

UNIVERSITY OF CALIFORNIA

Los Angeles

A Three-Dimensional Finite Element Model
for Mantle Convection

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Geophysics and Space Physics

by

John Rudolph Baumgardner

1983

© Copyright by

John Rudolph Baumgardner

1983

The dissertation of John Rudolph Baumgardner is approved:

Orson L Anderson

O. L. Anderson

Rich H. Busse

F. H. Busse

Carl R Mechoso

C. R. Mechoso

L A Schmit Jr.

L. A. Schmit, Jr.

G. Peter Bird

G. P. Bird, Committee Chair

University of California, Los Angeles

1983

DEDICATION

I dedicate this dissertation to Jesus of Nazareth, the Messiah, in whom are hidden all the treasures of wisdom and knowledge (Colossians 2:3).

CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	vi
VITA	viii
ABSTRACT OF THE DISSERTATION	ix
1. INTRODUCTION	1
2. BASIC EQUATIONS	5
3. THE GALERKIN METHOD	12
4. THE FINITE-ELEMENT DISCRETIZATION	17
Dyadic Spherical Icosahedral Grid	17
Indexing Conventions	21
Spherical Barycentric Coordinates	26
Approximation Space for the Two-Sphere	28
Approximation Error	30
Solution of a Differential Equation on the Two-Sphere	32
3-D Approximation Spaces	42
3-D Discretization of the Laplacian Operator	45
Discretization of ∇p	50
Discretization of the Buoyancy Force	52
Discrete Form of the Force Balance Equation	54
5. MULTIGRID SOLUTION TECHNIQUE FOR FORCE BALANCE EQUATION	56
The Multigrid Algorithm	56
Least-Squares Local Approximate Inverse	59
6. DISCRETE FORMULATION OF MASS AND ENERGY CONSERVATION	62
Cell-Wall Advection	62
Discretization of Thermal Conduction	64
Second-Order Runge-Kutta Time Integration	70

7.	ILLUSTRATIVE EXAMPLES OF CONVECTION IN SPHERICAL SHELLS	73
	Shell Heated from Below: The Preferred Solution	74
	Shell Heated from Below: Variation with Rayleigh Number	108
	Shell Heated from Within with Random Initial Temperatures	130
	Shell Heated Both from Below and from Within	153
8.	APPLICATION TO THE EARTH'S MANTLE	165
	Mantle Convection with $L = 3$ Initial Temperatures	166
	Mantle Convection with Random Initial Conditions	179
	Mantle Convection with Ridge/Trench Distribution Specifying Initial Temperatures	190
9.	CONCLUSIONS	266
	REFERENCES	269

ACKNOWLEDGEMENTS

I would like to express my gratitude to the individuals in the Institute of Geophysics and Planetary Physics of the University of California, especially to Dr. Orson Anderson, its Director, for providing the grant that supported a major portion of this research at Los Alamos National Laboratory. I thank Dr. Anderson for the special encouragement and advice he gave throughout the project.

I also would like to acknowledge the generous support provided by the Computer Research and Applications Group at Los Alamos National Laboratory and the contributions of Dr. Paul Frederickson in particular. Apart from the genius of Paul's multigrid approximate inverse algorithm and the months of highly stimulating collaboration with Paul, it is difficult to imagine how this undertaking could have succeeded.

I want further to thank Dr. Peter Bird, my research advisor at UCLA, for his encouragement, example, wise counsel, and friendship over the past four years. I am grateful to Peter for the research assistantship support he provided and for the initial introduction he gave me to the finite element method.

I also would like to express my appreciation to the management of Rocketdyne Division of Rockwell International, especially to Phil Briggs, for their generosity in allowing me a schedule to accomodate the many two and three week visits to Los Alamos.

I want to acknowledge the very special role my wife Jean played by her joyful spirit and constructive advice and unfailing confidence as my companion in this effort.

Most of all, I want to acknowledge and thank the God of Abraham, Isaac, and Jacob for the unspeakable favor He granted and the incredible doors He opened during the course of this endeavor.

VITA

January 23, 1944--Born, Marshall, Texas

1968--B.S., Electrical Engineering, Texas Tech University

1970--M.S., Electrical Engineering, Princeton University

1981--M.S., Geophysics and Space Physics, University of California,
Los Angeles

ABSTRACT OF THE DISSERTATION

A Three-Dimensional Finite Element Model for Mantle Convection

by

John Rudolph Baumgardner

Doctor of Philosophy in Geophysics and Space Physics

University of California, Los Angeles, 1983

Professor G. P. Bird, Chair

This dissertation presents a highly efficient numerical approach for solving the conservation equations for momentum, mass, and energy in a spherical shell. The treatment is limited to an infinite Prandtl number Newtonian fluid, although the method is not. An almost uniform discretization is described based on successive dyadic refinements of the mesh obtained by projection of the regular icoshedron onto the sphere. The finite element formulation utilizes a new spherical element defined in terms of spherical barycentric coordinates. A fast multigrid elliptic solver, which takes advantage of the nested character of successive refinements of the icosahedral grid, is used to

solve the equation of motion for the velocity field in order(n) operations, where n is the number of nodes in the three-dimensional grid. The order(n) speed of the algorithm means that high-resolution 3-D problems are practical on current vector computers such as the Cray-1. Time-dependent convection experiments that use a grid with 43,554 nodes are presented.

Application of the numerical method is limited in the dissertation to the case of infinite Prandtl number, almost incompressible convection in non-rotating spherical shells with undeformable, free-slip boundaries, spatially uniform material properties, and a linear, isotropic rheology. Several new results are obtained for a shell radius ratio 0.5 and linearly increasing gravity with radius. When heating is entirely from below, the Nusselt number varies as the Rayleigh number R to the 0.286 power for R greater than 25,000. A preferred pattern with 3 cells which is stable at least to $R = 100,000$ is identified for this mode of heating.

Applied to the earth's mantle using constant gravity and a radius ratio of 0.547, the method yields solutions at $R = 1,000,000$ with plume-like character. Cases initialized with temperatures warmer beneath present ridges and cooler adjacent to present trenches give patterns with upwelling localized to seven plumes that correlate with current mid-ocean ridge and volcanic features. Scaling of these whole-mantle flow results to $R = 20,000,000$ gives a RMS surface velocity of 37 mm/yr.

1. INTRODUCTION

Key to understanding the dynamics of the earth's tectonic history is the ability to model thermal convection in the mantle. Because the problem is nonlinear, three-dimensional, and time-dependent, with rheology a strong function of temperature, its detailed solution has been considered well beyond the reach of existing computers (Schubert, 1979; Torrence, 1979; Boss, 1983). This dissertation represents a step in overcoming the computational barrier by describing a numerical approach with the efficiency to treat three-dimensional, high Rayleigh number convection on currently existing machines. Although the approach has the generality to handle compressible convection with spatially varying (e.g., temperature-dependent) rheology, the dissertation restricts its scope to the case of almost incompressible, constant viscosity flow.

Thermal convection in spherical shells has been studied several investigators. Chandrasekhar (1961) applied linear theory to find the

critical Rayleigh numbers for a variety of spherical cases. Young (1974) used a spherical harmonic representation for tangential field components and a finite-difference representation for the radial to investigate three-dimensional finite-amplitude convection of a Boussinesq fluid in spherical shells of radius ratios 0.3 and 0.6 and Prandtl numbers 1 and 5 for Rayleigh numbers up to about 5 times critical. He found that axisymmetric solutions in general are not the preferred solutions and that the dominant harmonic modes at finite amplitude are usually the modes most unstable to the onset of convection.

Busse (1975) showed that the preferred patterns of convection near the onset of convection can be found from the solvability conditions generated when nonlinear terms are added as perturbations to the linear problem. He found that solutions exhibiting the symmetries of four of the five Platonic bodies are preferred among the various $L = 4$ and $L = 6$ spherical harmonic patterns. This analysis, extended to spherical harmonics of odd degree (Busse and Riahi, 1982), indicated that a pattern with tetrahedral symmetry is preferred at the onset of convection when $L = 3$. Expansion of the dependent variables in the convection equations in powers of the convective amplitude was applied by Riahi et al. (1982) to treat three-dimensional small amplitude convection in spherical shells with radius ratios of 0.33 and 0.5. This study found the $L = 3$ tetrahedral solution to be preferred for the radius ratio 0.5 near the onset of convection.

Zebib et al. (1980) used a Galerkin technique with Legendre polynomial basis functions to investigate axisymmetric, infinite Prandtl number, Boussinesq convection in a spherical shell of radius ratio 0.5 heated only from below for Rayleigh numbers up to about 10 times critical. This study found axisymmetric $L = 3$ and $L = 4$ solutions, but linear stability analysis indicated the $L = 4$ solutions are not stable. The same numerical procedure and assumptions applied to the case of strictly internal heating (Schubert and Zebib, 1980) showed that the only possible steady solutions are fully three-dimensional. These axisymmetric calculations were extended to include other radius ratios by Zebib et al. (1983) for Rayleigh numbers up to about five times critical and reveal many interesting features.

The numerical approach presented in this dissertation exploits the flexibility of the finite element method. Its novelty is the use of an almost uniform triangular discretization of the sphere together with a multigrid technique for solving the equation of motion. The multigrid algorithm solves the system of n equations in $O(n)$ machine operations and thus affords a savings in computational cost similar to that realized by the fast Fourier transform in spectral computations. A new type of finite element for the sphere based on spherical barycentric coordinates is described. An important benefit of the discretization is a data structure that can be readily vectorized on a computer such as the Cray-1 and partitioned for efficient implementation on computers with multiple processors.

The dissertation describes the numerical method in considerable detail. It provides estimates of its accuracy and illustrates its capabilities with examples of convection in spherical shells with undeformable, free-slip, isothermal boundaries. Finally, results for some simplified models of the earth's mantle are presented.

2. BASIC EQUATIONS

In treating mantle convection in the terrestrial planets, it is appropriate to neglect both inertial and Coriolis forces, that is, to assume both the Prandtl number and Ekman number are infinite. The Prandtl number measures the relative magnitude of viscous forces to inertial forces, while the Ekman number describes the relative magnitude of viscous forces to Coriolis forces. Viscous forces are relatively large because of the large values of viscosity associated with the solid state. For the earth's mantle, the Prandtl number is on the order of 10^{23} , and the Ekman number is on the order of 10^9 or greater. Therefore, inertial and Coriolis forces are omitted in the formulation which follows. The centrifugal body force is also neglected, and the planetary mantle is treated as a spherical, instead of a slightly spheroidal, shell.

With these assumptions the following equations, written in Cartesian tensor notation, describe the balance of forces and the conservation of mass and energy for an infinitesimal volume of fluid in a Eulerian reference frame:

$$\partial_I \sigma_{IJ} + \rho g_J = 0 \quad (2.1)$$

$$\partial_t \rho + \partial_I (\rho u_I) = 0 \quad (2.2)$$

$$\partial_t (\rho e) + \partial_I (\rho e u_I) = -\partial_I q_I + H + \sigma_{IJ} \dot{\epsilon}_{IJ} \quad (2.3)$$

Here σ_{ij} represents the stress tensor, ρ the density, g_j the gravitational acceleration, u_i the velocity, e the internal energy, q_i the heat flux per unit area, H the volumetric radiogenic heat production rate, and $\epsilon_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ the rate of strain tensor. The summation convention, $a_i b_i = \sum_{i=1}^3 a_i b_i$, is used here and throughout the dissertation.

It is convenient to resolve the stress tensor σ_{ij} into hydrostatic and viscous components

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij} , \quad (2.4)$$

where p is pressure, δ_{ij} is the Kronecker delta, and τ_{ij} is the viscous stress tensor.

To relate stress to the velocity field of the fluid a constitutive relation is needed. At very low rates of strain, diffusion creep is the dominant mechanism for solid-state flow. This type of creep has a linear, or Newtonian, constitutive law which may be expressed in terms of a dynamic shear viscosity μ and bulk viscosity β as

$$\tau_{ij} = 2\mu(\dot{\epsilon}_{ij} - \delta_{ij}\dot{\epsilon}_{kk}/3) + \beta\delta_{ij}\dot{\epsilon}_{kk} , \quad (2.5)$$

where

$$\mu = \frac{kT R_c^2}{10D_o V_a} \exp \left[\frac{E^* + pV^*}{kT} \right] \quad (2.6)$$

(Herring, 1950). Here k is Boltzmann's constant, T absolute temperature, R_c the crystal radius, D_0 a reference diffusion coefficient, V_a the mean atomic volume, E^* an activation energy, and V^* an activation volume. The bulk viscosity β arises from volume changes which are rapid relative to molecular relaxation processes. For time scales related to mantle flow, the bulk viscosity term therefore may be neglected.

At higher rates of strain, dislocation creep likely plays a significant role. In this case the constitutive law is nonlinear with the stress varying approximately with the one-third power of the strain rate (Weertman and Weertman, 1975). Although strain rates in the mantle almost certainly are high enough for dislocation creep to be occurring, numerical experiments (Christensen) indicate that convection with a nonlinear rheology can be approximated quite well using a linear constitutive law with modified values for the activation energy and volume. This dissertation will restrict treatment to a linear and isotropic constitutive law given by

$$\tau_{IJ} = 2\mu(\dot{\epsilon}_{IJ} - \delta_{IJ}\dot{\epsilon}_{kk}/3) . \quad (2.7)$$

From (2.4) and (2.7), the divergence of the stress tensor in (2.1) may be expressed in terms of velocity as

$$\partial_I \sigma_{IJ} = -\partial_J p + \partial_I \mu \partial_I u_J + \partial_I \mu \partial_J u_I - 2\partial_J \mu \partial_I u_I / 3 . \quad (2.8)$$

Similarly, the shear heating term $\sigma_{ij}\dot{\epsilon}_{ij}$ in the energy conservation equation (2.3) may be written in terms of velocity as

$$\sigma_{ij}\dot{\epsilon}_{ij} = -p\partial_i u_i + \mu(\partial_i u_j + \partial_j u_i)(\partial_i u_j) - 2\mu(\partial_i u_i)^2/3 \quad (2.9)$$

Equation (2.3) can be further expressed in terms of temperature by use of thermodynamic identities. We note that $de = Tds - pdv$ and $ds = (c_p/T)dT - (\alpha/\rho)dp$, where s represents specific entropy, v specific volume, c_p specific heat at constant pressure, and α the volume coefficient of thermal expansion. We also note that (2.2) implies that $\partial_i u_i = (\partial_t v + u_i \partial_i v)\rho$. Finally, with Fourier's law of heat conduction, $q_i = -k\partial_i T$, where k is thermal conductivity, (2.3) may be written

$$\begin{aligned} c_p \partial_t (\rho T) - \frac{\alpha T}{\rho} \partial_t (\rho p) + c_p \partial_i (\rho T u_i) - \frac{\alpha T}{\rho} \partial_i (\rho p u_i) \\ = \partial_i k \partial_i T + H + \mu[(\partial_i u_j + \partial_j u_i)(\partial_i u_j) - \frac{2}{3}(\partial_i u_i)^2] \end{aligned} \quad (2.10)$$

The force balance and conservation of mass and energy under the approximations made to this point, namely, infinite Prandtl number, no rotational forces, and a linear isotropic constitutive law with zero bulk viscosity, may be expressed

$$\partial_i \mu \partial_i u_j + \partial_i \mu \partial_j u_i - \frac{2}{3} \partial_j \mu \partial_i u_i - \partial_j p + \rho g_j = 0 \quad (2.11)$$

$$\partial_t \rho = -\partial_i (\rho u_i) \quad (2.12)$$

$$\begin{aligned} a_+(pT) = & -a_1(pTu_1) + \frac{1}{c_p} \{a_1 k a_1 T + \frac{\alpha T}{\rho} [a_+(p\rho) + a_1(\rho p u_1)] \\ & + H + \mu[(a_1 u_j + a_j u_1)(a_1 u_j) - \frac{2}{3}(a_1 u_1)^2]\} \end{aligned} \quad (2.13)$$

To solve this system of equations an equation of state is needed to relate pressure to density and temperature. The following, based on a Morse potential is an example of the several which have been applied in mantle calculations (Stacey et al., 1981):

$$\begin{aligned} p = & \frac{3K_0}{(K'_0 - 1)} \left(\frac{\rho}{\rho_0}\right)^{2/3} \{\exp [2(K'_0 - 1)(1 - (\rho/\rho_0)^{-1/3})] \\ & - \exp [(K'_0 - 1)(1 - (\rho/\rho_0)^{-1/3})]\} + \alpha K_T T \end{aligned} \quad (2.14)$$

where K_0 is the bulk modulus, K'_0 is the pressure derivative of K_0 , and ρ_0 is the density, all at zero temperature and zero pressure. K_T is the bulk modulus at high temperature (above the Debye temperature) and zero pressure. The thermal contribution to the pressure is given by the term $\alpha K_T T$. When T greatly exceeds the Debye temperature, as is the case for the interiors of the terrestrial planets, the product αK_T is largely independent of temperature and volume, and this simple form for the thermal pressure appears to provide an accurate representation (Anderson and Baumgardner, 1980).

The system of equations (2.11)-(2.14) thus contains the essential physics for treating mantle convection in terms of a linear viscous

rheology. In particular, they account for compressibility and admit spatial variation in all the material parameters.

The finite-element formulation to be described in the following chapters possesses the flexibility and efficiency to include this degree of generality. However, in what is to follow, the treatment will be restricted to a model for which the fluid is almost incompressible and the viscosity and other material parameters have a constant value throughout the domain. This permits use of a scalar instead of a tensor operator in the force balance equation and results in a considerable savings in computational cost. It also allows a more direct comparison with finite-amplitude results of other investigators who have used the Boussinesq approximation.

These additional assumptions, of course, simplify the system of four equations significantly. In the case of the force balance equation (2.11), the viscosity may be taken outside the derivatives and the terms containing the divergence of the velocity neglected. In the energy equation, the adiabatic compression terms may be dropped and the thermal conductivity moved outside the derivative. Shear heating will also be neglected since it can be shown (Turcotte et al., 1974) to be balanced by adiabatic compression terms. A simpler equation of state is now appropriate since only small changes in density are to be allowed. This simplified system may be written

$$\mu \partial_i \partial_i u_j - \partial_j p + \rho g_j = 0 \quad (2.15)$$

$$\partial_t \rho = -\partial_i (\rho u_i) \quad (2.16)$$

$$\partial_t(\rho T) = -\partial_i(\rho T u_i) + (k\partial_i\partial_i T + H)/c_p \quad (2.17)$$

$$p = K_0 \rho/\rho_0 + \alpha K_0 T \quad (2.18)$$

An evident strategy for solving this time-dependent system is to use (2.18) to provide the pressure field for (2.15), to solve (2.15) for the velocity field, to use this velocity field to find the time rates of change in density and temperature from (2.16) and (2.17), and then to take a time step and repeat the process. This is the approach followed in the discrete formulation. The most expensive step in this procedure is the solution of (2.15) for the velocity field. An efficient means for doing this will be described in Chapter 5.

3. THE GALERKIN METHOD

Suppose we are seeking a set $\{u_k\}$ of m unknown functions, each of which depends on m independent variables $\{x_k\}$, that satisfies a specified set of m linear differential equations

$$D_{jk}u_k + d_j = 0, \quad j = 1, \dots, m \quad (3.1)$$

within a domain Ω and also satisfies m linear specified boundary conditions

$$B_{jk}u_k + b_j = 0, \quad j = 1, \dots, m \quad (3.2)$$

on the boundaries Γ of the domain.

Since the set of differential equations (3.1) vanishes everywhere in Ω , it follows that for any arbitrary integrable function w on Ω

$$\int_{\Omega} w(D_{jk}u_k + d_j)d\Omega = 0, \quad j = 1, \dots, m. \quad (3.3)$$

The converse is also true, namely, that if (3.3) is satisfied for any integrable function w on Ω , then (3.1) must be satisfied at all points in Ω . This is verified by observing that if $D_{jk}u_k + d \neq 0$ at some point in Ω then a w can immediately be found which makes the integral expression (3.3) nonzero.

Similar statements can be made concerning the boundary conditions (3.2), namely, that

$$\int_{\Gamma} \bar{w}(B_{jk}u_k + b_j)d\Gamma = 0, \quad j = 1, \dots, m \quad (3.4)$$

for any integrable function \bar{w} on Γ is true if and only if (3.2) is satisfied.

For convenience let us combine (3.3) and (3.4) into a single expression

$$\int_{\Omega} w(D_{jk}u_k + d_j)d\Omega + \int_{\Gamma} \bar{w}(D_{jk}u_k + b_j)d\Gamma = 0, \\ j = 1, \dots, m, \quad (3.5)$$

where w and \bar{w} represent any integrable functions on Ω and Γ , respectively, and note that (3.5) is equivalent to (3.1) and (3.2).

Now let us apply an approximation procedure to this general problem and settle for an approximate solution, that is, a set $\{\hat{u}_k\}$, expressed as a linear combination of a finite set $\{N_\ell\}$ of n basis functions which also are functions of the independent variables $\{x_k\}$. That is,

$$\hat{u}_k = a_{k\ell} N_\ell, \quad k = 1, \dots, m. \quad (3.6)$$

The problem then becomes one of finding the unknown coefficients $a_{k\ell}$.

If we now substitute for u_k in (3.5) the representation for \hat{u}_k of (3.6), we obtain

$$\int_{\Omega} w(D_{jk}a_{k\ell}N_\ell + d_j)d\Omega + \int_{\Gamma} \bar{w}(D_{jk}a_{k\ell}N_\ell + b_j)d\Gamma = 0, \\ j = 1, \dots, m. \quad (3.7)$$

But since w and \bar{w} are arbitrary functions, we can choose sets $\{w_i\}$ and $\{\bar{w}_i\}$ of n linearly independent functions on Ω and Γ , respectively, to provide mn independent equations of the form (3.7):

$$\left[\int_{\Omega} w_i D_{jk} N_{\ell} d\Omega + \int_{\Gamma} \bar{w}_i B_{jk} N_{\ell} d\Gamma \right] a_{k\ell} + \int_{\Omega} w_i d_j d\Omega + \int_{\Gamma} \bar{w}_i b_j d\Gamma = 0, \quad \begin{matrix} j = 1, \dots, m \\ i = 1, \dots, n. \end{matrix} \quad (3.8)$$

Therefore, the problem of finding the mn unknown coefficients $a_{k\ell}$ reduces to that of solving a system of mn simultaneous linear equations

$$A_{ijkl} a_{k\ell} + c_{ij} = 0 \quad \begin{matrix} i = 1, \dots, n \\ j = 1, \dots, m \end{matrix}, \quad (3.9)$$

where

$$A_{ijkl} = \int_{\Omega} w_i D_{jk} N_{\ell} d\Omega + \int_{\Gamma} \bar{w}_i B_{jk} N_{\ell} d\Gamma \quad (3.10)$$

and

$$c_{ij} = \int_{\Omega} w_i d_j d\Omega + \int_{\Gamma} \bar{w}_i b_j d\Gamma. \quad (3.11)$$

The error, or residual, resulting from use of \hat{u}_k instead of u_k in (3.1) is simply $D_{jk} \hat{u}_k + d_j$. The error in (3.2) is $B_{jk} \hat{u}_k + b_j$. Expression (3.8) thus implies that the approximate solution \hat{u}_k obtained through this procedure yields zero integrated error when weighted by each of the functions w_i and \bar{w}_i on Ω and Γ , respectively. This approximation procedure is known therefore as the method of weighted

residuals. It has been applied in various forms since the end of the last century. A wide class of weighting or test functions may be used, but the most frequent choice today is that proposed by Galerkin in 1915. Galerkin selected for the weighting functions $\{w_i\}$ and $\{\bar{w}_i\}$ the basis functions $\{N_\ell\}$ used to represent the approximate solution $\{\hat{u}_k\}$. For this case, (3.9) may be expressed

$$A_{ijkl} a_{k\ell} + c_{ij} = 0, \quad \begin{matrix} i = 1, \dots, n, \\ j = 1, \dots, m, \end{matrix} \quad (3.12)$$

where

$$A_{ijkl} = \int_{\Omega} N_i D_{jk} N_\ell d\Omega + \int_{\Gamma} N_i B_{jk} N_\ell d\Gamma \quad (3.13)$$

and

$$c_{ij} = \int_{\Omega} N_i d_j d\Omega + \int_{\Gamma} N_i b_j d\Gamma. \quad (3.14)$$

The approximate solution $\hat{u}_k = a_{k\ell} N_\ell$ to the linear problem described by (3.1) and (3.2) merely involves solving the linear system of algebraic equations (3.12) for the coefficients $a_{k\ell}$.

Although the class of admissible basis functions for this method is large, restrictions are imposed by the order of the differentiation in the operators D_{jk} and B_{jk} . For the integrals of (3.10) to be well behaved, the basis functions $\{N_\ell\}$ must have their first $p - 1$ derivatives continuous, where p is the highest order derivative in D_{jk} or B_{jk} . Often, however, it is possible to perform an integration by parts to obtain a form with lower order derivatives. This then

reduces the basis function continuity requirements.

The basis functions may be global in extent as, for example, the various harmonic functions widely used in Galerkin-Fourier spectral methods. Or they may be limited to local subdomains of Ω . The latter case is commonly referred to as the finite-element method. It has great generality and is currently being applied to a broad spectrum of continuum problems in engineering and the physical sciences (Zienkiewicz, 1977). Much of its appeal arises from the flexibility that local basis functions provide in handling complex boundaries and spatially varying material properties.

An example in the field of geophysics is a recent finite-element model for the tectonic deformation of southern California that represents the regional network of faults with special curved elements and includes the effects of non-linear rheology, local variations in elevation and heat flow, as well as various patterns of horizontal mantle-drag on the base of the crust (Bird and Baumgardner, 1983).

The finite-element method is the approach developed in chapters to follow for the problem of mantle convection.

4. THE FINITE-ELEMENT DISCRETIZATION

One of the first tasks in applying the procedure described in the previous section is the selection of the finite-dimensional approximation space within which the solution is to be represented. To describe this space it is customary to specify a grid on the domain Ω of the problem and then to associate a local basis function with each node and/or cell in the grid. We begin with a discussion of the grid.

4.1 DYADIC SPHERICAL ICOSAHEDRAL GRID

The mantle is treated as a thick spherical shell. For reasons which will become evident, a grid based on the regular icosahedron is selected to discretize this spherical problem. Such a grid was applied by Vestine et al. (1963) in an analysis of the geomagnetic field. Sadourny, Arakawa, and Mintz (1968) and Williamson (1968) used grids constructed from the regular icosahedron in finite-difference integration on the sphere of the nondivergent barotropic vorticity equation. Cullen (1974) employed a similar grid with a finite-element approach to integrate the shallow water equations on the sphere. These investigators all point out the benefits of this discretization, which has elements nearly equal in area and shape.

There are several ways to construct a grid based on the icosahedron. The method to be described here is somewhat different from those just mentioned and leads to several additional advantages. The initial step is to project onto the unit sphere the twelve vertices of the regular icosahedron. These twelve points define a mesh consisting of twenty equal spherical triangles (Figure 4.1a).

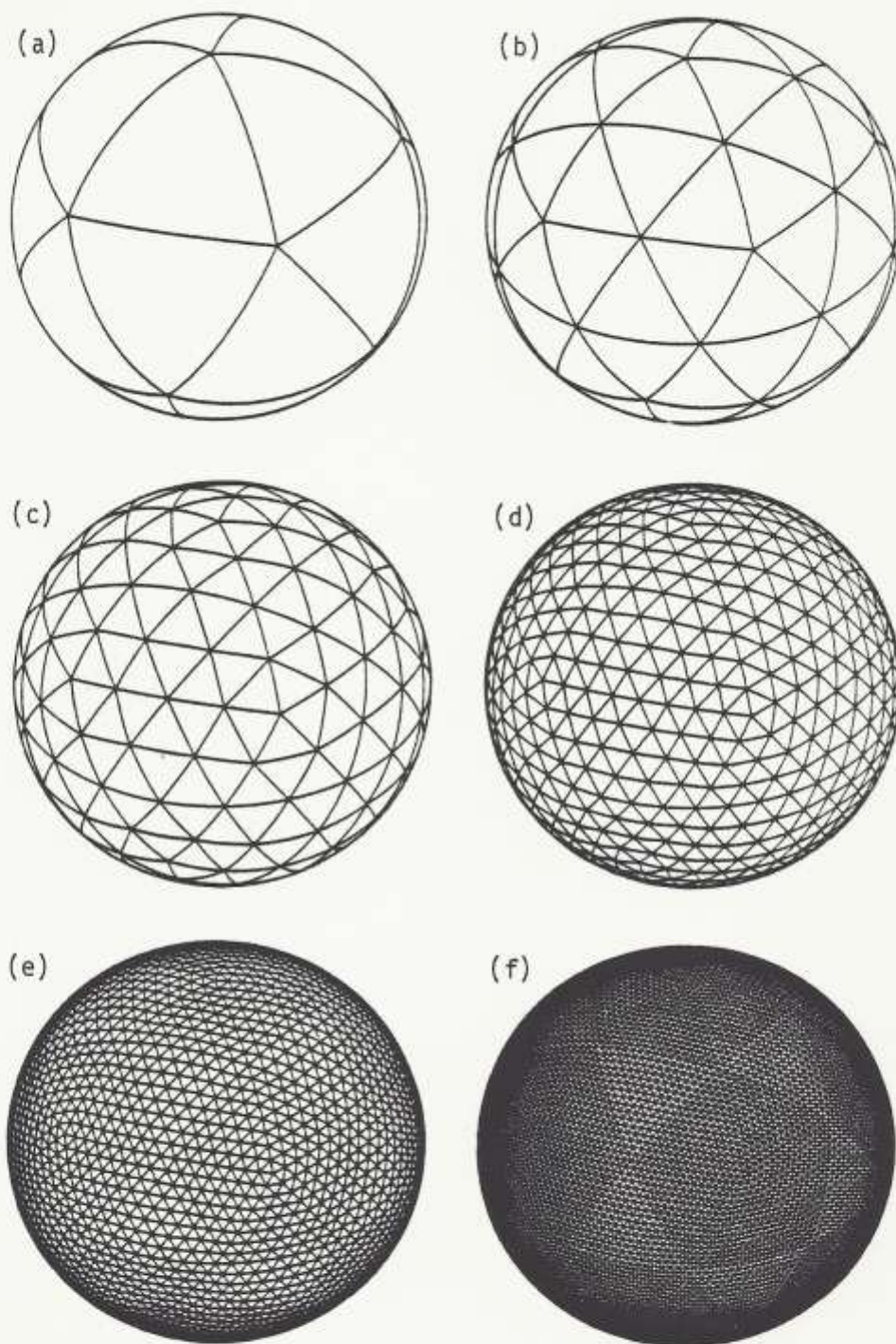


Figure 4.1. Dyadic refinements of grid produced by projection of the regular icoahedron onto the sphere: (a) original grid, (b) first refinement, (c) second refinement, (d) third refinement, (e) fourth refinement, (f) fifth refinement.

A dyadic mesh refinement procedure is then used to generate a mesh of the desired resolution. The refinement technique simply connects the midpoints of the three sides of each spherical triangle with geodesic, or great circle, arcs to divide it into four subtriangles. This produces a refined mesh with four times the number of spherical triangles and approximately four times the number of nodes. Figure 4.1b-f shows five such successive dyadic refinements of the original spherical icosahedron. We note that there are $2 + 10n^2$ nodes and $20n^2$ elementary triangles in the k^{th} refinement, where $n = 2^k$. Each node is surrounded by six triangles except for the twelve icosahedral vertices of the initial mesh. These are surrounded by five triangles and thus will be referred to as pentagonal nodes.

4.2 INDEXING CONVENTIONS

This icosahedral grid for the sphere leads to a convenient data structure for numerical calculations. Suppose, as illustrated in Figure 4.2, that two antipodal pentagonal vertices are designated north and south poles, respectively. Then ten separate pairs of spherical icosahedral triangles may be identified, each of which forms a rhombus or diamond, with five such diamonds surrounding the north pole and five the south pole. These may be indexed as shown in Figure 4.2. Furthermore, the nodes associated with a given diamond can be indexed with a pair of indices (i_1, i_2) as shown in Figure 4.3. These same indices may also be used to label the triangles or elements in the grid. Two triangles may be identified uniquely with each node (except for the polar nodes). A third index, designated i_e , is utilized to distinguish the two elements. The convention chosen is to assign the triangle closest to the pole the index $i_e = 1$ and the other the index $i_e = 2$. Figure 4.3 illustrates this convention for the two triangles associated with node $(6, 1)$.

For programming convenience, $(n + 2) \times (n + 2)$ nodes are indexed on each diamond. This provides a complete set of boundary points for the $n \times n$ points naturally identified with each diamond. With periodic refreshing of field values at these boundary points, much of the computation for a given diamond can be done independently from the others. This suggests a logical way to partition a problem for a computer with multiple processors. One diamond, or a small number of connected diamonds, is assigned to each processor. During the course

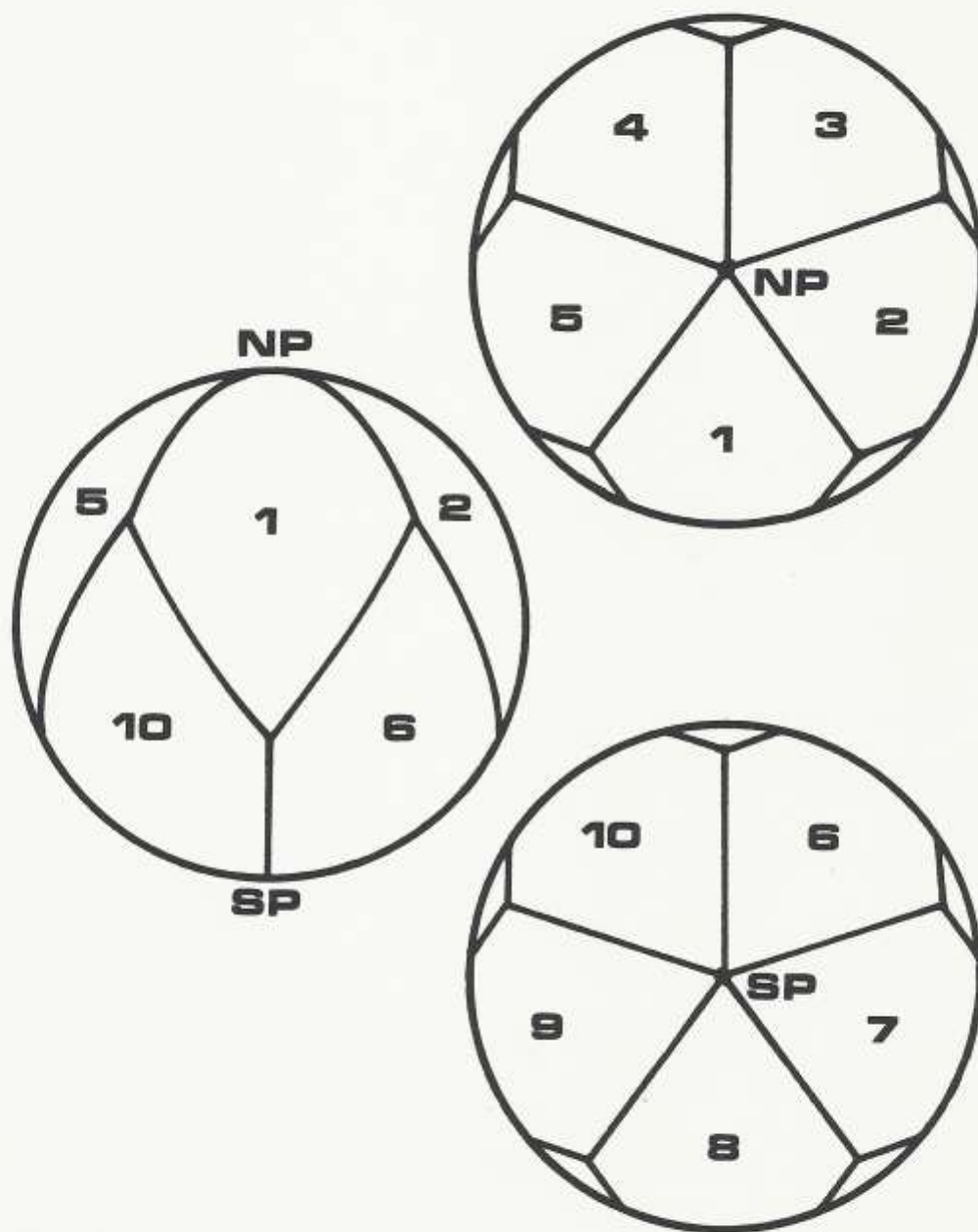


Figure 4.2. Indexing convention for the ten icosahedral diamonds.

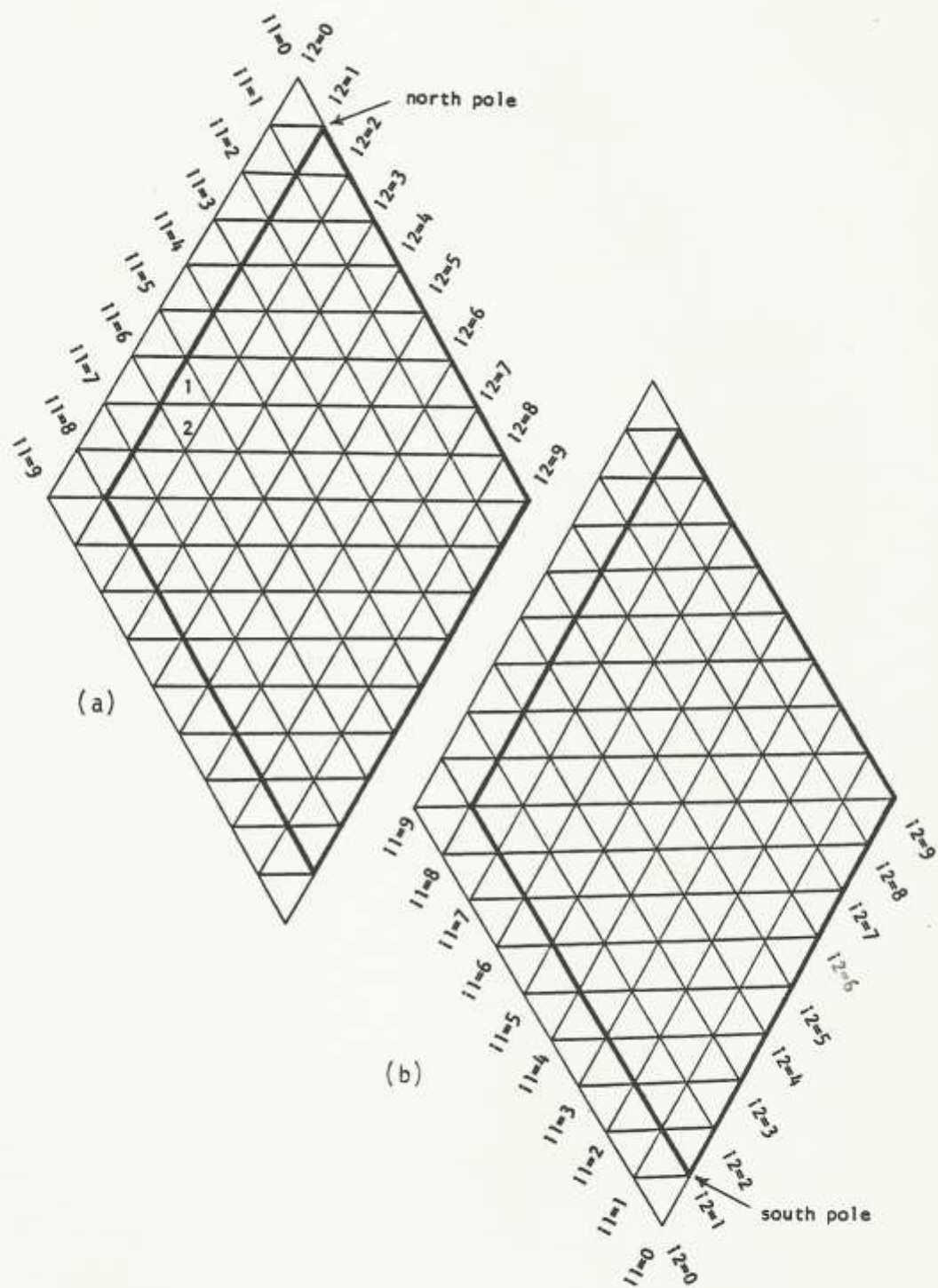


Figure 4.3. Indexing convention for the nodes and triangles of an icosahedral diamond associated with (a) the north pole and (b) the south pole.

of the calculation, boundary information is periodically exchanged among processors. Since the boundaries are of lower dimensionality than the diamonds themselves, relatively high efficiency may be achieved. For a vector processor, the two indices (i_1, i_2) can frequently be combined into a single index to obtain long vectors of field data with consequent improvement in computational speed.

An attractive feature of the icosahedral grid to be noted at this point is its symmetry. A scalar operator like the Laplacian, for example, need be generated and stored only for one of the ten diamonds. Indeed, each diamond itself has additional symmetries which can be exploited with even further savings in operator storage.

Still another benefit of the hierarchy of grids constructed by successive dyadic refinements is that all the nodes of the coarser grids are contained in the finer grids, and the nodes in the finer grids are related in a simple way to the nodes in the coarser grids. This provides for simple and inexpensive interpolation and projection of fields among grid levels. This feature is a key to the high performance of the multigrid elliptic solver to be considered in Chapter 5.

The dyadic icosahedral mesh can be replicated in the radial direction to discretize a thick spherical shell. The procedure is to assign a set of the $2 + 10n^2$ nodes belonging to a given dyadic refinement to each of $n + 1$ radial positions in the shell. Two of these radial positions are the inner and outer shell boundaries. The indexing convention shown in Figure 4.4 assigns index $i_3 = 1$ to the set of nodes at the inner boundary and $i_3 = n + 1$ to the set of nodes

at the outer boundary. These $n + 1$ sets of nodes are aligned so that all nodes lie along $2 + 10n^2$ distinct radial vectors. The three-dimensional elements defined by this grid are the triangular prisms evident in Figure 4.4. There are n layers of such elements. Note that the spacing of the $n - 1$ radial node positions in the shell interior may be nonuniform. Indeed, because of the boundary layer character of the convective flow, it is advantageous to concentrate the nodes near the shell boundaries. Before considering further the discretization of the full spherical shell, however, it will be useful to focus attention on some additional features of the icosahedral discretization of the two-sphere itself.

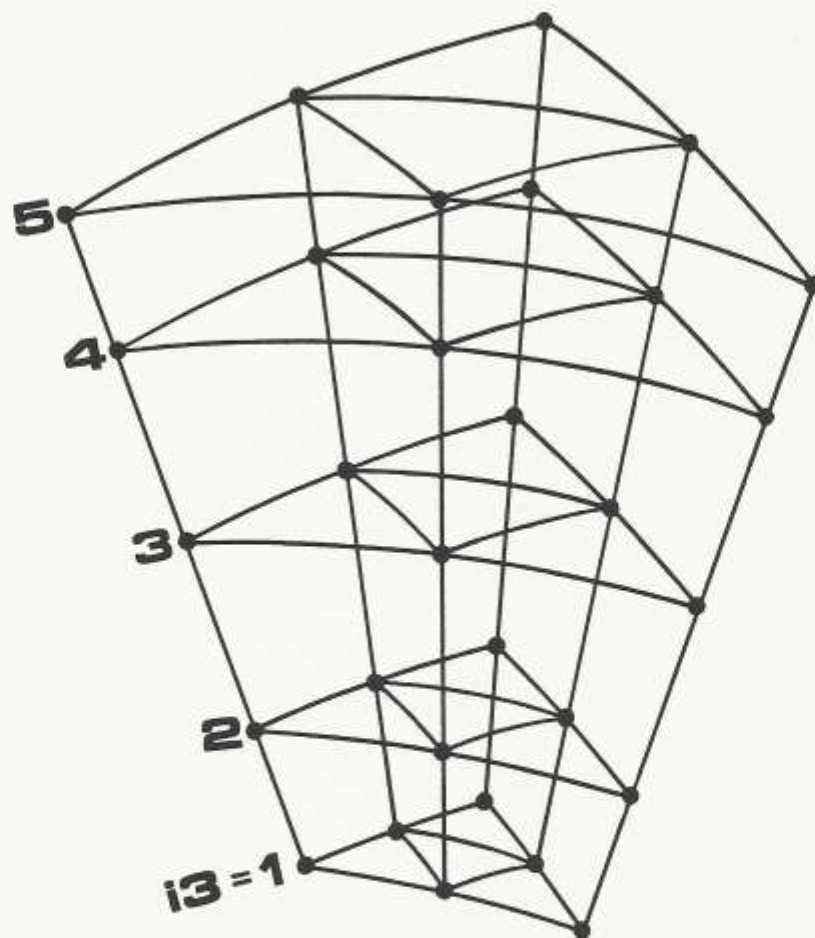


Figure 4.4. Indexing convention for radial discretization of a spherical shell.

4.3 SPHERICAL BARYCENTRIC COORDINATES

The dyadic refinement procedure provides a convenient means for determining what will be referred to as spherical barycentric coordinates for the spherical triangles. These spherical barycentric coordinates in turn are used to define the special finite elements on which the discrete formulation is based.

Definition 4.1: The vector function $\eta = (\eta_1, \eta_2, \eta_3)$ is defined as the barycentric coordinates for the triangle T if it satisfies the following four conditions:

- i) η is an affine function of arc length on the three edges of T
- ii) η takes the values $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$ respectively at vertices 1, 2, and 3 of T .
- iii) η is continuous on T .
- iv) η is related to the barycentric coordinates η^1 , η^2 , η^3 , and η^4 on the four subtriangles T_1 , T_2 , T_3 , and T_4 of the dyadic subdivision of T by the equation (see Figure 4.5 for the convention used in labeling the subtriangles).

$$\eta = (\eta_1, \eta_2, \eta_3) = \begin{cases} (\frac{1}{2} + \frac{1}{2}\eta_1^1, \frac{1}{2}\eta_2^1, \frac{1}{2}\eta_3^1) & \text{on } T_1 \\ (\frac{1}{2}\eta_1^2, \frac{1}{2} + \frac{1}{2}\eta_2^2, \frac{1}{2}\eta_3^2) & \text{on } T_2 \\ (\frac{1}{2}\eta_1^3, \frac{1}{2}\eta_2^3, \frac{1}{2} + \frac{1}{2}\eta_3^3) & \text{on } T_3 \\ (\frac{1}{2} - \frac{1}{2}\eta_1^4, \frac{1}{2} - \frac{1}{2}\eta_2^4, \frac{1}{2} + \frac{1}{2}\eta_3^4) & \text{on } T_4 \end{cases} \quad (4.1)$$

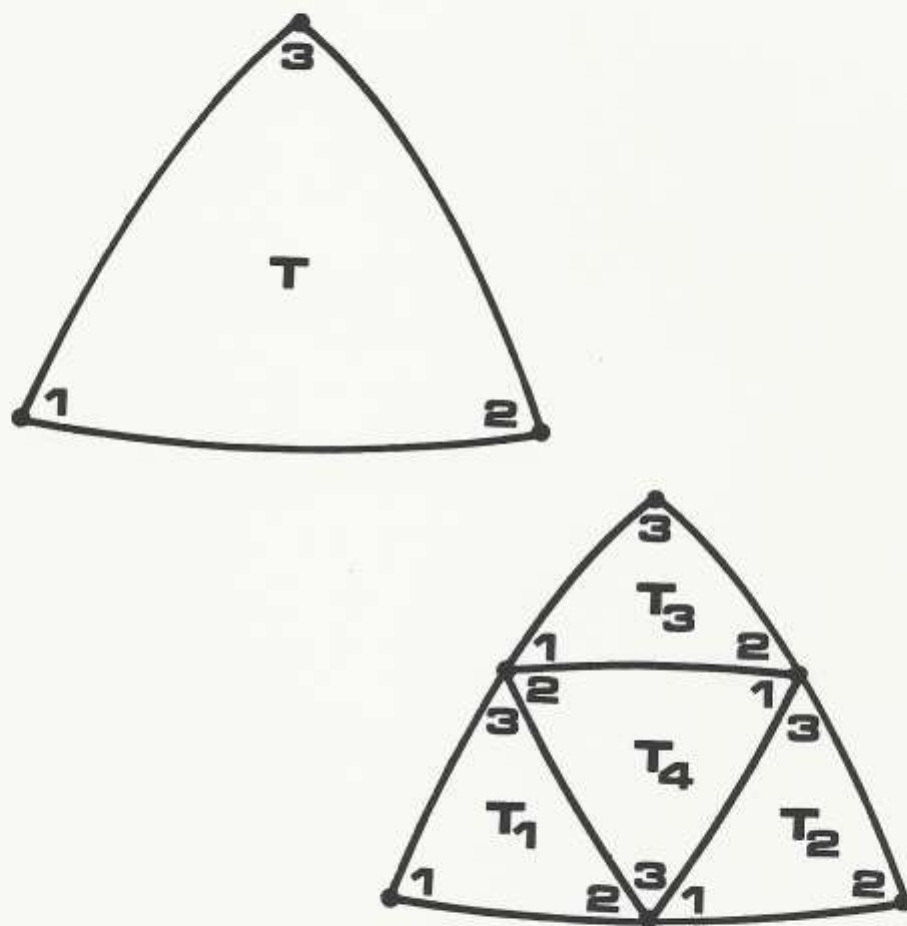


Figure 4.5. Convention for labeling triangles and vertices in the dyadic subdivision of a spherical triangle.

We note that all four conditions of Definition 4.1 are satisfied by barycentric coordinates on a plane triangle. Condition (i) is weaker than the usual requirement that η be affine within the triangle as well as on its edges. This is the reason for the other three conditions and condition (iv) in particular. One observes that (i) and (ii) together imply that η assumes the values $(0, \frac{1}{2}, \frac{1}{2})$, $(\frac{1}{2}, 0, \frac{1}{2})$, and $(\frac{1}{2}, \frac{1}{2}, 0)$ at the midpoints of the three sides of T . Furthermore, (iv) together with (i) imply that η is affine on the three geodesic arcs connecting these vertices. Moreover, (iv) allows us to connect the midpoints of the sides of the smaller triangles in the same fashion. If we can continue the dyadic refinement process, we find that η is defined on a dense network of geodesic arcs within triangle T , and the continuity specified in (iii) defines η uniquely everywhere with T .

4.4 APPROXIMATION SPACE FOR THE TWO-SPHERE

The spherical barycentric coordinates just described immediately provide a means for defining a finite-dimensional approximation space on the two-sphere.

Definition 4.2: For any given spherical triangulation define the space S to be the space of all continuous real functions s on the sphere with the property that for each triangle T in the triangulation there is a triple (a_1, a_2, a_3) of real numbers such that

$$s(x) = a_1 n_1 + a_2 n_2 + a_3 n_3 \quad (4.2)$$

for all points x in T . These functions s belonging to S vary in an affine manner along the geodesic arcs which define the triangulation just as do the barycentric coordinates from which they may be constructed. In general, they display slope discontinuities at triangle boundaries. They are analogous to the piecewise linear functions defined on triangulations of the plane.

A local basis for S in terms of the nodes of the triangulation may readily be defined.

Definition 4.3: For each node i of a given spherical triangulation, denote by L_i that element of the spherical finite-element space S which assumes the value 1 at node i and vanishes at all other nodes. The set of functions $\{L_i\}$ define a basis for S ; that is, for every $s \in S$ there is an array of coefficients $\{s_i\}$, one for each node, such that

$$s(x) = \sum_i s_i L_i(x) \quad (4.3)$$

for all points x on the sphere.

That $\{L_i\}$ is a basis for S is evident, since each barycentric coordinate for each of the triangles is represented in $\{L_i\}$. Indeed, the basis function L_i assumes identically the value of the barycentric coordinate associated with node i in each of the five or six surrounding triangles.

4.5 APPROXIMATION ERROR

One measure of the resolution of a spherical triangulation T is the maximum arc length h of any edge in the triangulation. A slightly more sensitive measure is \bar{h} given by

$$\bar{h}(T) = \lim_{m \rightarrow \infty} 2^m h(T^m), \quad (4.4)$$

where T^m denotes the m^{th} dyadic refinement of the triangulation T . Numerically it may be computed that $\bar{h} = 1.32317 \cdot 2^{-k} r$ for the k^{th} dyadic refinement of the spherical icosahedron with radius r .

A bound on the error in the representation of a smooth function u in the finite-element space S is provided by the following theorem:

Theorem 4.1: For any dyadic refinement T of the regular spherical icosahedron and any function u with a bounded second derivative u'' , there is a function s in the spherical finite-element space S such that

$$\|s - u\|_{\infty} \leq \frac{\bar{h}^2}{6} \|u''\|_{\infty}, \quad (4.5)$$

where \bar{h} is the mesh constant of the triangulation.

Proof: Let s be the element of S which agrees at all nodes of the triangulation. At the midpoint of any side in the triangulation, one has

$$|s - u| \leq \frac{\bar{h}^2}{8} \|u''\|_{\infty} \quad (4.6)$$

since $|s-u|$ can have no more than parabolic variation between two nodes with its maximum value given by $\frac{1}{2}\|u''\|_{\infty}(\bar{h}/2)^2$. At midpoints of the edges of a further dyadic refinement T' of T one has

$$|s - u| \leq \frac{\bar{h}^2}{8} + \frac{\bar{h}^2}{32} \|u''\|_{\infty} . \quad (4.7)$$

Repeating the process, one finds that at the nodes of the m^{th} refinement

$$|s - u| \leq \frac{\bar{h}^2}{8} (1 + \dots + \frac{1}{4^m}) \|u''\|_{\infty} . \quad (4.8)$$

The bound $\|u''\|_{\infty} \bar{h}^2/6$ follows upon taking the limit, since both u and s are continuous.

Corollary: The space S_n of spherical finite elements on the dyadic icosahedral triangulation with $20n^2$ elements of the unit sphere has a member s which satisfies

$$\|s - u\|_{\infty} < 0.221 n^{-2} \|u''\|_{\infty} . \quad (4.9)$$

Proof: It may be determined numerically that $\bar{h} \leq 1.3232/n$ for the dyadic triangulations of the unit sphere.

We note at this point that Theorem 4.1 implies that the spherical finite-element spaces of Definition 4.2 approximate smooth functions to second order. That is, each dyadic refinement reduces the approximation error by a factor of four.

4.6 SOLUTION OF A DIFFERENTIAL EQUATION ON THE TWO-SPHERE

Next let us consider the discretization of a differential equation using the spherical finite-element approximation spaces just described. Let us apply the Galerkin formulation (3.12), assume free boundary conditions, and take the differential operation D_{jk} to be the Laplacian, $\partial_m \partial_m$, with the Cartesian spatial coordinates $x = (x_1, x_2, x_3)$ as independent variables. The system then for which we seek a solution is

$$\partial_m \partial_m u_k(x) + d_k(x) = 0, \quad k = 1, 2, 3. \quad (4.10)$$

At this point we allow d_k to be any integrable function on the sphere. The Galerkin procedure yields an approximate solution $\hat{u}_k(x)$ in terms of the spherical finite-element basis function $\{L_i\}$,

$$\hat{u}_k(x) = a_{kj} L_j(x). \quad (4.11)$$

The coefficients a_{kj} are obtained by solving the linear systems

$$A_{ij} a_{kj} + c_{ik} = 0, \quad i, j = 1, \dots, 2+10n^2, \quad k = 1, 3, \quad (4.12)$$

where

$$A_{ij} = \int_{\text{sphere}} L_i(x) \partial_m \partial_m L_j(x) da \quad (4.13)$$

and

$$c_{ik} = \int_{\text{sphere}} L_i(x) d_k da. \quad (4.14)$$

Note that this formulation is simplified compared with (3.12) because the Laplacian is a scalar operator. Here a denotes area and $n = 2^p$, where p is the dyadic refinement index. Integrating (4.13) by parts yields

$$A_{IJ} = - \int_{\text{sphere}} [\partial_m L_I(x)] [\partial_m L_J(x)] da . \quad (4.15)$$

Representation (4.15) is sometimes referred to as the weak form of representation (4.13). The weak form is required here since the spherical basis functions have only C_0 continuity.

Let us examine how the matrices A_{ij} and c_{ik} may be computed. For notational simplicity let $\langle y, z \rangle$ denote integration of the inner product $y \cdot z$ over the two-sphere. A_{ij} and c_{ik} may then be expressed

$$A_{IJ} = - \langle \partial_m L_I, \partial_m L_J \rangle \quad (4.16)$$

$$c_{ik} = \langle L_I, d_k \rangle . \quad (4.17)$$

As noted previously, the basis function L_i assumes the value of the barycentric coordinate associated with node i in each of the surrounding five or six triangles. That is, if triangle T has node i as a vertex and $x \in T$, then

$$L_i(x) = \eta_i^T(x) . \quad (4.18)$$

It is therefore sufficient to compute the quantities $\langle \partial_m \eta_i, \partial_m \eta_j \rangle$ and $\langle \eta_i, d_k \rangle$ on each of the triangles and then use these quantities to assemble the matrices $\langle \partial_m L_i, \partial_m L_j \rangle$ and $\langle L_i, d_k \rangle$.

To achieve a high level of accuracy it is convenient to compute $\langle \partial_m \eta_i, \partial_m \eta_j \rangle$ and $\langle \eta_i, d_k \rangle$ on a grid much finer than that used for (4.3). Simple recursion formulas, based on the relationship of the barycentric coordinates of successive dyadic refinements given in Definition 4.1, may be derived for interpolating these quantities to coarser meshes. Referring to equation (4.1), one finds, for example, that $\langle \partial_m \eta_i, \partial_m \eta_j \rangle$ for a triangle on a given mesh can be expressed in terms of the four subtriangles on the next dyadic refinement as

$$\langle \partial_m \eta_i, \partial_m \eta_j \rangle = \frac{1}{4} \sum_{p=1}^4 \langle \partial_m \eta_i^p, \partial_m \eta_j^p \rangle \quad . \quad (4.19)$$

In a similar fashion one obtains for $\langle \eta_i, d_k \rangle$ the relation

$$\langle \eta_i, d_k \rangle = \frac{1}{2} \left[\sum_{p=1}^4 \langle \eta_i^p, d_k \rangle + \langle 1, d_k \rangle - \langle \eta_1^4, d_k \rangle \right] \quad . \quad (4.20)$$

Table 4.1 provides a complete list of the recursion formulas for zero-, first-, and second-order operators in terms of the approximation space S .

The task of computing $\langle \partial_m \eta_i, \partial_m \eta_j \rangle$ on a spherical triangle will now be addressed. The quantity $\partial_m \eta_i$ represents the spatial gradient of

Table 4.1. Recursion relations for elementary operators on the two-sphere.

$$\langle 1, 1 \rangle = \sum_{p=1}^4 \langle 1, 1 \rangle^p$$

$$\langle \eta_i, 1 \rangle = \frac{1}{2} \left[\sum_{p=1}^3 \langle \eta_i, 1 \rangle^p - \langle \eta_i, 1 \rangle^4 + \langle 1, 1 \rangle^i + \langle 1, 1 \rangle^4 \right]$$

$$\langle \hat{r}_k, 1 \rangle = \sum_{p=1}^4 \langle \hat{r}_k, 1 \rangle^p$$

$$\langle \partial_k \eta_i, 1 \rangle = \frac{1}{2} \left[\sum_{p=1}^3 \langle \partial_k \eta_i, 1 \rangle^p - \langle \partial_k \eta_i, 1 \rangle^4 \right]$$

$$\langle \eta_i, \hat{r}_k \rangle = \frac{1}{2} \left[\sum_{p=1}^3 \langle \eta_i, \hat{r}_k \rangle^p - \langle \eta_i, \hat{r}_k \rangle^4 + \langle \hat{r}_k, 1 \rangle^i + \langle \hat{r}_k, 1 \rangle^4 \right]$$

$$\langle \partial_k \eta_i, \partial_k \eta_j \rangle = \frac{1}{4} \left[\sum_{p=1}^4 \langle \partial_k \eta_i, \partial_k \eta_j \rangle^p \right]$$

$$\begin{aligned} \langle \eta_i, \eta_j \rangle &= \frac{1}{4} \left[\sum_{p=1}^4 \langle \eta_i, \eta_j \rangle^p + \langle 1, 1 \rangle^i \delta_{ij} + \langle \eta_i, 1 \rangle^j + \langle \eta_j, 1 \rangle^i + \langle 1, 1 \rangle^4 \right. \\ &\quad \left. - \langle \eta_i, 1 \rangle^4 - \langle \eta_j, 1 \rangle^4 \right] \end{aligned}$$

$$\langle \hat{r}_k, \hat{r}_m \rangle = \sum_{p=1}^4 \langle \hat{r}_k, \hat{r}_m \rangle^p$$

$$\langle \eta_i, \partial_k \eta_j \rangle = \frac{1}{4} \left[\sum_{p=1}^4 \langle \eta_i, \partial_k \eta_j \rangle^p + \langle \partial_k \eta_j, 1 \rangle^i - \langle \partial_k \eta_j, 1 \rangle^4 \right]$$

$$\begin{aligned} \langle \eta_i, \hat{r}_k \eta_j \rangle &= \frac{1}{4} \left[\sum_{p=1}^4 \langle \eta_i, \hat{r}_k \eta_j \rangle^p + \langle \hat{r}_k, 1 \rangle^i \delta_{ij} + \langle \eta_i, \hat{r}_k \rangle^j + \langle \eta_j, \hat{r}_k \rangle^i \right. \\ &\quad \left. + \langle \hat{r}_k, 1 \rangle^4 - \langle \eta_i, \hat{r}_k \rangle^4 - \langle \eta_j, \hat{r}_k \rangle^4 \right] \end{aligned}$$

Table 4.1 (Cont.)

$$\langle \partial_{k\eta_i}, \hat{r}_m \rangle = \frac{1}{2} \left[\sum_{p=1}^3 \langle \partial_{k\eta_i}, \hat{r}_m \rangle^p - \langle \partial_{k\eta_i}, \hat{r}_m \rangle^4 \right]$$

$$\langle \hat{r}_{k\eta_i}, \hat{r}_m \rangle = \frac{1}{2} \left[\sum_{p=1}^3 \langle \hat{r}_{k\eta_i}, \hat{r}_m \rangle^p - \langle \hat{r}_{k\eta_i}, \hat{r}_m \rangle^4 + \langle \hat{r}_k, \hat{r}_m \rangle^i + \langle \hat{r}_k, \hat{r}_m \rangle^4 \right]$$

$$\langle \partial_{k\eta_i}, \partial_{m\eta_j} \rangle = \frac{1}{4} \sum_{p=1}^4 \langle \partial_{k\eta_i}, \partial_{m\eta_j} \rangle^p$$

$$\langle \partial_{k\eta_i}, \hat{r}_{m\eta_j} \rangle = \frac{1}{4} \left[\sum_{p=1}^4 \langle \partial_{k\eta_i}, \hat{r}_{m\eta_j} \rangle^p + \langle \partial_{k\eta_i}, \hat{r}_m \rangle^j - \langle \partial_{k\eta_i}, \hat{r}_m \rangle^4 \right]$$

$$\begin{aligned} \langle \hat{r}_{k\eta_i}, \hat{r}_{m\eta_j} \rangle &= \frac{1}{4} \left[\sum_{p=1}^4 \langle \hat{r}_{k\eta_i}, \hat{r}_{m\eta_j} \rangle^p + \langle \hat{r}_k, \hat{r}_m \rangle^i \delta_{ij} + \langle \hat{r}_{k\eta_i}, \hat{r}_m \rangle^j \right. \\ &\quad \left. + \langle \hat{r}_{m\eta_j}, \hat{r}_k \rangle^i + \langle \hat{r}_k, \hat{r}_m \rangle^4 - \langle \hat{r}_{k\eta_i}, \hat{r}_m \rangle^4 - \langle \hat{r}_{m\eta_j}, \hat{r}_k \rangle^4 \right] \end{aligned}$$

NOTE: Subtriangles are defined as shown in Figure 4.5. The quantity \hat{r}_k is the unit vector in the radial direction. Integration is over individual triangles.

the spherical barycentric coordinate η_i . In the case of a plane triangle, the gradient of a barycentric coordinate has a direction perpendicular to its associated side, lies in the plane of the triangle, and has a constant magnitude equal to one divided by the perpendicular distance between the side and the node. For a spherical triangle, on the other hand, the direction of the gradient varies over the spherical surface. As an integrated or averaged value is required, a quadrature approach is taken. The scheme uses a weighted sum of unit vectors normal to the planes defined by the triangle edges and the center of the sphere and pointing inward relative to the triangle. Let q_{jm} denote the j^{th} component of the unit normal associated with triangle side m , θ_m the angle subtended by triangle side m relative to the center of the sphere, i' and i'' the nodes in the triangle different from i , ℓ' and ℓ'' the nodes different from ℓ , and A the area of the triangle. The quadrature formula may then be expressed

$$\begin{aligned} \langle \partial_j \eta_i, \partial_j \eta_\ell \rangle &= \sum_{j=1}^3 (\theta_1 q_{j1}/3 - \theta_{1'} q_{j1'}/6 - \theta_{1''} q_{j1''}/6) \\ &\quad \cdot (\theta_\ell q_{j\ell}/3 - \theta_{\ell'} q_{j\ell'}/6 - \theta_{\ell''} q_{j\ell''}/6) / A \end{aligned} \quad (4.21)$$

It may readily be verified that in the limit the formula reduces to the result obtainable for a plane triangle. Numerical experience shows this formulation to yield highly satisfactory results. Table 4.2 gives a list of formulas for computing each of the zero-, first-, and second-order differential operators for the spherical icosahedral grid. Some of these like (4.21) are approximations, while others are exact.

Table 4.2. Formulas for elementary operators on the two-sphere

$$\langle 1, 1 \rangle = A$$

$$\langle n_i, 1 \rangle = A/3$$

$$\langle \hat{r}_k, 1 \rangle = A(x_{k1} + x_{k2} + x_{k3})/3$$

$$\langle \partial_k n_i, 1 \rangle = (\phi_i q_{ki} - \frac{1}{2} \phi_i q_{ki} - \frac{1}{2} \phi_i q_{ki})/3$$

$$\langle n_i, \hat{r}_k \rangle = A \bar{r}_{ki}$$

$$\langle \partial_k n_i, \partial_k n_j \rangle = \langle \partial_k n_i, 1 \rangle \langle \partial_k n_j, 1 \rangle / A$$

$$\langle n_i, n_j \rangle = A(1 + \delta_{ij})/12$$

$$\langle \hat{r}_k, \hat{r}_m \rangle = \langle \hat{r}_k, 1 \rangle \langle \hat{r}_m, 1 \rangle / A$$

$$\langle n_i, \partial_k n_j \rangle = \langle \partial_k n_j, 1 \rangle / 3$$

$$\langle n_i, \hat{r}_k n_j \rangle = A \bar{r}_{kj} (1 + \delta_{ij}) / 12$$

$$\langle \partial_k n_i, \hat{r}_m \rangle = \langle \partial_k n_i, 1 \rangle \bar{r}_{mi}$$

Table 4.2 (Cont.)

$$\langle \hat{r}_{n_i}, \hat{r}_m \rangle = A \bar{r}_{ki} \bar{r}_{mi} / 3$$

$$\langle \partial_{k n_i}, \partial_{m n_j} \rangle = \langle \partial_{k n_i}, 1 \rangle \langle \partial_{m n_j}, 1 \rangle / A$$

$$\langle \partial_{k n_i}, \hat{r}_{m n_j} \rangle = \langle \partial_{k n_i}, 1 \rangle \bar{r}_{mj} / 3$$

$$\langle \hat{r}_{k n_i}, \hat{r}_{m n_j} \rangle = A \bar{r}_{ki} \bar{r}_{mj} (1 + \delta_{ij}) / 12$$

ϕ_i = angle subtended relative to center of sphere by triangle side i

x_{ki} = Cartesian coordinate k of triangle vertex i

q_{ki} = Cartesian component k of the unit normal to the plane containing triangle side i and the center of the sphere, pointing inward relative to the triangle i

$\bar{r}_{ki} = (x_{ki} + \frac{1}{2}x_{ki'} + \frac{1}{2}x_{ki''})/2$, where i' and i'' denote the triangle vertices different from i

$$A = 4 \tan^{-1} \left\{ \left[\tan[(\phi_1 + \phi_2 + \phi_3)/4] \tan[(\phi_1 + \phi_2)/4] \tan[(\phi_2 + \phi_3)/4] \right. \right. \\ \left. \left. \cdot \tan[(\phi_3 + \phi_1)/4] \right]^{1/2} \right\}$$

δ_{ij} = Kronecker delta

To illustrate the practicality of the approach described thus far, a numerical experiment will now be presented (Baumgardner and Frederickson, 1985). Consider the differential equation

$$\partial_m \partial_m u_k = S_k, \quad (4.22)$$

where S_k is a vector (toroidal) spherical harmonic of Legendre degree L . Since the spherical harmonics are eigenfunctions of the Laplacian operator on the two-sphere, with eigenvalue $L(L + 1)$,

$$\partial_m \partial_m S_k = L(L + 1)S_k, \quad (4.23)$$

we have a convenient analytical check for the numerical solutions.

The system of equations (4.12) is solved using the fast multigrid elliptic solver described in Chapter 5. The function d_k in (4.17) is the spherical harmonic S_k . Results for three harmonic degrees and four mesh refinements are summarized in Table 4.3.

In all cases, the discrete operator $\langle \partial_m L_i, \partial_m L_j \rangle$ was generated on the $n = 128$ grid and projected to the level at which it was used. Similarly, the spherical harmonic was generated at $n = 256$, and a quadrature procedure used to evaluate $\langle L_i, S_k \rangle$. The intent was to make these two sources of error small relative to the intrinsic error treated in Theorem 4.1.

Two measures of error are given in Table 4.3. The first is the quotient $\langle S_k, S_k \rangle / \langle \hat{u}_k, S_k \rangle$ of inner products that would equal the eigenvalue of the toroidal spherical harmonic had the computation been

Table 4.3. Performance of discrete Laplacian operator on the two-sphere using spherical harmonics of degree L .

Grid n	$L = 2$		$L = 4$		$L = 8$		Cray-1 CPU Seconds
	$\frac{\langle S_k, S_k \rangle}{\langle \hat{u}_k, S_k \rangle}$	$\frac{\ u - \hat{u}\ _\infty}{\ u\ _\infty}$	$\frac{\langle S_k, S_k \rangle}{\langle \hat{u}_k, S_k \rangle}$	$\frac{\ u - \hat{u}\ _\infty}{\ u\ _\infty}$	$\frac{\langle S_k, S_k \rangle}{\langle \hat{u}_k, S_k \rangle}$	$\frac{\ u - \hat{u}\ _\infty}{\ u\ _\infty}$	
8	6.03747	0.01096	20.5137	0.03693	79.1670	0.12285	0.18
16	6.00927	0.00278	20.1268	0.00988	73.7144	0.03408	0.35
32	6.00223	0.00075	20.0313	0.00264	72.4231	0.00892	0.95
64	6.00047	0.00020	20.0075	0.00070	72.1045	0.00227	3.22

done exactly. The inner products were evaluated at two dyadic refinements above the level at which the solution was obtained, which amounts to using a quadrature formula with 15 points per triangle. The second measure of error was simply $\|u - \hat{u}\|_{\infty} / \|u\|_{\infty}$, the maximum relative error between the exact solution u and the computed solution \hat{u} . Note that both measures of the error display the second-order behavior predicted by Theorem 4.1. The observed error is only slightly greater than the bound on the basis function discretization error given by Theorem 4.1. The difference is due to the additional small sources of error involved in the experiment.

A Cray-1 computer was used to perform these calculations. The times shown are for ten iterations of the multigrid algorithm, which in every case reduced the residual by a factor of at least 10^{-8} — more than sufficient for the precision shown. We note that the $n = 64$ case involves a system of $3(2 + 10 n^2) = 122886$ simultaneous linear equations. This experiment thus provides confirmation of second-order accuracy of the spherical finite-element formulation, is a strong test of the correctness of the software, and demonstrates the feasibility of large scale numerical calculations in spherical geometries.

4.7 3-D APPROXIMATION SPACES

The finite-element formulation for the two-sphere may be extended to the spherical shell by adding the radial dimension. Since the radial coordinate is everywhere orthogonal to the tangential spherical barycentric coordinates, it is possible to retain the tangential discretization with all its features and introduce a convenient radial

discretization. The fully three-dimensional nodal basis functions then become Cartesian products of the tangential basis functions $\{L_i\}$ previously described and a set of radial basis functions which may be denoted $\{M_j\}$ and defined as follows.

Definition 4.4: Let $\{r_j\}$ represent any set of $n + 1$ distinct points on the real interval (r_1, r_{n+1}) . Let R be the space of all continuous real functions on the interval (r_1, r_{n+1}) which are piecewise linear between neighboring points in $\{r_j\}$. The space R is chosen as the radial approximation space for a grid with nodes at radial positions given by $\{r_j\}$.

Definition 4.5: For each radial index j of a given radial discretization $\{r_j\}$, denote by M_j that element of R which assumes the value one at r_j and vanishes at all other radial nodal positions. The set of functions $\{M_j\}$ define the radial finite-element basis for R .

This space of one-dimensional piecewise linear functions is well known (Zienkiewicz, p. 34) to have second-order convergence properties.

An approximation space Q of nodal functions for the spherical shell may now be defined as the Cartesian product of the spaces S and R of Definitions 4.2 and 4.4, respectively. Furthermore, a set of basis functions $\{N_k\}$ for the space Q is provided by the Cartesian product of the basis functions $\{L_i\}$ and $\{M_j\}$ of Definitions 4.3 and 4.5, respectively.

An approximation space on the cells or elements of the three-dimensional grid may also be specified.

Definition 4.6: Let $\{c_i\}$ represent the $20n^3$ subdomains or cells associated with the spherical icosahedral grid for a spherical shell. Let P be the space of all real functions on the volume of the shell which have constant values on each of the subdomains in $\{c_i\}$. This space P is defined as the finite-element approximation space for the cells of the grid.

Definition 4.7: For each subdomain or cell c_i in the icosahedral grid for a spherical shell, denote by K_i that element of P which assumes the value one on c_i and vanishes elsewhere. The set of functions $\{K_i\}$ defines a basis for P .

Hence functions in the approximate space P for quantities defined on the cells are piecewise constant.

At this point it is possible to assign a finite-element approximation space to each of the variables in the system of equations (2.15)-(2.18) used to model the thermal convection. The velocity u_i is approximated in terms of the piecewise linear nodal approximation space Q . The density ρ , temperature T , pressure p , and radiogenic heat generation H , on the other hand, are approximated in terms of the piecewise constant cell approximation space P . The gravitational acceleration g_i will be assumed spherically symmetric and be represented in terms of a low-order polynomial in r .

4.8 3-D DISCRETIZATION OF THE LAPLACIAN OPERATOR

Let us now consider the discretization of the Laplacian operator in the space Q . In a manner analogous to the development of section 4.6, the finite-element representation of the weak form for the Laplacian operator may be expressed

$$A_{ij} = - \int_{\text{shell}} (\partial_m N_i)(\partial_m N_j) dV, \quad (4.24)$$

or in more compact notation, $A_{ij} = -\langle \partial_m N_i, \partial_m N_j \rangle$, where the \langle, \rangle notation here implies the inner product integrated over the volume of the spherical shell. Since the basis functions $\{N_i\}$ are products of tangential and radial functions, it will be convenient for a while to switch from Cartesian tensor notation to a spherical coordinate notation, where η is used to denote the tangential coordinates and r the radial coordinate. The basis functions then may be written

$$N_i(\eta, r) = L_i(\eta) M_i(r). \quad (4.25)$$

Expressing A_{ij} in ∇ notation as $-\langle \nabla N_i, \nabla N_j \rangle$, noting that in spherical coordinates $\nabla = (1/r)\nabla_\eta + \hat{r}(\partial/\partial r)$, and using (4.25), we find

$$\begin{aligned} A_{ij} &= -\langle \left(\frac{1}{r} M_i \nabla_\eta L_i + \hat{r} L_i \frac{\partial}{\partial r} M_i \right), \left(\frac{1}{r} M_j \nabla_\eta L_j + \hat{r} L_j \frac{\partial}{\partial r} M_j \right) \rangle \\ &= -\left[\left\langle \frac{1}{r} M_i, \frac{1}{r} M_j \right\rangle_r \langle \nabla_\eta L_i, \nabla_\eta L_j \rangle_\eta \right. \\ &\quad \left. + \left\langle \frac{\partial}{\partial r} M_i, \frac{\partial}{\partial r} M_j \right\rangle_r \langle L_i, L_j \rangle_\eta \right], \end{aligned} \quad (4.26)$$

where ∇_η represents the tangential gradient, \hat{r} the unit vector in the radial direction, $\langle y, z \rangle$ the volume integral $\int y \cdot z \, r^2 dr$, and $\langle y, z \rangle_\eta$ the surface integral of the inner product $y \cdot z$ over the sphere. Since the functions $\{L_i\}$ vary only in the tangential direction, the full gradient operator can be used in place of ∇_η , and 4.26 may be rewritten

$$A_{ij} = - \left[\left\langle \frac{1}{r} M_i, \frac{1}{r} M_j \right\rangle_r \left\langle \partial_m L_i, \partial_m L_j \right\rangle_\eta + \left\langle \frac{\partial}{\partial r} M_i, \frac{\partial}{\partial r} M_j \right\rangle_r \left\langle L_i, L_j \right\rangle_\eta \right] \quad (4.27)$$

We note this expression for A_{ij} contains the tangential operator $\langle \partial_m L_i, \partial_m L_j \rangle_\eta$ obtained in Section 4.6 for the discrete Laplacian on the two-sphere. The procedure for computing this operator has already been described. The tangential operator $\langle L_i, L_j \rangle_\eta$ may be computed in a similar manner from the formula given in Table 4.2. The radial factors $\langle (1/r)M_i, (1/r)M_j \rangle_r$ and $\langle (\partial/\partial r)M_i, (\partial/\partial r)M_j \rangle_r$ involve integration of low-order polynomials in r . The results are summarized as follows:

$$\langle (1/r)M_i, (1/r)M_j \rangle_r = \begin{cases} (r_{i+1} - r_i)/6, & \text{node } j \text{ in layer} \\ & \text{of nodes just above} \\ & \text{that of } i \\ (r_{i+1} - r_i)/3 \\ + (r_i - r_{i-1})/3, & \text{node } j \text{ in same} \\ & \text{layer as } i \\ (r_i - r_{i-1})/6, & \text{node } j \text{ in layer just} \\ & \text{below that of } i \end{cases} \quad (4.28)$$

$$\begin{aligned}
\langle (\partial/\partial r)M_i, (\partial/\partial r)M_j \rangle = & \left\{ \begin{aligned} & \frac{r_{i+1}^2 + r_{i+1}r_i + r_i^2}{3(r_{i+1} - r_i)}, \text{ node } j \text{ in layer} \\ & \hspace{10em} \text{of nodes just} \\ & \hspace{10em} \text{above that of } i \\ & \frac{r_{i+1}^2 + r_{i+1}r_i + r_i^2}{3(r_{i+1} - r_i)} \\ & + \frac{r_i^2 + r_i r_{i-1} + r_{i-1}^2}{3(r_i - r_{i-1})}, \text{ node } j \text{ in same} \\ & \hspace{10em} \text{layer as } i \\ & - \frac{r_i^2 + r_i r_{i-1} + r_{i-1}^2}{3(r_i - r_{i-1})}, \text{ node } j \text{ in layer} \\ & \hspace{10em} \text{just below that} \\ & \hspace{10em} \text{of } i \end{aligned} \right. \quad (4.29)
\end{aligned}$$

Here r_i denotes the radius of the layer of nodes containing node i . At the boundaries, terms involving a nonexistent layer are dropped. Table 4.4 provides a list of radial factors from which a variety of three-dimensional operators may be constructed.

Since A_{ij} may be assembled inexpensively from two-dimensional tangential operators and one-dimensional radial operators, it is logical to store only the lower-dimensional components and assemble the three-dimensional operator as it is applied. This reduces significantly the storage requirements. As indicated earlier, the operator need be created for only one icosahedral diamond of the ten, which cuts the storage by an additional factor of ten. It also trims by a factor of ten the overhead for assembling the 3-D operator.

We note that for a given node i , the finite-element operator A_{ij} in general is nonzero for only 21 values of the node index j . These 21 nodes include node i and the six nodes surrounding i at the same radius plus corresponding nodes in the layer above and the layer

Table 4.4. Radial finite-element operators

Operator	Node i on Inner Radius of Cell m	Node i on Outer Radius of Cell m
$\langle r M_i, K_m \rangle$	$(r_{i+1}^3 + 2r_{i+1}^2 r_i + 3r_{i+1} r_i^2 + 4r_i^3)(r_{i+1} - r_i)/20$	$(4r_i^3 + 3r_i^2 r_{i-1} + 2r_i r_{i-1}^2 + r_{i-1}^3)(r_i - r_{i-1})/20$
$\langle M_i, K_m \rangle$	$(r_{i+1}^2 + 2r_{i+1} r_i + 3r_i^2)(r_{i+1} - r_i)/12$	$(3r_i^2 + 2r_i r_{i-1} + r_{i-1}^2)(r_i - r_{i-1})/12$
$\langle (1/r) M_i, K_m \rangle$	$(r_{i+1} + 2r_i)(r_{i+1} - r_i)/6$	$(2r_i + r_{i-1})(r_i - r_{i-1})/6$
$\langle (a/a_r) M_i, K_m \rangle$	$-(r_{i+1}^2 + r_{i+1} r_i + r_i^2)/3$	$(r_i^2 + r_i r_{i-1} + r_{i-1}^2)/3$

Operator	Node j in Layer Just Above That of i	Node j in Same Layer of Nodes as i	Node j in Layer Just Below That of i
$\langle M_i, M_j \rangle$	$(3r_{i+1}^2 + 4r_{i+1} r_i + 3r_i^2) \cdot (r_{i+1} - r_i)/60$	$(r_{i+1}^2 + 3r_{i+1} r_i + 6r_i^2)(r_{i+1} - r_i)/30 + (6r_i^2 + 3r_i r_{i-1} + r_{i-1}^2)(r_i - r_{i-1})/30$	$(3r_i^2 + 4r_i r_{i-1} + 3r_{i-1}^2) \cdot (r_i - r_{i-1})/60$
$\langle M_i, (1/r) M_j \rangle$	$(r_{i+1}^2 - r_i^2)/12$	$(r_{i+1} + 3r_i)(r_{i+1} - r_i)/12 + (3r_i + r_{i-1})(r_i - r_{i-1})/12$	$(r_i^2 - r_{i-1}^2)/12$

Table 4.4 (Cont.)

Operator	Node j in Layer Just Above That of i	Node j in Same Layer of Nodes as i	Node j in Layer Just Below That of i
$\langle M_i, (a/a_r) M_j \rangle$	$(r_{i+1}^2 + 2r_{i+1}r_i + 3r_i^2)/12$	$-(r_{i+1}^2 + 2r_{i+1}r_i + 3r_i^2)/12$ $+(3r_i^2 + 2r_{i+1}r_{i-1} + r_{i-1}^2)/12$	$-(3r_i^2 + 2r_{i+1}r_{i-1} + r_{i-1}^2)/12$
$\langle (1/r) M_i, (1/r) M_j \rangle$	$(r_{i+1} - r_i)/6$	$(r_{i+1} - r_{i-1})/3$	$(r_i - r_{i-1})/6$
$\langle (1/r) M_i, (a/a_r) M_j \rangle$	$(r_{i+1} + 2r_i)/6$	$(r_{i+1} - r_{i-1})/6$	$-(2r_i + r_{i-1})/6$
$\langle (a/a_r) M_i, (a/a_r) M_j \rangle$	$-(r_{i+1}^3 - r_i^3)/3(r_{i+1} - r_i)^2$	$(r_{i+1}^3 - r_i^3)/3(r_{i+1} - r_i)^2$ $+(r_i^3 - r_{i-1}^3)/3(r_i - r_{i-1})^2$	$-(r_i^3 - r_{i-1}^3)/3(r_i - r_{i-1})^2$

Note: r_i denotes the radius of the layer of nodes containing node i

below. Exceptions are at the shell boundaries and the pentagonal nodes, which have but five surrounding nodes. Therefore, the cost in machine multiplications for performing the matrix multiplication $A_{ij}y_j$ is about 21 times the number of nodes in the grid.

4.9 DISCRETIZATION OF ∇P

As indicated in Section 4.7 the pressure p is discretized in the piecewise constant approximation space P . In terms of the basis functions $\{K_m\}$ the pressure may be expressed as $p_m K_m$, where p_m is the pressure associated with cell m . The Galerkin representation for the pressure gradient term in (2.15), here denoted h_{ik} , is

$$h_{ik} = \langle N_i, \partial_k K_m \rangle p_m \quad (4.30)$$

where \langle , \rangle denotes the volume integral of the product. Since K_m does not have C_0 continuity, (4.30) is integrated by parts to yield

$$h_{ik} = -\langle \partial_k N_i, K_m \rangle p_m + p_m \int_A N_i \hat{n}_k dA \quad (4.31)$$

where A represents the boundary surfaces of the cell and n_k is k^{th} Cartesian component of the unit normal to the cell surface.

The surface integrals resulting from the integration by parts cancel identically at all nodes i in the shell interior. On the shell boundaries, these integrals yield radial nodal forces which are balanced by the external pressure on the shell. However, in the applications treated in this dissertation, the shell boundaries are

assumed undeformable, and the radial forces at the boundary nodes do not enter the problem.

The operator $\langle \partial_k N_i, K_m \rangle$ may be written as a sum of products of tangential and radial components

$$\langle \partial_k N_i, K_m \rangle = \langle \partial_k L_i, K_m \rangle_\eta \langle (1/r) M_i, K_m \rangle_r + \langle \hat{r}_k L_i, K_m \rangle_\eta \langle (\partial/\partial r) M_i, K_m \rangle_r \quad (4.32)$$

$$= \langle \partial_k \eta_i^m, 1 \rangle_\eta \langle (1/r) M_i, K_m \rangle_r + \langle \eta_i^m, \hat{r}_k \rangle_\eta \langle (\partial/\partial r) M_i, K_m \rangle_r, \quad (4.33)$$

where η_i^m is the barycentric coordinate associated with node i in cell m , \hat{r}_k is the unit vector in the radial direction, and \langle , \rangle_η and \langle , \rangle_r are as defined in the previous section.

The tangential operators $\langle \partial_k \eta_i, 1 \rangle_\eta$ and $\langle \eta_i, \hat{r}_k \rangle_\eta$ may be computed from the formulas in Table 4.2 on a very fine grid and then projected to the grid on which the convection calculation is to be performed using the recursion relations in Table 4.1. Formulas for the radial operators $\langle (1/r) M_i, K_m \rangle_r$ and $\langle (\partial/\partial r) M_i, K_m \rangle_r$, obtained by integrating low-order polynomials in r , are listed in Table 4.4. As in the case of the discrete Laplacian operator, the 3-D cell-to-node gradient operator $\langle \partial_k N_i, K_m \rangle$ may be stored in unassembled form and assembled as it is applied.

Two tests were used to evaluate the correctness and accuracy of this operator. In the first, the gradient operator was applied to a constant unit-amplitude scalar field. The RMS value of the resulting

vector field at the nodes (except for those at the shell boundaries) was multiplied by the number of nodes, divided by the shell volume, and multiplied by the shell radius for normalization. Since the gradient of a constant field is zero, departure from zero is a measure of the error. For an $n = 8$ grid, this test gave a normalized error of 9.76×10^{-6} , and for an $n = 16$ grid, 1.52×10^{-5} . In the second test, a set of random numbers, uniformly distributed on the interval $(0.0, 2.0)$, was used to specify the scalar field. The components of the vector field obtained by applying the gradient operator to this scalar field were summed and the sum divided by shell volume and multiplied by the shell radius. Since the integral of the gradient of any such function with compact support vanishes, the exact solution for this test is also zero. The results were 5.49×10^{-16} for an $n = 8$ grid and 6.10×10^{-16} for an $n = 16$ grid. Both tests imply an acceptable level of accuracy for this operator.

4.10 DISCRETIZATION OF THE BUOYANCY FORCE

Like the pressure, the density ρ is also discretized in the piecewise constant approximation space P . The discretized density field, expressed in terms of the basis functions $\{K_m\}$, is $\rho_m K_m$, where ρ_m is the density associated with cell m . The gravitational acceleration g_k , for simplicity, may be assumed spherically symmetric and be represented by a low-order polynomial $\gamma(r)$ such that

$$g_k = g_0 \hat{r}_k \gamma(r) \quad (4.34)$$

The finite-element representation of the buoyancy force in (2.15) may then be written

$$f_{ik} = g_0 \langle N_i, \hat{r}_k \gamma(r) K_m \rangle \rho_m \quad (4.35)$$

The operator $\langle N_i, \hat{r}_k \gamma(r) K_m \rangle$ can be factored into tangential and radial components

$$\langle N_i, \hat{r}_k \gamma(r) K_m \rangle = \langle L_i, \hat{r}_k K_m \rangle_\eta \langle M_i \gamma(r), K_m \rangle_r \quad (4.36)$$

$$= \langle \eta_i^m, \hat{r}_k \rangle_\eta \langle M_i \gamma(r), K_m \rangle_r \quad (4.37)$$

where η_i^m is the spherical barycentric coordinate associated with node i in cell m . The tangential operator is computed using Tables 4.1 and 4.2. For the case of constant gravitational acceleration, $\gamma(r) = 1$, and the radial factor is $\langle M_i, K_m \rangle_r$. Similarly, for g_k increasing linearly with r , $\gamma(r) = r/r_{\max}$. In both cases the radial factors represent integrals of low-order polynomials in r which are listed in Table 4.4. Higher-order polynomial representations for g_k may be handled in like fashion.

This operator was checked for correctness and accuracy using two tests. The first applied the buoyancy operator to a shell with a constant unit-amplitude density field and a unit-amplitude gravity field. The vector sum over all nodes, normalized by the volume of the shell, of the resulting vector field corresponds to the normalized net buoyancy, which vanishes for the exact case. For an $n = 8$ grid, the

magnitude of this normalized net buoyancy was 3.40×10^{-14} , and for an $n = 16$ grid, it was 2.08×10^{-13} .

The second test compared the magnitude of the force acting on individual cells as computed by the operator against a simpler and less accurate method which integrates the gravity field magnitude over the cell volume and then multiplies the result by the cell density. For the $n = 8$ case, the fractional difference between the two methods was 9.03×10^{-4} both for uniform gravity and for gravity increasing linearly with radius. For the $n = 16$ case, the fractional difference was 2.28×10^{-4} . Most of the error is in the simpler method which does not account for the central nature of the vector gravity field. These tests imply an acceptable level of accuracy for this operator.

4.11 DISCRETE FORM OF THE FORCE BALANCE EQUATION

It is now possible to express the finite-element equivalent of (2.15). Let the approximation to the velocity field in terms of the nodal basis functions be written

$$\tilde{u}_k = u_{kJ} N_J \quad (4.38)$$

The discrete version of (2.15) then consists of the following system of linear equations

$$\mu A_{IJ} u_{kJ} - h_{Ik} + f_{Ik} = 0, \quad \begin{matrix} i = 1, \dots, 2 + 10 n^2 \\ k = 1, 3 \end{matrix} \quad (4.39)$$

where A_{ij} , h_{ik} , and f_{ik} are defined by (4.27), (4.31), and (4.35), respectively, and μ is the dynamic shear viscosity. Here n depends on the dyadic refinement of the grid as discussed in Section 4.1. The next chapter describes a technique for solving this elliptic system of equations in an efficient fashion for the unknown coefficients u_{kj} .

The mass and energy conservation equations (2.16) and 2.17) utilize the same finite-element approximation spaces described in Section 4.7. Details of their finite-element implementation and method of solution, however, are reserved for Chapter 6.

5. MULTIGRID SOLUTION TECHNIQUE FOR FORCE BALANCE EQUATION

The multigrid technique used to solve the linear system (4.39) was conceived and developed by Paul O. Frederickson, presently at Los Alamos National Laboratory. A detailed discussion of the algorithm together with examples of its performance are included in a paper to appear (Frederickson). Its application to the 3-D spherical problem is described below.

5.1 THE MULTIGRID ALGORITHM

Suppose a solution is sought for the discrete system

$$Wv = a, \tag{5.1}$$

where W is a linear operator on the nodal approximation space for a 3-D grid constructed from the m^{th} dyadic refinement of the spherical icosahedron. Let us denote this space as Q_m . By inspection of the basis functions $\{L_i\}$ and $\{M_i\}$, whose Cartesian product forms the basis functions $\{N_i\}$ (see Chapter 4), it is simple to show that $Q_0 \subset Q_1 \subset \dots \subset Q_k \subset \dots \subset Q_m$. That is, the approximation space Q_k for a given dyadic refinement k is a subspace of the approximation spaces of all further dyadic refinements.

This sequence of nested approximation spaces is ideally suited for applying an iterative multigrid algorithm to solve system (5.1) in a highly efficient manner. As will become evident in the following discussion the efficiency is obtained because only an inexpensive local

approximate inverse is needed to refine the solution interpolated from the next lower approximation space.

Let us now examine how such an algorithm works. For notational simplicity let superscript (n) represent the iteration number and subscript k the dyadic refinement index for the approximation space involved. At the beginning of iteration n , we have an approximate solution $v_m^{(n-1)}$ from the preceding iteration. (To start the procedure, the null vector can be used for $v^{(0)}$.)

The first step in the algorithm is to compute the residual r_m

$$r_m = a - W_m v_m^{(n-1)}. \quad (5.2)$$

This residual r_m is then projected onto each of the lower-order approximation spaces Q_{m-1}, \dots, Q_0 to provide coarser representations of the residual, r_{m-1}, \dots, r_0 .

In the lowest-order space Q_0 , the system

$$W_0 z_0 = r_0 \quad (5.3)$$

is solved by standard algebraic techniques (inexpensive since the system is small) for the correction field z_0 . This correction z_0 is then interpolated onto space Q_1 as z_1 and refined by use of the residual r_1 . The refined correction field, denoted z_1' , is computed by finding a second-order residual

$$y_1 = r_1 - W_1 z_1 \quad (5.4)$$

to which an approximate inverse \tilde{W}_1^{-1} is applied and the result added to z_1 , that is,

$$z_1' = z_1 + \tilde{W}_1^{-1} y_1 \quad (5.5)$$

The refined correction z_1' on Q_1 is now interpolated onto Q_2 and further refined using r_2 . This procedure is continued until a refined correction z_m' is obtained on space Q_m . This correction is now added to the previous estimate $v_m^{(n-1)}$ for the solution to obtain an improved solution $v_m^{(n)}$:

$$v_m^{(n)} = v_m^{(n-1)} + z_m' \quad (5.6)$$

The process converges rapidly, typically with a factor of three to five reduction in the RMS norm of the residual r_m per iteration for the 3-D problem.

The cost per iteration for this algorithm corresponds to less than that of four applications at the operator W_m . How this is true may be understood as follows. First of all, the projections and interpolations are simple and inexpensive because of the nestedness of the approximation spaces. Further, since the size of the linear system decreases by a factor of eight for a 3-D grid each time the dyadic index decreases by one, the vast majority of the computational cost is

associated with the finest discretization Q_m . Finally, as will be discussed in the next section, the approximate inverse operators \tilde{W}_k^{-1} are chosen to have the same graph (and hence the same cost to apply) as the forward operators W_k . In the approximation space Q_m , there are three applications of the operators W_m and \tilde{W}_m^{-1} .

A 21-point scalar operator like the 3-D Laplacian applied to a field with three spatial components requires close to $63 N$ multiplications, where N is dimensionality of the approximation space. For such an operator the cost per iteration of the multigrid algorithm is therefore about $250 N$ multiplications. The fact that the cost is linear with N means that the multigrid approach offers significant savings in computational effort for large problems compared with other numerical techniques currently in use.

5.2 LEAST-SQUARES LOCAL APPROXIMATE INVERSE

A crucial feature in the multigrid algorithm just described is the local approximate inverse \tilde{W}_k^{-1} . This section outlines a procedure for computing the optimum inverse operator in the least squares sense, subject to the constraint that the inverse operator have the same graph as the forward operator.

The problem is to find the operator Y which minimizes the functional

$$f(Y) = \sum_{l=1}^N \text{diag} [(I - YW)(I - YW)^T] \quad , \quad (5.7)$$

where W is a linear operator on the N -dimensional approximation space and Y has the same graph as W . We shall use the fact that for this optimum Y ,

$$g_{mn}(Y) = \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} [f(Y - \epsilon E^{mn}) - f(Y)] = 0 \quad (5.8)$$

for all values of m and n , where E^{mn} is the matrix which has unit value for its mn^{th} element and is zero otherwise. With double subscripts to denote output and input indices for the operators, (5.7) may be rewritten

$$f(Y) = N - Y_{IJ} W_{JI} - W_{JI} Y_{IJ} + Y_{IJ} W_{JK} W_{LK} Y_{IL} \quad , \quad (5.9)$$

and $g_{mn}(Y)$ may be expressed

$$g_{mn}(Y) = \lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} [-2\epsilon W_{nm} + \epsilon W_{nk} W_{lk} Y_{ml} + \epsilon Y_{mj} W_{jk} W_{nk} + O(\epsilon^2)] \quad . \quad (5.10)$$

We observe that for g_{mn} to vanish, we need Y to satisfy

$$W = W W^T Y^T \quad . \quad (5.11)$$

Given the operator W , one may use standard linear algebra routines to compute Y from (5.11). Since W for the problems of interest is a local operator, the cost of this procedure is relatively modest. For the 3-D Laplacian, this involves solving a rank 21 linear system at each node

for one diamond in the grid. Some additional features of local least-squares approximate inverses are described by Benson and Frederickson (1982).

6. DISCRETE FORMULATION OF MASS AND ENERGY CONSERVATION

This chapter describes the methods used to treat the advection, thermal conduction, and time integration of the mass and energy conservation equations (2.16) and (2.17) in terms of the discretization of Chapter 4.

6.1 CELL-WALL ADVECTION

The discrete method selected for treating the advection of mass and heat is referred to as cell-wall advection since mass and heat are transferred only between cells that share a common face. The advection depends on the sign and magnitude of the normal velocity integrated over the area of the common face as well as the values of density, temperature, and specific heat of the cell for which the integrated normal velocity is outward. This method offers the advantages that it is simple to implement and that mass and heat are strictly conserved.

As may be noted from Figure 4.4, cells in the 3-D icosahedral grid have the form of triangular prisms with three planar faces and two spherical ones. Advection through the planar faces will be referred to as tangential and that through the spherical faces as radial. In both cases, computing the integral of the normal velocity over the face area is accomplished by multiplying the normal velocities at the vertices of the face by appropriate weighting factors and summing. For the planar faces, the weighting factors for the nodes at the inner radius r_i of the face are given by $(2r_i + r_o)(r_o - r_i)\phi/12$

while for the nodes at the outer radius r_o they are $(r_i + 2r_o)(r_o - r_i)\phi/12$, where ϕ is the angle subtended by the face. For the spherical faces, the weighting factors correspond to the tangential finite-element operator $\langle \eta_i, \hat{r}_k \rangle$ described in Section 4.9.

Performing the advection calculation first involves resolving the velocity field into radial and tangential components at each of the nodes. Next, integrated normal velocities are computed for each face. Finally, rates of change of mass and heat are calculated for all the cells by accumulating the contributions across all the faces. The rate of mass advection through a given face is the integrated normal velocity times the density of the donor cell. (Recall the discretization of density assigns a constant value to each cell.) Similarly, the rate of heat advection is the product of the donor cell density, specific heat, and temperature, and the face integrated normal velocity.

This method introduces a slight error in the integration of the normal velocity on the spherical faces. The error becomes negligible as the discretization becomes fine and the spherical triangles have only small variation in normal direction. A numerical experiment to quantify the magnitude of this error uses a uniform translational velocity field across a spherical shell of uniform density. The rate of change of mass in the cells not adjacent to the shell boundaries should be zero. The computed rate of change in mass compared with the rate of mass flow through these cells is a measure of the error. The RMS value of this error for an $n = 8$ grid is 1.8×10^{-8} and for an $n = 16$ grid is 9.8×10^{-8} . The same experiment was performed for heat advection with identical results. Since the translational velocity field produces

advective flow through cells having a variety of spatial orientations with respect to the flow, this experiment provides a strong test on the correctness of the software as well. It may be concluded from the experiment that the method used for the advection is accurate as well as conservative.

6.2 DISCRETIZATION OF THERMAL CONDUCTION

Thermal conduction is treated in a manner similar to advection in that heat is transferred only between cells that share a common face. The method uses cell-to-cell conductivity factors based on the geometry of the grid. These factors may be precomputed once the grid is specified. The rate of conductive heat flow between adjacent cells is found by multiplying the temperature difference between the cells by the appropriate conductivity factor.

As in the case of cell-wall advection, separate treatments are required for the tangential and radial conduction associated with the plane and spherical faces, respectively. For radial conduction, the conductivity factor ξ_r is given by the formula

$$\xi_r = kA\bar{r}_1\bar{r}_2/(\bar{r}_2 - \bar{r}_1) \quad (6.1)$$

where k is the thermal conductivity and A is the area of the spherical face at unit radius. \bar{r}_1 and \bar{r}_2 are defined by

$$\bar{r}_1 = \frac{2(r_{i-1}^2 + r_{i-1}r_i + r_i^2)}{3(r_{i-1} + r_i)} \quad (6.2)$$

$$\text{and } \bar{r}_2 = \frac{2(r_i^2 + r_i r_{i+1} + r_{i+1}^2)}{3(r_i + r_{i+1})} \quad (6.3)$$

Here r_{i-1} denotes the inner radius of the lower cell, r_i the radius of the common face, and r_{i+1} the outer radius of the upper cell.

The radii \bar{r}_1 and \bar{r}_2 have special significance in themselves. Consider the problem of steady state conduction across a spherical layer with isothermal boundaries at r_{i-1} and r_i . The radius \bar{r}_1 is the position in the layer at which the temperature is equal to the mean temperature of the layer. The radius \bar{r}_2 has a similar significance for a layer with isothermal boundaries at r_i and r_{i+1} . With this understanding it is clear how to treat the temperature boundary conditions. The value for \bar{r}_2 at the outer shell boundary is simply the outer radius of the shell, and the value for \bar{r}_1 at the inner shell boundary is the inner radius of the shell. Boundary temperatures are then handled in the same fashion as cell temperatures. For the test problem of steady state conduction across a spherical shell with isothermal boundaries, it is readily verified that this discrete formulation gives a result which agrees exactly with the analytical solution.

For tangential conduction, the conductivity factor ξ_η is given by the expression

$$\xi_\eta = k\phi(r_o - r_i)/\delta \quad (6.4)$$

where ϕ is the angle subtended by the planar face, r_o the outer radius of the face, and δ is the distance between cell centers when both are projected to unit radius.

A test problem to check the accuracy of this approximate formulation for the tangential conduction uses a spherical shell of thickness t with conical sections removed at the north and south poles as illustrated in Figure 6.1. The spherical surfaces are taken to be perfectly insulating and the conical surfaces isothermal with a temperature difference ΔT between them. The resulting heat flux Q for this problem is given by

$$Q = -\frac{\pi k t \Delta T}{\ln \tan(\phi_o/2)}, \quad (6.5)$$

where ϕ_o is the half-angle of the conical surface with respect to the center of the shell. The temperature distribution $T(\phi)$ in the shell as a function of colatitude ϕ , when the temperature $T(\phi_o)$ on the conical surface at the north pole is $\frac{1}{2}\Delta T$, is

$$T(\phi) = T(\phi_o) \frac{\ln \tan(\phi/2)}{\ln \tan(\phi_o/2)}. \quad (6.6)$$

It is possible to check the numerical solution both for the total heat flux Q and the temperature profile $T(\phi)$. The $n = 8$ discretization yields a heat flux value which is 0.55 percent less than the analytical value, while the $n = 16$ discretization gives a value of 0.39 percent less. These results are for a value of ϕ_o of 30 degrees. Meridional temperature distributions are listed in Table 6.1. Since

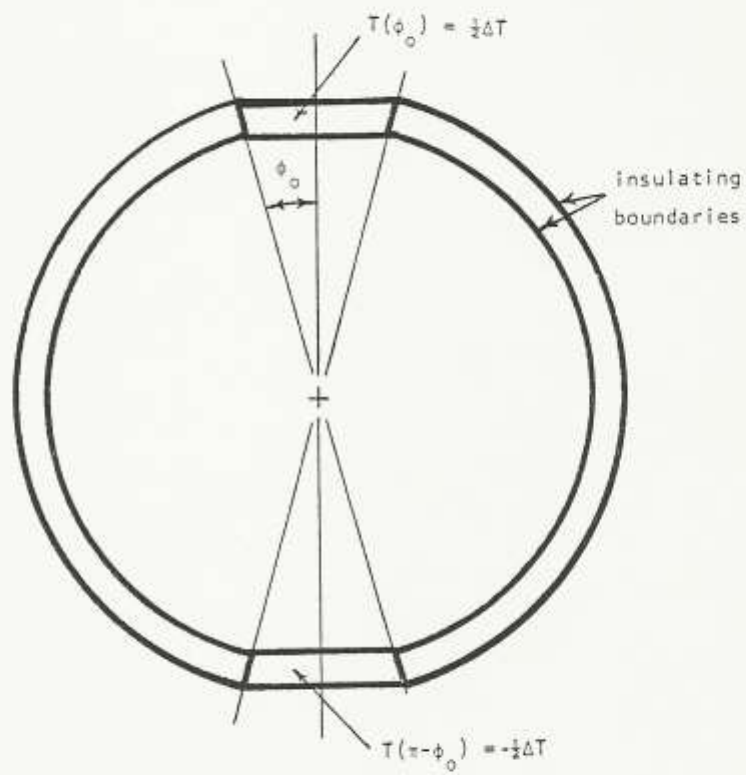


Figure 6.1. Test problem for evaluation of tangential conduction.

tangential conduction plays only a secondary role in the mantle convection problem, the accuracy indicated here is deemed satisfactory. Part of the error in this experiment is associated with the piecewise constant representation of the temperature field and ought not be attributed to the technique used to model the conduction itself.

Table 6.1. Meridional temperature distributions for tangential conduction experiment ($\phi_0 = 30$ degrees).

Colatitude (degrees)	$T_{n=8}$	$T_{n=16}$	T_{exact}
31.95	0.939	0.936	0.950
37.38	0.815	0.817	0.823
48.01	0.610	0.614	0.614
53.17	0.526	0.524	0.526
63.39	0.363	0.365	0.366
68.55	0.290	0.289	0.291
79.19	0.144	0.144	0.144
84.62	0.073	0.073	0.071

6.3 SECOND-ORDER RUNGE-KUTTA TIME INTEGRATION

The preceding two sections outlined methods for computing the time rates of change on the cells of mass and of heat due to advection and conduction. Addition of a radiogenic heat source in each cell completes the description of the spatial discretization of the mass and energy conservation equations. Attention will now be directed to their time discretization.

The discrete time integration procedure selected provides second-order accuracy but requires derivatives at two points in each interval. For purposes of illustration, consider the differential equation

$$\frac{\partial \zeta}{\partial t} = f(\zeta) \quad . \quad (6.7)$$

The method can be summarized as follows, where superscript (n) is used to denote the value at the end of time interval n, Δt is the length of the interval, and θ is a scalar constant between zero and one:

$$\left. \frac{\partial \zeta}{\partial t} \right|_0 = f[\zeta^{(n)}] \quad (6.8)$$

$$\left. \frac{\partial \zeta}{\partial t} \right|_\theta = f[\zeta^{(n)} + \theta \left. \frac{\partial \zeta}{\partial t} \right|_0 \Delta t] \quad (6.9)$$

$$\zeta^{(n+1)} = \zeta^{(n)} + [(1 - \theta) \left. \frac{\partial \zeta}{\partial t} \right|_0 + \theta \left. \frac{\partial \zeta}{\partial t} \right|_\theta] \Delta t \quad . \quad (6.10)$$

This is the second-order Runge-Kutta formula. It is applied in a straightforward fashion to the discretized mass and energy conservation equations using a value for θ of 0.75.

Of course, to do the convection problem, the discretized version of the complete system of equations (2.15)-(2.18) must be solved together. Once the mass and energy conservation equations have been integrated to yield updated distributions for density and temperature, these are processed by the equation of state to provide an updated pressure field, which in turn is required in solving the force balance equation for a revised velocity field. In this overall process, the time step Δt is continually adjusted so that only one iteration of the multigrid algorithm is needed to maintain the residual error in the force balance equation below some specified level (typically 10^{-4}). At the same time, an upper bound is also set on the time step such that the mass advected through any cell in the grid in a single time step never exceeds the volume of the cell.

To reduce the numerical stiffness arising from the large values typically used for the bulk modulus, a damping factor for the rate of cell density change is introduced. This damping factor scales down the net change in density during a time step and serves to eliminate the pressure-driven oscillations which would occur otherwise. It was found in most cases that a damping factor of 0.001 suppresses the oscillations without significantly altering the trajectory of the convection solution. With this technique, the convergence rate of the multigrid algorithm is fast enough and the accuracy of the time integration is high enough so that, after the initial transient, time

steps commonly on the order of 0.5 times the upper bound are obtained.

Cost per time step in Cray-1 CPU seconds to solve the system of equations (2.15)-(2.18) is approximately 0.5 for the $n = 8$ grid and 4.0 for the $n = 16$ grid.