\Delta I = \int_m [2\ r \cdot u\ 1 - (ru + ur)]\ dm. \tag{6.7}$$

The changes in the inertia tensor can be expanded as:

$$\int_s \rho_0\ n \cdot u\ [r^2\ 1 - rr]\ ds - \int_v [u \cdot \nabla \rho_0 + \rho_0 \nabla \cdot u]\ [r^2\ 1 - rr]\ dv, \tag{6.8}$$

where the first terms express displacement in the boundaries, the second term describes departures from homogeneity, and the final term depicts flow that departs from incompressible flow. The surface integral is over the inner core boundary and the core-mantle boundary. When the boundaries are rigid, and the fluid is homogeneous and incompressible, then no changes to the inertia tensor can occur and only the changes in the flow h need to be considered:

$$h = i\omega \int_m r \times u\ dm. \tag{6.9}$$

As with the linear momentum, only those cases where $m = 0, \pm 1$ need to be filtered out using Lagrange multipliers.

6.2.3 More Realistic Boundary Conditions

The boundary condition used here was a rigid spherical core-mantle boundary. For a more realistic model both the inner core and the core-mantle boundary must be deformable and nonspherical. A nonspherical boundary would allow a transfer of torque between the core and the mantle.

6.2.4 Azimuthal Modes

In this work the azimuthal modes were ignored by considering only the case when $m = 0$. If these modes are to be considered, both the trial function and the functionals used must be changed. For example, for $m \neq 0$,the functional for implementing the variational principle with rigid fixed boundaries but non-neutral stratification becomes:

$$F = \sigma^2 \int_v \rho_o' |\nabla\chi|^2 dv - \int_v \rho_o' |e_3 \cdot \nabla\chi|^2 dv - \int_v \frac{\rho_o'}{B} |C \cdot \nabla\chi|^2 dv$$
$$+ m\sigma \int_v \frac{\rho_o'}{a^2} |\chi|^2 \frac{(g_o \times e_3)\cdot \hat{\phi}}{R} dv + \rho_o'(S_-) \int_{S_-} |\chi|^2 \frac{m\sigma (e_3 \times n)\cdot \hat{\phi}}{R} ds \qquad (6.10)$$

which is essentially equation (86) from Rochester (1989).

Similarly, the Galerkin method will also include additional terms. The trial function will now be:

$$\chi = \sum_{p=1} c_p \phi_p (R, Z) e^{im\phi}. \qquad (6.11)$$

Another point is to remove the restriction placed on the trial function forcing it to satisfy:

$$\frac{\partial \chi}{\partial R} = 0 \quad on\ R = 0 \quad and \quad \frac{\partial \chi}{\partial Z} = 0 \quad on\ Z = 0, \qquad (6.12)$$

as was done in Chapter 2. Modes involving motion across the equatorial axis and plane are possible.

6.2.5 Improved Representations of Density Profile

The stability parameter used here was $\beta = 0$ or a small negative number. Positive values of β cannot be considered as gravity waves will not exist when the fluid is nowhere stably stratified. In any case, β is most likely a complicated function of radius. This aspect needs to be considered so that regions of stability can be dealt with separately from those that may be unstable.

As was noted in Chapter 5, the two density profiles used, ρ_o = PREM and ρ_o = NEUT, lead to a neutrally stratified region. As β is made negative, the density profile must be changed to ensure that the model is consistent.

6.2.6 Viscosity

In this thesis the problem of viscosity has been ignored. This has been considered by Rieutord (1987, 1990). The presence of viscosity will selectively damp the smaller wavelength modes and reduce the severity of the truncation to small values of N necessary for computational purposes. Viscosity will also help in removing the degeneracy of the ill-posedness which seems to introduce discontinuous eigenfunctions in the spherical shell geometry.

6.2.7 Density of the Eigenspectrum

According to Valette (1989) the inertia/gravity eigenspectrum for an elastic, uniformly rotating, self-gravitating body with a fluid inclusion is a continuous one. However an unpublished proof by Rochester demonstrates that the eigenspectrum should be discrete (but dense) if the density profile of the liquid core is neutral or stable everywhere. This issue needs further investigation.

6.3 Summary of Research Results

Whereas the scope of this thesis is limited, it has yielded many useful results. First of all, the variational principle used is based on a more general functional than that used by Aldridge (1972). He used a stream function formulation, which while more desirable mathematically, is limiting in the amount of physics that it elucidates, as it cannot be used when the fluid is compressible. The variational principle presented here can be used in an inhomogeneous

compressible fluid. Aldridge (1972) also only considered those modes where $m = 0$. The variational principle used in this thesis for axisymmetric modes can also be used when m is nonzero. So far as I am aware, no published work has yet reported numerical results for axisymmetric oscillations based on this variational principle.

The second original contribution of this thesis is a numerical test of the perturbation treatment of the effect, on the eigenspectrum of a neutrally stratified fluid, of very small negative ß. It appears that first-order perturbation theory is useful only for values of -ß which are so small as to be comparatively uninteresting.

Finally this thesis makes one further original contribution in the numerical application of a Galerkin procedure for studying the effects of increasingly negative ß on the eigenspectrum, namely increase in the inertial mode frequencies and the appearance of gravity modes.

REFERENCES

Aldridge, K.D. (1967) An experimental study of axisymmetric inertial oscillations of a rotating liquid sphere. *Ph.D. Dissertation*, Massachusetts Institute of Technology, 128 pp.

Aldridge, K.D. (1972) Axisymmetric inertial oscillations of a fluid in a rotating spherical shell. *Mathematika* **19**, 163-168

Aldridge, K.D. & Lumb, L.I. (1987) Inertial waves identified in the Earth's fluid outer core. *Nature* **325**, 421-423

Aldridge, K.D. & Toomre, A. (1969) Axisymmetric inertial oscillations of a fluid in a rotating spherical container. *J. Fluid Mech.* **37**, 307-323

Alterman, Z., Jarosch, H. & Pekeris, C.L. (1959) Oscillations of the earth. *Proc. R. Soc. Lond.* **A252**, 80-95

Anderson, D. L., & Dziewonski, A. M., (1984) Seismic Tomography. *Scientific American* **251**, no. 4, 58-66

Benioff, H., Gutenburg, B., & Richter, C. F. (1954) *Trans. Am. Geophys. Union* **35**, 979

Brush, S.G. (1980) Discovery of the Earth's core. *Amer. J. Phys.* **48**, 705-724

Bryan, G.H. (1889) The waves on a rotating liquid spheroid of finite elipticity. *Phil. Trans. R Soc. Lond.*, **A 180**, 187-219

Bullard, E. C., (1957) The density within the Earth, *Verh. geol. mijnb. Genoot. Ned. Kolon.*, **18**, 23-41

Bullen, K. E., (1950) An Earth model based on a compressibility-pressure hypothesis, *Mon. Not. R. astr. Soc., Geophys. Suppl.*, **6**, 50-59

Crossley, D.J. (1984) Oscillatory flows in the liquid core. *Physics of the Earth and Planet. Interiors* **36**, 1-16

Crossley, D.J. & Smylie, D.E. (1975) Electromagnetic and viscous damping of core oscillations, *Geophys. J. R astr. Soc.*, **42**, 1011-1033

Crossley, D. J., & Rochester, M. G., (1991) The subseismic approximation in core dynamics. *Geophys. J. Int.* (in press)

Dziewonski, A.M. & Anderson, D.L. (1981) Preliminary reference Earth model. *Phys. Earth Planet. Interiors* **25**, 297-356

Dziewonski, A.M. & Gilbert, F. (1972) Observations of normal modes from 84 recordings of the Alaskan earthquake of 1964 March 28, *Geophys. J. R. astr. Soc.*, **27**, 393-446

Elsasser, W. M., (1950) The Earth's interior and geomagnetism, *Revs. Mod. Phys.*, **22**, 1-35

Friedlander, S. (1985) Internal oscillations in the Earth's fluid core. *Geophys. J. R astr. Soc.* **80**, 345-361.

Friedlander, S. (1987) Inertial waves in a rotating stratified spherical shell: asymptotic solutions. *Geophys. J. R. astr. Soc.* **89**, 637-655.

Friedlander, S. (1988) Stability and waves in the Earth's fluid core. In: B. Straughn (ed.) *Proc. U.S. - Italy Conference on Energy Stability and Convection*. Longman Sci. Press Pitman Research Notes in Mathematics, **168**, 325-345.

Greenspan, H.P. (1968) *The Theory of Rotating Fluids*. Cambridge Univ. Press, London, 323 pp.

Gubbins, D., Thomson, C.J., & Whaler, K.A. (1982) Stable regions in the Earth's liquid core. *Geophys. J. R. astr. Soc.* **68**, 241-251.

Jeffreys, H., (1926) The rigidity of the Earth's central core, *Mon. Notes R. astr. Soc., Geophys. Suppl.*, **1**, 371-383

Johnson, I. M., & Smylie, D. E., (1977) A variational approach to whole-Earth dynamics, *Geophys. J. R. astr. Soc.*, **50**, 35-54

Kudlick, M.D. (1966) On the transient motions in a contained rotating fluid. *Ph.D. dissertation*, Massachusetts Institute of Technology.

Love, A. E. H., (1911) *Some problems in Geodynamics*, Cambridge University Press

Masters, G., (1979) Observational constraints on the chemical and thermal structure of the Earth's deep interior. *Geophys. J. R. astr. Soc.* **57**, 507-534.

Melchior, P. (1986) *Physics of the Earth's Core*. Pergamon Press, New York, pg 33-38.

Melchior, P. & Ducarme, B. (1986) Detection of inertial gravity oscillations in the Earth's core with a superconducting gravimeter at Brussels. *Phys. Earth Planet. Interiors* **42**, 129-134.

Pekeris, C.L. & Accad, Y. (1972) Dynamics of the liquid core of the Earth. *Phil. Trans. R. Soc. Lond.* **A273**, 237-260

Poincaré, H., (1885) Sur l'équilibre d'une masse fluide animée d'un mouvement de rotation, *Acta Mathematica*, **7**, 259-380

Rieutord, M., (1987) Linear theory of rotating fluids using spherical harmonics - I, Geophys. *Astrophys. Fluid Dyn.* **39**, 163-182

Rieutord, M., (1991) Linear theory of rotating fluids using spherical harmonics - II, *Geophys. Astrophys. Fluid Dyn.*, **59**, 185-208

Rochester, M.G. (1989) Normal modes of rotating self-gravitating compressible stratified fluid bodies: The subseismic wave equation. In: G.A.C. Graham and S.K. Malik (eds.) *Continuum Mechanics and its Applications*. Hemisphere Publ. Corp., New York, pp 797-823.

Singer, H.A. & Olson, P.L. (1984) Dynamo action in a stably stratified core, *Geophys. J. R. astr. Soc.* **78**, 371-387

Smylie, D.E. (1974) Dynamics of the Earth's outer core. *Veroffentlichungen des Zentralinstituts fur Physik der Erde, Akad. Wissensch. D.D.R.*, **30**, 91-104

Smylie, D.E., Szeto, A.M.K. & Rochester, M.G. (1984) The dynamics of the Earth's inner and outer core. *Rep. Prog. Phys.* **47**, 855-906

Smylie, D.E. & Rochester, M.G. (1981) Compressibility, core dynamics and the subseismic wave equation. *Phys. Earth Planet. Interiors* **24**, 308-319

Smylie, D.E. & Rochester, M.G. (1986a) Long period core dynamics. In: A. Cazenave (ed.) *Earth Rotation and Unsolved Problems*. NATO Adv. Studies Inst. Series C, v. 187, Reidel, Dordrecht, pp. 297-324

Smylie, D.E. & Rochester, M.G. (1986b) A variational principle for the subseismic wave equation. *Geophys. J. R. astr. Soc.* **86**, 553-561

Stacey, F.D. (1977) *Physics of the Earth*. 2nd ed. John Wiley and Sons, New York, 414pp

Stewartson, K., and Rickard, J.A. (1969) Pathological oscillation of a rotating fluid, *J. Fluid Mech.*, **35**, 759-773

Valette, B., (1989) Spectre des vibrations propres d'un corps elastique, auto-gravitant, en rotation uniforme et contenant une partie fluide, *C. R. Acad. Sci. Paris*, **309**, Ser. I, 419-422

Wu, W.J. & Rochester, M.G. (1990) Core dynamics: the two-potential description and a new variational principle. *Geophys. J. Int.* **103**, 697-706

Zürn, W., Richter, B., Rydelek, P.A., & Neuberg, J. (1987) Comment on "Detection of inertial gravity oscillations in the Earth's core with a superconducting gravimeter at Brussels." by Melchior, P. & Ducarme, B. *Phys. Earth Planet. Interiors* **49**, 176-178

APPENDIX A: Three methods for computing an integral.

The integral under consideration is:

$$\sum_{p}\sum_{q}\int_{0}^{1} z^{2p}(1-z^2)^q\,dz \tag{A.1}$$

This integral can be evaluated in three different ways. The first uses the half integer gamma function, the second employs binomial coefficients and the third utilises factorials for a solution.

1) Half Integer Gamma

This solution car be expressed as:

$$\sum_{p=0}^{P}\sum_{q=0}^{Q}\int_{0}^{1} z^{2p}(1-z^2)^q\,dz = \sum_{p=0}^{P}\sum_{q=0}^{Q}\frac{\Gamma(p+\frac{1}{2})\Gamma(q+1)}{2\Gamma(p+q+\frac{3}{2})} \tag{A.2}$$

This solution can be programmed in FORTRAN using the IMSL subroutine DGAMMA.

2) Binomial Coefficients

This solution involves the inclusion of another summation:

$$\sum_{p=0}^{P}\sum_{q=0}^{Q}\int_0^1 z^{2p}(1-z^2)^q\,dz = \sum_{p=0}^{P}\sum_{q=0}^{Q}\sum_{r=0}^{q}\frac{q!(-1)^r}{r!(q-r)!}\int_0^1 z^{(2r+2p)}\,dz$$
$$= \sum_{p=0}^{P}\sum_{q=0}^{Q}\sum_{r=0}^{q}\frac{q!(-1)^r}{r!(q-r)!}\,\frac{1}{2(r+p)+1} \qquad \text{(A.3)}$$

This can then be programmed using the IMSL routine DBINOM.

3) Factorial Solution

This solution is given by:

$$\sum_{p=0}^{P}\sum_{q=0}^{Q}\int_0^1 z^{2p}(1-z^2)^q\,dz = \sum_{p=0}^{P}\sum_{q=0}^{Q}\frac{(2p)!\,q!\,(p+q+1)!\,2^{2q+1}}{p!\,(2(p+q+1))!} \qquad \text{(A.4)}$$

This can also be programmed using the IMSL subroutine DFAC, however numbers may get very large. To alleviate this problem, logarithms can be used to make the numbers more tractable.

APPENDIX B: Program to calculate the eigenfrequencies of the axisymmetric normal modes of the Poincaré problem in a liquid sphere with rigid fixed boundaries.

```
C                    sp5.f
C      Program to calculate eigenfrequencies of axisymmetric normal modes of the Poincare
C       problem in a liquid sphere.  Program calls on the subroutines for calculating the

C      determinant of a matrix which has no zeroes on the main diagonal.
C      The IMSL subroutine DZREAL, is used to locate the zeroes of the determinant,
C      with subroutine DET and DETM to evaluate the determinant.
C      Many write statements are included, but have been commented out.  The were used to
C      ensure that values were correct.
C      Reminder:  Change dimension and data specifications in the function DETM when
C      changing NN, so that C, F, H, are MM by MM.  NOTE:  MM = NN*(NN+3)/2
C      Both NN, MMM and steps may need to be changed for different runs.
       implicit  real*8(a-h, o-z)
         parameter (NN = 5, MMM = NN*(NN+3)/2)
         parameter  (nroot = NN*(NN-1)/2)
         integer    info(nroot)
         real*8     detm, x(nroot), xguess(nroot)
         external   wrrrn, dzreal
       external  detm
         common/a/ C(MMM,MMM), F(MMM,MMM), MM
       MM = MMM
C      NN is the size of the summation in the trial function,
```

```
C     MM the number of terms in the trial function.
C     FACTNN is a factor pulled out of the calculation.
C     FLOG is a factor multiplied in to make the numbers more managable.
      FLOG = 0.D0
      DO 5,  LL = 1, NN
        DI = 2*LL + 1
        FLOG = FLOG + DLOG10(DI)
5     CONTINUE
      FACTNN = 10.D0**FLOG
C                 Matrix elements
C     The matrix elements have two indicies, JJ and LL, for row and column
C     respectively.  (JJ depends on I and J, and LL depends on K and L.)
C     Since the row increases slowest, this loop starts first.
C     Starting the outer loop, to determine the row number.
      do 10   I = 0, NN
        kk=0
        if (I.eq.0)  kk=1
      do 20   J = kk, NN-I
C     Starting the inner loop, to determine the column number.
    do 30   K = 0, NN
        kk=0
        if (K.eq.0)  kk=1
    do 40   L = kk, NN-k
C     Start the calculation.
```

```
            csumm = 0
            fsumm = 0
          do 50  M = 0, I + K
           BINO = BINOM(I + K, M)
            IM = (-1)**M
           CBOTM = 2*(J + L + M) + 1
            FBOTM = CBOTM*(2*(J + L + M) - 1)
           CSUMM = CSUMM + BINO * IM/CBOTM
            FSUMM = FSUMM + BINO * IM/FBOTM
50         CONTINUE
          IF (I + K .EQ. 0) THEN
            CC = 0
           ELSE
            CC = I*K* CSUMM/((I+K)*2)
          ENDIF
          FF = J * L * FSUMM/(I + K +1)
C         Set up the indexing
          JJ = J + I * (2 * NN + 3 - I)/2
          LL = L + K * (2 * NN + 3 - K)/2
            C(JJ,LL) = FACTNN * CC
            F(JJ,LL) = FACTNN * FF
40         continue
30         continue
20         continue
```

```
10      continue
C       This will use the IMSL subroutine DZREAL to find the roots from DETM
        data xguess/0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99/
        EPS     = 1.0D-8
          ERRABS    = 1.0D-11
          ERRREL    = 1.0D-11
        ETA     = 1.0D-7
        ITMAX     = 300
           Call dzreal (detm, errabs, errrel, eps, eta, nroot, itmax, xguess,
     &           x, info)
           Call dwrrrn ('the zeroes are', 1, nroot, x, 1, 0)
           STOP
           END
C       CALCULATES DETERMINANT OF N BY N MATRIX WITH NO ZERO
C       ELEMENTS ON THE MAIN DIAGONAL.  THE MATRIX IS AN N BY N ARRAY
C       VALUE OF DETERMINANT IS DETT TIMES (10 TO THE IE'TH POWER)
C       (THIS IS DONE BY TRIANGULARIZING THE MATRIX)
           SUBROUTINE  DET(A,N,IE,DETT)
           IMPLICIT  REAL*8(A-H,O-Z)
           DIMENSION A(N,N)
           NM1 = N - 1
           DETLOG = 0.D0
           SIGN = 1.D0
        DO 1  K = 1, NM1
```

```
      KP1 = K + 1
      R = 1.D0/A(K,K)
      IF (A(K,K).GT.0.D0)  GO TO 3
      SIGN = -SIGN
3     DETLOG = DETLOG + DLOG10(DABS(A(K,K)))
      DO 2  J = KP1, N
2     A(K,J) = R*A(K,J)
      DO 1  I = KP1, N
      S = A(I,K)
      DO 1  L = KP1, N
1     A(I,L) = A(I,L) - S*A(K,L)
      IF (A(N,N).GT.0.D0)  GO TO 4
      SIGN = -SIGN
4     DETLOG = DETLOG + DLOG10(DABS(A(N,N)))
      IE = IDINT(DETLOG)
      DETLOG = DETLOG - DFLOAT(IE)
      DETT = SIGN*(10.D0**DETLOG)
      RETURN
      END
      REAL*8 FUNCTION  DETM(X)
      IMPLICIT  REAL*8(A-H, O-Z)
      parameter (NN = 5, MMM = NN*(NN+3)/2)
      COMMON/A/ C(MMM,MMM), F(MMM,MMM), MM
      DIMENSION H(MMM,MMM)
```

```
      DO 1  J = 1, MM
      DO 2  L = 1, MM
     IF (J.LE.5)  GO TO 3
      IF (J.EQ.6)  GO TO 4
      IF (J.EQ.11)  GO TO 4
      IF (J.EQ.15)  GO TO 4
      IF (J.EQ.18)  GO TO 4
      IF (J.EQ.20)  GO TO 4
      GO TO 5
3     H(J,L) = F(J,L)
      GO TO 6
4     H(J,L) = C(J,L)
      GO TO 6
5     H(J,L) = X*X*(C(J,L) + F(J,L)) - F(J,L)
6     CONTINUE
2     CONTINUE
1     CONTINUE
      CALL DET(H,MM,IE,DETT)
      DETM = DETT * (10.D0**IE)
      RETURN
      END
```

APPENDIX C: Program to calculate the eigenfrequencies of the axisymmetric normal modes of the Poincaré problem in a liquid sphere including a neutral density profile from PREM.

```
C                    spden5.f
C     Program to calculate eigenfrequencies of axisymmetric normal modes of the Poincare
C     problem in a liquid sphere INCLUDING DENSITY from PREM.  Program calls on a
C     subroutine for calculating the determinant of a matrix which has no zeroes on the main
C    diagonal. The IMSL subroutine DZREAL, is used to locate the zeroes of the determinant,
C     using the external function DETM(X).  NN is the size of the summation in the trial
C     function, MM the number of terms in the trial function.  NOTE:  MM = NN*(NN+3)/2
C          NN will need to be changed for different runs.
C          NN must be changed in the parameter statements both on top
C            and in the external function
C          in DETM(X) must also change the goto statements for different
C          numbers of zeros.
      implicit  real*8(a-h, o-z)
         parameter (NN = 5, MMM = NN*(NN+3)/2)
      parameter  (nroot = NN*(NN-1)/2)
         integer    info(nroot)
         real*8     detm, x(nroot), xguess(nroot), densty(4)
         external   dwrrrn, dzreal
      external  detm
      common/a/ C(MMM,MMM), F(MMM,MMM), MM
        open(10, file = 'spd5zeroes',status = 'unknown')
     MM = MMM
```

```
C     This calculates the values of the matrix needed in the program.
C     This is a factor to make the values of the matrix tractable.
      FLOG = 0.D0
        DO 5,  LL = 1, NN
          DI = 2*LL + 1
          FLOG = FLOG + DLOG10(DI)
5       CONTINUE
        FACTNN = 10.D0**FLOG
C     These are the coefficients for the density polynomial given by D&A (renormalized)
      densty(1) = 12.5815
        densty(2) = -1.2638 * .546225
        densty(3) = -3.6426 * .546225 * .546225
        densty(4) = -5.5281 * .546225 * .546225 * .546225
C                   Matrix elements
C     The matrix elements have two indicies, JJ and LL, for row and column
C     respectively.  (JJ depends on I and J, and LL depends on K and L.)
C     Since the row increases slowest, this loop starts first.
C     Starting the outer loop, to determine the row number.
        do 10   I = 0, NN
          kk=0
          if (I.eq.0)  kk=1
        do 20   J = kk, NN-I
C       Starting the inner loop, to determine the column number.
      do  30   K = 0, NN
```

```
      kk=0
      if (K.eq.0)  kk=1
    do  40   L = kk, NN-k
    csumm = 0
      fsumm = 0
    do 50   M = 0, I + K
     FBINO = DBINOM(I + K, M)
       IM = (-1)**M
       FBOTM = 2*(J + L + M) - 1
       FSUMM = FSUMM + FBINO * IM/FBOTM
50    Continue
    do 55  M = 0, I + K - 1
       CBINO = DBINOM((I + K -1), M)
       IM = (-1)**M
       CBOTM = 2*(J + L + M) + 1
       CSUMM = CSUMM + CBINO * IM/CBOTM
55    Continue
C     This loop accounts for the "density effect"
    SUMN = 0
      DO  60  N = 1,4
     DSUMN = N + 2*(I + J + K + L)
     SUMN = SUMN + DENSTY(N)/DSUMN
60    CONTINUE
C     Calculate the elements.
```

```
            CC = I * K * CSUMM * SUMN
            FF = J * L * FSUMM * SUMN
C       Set up indexing....
        JJ =  J + I * (2 * NN + 3 - I)/2
        LL =  L + K * (2 * NN + 3 - K)/2
        C(JJ,LL) = CC*500.0
        F(JJ,LL) = FF*500.0
40      continue
30      continue
20      continue
10      continue
C       This will use the IMSL subroutine dzreal to find the roots from the
C       function detm.  Guesses must be the correct number, but need not be close to the actual
C       value.  First set up the guesses.
        do   4999  kkk=1,nroot
                   xguess(kkk) = dble(kkk)/11.0
4999    continue
C       EPS and ETA spread criteria for multiple zeroes.
C       ERRABS  absolute error (a stopping criterion)
C       ERRREL  relative error (second stopping criterion)
C       ITMAX   maximum number of iterations per zero
C       INFO    number of interations actually used
        EPS       = 1.0D-11
          ERRABS     = 1.0D-12
```

```
      ERRREL    = 1.0D-12
     ETA        = 1.0D-10
      ITMAX     = 400
      Call dzreal (detm, errabs, errrel, eps, eta, nroot, itmax, xguess,
     &          x, info)
     Call dwrrrn ('the zeroes are', 1, nroot, x, 1, 0)
      STOP
     END
C    CALCULATES DETERMINANT OF N BY N MATRIX WITH NO ZERO ELEMENTS
C    ON THE MAIN DIAGONAL.  THE MATRIX IS STORED IN N BY N ARRAY.
C    THE VALUE OF THE DETERMINANT IS DETT TIMES (10 TO THE IE'TH
C    POWER).  THIS IS DONE BY TRIANGULARIZING THE MATRIX.
      SUBROUTINE  DET(A,N,IE,DETT)
      IMPLICIT  REAL*8(A-H,O-Z)
      DIMENSION A(N,N)
      NM1 = N - 1
      DETLOG = 0.D0
      SIGN = 1.D0
      DO 1  K = 1, NM1
        KP1 = K + 1
        R = 1.D0/A(K,K)
        IF (A(K,K).GT.0.D0)  GO TO 3
        SIGN = -SIGN
3      DETLOG = DETLOG + DLOG10(DABS(A(K,K)))
```

```
      DO 2  J = KP1, N
2     A(K,J) = R*A(K,J)
      DO 1  I = KP1, N
        S = A(I,K)
      DO 1  L = KPI, N
1     A(I,L) = A(I,L) - S*A(K,L)
      IF (A(N,N).GT.0.D0)  GO TO 4
      SIGN = -SIGN
4     DETLOG = DETLOG + DLOG10(DABS(A(N,N)))
      IE = IDINT(DETLOG)
      DETLOG = DETLOG - DFLOAT(IE)
      DETT = SIGN*(10.D0**DETLOG)
      RETURN
      END
      REAL*8 FUNCTION  DETM(X)
      IMPLICIT  REAL*8(A-H, O-Z)
      parameter (NN = 5, MMM = NN*(NN+3)/2)
      COMMON/A/ C(MMM,MMM), F(MMM,MMM), MM
      DIMENSION H(MMM,MMM)
C     DATA H/81*0.D0/
      DO 1  J = 1, MM
      DO 2  L = 1, MM
        IF (J.LE.5)  GO TO 3
        IF (J.EQ.6)  GO TO 4
```

```
      IF (J.EQ.11)  GO TO 4
      IF (J.EQ.15)  GO TO 4
      IF (J.EQ.18)  GO TO 4
      IF (J.EQ.20)  GO TO 4
      GO TO 5
3     H(J,L) = F(J,L)
      GO TO 6
4     H(J,L) = C(J,L)
      GO TO 6
5     H(J,L) = X*X*(C(J,L) + F(J,L)) - F(J,L)
6     CONTINUE
2     CONTINUE
1     CONTINUE
      CALL DET(H,MM,IE,DETT)
      DETM = DETT * (10.D0**IE)
      RETURN
      END
```

APPENDIX D: This program finds the eigenvector coefficients for the eigenfrequencies previously calculated.

```
C                    wspdlinsys5.f
C      THIS PROGRAM USES A LINEAR SYSTEM ROUTINE TO FIND THE
C      EIGENVECTOR COEFFICIENTS FOR THE PREVIOUSLY DETERMINED
C      EIGENFREQUENCIES.  FIRST PART OF THE PROGRAM IS EXACTLY AS IN
C      SPDEN5.F,  NN is the size of the summation in the trial function,
C      MM the number of terms in the trial function.  NOTE:  MM = NN*(NN+3)/2
C           NN will need to be changed for different runs.
C           NN must be changed in the parameter statements both on top
C             and in the external function
C           in DETM(X) must also change the goto statements for different runs.
      implicit  real*8(a-h, o-z)
         parameter (NN = 5, MMM = NN*(NN+3)/2)
         parameter  (nroot = NN*(NN-1)/2)
         integer    info(nroot)
         real*8     detm, x(nroot), xguess(nroot), densty(4)
         real*8     D(MMM-1, MMM-1), E(MMM-1), EIG(MMM-1), HH(MMM,MMM)
         external   dwrrrn, dzreal
      external  detm
         common/a/ C(MMM,MMM), F(MMM,MMM), MM
      open(1, file='eigenvalues.out',status='unknown')
         open(2, file='eigenvector.out',status='unknown')
      MM = MMM
```

```
C       This calculates the values of the matrix needed in the program.
C       This is a factor to make the values of the matrix tractable.
          FLOG = 0.D0
          DO 5,  LL = 1, NN
            DI = 2*LL + 1
            FLOG = FLOG + DLOG10(DI)
5       CONTINUE
          FACTNN = 10.D0**FLOG
C       These are the coefficients for the density polynomial given by D&A (renormalized)
        densty(1) = 12.5815
          densty(2) = -1.2638 * .546225
          densty(3) = -3.6426 * .546225 * .546225
          densty(4) = -5.5281 * .546225 * .546225 * .546225
C       The matrix elements have two indicies, JJ and LL, for row and column
C       respectively.  (JJ depends on I and J, and LL depends on K and L.)
C       Since the row increases slowest, this loop starts first.
C       Starting the outer loop, to determine the row number.
        do 10   I = 0, NN
            kk=0
            if (I.eq.0)  kk=1
          do 20   J = kk, NN-I
        do  30   K = 0, NN
            kk=0
            if (K.eq.0)  kk=1
```

```
      do  40   L = kk, NN-k
           csumm = 0
           fsumm = 0
      do 50   M = 0, I + K
        FBINO = DBINOM(I + K, M)
           IM = (-1)**M
           FBOTM = 2*(J + L + M) - 1
           FSUMM = FSUMM + FBINO * IM/FBOTM
50    CONTINUE
      do 55  M = 0, I + K - 1
           CBINO = DBINOM((I + K -1), M)
           IM = (-1)**M
           CBOTM = 2*(J + L + M) + 1
           CSUMM = CSUMM + CBINO * IM/CBOTM
55    Continue
        SUMN = 0
        DO  60  N = 1,4
          DSUMN = N + 2*(I + J + K + L)
       SUMN = SUMN + DENSTY(N)/DSUMN
60    CONTINUE
C     Calculate the elements.
        CC = I * K * CSUMM * SUMN
        FF = J * L * FSUMM * SUMN
C     Set up indexing.
```

```
      JJ = J + I * (2 * NN + 3 - I)/2
      LL = L + K * (2 * NN + 3 - K)/2
      C(JJ,LL) = CC*1000.0
         F(JJ,LL) = FF*1000.0
40    continue
30    continue
20    continue
10    continue
C     This will use the IMSL subroutine dzreal to find the roots from the function detm.
C     Guesses must be the correct number, but need not be close to the actual value.
      EPS      = 1.0D-9
         ERRABS    = 1.0D-12
         ERRREL    = 1.0D-12
      ETA      = 1.0D-8
         ITMAX     = 300
C     first establish the initial guesses
      do   4999 kkk=1,nroot
               xguess(kkk) = dble(kkk)/11.0
4999  continue
         Call dzreal (detm, errabs, errrel, eps, eta, nroot, itmax, xguess,
&              x, info)
         Call dwrrrn ('THE EIGENVALUES ARE:', 1, nroot, x, 1, 0)
C     THIS SECTION WILL FIND THE EIGENVECTORS USING A LINEAR SYSTEM
C     SUBROUTINE FROM IMSL.  NOTE THAT THE SYSTEM SOLVED IS OF ONE
```

```
C     ORDER LESS THAN THE ORIGINAL SYSTEM.  THIS IS BECAUSE THE FIRST
C     CONSTANT HAS BEEN CHOSEN AS ONE.  FIRST THE EIGENVALUES JUST
C     FOUND ARE SUBBED INTO THE MATRIX HH.
        Do  100  i = 1, nroot
      Do  110  J = 1, MM
      Do  120  L = 1, MM
          IF (J.LE.5)  GO TO 130
          IF (J.EQ.6)  GO TO 140
          IF (J.EQ.11)  GO TO 140
          IF (J.EQ.15)  GO TO 140
          IF (J.EQ.18)  GO TO 140
          IF (J.EQ.20)  GO TO 140
          GO TO 150
130     HH(J,L) = F(J,L)
        GO TO 160
140     HH(J,L) = C(J,L)
        GO TO 160
150     HH(J,L) = X(i)*X(i)*(C(J,L) + F(J,L)) - F(J,L)
160   CONTINUE
120   CONTINUE
110   CONTINUE
        Do  170  J = 1, MM - 1
        Do  180  L = 1, MM - 1
          D(J,L) = HH(J+1, L+1)
```

```
            E(J) = -HH(J+1, 1)
180      Continue
170      Continue
           Call  DLSASF(MM-1, D, MM-1, E, EIG)
         Call  DWRRRN('THE EIGENVECTOR IS:', 1, MM-1, EIG, 1, 0)
         Do 200  kkk = 1, MM-1
           write(2,*) eig(kkk)
200      Continue
           write(2,*)
100      Continue
           STOP
           END
C        CALCULATES DETERMINANT OF N BY N MATRIX WITH NO ZERO
C        ELEMENTS ON THE MAIN DIAGONAL.  THE MATRIX IS STORED IN AN N BY
C        N ARRAY.  THE VALUE OF THE DETERMINANT IS DETT TIMES (10 TO THE
C        IE'TH POWER).  THIS IS DONE BY TRIANGULARIZING THE MATRIX.
           SUBROUTINE  DET(A,N,IE,DETT)
           IMPLICIT  REAL*8(A-H,O-Z)
           DIMENSION A(N,N)
         NM1 = N - 1
           DETLOG = 0.D0
           SIGN = 1.D0
           DO 1  K = 1, NM1
             KP1 = K + 1
```

```
            R = 1.D0/A(K,K)
            IF (A(K,K).GT.0.D0)  GO TO 3
              SIGN = -SIGN
 3          DETLOG = DETLOG + DLOG10(DABS(A(K,K)))
           DO 2  J = KP1, N
 2          A(K,J) = R*A(K,J)
            DO 1  I = KP1, N
              S = A(I,K)
            DO 1  L = KPI, N
 1          A(I,L) = A(I,L) - S*A(K,L)
              IF (A(N,N).GT.0.D0)  GO TO 4
              SIGN = -SIGN
 4          DETLOG = DETLOG + DLOG10(DABS(A(N,N)))
            IE = IDINT(DETLOG)
            DETLOG = DETLOG - DFLOAT(IE)
            DETT = SIGN*(10.D0**DETLOG)
            RETURN
            END
            REAL*8 FUNCTION  DETM(X)
            IMPLICIT  REAL*8(A-H, O-Z)
            parameter (NN = 5, MMM = NN*(NN+3)/2)
            COMMON/A/ C(MMM,MMM), F(MMM,MMM),  MM
            DIMENSION H(MMM,MMM)
C           DATA H/81*0.D0/
```

```
      DO 1  J = 1, MM
      DO 2  L = 1, MM
        IF (J.LE.5)  GO TO 3
        IF (J.EQ.6)  GO TO 4
        IF (J.EQ.11)  GO TO 4
        IF (J.EQ.15)  GO TO 4
        IF (J.EQ.18)  GO TO 4
        IF (J.EQ.20)  GO TO 4
       GO TO 5
3     H(J,L) = F(J,L)
        GO TO 6
4     H(J,L) = C(J,L)
        GO TO 6
5     H(J,L) = X*X*(C(J,L) + F(J,L)) - F(J,L)
6    CONTINUE
2    CONTINUE
1    CONTINUE
      CALL DET(H,MM,IE,DETT)
      DETM = DETT * (10.D0**IE)
      RETURN
      END
```

APPENDIX E: This program evaluates equation (82) from Rochester (1989).

```
C                    perturbation.f
C   This is a program for eqn(82) in R's '89 paper.  The eigenvalues and eigenvector
C   coefficients are read in from a file.  The first eigenvector value is always one
      parameter (ivalue = 10, ivectr = 20)
      implicit  real*8(a-h, o-z)
      real*8  eigval(ivalue), eigvec(ivalue,ivectr)
      real*8  cr(20), rho(4), rinten
      integer  p, q, qq
      open(1,file='eigenvalues.out',status='old')
      open(2,file='eigenvector.out',status='old')
      open(3,file='zeros.out', status='unknown')
      beta = -0.000010d0
C     Set up loops to read in eigenvalues/vector coefficients, the outer loop is for the
C     eigenvalues, each eigenvalue is associated with an eigenvector comprising ivectr entries.
      do 10  i = 1, ivalue
      read(1,*)  eigval(i)
      eigvec(i,1) = 1.0d0
      read(2,*)  (eigvec(i,j),j=2,ivectr)
10    continue
C   The constants needed:  alf* are coefficients of the polynomial representing 1/alpha^2, rho*
C   are renormalized values of PREM density for CMB = r = 1, cr* are coefficients of the
C   powers of r that have been integrated using MACSYMA and exported in FORTAN
C   format.  These are the coefficients for the 7th degree polynomial for 1/alpha^2 obtained
```

```
C     using the IMSL routine called drcurv. (r has been renormalized). Units are (sec/km)^2.
      alf1 = .0081914306880072d0
      alf2 = .0032968419226614d0
      alf3 = -.0016245386362791d0
      alf4 = .0055783009628760d0
      alf5 = -.0093306230249849d0
      alf6 = .019869926098212d0
      alf7 = -.0181699822952l8d0
      alf8 = .007563066301627 9d0
      rho1 = 12.5815d0
      rho2 =  -.6903192d0
      rho3 = -1.0868115d0
      rho4 =  -.9009295d0
      rho(1) = rho1
      rho(2) = rho2
      rho(3) = rho3
      rho(4) = rho4
C           Info. for the coefficient of go^2.  (16*pi^2*G^2)*b^2, (units are
(m^3/(kg*sec^2))^2)*km^2)
      G = 6.6732d-11
      b = 3.480d3
      pi = 3.1415926d0
      comega = 7.292115d-05
C     Coefficients of powers of r integrated by MACSYMA and imported in FORTRAN format.
```

```
      cr(18) = alf8*rho4**3/36.0d0
      cr(17) = rho4**2*(5*alf7*rho4+17*alf8*rho3)/180.0d0
      cr(16) = rho4*(25*alf6*rho4**2+85*alf7*rho3*rho4+100*alf8*rho2*rho
     1  4+96*alf8*rho3**2)/900.0d0
      cr(15) = (25*alf5*rho4**3+85*alf6*rho3*rho4**2+100*alf7*rho2*rho4*
     1  *2+125*alf8*rho1*rho4**2+96*alf7*rho3**2*rho4+225*alf8*rho2*rho
     2  3*rho4+36*alf8*rho3**3)/900.0d0
      cr(14) = (100*alf4*rho4**3+340*alf5*rho3*rho4**2+400*alf6*rho2*rho
     1  4**2+500*alf7*rho1*rho4**2+384*alf6*rho3**2*rho4+900*alf7*rho2*
     2  rho3*rho4+1120*alf8*rho1*rho3*rho4+525*alf8*rho2**2*rho4+144*al
     3  f7*rho3**3+504*alf8*rho2*rho3**2)/3600.0d0
      cr(13) = (100*alf3*rho4**3+340*alf4*rho3*rho4**2+400*alf5*rho2*rho
     1  4**2+500*alf6*rho1*rho4**2+384*alf5*rho3**2*rho4+900*alf6*rho2*
     2  rho3*rho4+1120*alf7*rho1*rho3*rho4+525*alf7*rho2**2*rho4+1300*a
     3  lf8*rho1*rho2*rho4+144*alf6*rho3**3+504*alf7*rho2*rho3**2+624*a
     4  lf8*rho1*rho3**2+585*alf8*rho2**2*rho3)/3600.0d0
      cr(12) = (100*alf2*rho4**3+340*alf3*rho3*rho4**2+400*alf4*rho2*rho
     1  4**2+500*alf5*rho1*rho4**2+384*alf4*rho3**2*rho4+900*alf5*rho2*
     2  rho3*rho4+1120*alf6*rho1*rho3*rho4+525*alf6*rho2**2*rho4+1300*a
     3  lf7*rho1*rho2*rho4+800*alf8*rho1**2*rho4+144*alf5*rho3**3+504*a
     4  lf6*rho2*rho3**2+624*alf7*rho1*rho3**2+585*alf7*rho2**2*rho3+14
     5  40*alf8*rho1*rho2*rho3+225*alf8*rho2**3)/3600.0d0
      cr(11) = (100*alf1*rho4**3+340*alf2*rho3*rho4**2+400*alf3*rho2*rho
     1  4**2+500*alf4*rho1*rho4**2+384*alf3*rho3**2*rho4+900*alf4*rho2*
```

```
     2  rho3*rho4+1120*alf5*rho1*rho3*rho4+525*alf5*rho2**2*rho4+1300*a
     3  lf6*rho1*rho2*rho4+800*alf7*rho1**2*rho4+144*alf4*rho3**3+504*a
     4  lf5*rho2*rho3**2+624*alf6*rho1*rho3**2+585*alf6*rho2**2*rho3+14
     5  40*alf7*rho1*rho2*rho3+880*alf8*rho1**2*rho3+225*alf7*rho2**3+8
     6  25*alf8*rho1*rho2**2)/3600.0d0
        cr(10) = (340*alf1*rho3*rho4**2+400*alf2*rho2*rho4**2+500*alf3*rho
     1  1*rho4**2+384*alf2*rho3**2*rho4+900*alf3*rho2*rho3*rho4+1120*al
     2  f4*rho1*rho3*rho4+525*alf4*rho2**2*rho4+1300*alf5*rho1*rho2*rho
     3  4+800*alf6*rho1**2*rho4+144*alf3*rho3**3+504*alf4*rho2*rho3**2+
     4  624*alf5*rho1*rho3**2+585*alf5*rho2**2*rho3+1440*alf6*rho1*rho2
     5  *rho3+880*alf7*rho1**2*rho3+225*alf6*rho2**3+825*alf7*rho1*rho2
     6  **2+1000*alf8*rho1**2*rho2)/3600.0d0
        cr(9) = (400*alf1*rho2*rho4**2+500*alf2*rho1*rho4**2+384*alf1*rho3
     1  **2*rho4+900*alf2*rho2*rho3*rho4+1120*alf3*rho1*rho3*rho4+525*a
     2  lf3*rho2**2*rho4+1300*alf4*rho1*rho2*rho4+800*alf5*rho1**2*rho4
     3  +144*alf2*rho3**3+504*alf3*rho2*rho3**2+624*alf4*rho1*rho3**2+5
     4  85*alf4*rho2**2*rho3+1440*alf5*rho1*rho2*rho3+880*alf6*rho1**2*
     5  rho3+225*alf5*rho2**3+825*alf6*rho1*rho2**2+1000*alf7*rho1**2*r
     6  ho2+400*alf8*rho1**3)/3600.0d0
        cr(8) = (500*alf1*rho1*rho4**2+900*alf1*rho2*rho3*rho4+1120*alf2*r
     1  ho1*rho3*rho4+525*alf2*rho2**2*rho4+1300*alf3*rho1*rho2*rho4+80
     2  0*alf4*rho1**2*rho4+144*alf1*rho3**3+504*alf2*rho2*rho3**2+624*
     3  alf3*rho1*rho3**2+585*alf3*rho2**2*rho3+1440*alf4*rho1*rho2*rho
     4  3+880*alf5*rho1**2*rho3+225*alf4*rho2**3+825*alf5*rho1*rho2**2+
```

```
     5  1000*alf6*rho1**2*rho2+400*alf7*rho1**3)/3600.0d0
        cr(7) = (1120*alf1*rho1*rho3*rho4+525*alf1*rho2**2*rho4+1300*alf2*
     1  rho1*rho2*rho4+800*alf3*rho1**2*rho4+504*alf1*rho2*rho3**2+624*
     2  alf2*rho1*rho3**2+585*alf2*rho2**2*rho3+1440*alf3*rho1*rho2*rho
     3  3+880*alf4*rho1**2*rho3+225*alf3*rho2**3+825*alf4*rho1*rho2**2+
     4  1000*alf5*rho1**2*rho2+400*alf6*rho1**3)/3600.0d0
        cr(6) = (1300*alf1*rho1*rho2*rho4+800*alf2*rho1**2*rho4+624*alf1*r
     1  ho1*rho3**2+585*alf1*rho2**2*rho3+1440*alf2*rho1*rho2*rho3+880*
     2  alf3*rho1**2*rho3+225*alf2*rho2**3+825*alf3*rho1*rho2**2+1000*a
     3  lf4*rho1**2*rho2+400*alf5*rho1**3)/3600.0d0
        cr(5) = (160*alf1*rho1**2*rho4+288*alf1*rho1*rho2*rho3+176*alf2*rh
     1  o1**2*rho3+45*alf1*rho2**3+165*alf2*rho1*rho2**2+200*alf3*rho1*
     2  *2*rho2+80*alf4*rho1**3)/720.0d0
        cr(4) = rho1*(176*alf1*rho1*rho3+165*alf1*rho2**2+200*alf2*rho1*rh
     1  o2+80*alf3*rho1**2)/720.0d0
        cr(3) = rho1**2*(5*alf1*rho2+2*alf2*rho1)/18.0d0
        cr(2) = alf1*rho1**3/9.0d0
C       Start the loop for the eigenvalue usage.
        do 80  isig = 1,ivalue
        rinten = 0.0d0
        rinted = 0.0d0
        zinten = 0.0d0
        zinted = 0.0d0
        signum = 0.0d0
```

```
      sigdem = 0.0d0
      signew = 0.0d0
      sigtwo = 0.0d0
      do 100  l = 0,5
         nn=0
            if ( l .eq. 0 ) nn = 1
      do 110  n = nn,5-l
      do 120  p = 0,5
         qq=0
            if ( p .eq. 0 ) qq = 1
      do 130  q = qq,5-p
C     Set up conversion from double index to single index
      jj = n + l*5 - (l-3)*l/2
      kk = q + p*5 - (p-3)*p/2
      rinten = 0.0d0
      do 140  ii = 2,18
      rinten=rinten+cr(ii)/(2.0d0*dble(l+p+n+q)+dble(ii)+1.0d0)
140   continue
C     Calculate the z integral of the numberator (zinten)
      zntop=DGAMMA(dble(n+q)+0.5d0)*DGAMMA(dble(l+p)+1.0d0)
      znbot = DGAMMA(dble(n + q + l + p) + 1.5d0)
      zinten = zntop/znbot
C     Calculate the coefficient from C*gradX for the numberator
      ccgrad = (eigval(isig)**4.0*l*p-(eigval(isig)**2.0)*
```

```
     & (1.0d0-eigval(isig)**2.0)*(l*q+n*p)
     & + (1.0d0-eigval(isig)**2.0)**2.0*n*q)

C    Choosing the correct eigenvector value to multiply in to the numberator.
       cjj = eigvec(isig,jj)
       ckk = eigvec(isig,kk)
     signum = signum + (cjj*ckk*ccgrad*zinten*rinten)
C    Calculating the denomonator.
     rinted = 0.0d0
     do 150 ir = 1,4
       rinted = rinted + rho(ir)/(2.0d0*dble(l+n+p+q)+dble(ir))
150  continue
     if ((l+p).EQ.0)then
          zdtop1 = 0.0d0
     else
       zdtop1=DGAMMA(dble(n+q)+0.5d0)*DGAMMA(dble(l+p))
     endif
     zdbot1 = DGAMMA(dble(n+q+l+p)+0.5d0)
     if ((n+q).EQ.0) then
          zdtop2 = 0.0d0
     else
       zdtop2=DGAMMA(dble(n+q)-0.5d0)*DGAMMA(dble(l+p)+1.0d0)
     endif
     zdbot2 = DGAMMA(dble(n+q+l+p)+0.5d0)
```

```
      zinted = l*p*(zdtop1/zdbot1) + n*q*(zdtop2/zdbot2)
      sigdem = sigdem + cjj*ckk*rinted*zinted
130   continue
120   continue
110   continue
100   continue
      coetop = 8.0d0*beta*((pi*G*b)**2.0d0)*1.0d6/
     & ((comega**2.0d0)*eigval(isig)**2.0d0*(eigval(isig)**2.0d0-1.0d0))
         coebot = 4.0d0*eigval(isig)
      sigtwo = coetop*signum/(coebot*sigdem)
         signew = sigtwo + eigval(isig)
         write(*,900)  sigtwo, signew, eigval(isig)
         write(3,920)  signew
80    continue
900   format(f10.8, 5x, f10.8, 5x, f10.8)
920   format(d10.4)
      stop
      end
```

APPENDIX F: These programs evaluate negative betas using the Galerkin method.

```
C                Biggerbetas.f
C     THIS SETS UP THE MATRICIES TO BE READ BY readbb.f, MAKE SURE THAT
C     THE VALUE FOR BETA IS THE SAME IN BOTH PROGRAMS. This program is used
C     to find eigenvalues for the problem with nonzero negative betas. The equations are from
C     solving the SSWE when B can vanish in a sphere and allows for larger negative betas to
C     be used than the perturbation approach does. All integers are prefaced by "i's", except in
C     Functions, where they are prefaced by "n's", all arrays are prefaced by "A's" outside loops
C     range from 50 to 90 write loops range from 10 to 40  (for checking) outer index loops
C     range from 100 to 190, inner summation loops range from 200 up
      implicit  real*8(a-h, o-z)
      parameter (NN = 5, LL = NN*(NN+3)/2, KK = LL)
      real*8  AA(LL,KK), AB(LL,KK), AC(LL,KK), AD(LL,KK), AE(LL,KK)
      real*8  AG(LL,KK), AP(LL,KK), AQ(LL,KK), AR(LL,KK), AT(LL,KK)
      real*8  AU(LL,KK), AV(LL,KK), AX(LL,KK), AY(LL,KK)
      real*8   ArhoP(4), ArhoR(10), Arho(10)
      real*8   Aalpha(8)
      open(11, file = 'AA.dat',status = 'unknown')
      open(12, file = 'AB.dat',status = 'unknown')
      open(13, file = 'AC.dat',status = 'unknown')
      open(14, file = 'AD.dat',status = 'unknown')
      open(15, file = 'AE.dat',status = 'unknown')
      open(16, file = 'AG.dat',status = 'unknown')
      open(17, file = 'AP.dat',status = 'unknown')
```

```
      open(18, file = 'AQ.dat',status = 'unknown')
      open(19, file = 'AR.dat',status = 'unknown')
      open(20, file = 'AT.dat',status = 'unknown')
      open(21, file = 'AU.dat',status = 'unknown')
      open(22, file = 'AV.dat',status = 'unknown')
      open(23, file = 'AX.dat',status = 'unknown')
      open(24, file = 'AY.dat',status = 'unknown')
      LLL = LL
C     Choose value for Beta
        beta = 0.000d0
C     Choose values for irho (either 4 (PREM) or 10 (NEUT))
        irho = 4
C       irho = 10
        crho0 = 12.5815d3
        calph0 = .008191430 6d0
        Aalpha(1) = .0081914306880072d0/calph0
        Aalpha(2) = .0032968419226614d0/calph0
        Aalpha(3) = -.0016245386362791d0/calph0
        Aalpha(4) = .0055783009628760d0/calph0
        Aalpha(5) = -.0093306230249849d0/calph0
        Aalpha(6) = .019869926098212d0/calph0
        Aalpha(7) = -.018169982295218d0/calph0
        Aalpha(8) = .0075630663016279d0/calph0
C     Density from PREM
```

```
      ArhoP(1) = 12.5815d3/crhoP0
         ArhoP(2) = -.6903192d3/crhoP0
         ArhoP(3) = -1.0868115d3/crhoP0
         ArhoP(4) = -.9009295d3/crhoP0
C     Density = NEUT
      ArhoR(1) = .1134369d1
      ArhoR(2) = 0.0d0
      ArhoR(3) = -.1959092d0
      ArhoR(4) = -.5251182d-1
      ArhoR(5) = .4362577d-1
      ArhoR(6) = -.1851055d-1
      ArhoR(7) = .2536884d-2
      ArhoR(8) = -.2360454d-1
      ArhoR(9) = .2196424d-1
      ArhoR(10) = -.8309286d-2
C     Decide which density profile is to be used.
      do 50 i = 1, irho
        if (irho.eq.10) then
            Arho(i) = ArhoR(i)
      else
            Arho(i) = ArhoP(i)
      endif
50    continue
C     Starting the loops for the calculations of the 14 matricies
```

```
      do 100  ii = 0,NN
            ijj = 0
        if(ii.eq.0) ijj = 1
      do 110  ij = ijj,NN-ii
      do 120  ip = 0,NN
            iqq = 0
        if(ip.eq.0) iqq = 1
      do 130  iq = iqq,NN-ip
        il = ii*(2*NN+3-ii)/2  + ij
        ik = ip*(2*NN+3-ip)/2  + iq
C     Start filling in the 14 arrays needed
C     Array "AA"
      arhosum = 0.0d0
         azint = 0.0d0
         do 200  im = 1, irho
      arhosum =  arhosum + Arho(im)/(2.0d0*(ii+ij+ip+iq)+im)
200    continue
      if ((ii + ip).eq.0) then
            aFIZ1 = 0.0d0
      else
        aFIZ1 = FIZ(ii + ip - 1, ij + iq)
      endif
        if((ij + iq).eq.0)then
          aFIZ2 = 0.0d0
```

```
      else
        aFIZ2 = FIZ(ii + ip, ij + iq - 1)
      endif
         azint = ii*ip*aFIZ1 + ij*iq*aFIZ2
         AA(il,ik) = -arhosum*azint
      write(11,900) il, ik, AA(il,ik)
C     Array "AB"
      brhosum = 0.0d0
         bzint = 0.0d0
         do 210  im = 1, irho
         do 220  in = 1, irho
         do 230  is = 1,8
      bdenom = (in+2.0d0)*(2.0d0*(ii+ij+ip+iq)+im+in+is)
      brhosum =  brhosum + Arho(im)*Arho(in)*Aalpha(is)/bdenom
230   continue
220   continue
210   continue
      bFIZ1 = FIZ(ii + ip, ij + iq)
         bzint = (ip + iq)*bFIZ1
      AB(il,ik) = -brhosum*bzint
      write(12,900) il, ik, AB(il,ik)
C     Array "AC"
      crhosum = 0.0d0
         czint = 0.0d0
```

```
          do 240  im = 1, irho
        crhosum =  crhosum + Arho(im)/(2.0d0*(ii+ij+ip+iq)+im)
240      continue
        if ((ii + ip).eq.0) then
            cFIZ1 = 0.0d0
        else
          cFIZ1 = FIZ(ii + ip - 1, ij + iq)
        endif
           if((ij + iq).eq.0)then
             cFIZ2 = 0.0d0
        else
          cFIZ2 = FIZ(ii + ip, ij + iq - 1)
        endif
           czint = ii*ip*cFIZ1 + 2.0d0*ij*iq*cFIZ2
           AC(il,ik) = crhosum*czint
        write(13,900) il, ik, AC(il,ik)
C       Array "AD"
        drhosum = 0.0d0
           dzint = 0.0d0
           do 250  im = 1, irho
           do 260  in = 1, irho
           do 270  is = 1,8
        ddenom = (in+2.0d0)*(2.0d0*(ii+ij+ip+iq)+im+in+is)
        drhosum =  drhosum + Arho(im)*Arho(in)*Aalpha(is)/ddenom
```

```
270    continue
260    continue
250    continue
       dFIZ1 = FIZ(ii + ip, ij + iq)
          dzint = (ip + 2.0d0*iq)*dFIZ1
       AD(il,ik) = drhosum*dzint
       write(14,900) il, ik, AD(il,ik)
C      Array "AE"
       erhosum = 0.0d0
          ezint = 0.0d0
          do 280  im = 1, irho
          do 290  in = 1, irho
          do 300  ir = 1, irho
          do 310  is = 1,8
       edenom=(in+2.0d0)*(ir+2.0d0)
     &   *(2.0d0*(ii+ij+ip+iq)+im+in+ir+is-1.0d0)
       erhosum=erhosum + Arho(im)*Arho(in)*Arho(ir)*Aalpha(is)/edenom
310    continue
300    continue
290    continue
280    continue
       if ((ii + ip).eq.0) then
            eFIZ1 = 0.0d0
       else
```

```
          eFIZ1 = FIZ(ii + ip - 1, ij + iq + 1)
        endif
        if ((ii + ip).eq.0) then
            eFIZ2 = 0.0d0
        else
          eFIZ2 = FIZ(ii + ip - 1, ij + iq)
        endif
        eFIZ3 = FIZ(ii + ip, ij + iq)
          if((ij + iq).eq.0)then
            eFIZ4 = 0.0d0
        else
          eFIZ4 = FIZ(ii + ip, ij + iq - 1)
        endif
          ezint1 = -ip*(3.0d0*ii+2.0d0*ip-1.0d0)*eFIZ1
          ezint2 = ip*(ii+2.0d0*ip-1.0d0)*eFIZ2
        ezint3=(3.0d0*ij*(ip+iq)+iq*(3.0d0*ii+4.0d0*ip+2.0d0*iq-1.0d0)
     &   +2.0d0*(ip+iq))*eFIZ3
          ezint4 = -2.0d0*ij*iq*eFIZ4
        ezint = ezint1 + ezint2 + ezint3 + ezint4
          AE(il,ik) = erhosum*ezint
        write(15,900) il, ik, AE(il,ik)
C       Array "AG"
        grhosum = 0.0d0
          gzint = 0.0d0
```

```
            do 320  im = 1, irho
            do 330  in = 1, irho
            do 340  ir = 1, irho
            do 350  is = 1,8
            do 360  it = 1, irho
            do 370  iw = 1,8
          gnumer=Arho(im)*Arho(in)*Arho(ir)*Aalpha(is)*Arho(it)*Aalpha(iw)
          gdenom = (in+2.0d0)*(ir+2.0d0)*(it+2.0d0)*
     &   (2.0d0*(ii+ij+ip+iq)+im+in+ir+is+it+iw-1.0d0)
          grhosum =  grhosum + gnumer/gdenom
370       continue
360       continue
350       continue
340       continue
330       continue
320       continue
          gFIZ1 = FIZ(ii + ip, ij + iq)
            gzint = (ip + iq)*gFIZ1
            AG(il,ik) = -grhosum*gzint
          write(16,900) il, ik, AG(il,ik)
C         Array "AP"
          prhosum = 0.0d0
            pzint = 0.0d0
            do 380  im = 1, irho
```

```
      prhosum =  prhosum + Arho(im)/(2.0d0*(ii+ij+ip+iq)+im)
380    continue
         if((ij + iq).eq.0)then
           pFIZ1 = 0.0d0
       else
         pFIZ1 = FIZ(ii + ip, ij + iq - 1)
       endif
       pzint = ij*iq*pFIZ1
       AP(il,ik) = -prhosum*pzint
       write(17,900) il, ik, AP(il,ik)
C      Array "AQ"
       qrhosum = 0.0d0
          qzint = 0.0d0
          do 390  im = 1, irho
          do 400  in = 1, irho
          do 410  is = 1,8
       qdenom = (in+2.0d0)*(2.0d0*(ii+ij+ip+iq)+im+in+is)
       qrhosum =  qrhosum + Arho(im)*Arho(in)*Aalpha(is)/qdenom
410    continue
400    continue
390    continue
       qFIZ1 = FIZ(ii + ip, ij + iq)
          qzint = iq*qFIZ1
          AQ(il,ik) = -qrhosum*qzint
```

```
      write(18,900) il, ik, AQ(il,ik)
C     Array "AR"
      rrhosum = 0.0d0
         rzint = 0.0d0
         do 420  im = 1, irho
         do 430  in = 1, irho
         do 440  ir = 1, irho
         do 450  is = 1,8
      rdenom=(in+2.0d0)*(ir+2.0d0)
     &   *(2.0d0*(ii+ij+ip+iq)+im+in+ir+is-1.0d0)
      rrhosum=rrhosum + Arho(im)*Arho(in)*Arho(ir)*Aalpha(is)/rdenom
450   continue
440   continue
430   continue
420   continue
      if ((ii + ip).eq.0) then
           rFIZ1 = 0.0d0
      else
        rFIZ1 = FIZ(ii + ip - 1, ij + iq + 2)
      endif
      if ((ii + ip).eq.0) then
           rFIZ2 = 0.0d0
      else
        rFIZ2 = FIZ(ii + ip - 1, ij + iq + 1)
```

```
      endif
      rFIZ3 = FIZ(ii + ip, ij + iq + 1)
      rFIZ4 = FIZ(ii + ip, ij + iq)
        if((ij + iq).eq.0)then
          rFIZ5 = 0.0d0
      else
        rFIZ5 = FIZ(ii + ip, ij + iq - 1)
      endif
      rzint1 = ip*(2.0d0*ii+2.0d0*ip-1.0d0)*rFIZ1
        rzint2 = -ip*(2.0d0*ip-1.0d0)*rFIZ2
        rzint3=-(iq*(2.0d0*ii+2.0d0*ij+2.0d0*iq-1.0d0)
     &   +2.0d0*ip*(ij+2.0d0*iq)+4.0d0*(ip+iq))*rFIZ3
      rzint4=-(ij*(ip+2.0d0*iq)
     &   +iq*(3.0d0*ii+2.0d0*ip+2.0d0*iq-1.0d0))*rFIZ4
        rzint5 = 2.0d0*ij*iq*rFIZ5
        rzint = rzint1 + rzint2 + rzint3 + rzint4 + rzint5
        AR(il,ik) = rrhosum*rzint
      write(19,900) il, ik, AR(il,ik)
C     Array "AT"
      trhosum = 0.0d0
        tzint = 0.0d0
        do 460  im = 1, irho
        do 470  in = 1, irho
        do 480  ir = 1, irho
```

```
            do 490  is = 1,8
            do 500  it = 1, irho
            do 510  iw = 1,8
          tnumer=Arho(im)*Arho(in)*Arho(ir)*Aalpha(is)*Arho(it)*Aalpha(iw)
          tdenom = (in+2.0d0)*(ir+2.0d0)*(it+2.0d0)*
     &    (2.0d0*(ii+ij+ip+iq)+im+in+ir+is+it+iw-1.0d0)
          trhosum =  trhosum + tnumer/tdenom
  510     continue
  500     continue
  490     continue
  480     continue
  470     continue
  460     continue
          tFIZ1 = FIZ(ii + ip, ij + iq + 1)
          tFIZ2 = FIZ(ii + ip, ij + iq)
            tzint = (ip + iq)*tFIZ1 + iq*tFIZ2
            AT(il,ik) = trhosum*tzint
          write(20,900) il, ik, AT(il,ik)
C         Array "AU"
          urhosum = 0.0d0
            uzint = 0.0d0
            do 520  im = 1, irho
            do 530  in = 1, irho
            do 540  ir = 1, irho
```

```
      do 550  is = 1,8
      do 560  it = 1, irho
      do 570  iv = 1, irho
      do 580  iw = 1,8
     unumer=Arho(im)*Arho(in)*Arho(ir)*Aalpha(is)
     &   *Arho(it)*Arho(iv)*Aalpha(iw)
     udenom = (in+2.0d0)*(ir+2.0d0)*(it+2.0d0)*(iv+2.0d0)*
     &   (2.0d0*(ii+ij+ip+iq)+im+in+ir+is+it+iv+iw-2.0d0)
     urhosum =  urhosum + unumer/udenom
580   continue
570   continue
560   continue
550   continue
540   continue
530   continue
520   continue
     if((ii+ip).eq.0)then
          uFIZ1 = 0.0d0
     else
       uFIZ1 = FIZ(ii + ip - 1, ij + iq + 1)
     endif
     uFIZ2 = FIZ(ii + ip, ij + iq)
     if((ij+iq).eq.0)then
          uFIZ3 = 0.0d0
```

```
      else
        uFIZ3 = FIZ(ii + ip, ij + iq - 1)
      endif
      uzint=-ii*ip*uFIZ1+(ii*iq+ij*ip+ij*iq)*uFIZ2-ij*iq*uFIZ3
         AU(il,ik) = urhosum*uzint
      write(21,900) il, ik, AU(il,ik)
C     Array "AV"
      vrhosum = 0.0d0
         vzint = 0.0d0
         do 590  im = 1, irho
         do 600  in = 1, irho
         do 610  ir = 1, irho
         do 620  is = 1,8
      vnumer=Arho(im)*Arho(in)*Arho(ir)*Aalpha(is)
      vdenom = (in+2.0d0)*(ir+2.0d0)*
     &   (2.0d0*(ii+ij+ip+iq)+im+in+ir+is-1.0d0)
      vrhosum =  vrhosum + vnumer/vdenom
620    continue
610    continue
600    continue
590    continue
      vFIZ1 = FIZ(ii + ip, ij + iq + 1)
      vFIZ2 = FIZ(ii + ip, ij + iq)
         vzint1=(iq*(2.0d0*(ii+ij+ip+iq)-1.0d0)+4.0d0*iq)*tFIZ1
```

```
            vzint2 = -(ij*iq + 2.0d0*iq)*tFIZ2
          vzint = vzint1 + vzint2
            AV(il,ik) = vrhosum*vzint
          write(22,900) il, ik, AV(il,ik)
C         Array "AX"
          xrhosum = 0.0d0
            xzint = 0.0d0
            do 630  im = 1, irho
            do 640  in = 1, irho
            do 650  ir = 1, irho
            do 660  is = 1,8
            do 670  it = 1, irho
            do 680  iw = 1,8
          xnumer=Arho(im)*Arho(in)*Arho(ir)*Aalpha(is)*Arho(it)*Aalpha(iw)
          xdenom = (in+2.0d0)*(ir+2.0d0)*(it+2.0d0)*
     &    (2.0d0*(ii+ij+ip+iq)+im+in+ir+is+it+iw-1.0d0)
          xrhosum = xrhosum + xnumer/xdenom
680       continue
670       continue
660       continue
650       continue
640       continue
630       continue
          xFIZ1 = FIZ(ii + ip, ij + iq + 1)
```

```
            xzint = iq*xFIZ1
            AX(il,ik) = -xrhosum*xzint
          write(23,900) il, ik, AX(il,ik)
C         Array "AY"
          yrhosum = 0.0d0
            yzint = 0.0d0
            do 690  im = 1, irho
            do 700  in = 1, irho
            do 710  ir = 1, irho
            do 720  is = 1,8
            do 730  it = 1, irho
            do 740  iv = 1, irho
            do 750  iw = 1,8
          ynumer=Arho(im)*Arho(in)*Arho(ir)*Aalpha(is)
     &     *Arho(it)*Arho(iv)*Aalpha(iw)
          ydenom = (in+2.0d0)*(ir+2.0d0)*(it+2.0d0)*(iv+2.0d0)*
     &     (2.0d0*(ii+ij+ip+iq)+im+in+ir+is+it+iv+iw-2.0d0)
          yrhosum =  yrhosum + ynumer/ydenom
750       continue
740       continue
730       continue
720       continue
710       continue
700       continue
```

```
690   continue
      if((ii+ip).eq.0)then
           yFIZ1 = 0.0d0
      else
        yFIZ1 = FIZ(ii + ip - 1, ij + iq + 2)
      endif
      yFIZ2 = FIZ(ii + ip, ij + iq + 1)
      yFIZ3 = FIZ(ii + ip, ij + iq)
      yzint1 = ii*ip*yFIZ1
         yzint2 = -(ij*iq+ij*ip+ii*iq+2.0d0*(ip+iq))*yFIZ2
         yzint3 = (ij*iq+2.0d0*iq)*yFIZ3
         yzint = yzint1 + yzint2 + yzint3
         AY(il,ik) = yrhosum*yzint
      write(24,900) il, ik, AY(il,ik)
130   continue
120   continue
110   continue
100   continue
900    format(i3, 3x, i3, 3x, d15.8)
      stop
         end
C     Function to calculate the z integrals using the gamma function.
      real*8 function FIZ(ni,nj)
      Implicit real*8 (a-h, o-z)
```

```
      rni = dble(ni)
      rnj = dble(nj)
      call factlog(2*nj, cone)
      call factlog(ni, ctwo)
      call factlog(ni + nj + 1, cthree)
      call factlog(nj, cfour)
      call factlog(2*(ni + nj + 1), cfive)
      csix = dlog10(2.0**(2.0*rni + 1))
      clogs = cone + ctwo + cthree - cfour - cfive + csix
      FIZ = 10.d0**clogs
      return
      end
C     Calculates the base-10 logarithm of the factorial of an integer
      subroutine factlog(int, fctlg)
      implicit real*8(a-h, o-z)
      w = 0.d0
      do 100 i = 1, int
      di = dble(i)
      w = w + dlog10(di)
100   continue
      fctlg = w
      return
      end
```

```
C                    readbb.f
C       Similar to above program, but reads matricies in and does evaluation.
      implicit real*8(a-h, o-z)
      parameter (NN = 5, LL = NN*(NN+3)/2, KK = LL)
      character zer*15, iter*10, bet*10, den*35
C     Values for IMSL routine DZBREN
      real*8  ERRABS, ERRREL, A, B
      integer maxfn
      parameter  (nroot = 120)
         real*8     detm, xguess(nroot)
      external detm, DZBREN
         common/a/ LLL, AA(LL,KK), AB(LL,KK), AC(LL,KK), AD(LL,KK)
         common/a/ AE(LL,KK), AG(LL,KK), AP(LL,KK), AQ(LL,KK)
         common/a/ AR(LL,KK), AT(LL,KK), AU(LL,KK), AV(LL,KK)
         common/a/ AX(LL,KK), AY(LL,KK)
      common/a/ beta, gamma, f
         open(11, file = 'AA.dat',status = 'old')
         open(12, file = 'AB.dat',status = 'old')
         open(13, file = 'AC.dat',status = 'old')
         open(14, file = 'AD.dat',status = 'old')
         open(15, file = 'AE.dat',status = 'old')
         open(16, file = 'AG.dat',status = 'old')
         open(17, file = 'AP.dat',status = 'old')
         open(18, file = 'AQ.dat',status = 'old')
```

```
      open(19, file = 'AR.dat',status = 'old')
      open(20, file = 'AT.dat',status = 'old')
      open(21, file = 'AU.dat',status = 'old')
      open(22, file = 'AV.dat',status = 'old')
      open(23, file = 'AX.dat',status = 'old')
      open(24, file = 'AY.dat',status = 'old')
      open(26, file = 'Zeros.out',status = 'unknown')
      open(27, file = 'Hbefor.out',status = 'unknown')
      LLL = LL
      beta = 0.000d0
C     Select value for irho
      irho = 4
C     irho = 10
      zer = 'Zeroes'
      iter = 'Iterations'
      write(26, 1071) zer, iter
C     Values from PREM or NEUT, comega sec^-1, cG m^3/(kg sec^2), crhoP0 kg/m^3,
C     cb km, calph0 sec/km, cgatb km/sec-2)
      comega = 7.292115d-5
      cG = 6.6732d-11
      cpi = 3.1415927d0
      crhoP0 = 12.5815d3
      cb = 3480.0d0
      calph0 = .008191430 6d0
```

```
      cgatb = 1.06823d-2
    if (irho.EQ.4) then
      gammat = comega**2.0d0
        gammab = 2.0d0*cpi*cG*crhoP0
        gamma = gammat/gammab
    else
      gammat = 2.0d0*cb*comega**2.0d0
        gammab = 3.0d0*gatb
        gamma = gammat/gammab
    endif
    if (irho .EQ. 4) then
      ftop = 4.0d0*calph0*(cpi*cG*crhoP0*cd)**2.0d0
        fbot = comega**2.0d0
        f = ftop/fbot
    else
        ftop = 9.0d0*calph0*gatb**2.0d0
        fbot = 4.0d0*comega
      f = ftop/fbot
    endif
C     Starting the loops for the reading of the 14 matricies
    do 100  ii = 0,NN
      ijj = 0
    if(ii.eq.0) ijj = 1
    do 110  ij = ijj,NN-ii
```

```
      do 120  ip = 0,NN
         iqq = 0
      if(ip.eq.0) iqq = 1
      do 130  iq = iqq,NN-ip
      il = ii*(2*NN+3-ii)/2  + ij
      ik = ip*(2*NN+3-ip)/2  + iq
C     Start reading in the 14 arrays needed
      read(11,*) ia, ib, AA(il, ik)
      read(12,*) ia, ib, AB(il, ik)
      read(13,*) ia, ib, AC(il, ik)
      read(14,*) ia, ib, AD(il, ik)
      read(15,*) ia, ib, AE(il, ik)
      read(16,*) ia, ib, AG(il, ik)
      read(17,*) ia, ib, AP(il, ik)
      read(18,*) ia, ib, AQ(il, ik)
      read(19,*) ia, ib, AR(il, ik)
      read(20,*) ia, ib, AT(il, ik)
      read(21,*) ia, ib, AU(il, ik)
      read(22,*) ia, ib, AV(il, ik)
      read(23,*) ia, ib, AX(il, ik)
      read(24,*) ia, ib, AY(il, ik)
130   continue
120   continue
110   continue
```

```
100   continue
C     The value of the determinant of H is found from the subroutine detm.  Where this value
C     changes from pos to neg is where the zeroes are searched for.  To start with, nroot
C     guesses, are evaluated to see where the determinant changes sign.  Set up the guesses.
      do   4999  kkk= 1,nroot
         xguess(kkk) = dble(kkk)/(nroot+1)
4999  continue
C     Find the zeroes using the IMSL subroutine DZBREN
         ERRABS = 0.0d0
         ERRREL = 1.0d-6
         maxfn = 1.0d4
         A = 0.0d0
         B = 0.0d0
         do 5000  kkk = 1, nroot
         write(27, 1060) kkk, xguess(kkk), detm(xguess(kkk))
      A = xguess(kkk)
      if (kkk.EQ.nroot) then
            B = xguess(kkk)
      else
            B = xguess(kkk+1)
      endif
      Aval = detm(A)
         Bval = detm(B)
         if (Aval*Bval.LT.0.0d0)  then
```

```
            maxfn = 1000
            call DZBREN(detm, ERRABS, ERRREL, A, B, maxfn)
            write(26, 1070) B, maxfn
        else
           goto 5000
        endif
5000      continue
1060     format(i5, 3x, d15.8, 3x, d15.9)
1070     format(d15.4, 3x, i10)
1071     format(5x,a15, 3x, a10/)
           stop
           end
C       Real function Detm(x) used to find the determinant of H(l,k) using
C       the IMSL subroutines DLFTRG and DLFDRG.  This uses the LU factorization
C       of the matrix.  This seems to be ok, even though there are zeroes on
C       the main diagonal.
           REAL*8 FUNCTION  DETM(X)
           IMPLICIT  REAL*8(A-H, O-Z)
        parameter (NN = 5, LL = NN*(NN+3)/2, KK = LL)
           parameter (LDA = 20, LDFAC = 20, N = 20)
           integer  IPVT(N)
           real*8  H(LDA, LDA), DET1, DET2,  FAC(LDFAC,LDFAC)
           common/a/ LLL, AA(LL,KK), AB(LL,KK), AC(LL,KK), AD(LL,KK)
           common/a/ AE(LL,KK), AG(LL,KK), AP(LL,KK), AQ(LL,KK)
```

```
      common/a/ AR(LL,KK), AT(LL,KK), AU(LL,KK), AV(LL,KK)
      common/a/ AX(LL,KK), AY(LL,KK)
      common/a/ beta, gamma, f
      DO 1  L = 1,LLL
      DO 2  K = 1,LLL
      H(L,K)=1.0d5*(X**6.0d0*(AA(L,K)+gamma*beta*f*AB(L,K))
     &  +X**4.0d0*(AC(L,K)+gamma*beta*f*AD(L,K)+beta*f*AE(L,K)
     &  +gamma*beta*f*f*(2.0d0-beta)*AG(L,K))
     &  +X**2.0d0*(AP(L,K)+gamma*beta*f*AQ(L,K)+beta*f*AR(L,K)
     &  +gamma*beta*f*f*(2.0d0-beta)*AT(L,K)+beta*beta*f*f*AU(L,K))
     &  +beta*f*AV(L,K) + gamma*beta*f*f*(2.0d0-beta)*AX(L,K)
     &  +beta*beta*f*f*AY(L,K))
2     CONTINUE
1     CONTINUE
      call DLFTRG (N, H, LDA, FAC, LDFAC, IPVT)
      call DLFDRG (N, FAC, LDFAC, IPVT, DET1, DET2)
      detm = det1*10.0**det2
      RETURN
      END
```