7-8-2010

A Hybrid Radial Basis Function-Pseudospectral Method for Thermal Convection in a 3-D Spherical Shell

Grady Wright
Boise State University

Natasha Flyer
National Center for Atmospheric Research

David A. Yuen
University of Minnesota

Geochemistry
Geophysics
Geosystems
G3

Article
Volume 11, Number 7
8 July 2010
Q07003, doi:10.1029/2009GC002985
ISSN: 1525-2027

Published by AGU and the Geochemical Society

Click Here for Full Article

A hybrid radial basis function–pseudospectral method for thermal convection in a 3-D spherical shell

G. B. Wright
Department of Mathematics, Boise State University, Boise, Idaho 83725, USA (gradywright@boisestate.edu)

N. Flyer
Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, Boulder, Colorado 80305, USA (flyer@ucar.edu)

D. A. Yuen
Department of Geology and Geophysics, University of Minnesota, Minneapolis, Minnesota 55455, USA (daveyuen@gmail.com)

[1] A novel hybrid spectral method that combines radial basis function (RBF) and Chebyshev pseudospectral methods in a “2 + 1” approach is presented for numerically simulating thermal convection in a 3-D spherical shell. This is the first study to apply RBFs to a full 3-D physical model in spherical geometry. In addition to being spectrally accurate, RBFs are not defined in terms of any surface-based coordinate system such as spherical coordinates. As a result, when used in the lateral directions, as in this study, they completely circumvent the pole issue with the further advantage that nodes can be “scattered” over the surface of a sphere. In the radial direction, Chebyshev polynomials are used, which are also spectrally accurate and provide the necessary clustering near the boundaries to resolve boundary layers. Applications of this new hybrid methodology are given to the problem of convection in the Earth’s mantle, which is modeled by a Boussinesq fluid at infinite Prandtl number. To see whether this numerical technique warrants further investigation, the study limits itself to an isoviscous mantle. Benchmark comparisons are presented with other currently used mantle convection codes for Rayleigh number (Ra) 7 × 10^3 and 10^5. Results from a $Ra = 10^6$ simulation are also given. The algorithmic simplicity of the code (mostly due to RBFs) allows it to be written in less than 400 lines of MATLAB and run on a single workstation. We find that our method is very competitive with those currently used in the literature.

Components: 10,700 words, 7 figures, 6 tables.

Keywords: mantle convection; radial basis function; pseudospectral; isoviscous; spherical geometry.

Index Terms: 0560 Computational Geophysics: Numerical solutions (4255); 8121 Tectonophysics: Dynamics: convection currents, and mantle plumes; 0550 Computational Geophysics: Model verification and validation.

Received 1 December 2009; **Revised** 3 May 2010; **Accepted** 19 May 2010; **Published** 8 July 2010.

Wright, G. B., N. Flyer, and D. A. Yuen (2010), A hybrid radial basis function–pseudospectral method for thermal convection in a 3-D spherical shell, *Geochem. Geophys. Geosyst.*, *11*, Q07003, doi:10.1029/2009GC002985.

1. Introduction

[2] Mantle convection models in spherical geometry have seen a variety of numerical method implementations, from finite element and finite volume methods on a variety of grids such as icosahedral, cubed sphere, Yin-Yang, spiral, and hexahedral [*Baumgardner*, 1985; *Hernlund and Tackley*, 2003; *Yoshida and Kageyama*, 2004; *Harder and Hansen*, 2005; *Stemmer et al.*, 2006; *Hüttig and Stemmer*, 2008; *Zhong et al.*, 2000, 2008], to pseudospectral (PS) methods using spherical harmonics [*Bercovici et al.*, 1989; *Harder*, 1998]. The former methods can be cumbersome and tedious to program due to grid generation and treatment of the equations near element boundaries and are generally low order. The latter requires more nodes than basis functions (especially when dealiasing filters are used), since there are $2N + 1$ longitudinal Fourier modes for each latitudinal associated Legendre function of degree N, and does not easily allow for local refinement.

[3] A novel approach that is in its infancy of development is radial basis functions (RBFs), a mesh-less method that has the advantage of being spectrally accurate for arbitrary node layouts in multidimensions. Former studies, using this method on spherical surfaces, have shown it to be very competitive in comparison to numerical methods that are currently used in the geosciences, algorithmically simpler, and naturally permitting local node refinement [*Flyer and Wright*, 2009; *Flyer and Lehto*, 2010; *Flyer and Wright*, 2007; *Fornberg and Piret*, 2008]. However, given this early stage of development, numerical modeling experiments with RBFs are warranted before full-blown mantle convection models using RBFs are developed that can handle everything from variable viscosity to thermochemical convection. As a result, this paper is of an exploratory nature from the perspective of numerics. Since no 3-D model using RBF spatial discretization of partial differential equations (PDEs) in spherical geometry exists in the math or science literature, it follows that taking the simplest formulation for mantle convection, isoviscous flows at various Rayleigh numbers (as is done by *Bercovici et al.* [1989] and *Harder* [1998]) would be a good starting point.

[4] The paper is organized as follows: Section 2 describes the physical model; section 3 gives an introduction to RBFs; section 4 shows how the spatial operators are discretized using RBFs; section 5 reviews the concept of influence matrices that must be used for solving the coupled Poisson equations which result from writing the velocity in terms of a poloidal potential [see *Chandrasekhar*, 1961]; section 6 describes the time discretization; section 7 provides numerical results from two test cases with comparisons to those in the literature and results from a $Ra = 10^6$ simulation; section 8 gives timing results for the benchmark cases and section 9 discusses extensions of the method to high Ra, variable viscosity, and local node refinement. Appendices A and B give the steps for implementing the RBF-PS algorithm.

2. Physical Model

[5] We consider a thermal convection model of a Boussinesq fluid at infinite Prandtl number in a spherical shell that is heated from below. The governing equations are

$$\nabla \cdot \mathbf{u} = 0 \quad \text{(continuity)}, \tag{1}$$

$$\nabla \cdot \left[\eta\left(\nabla\mathbf{u} + \{\nabla\mathbf{u}\}^T\right)\right] + Ra\, T\, \hat{\mathbf{r}} = \nabla p \quad \text{(momentum)}, \tag{2}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla^2 T \quad \text{(energy)}, \tag{3}$$

where $\mathbf{u} = (u_r,\ u_\theta,\ u_\lambda)$ is the velocity field in spherical coordinates (θ = latitude, λ = longitude), p is pressure, T is temperature, $\hat{\mathbf{r}}$ is the unit vector in the radial direction, η is the viscosity, and Ra is the Rayleigh number. The boundary conditions on the velocity of the fluid at the inner and outer surfaces of the spherical shell are

$$\underbrace{u_r\big|_{r=R_\mathrm{i},R_\mathrm{o}} = 0}_{\text{impermeable}} \qquad \text{and}$$

$$\underbrace{r\frac{\partial}{\partial r}\left(\frac{u_\theta}{r}\right)\bigg|_{r=R_\mathrm{i},R_\mathrm{o}} = r\frac{\partial}{\partial r}\left(\frac{u_\lambda}{r}\right)\bigg|_{r=R_\mathrm{i},R_\mathrm{o}} = 0}_{\text{shear stress free}}, \tag{4}$$

where R_i is the radius of the inner surface of the shell and R_o is the radius of the outer surface as measured from the center of the earth. The boundary conditions on the temperature are

$$T(R_\mathrm{i}, \theta, \lambda) = 1 \quad \text{and} \quad T(R_\mathrm{o}, \theta, \lambda) = 0.$$

Equations (1)–(3) have been nondimensionalized with the length scale chosen as the thickness of the shell, $\Delta R = R_\mathrm{o} - R_\mathrm{i}$, the time scale chosen as the thermal diffusion time, $t = (\Delta R)^2/\kappa$ (κ = thermal diffusivity), and the temperature scale chosen as the difference between the temperature at the inner and outer boundaries, ΔT.

[6] In this study, we treat the fluid as isoviscous, $\eta =$ const. Thus, the dynamics of the fluid are governed by the *Ra*, which can be interpreted as a ratio of the destabilizing force due to the buoyancy of the heated fluid to the stabilizing force due to the viscosity of the fluid. It takes the specific form of

$$Ra = \frac{\rho g \alpha \Delta T (\Delta R)^3}{\kappa \eta},$$

where ρ is the density of the fluid, g is the acceleration due to gravity, and α is the coefficient of thermal expansion.

[7] *Chandrasekhar* [1961] [see also *Backus*, 1966] shows that any divergence-free field can be expressed in terms of a poloidal and toroidal potential, $\mathbf{u} = \nabla \times \nabla \times ((\Phi r)\hat{\mathbf{r}}) + \nabla \times (\Psi \hat{\mathbf{r}})$. If the fluid is isoviscous (or the viscosity stress tensor is spherically symmetric) and satisfies (4) then the field is purely poloidal (i.e., $\Psi \equiv 0$). As a result, the three-dimensional continuity and momentum equations (1) and (2) can be alternatively written as a system of two coupled Poisson equations. The nonlinear thermal convection model can then be written as

$$\Delta_s \Omega + \frac{\partial}{\partial r}\left(r^2 \frac{\partial \Omega}{\partial r}\right) = Ra\, r\, T, \qquad (5)$$

$$\Delta_s \Phi + \frac{\partial}{\partial r}\left(r^2 \frac{\partial \Phi}{\partial r}\right) = r^2 \Omega, \qquad (6)$$

$$\frac{\partial T}{\partial t} = -\left(u_r \frac{\partial T}{\partial r} + u_\theta \frac{1}{r}\frac{\partial T}{\partial \theta} + u_\lambda \frac{1}{r\cos\theta}\frac{\partial T}{\partial \lambda}\right) + \frac{1}{r^2}\Delta_s T + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial T}{\partial r}\right), \qquad (7)$$

where $\theta \in [-\pi/2, \pi/2]$, $\lambda \in (-\pi, \pi]$, and Δ_s is the surface Laplacian operator. The velocity boundary conditions (4) in terms of Φ are

$$\Phi|_{r=R_i,R_o} = 0 \quad \text{and} \quad \left.\frac{\partial^2 \Phi}{\partial r^2}\right|_{r=R_i,R_o} = 0. \qquad (8)$$

The components of the velocity $\mathbf{u} = (u_r, u_\theta, u_\lambda)$ are given by

$$\mathbf{u} = \nabla \times \nabla \times [(\Phi r)\hat{\mathbf{r}}] = \left(\frac{1}{r}\Delta_s \Phi, \frac{1}{r}\frac{\partial^2}{\partial r \partial \theta}(\Phi r), \frac{1}{r\cos\theta}\frac{\partial^2}{\partial r \partial \lambda}(\Phi r)\right). \qquad (9)$$

We separate the angular and radial directions of the operators as will be discussed in section 4 on spatial discretization. Sections 5 and 6 describe the various steps of the algorithm.

3. Introduction to RBFs

[8] We only intend to give a brief introduction to RBFs. For a good, in depth discussion see *Fasshauer* [2007]. The strength of RBFs lie in approximation problems in multidimensional space with scattered node layouts [*Fornberg et al.*, 2010]. In the context of solving partial differential equations (PDEs), the global RBF approach can be viewed as a major generalization of pseudospectral methods [*Fornberg et al.*, 2002, 2004]. The concept behind RBFs is that by abandoning the orthogonality of the basis functions, the nodes can be arbitrarily scattered over the domain, maintaining spectral accuracy with the ability to node refine in a completely grid-independent environment [*Flyer and Lehto*, 2010]. This allows for geometric flexibility with regard to the shape of the domain, as well as flexibility in allowing the nodes to be concentrated where greater resolution is needed. In addition, studies have shown that RBFs can take unusually long time steps in comparison to other methods, such as pseudospectral, spectral element and finite volume, for solving purely hyperbolic systems [*Flyer and Wright*, 2007, 2009; *Flyer and Lehto*, 2010].

[9] RBF spatial discretization is based on linear combinations of translates of a single radially symmetric function that collocates the data, as is illustrated in Figure 1. The argument of the RBF, d, is the Euclidean distance between where the RBF is centered $\mathbf{x}_j \in \mathbb{R}^n$ and where it is evaluated $\mathbf{x} \in \mathbb{R}^n$ with n being the dimension of the space, i.e., $d = \|\mathbf{x} - \mathbf{x}_j\|_2$ (from now on for simplicity we drop the subscript 2). Since its argument only depends on a scalar distance, independent of *coordinates*, *dimension* or *geometry*, RBFs are exceptionally simple to program with the algorithmic complexity of the code not increasing with dimension. For example, for two points on the surface of the unit sphere, $\mathbf{x}_1 = (x_1, y_1, z_1)$ and $\mathbf{x}_2 = (x_2, y_2, z_2)$ (or in spherical coordinates (θ_1, λ_1) and (θ_2, λ_2)), where $\mathbf{x}_1$ is the center of the RBF and $\mathbf{x}_2$ is where it is to be evaluated, the argument of the RBF is

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} = \sqrt{2(1 - \mathbf{x}_2^T \mathbf{x}_1)} = \sqrt{2(1 - \cos\theta_2 \cos\theta_1 \cos(\lambda_2 - \lambda_1) - \sin\theta_2 \sin\theta_1)}.$$

Notice that the distance is not measured as great arcs along the sphere but rather as a straight line through the sphere. Thus, the RBF has no "sense"

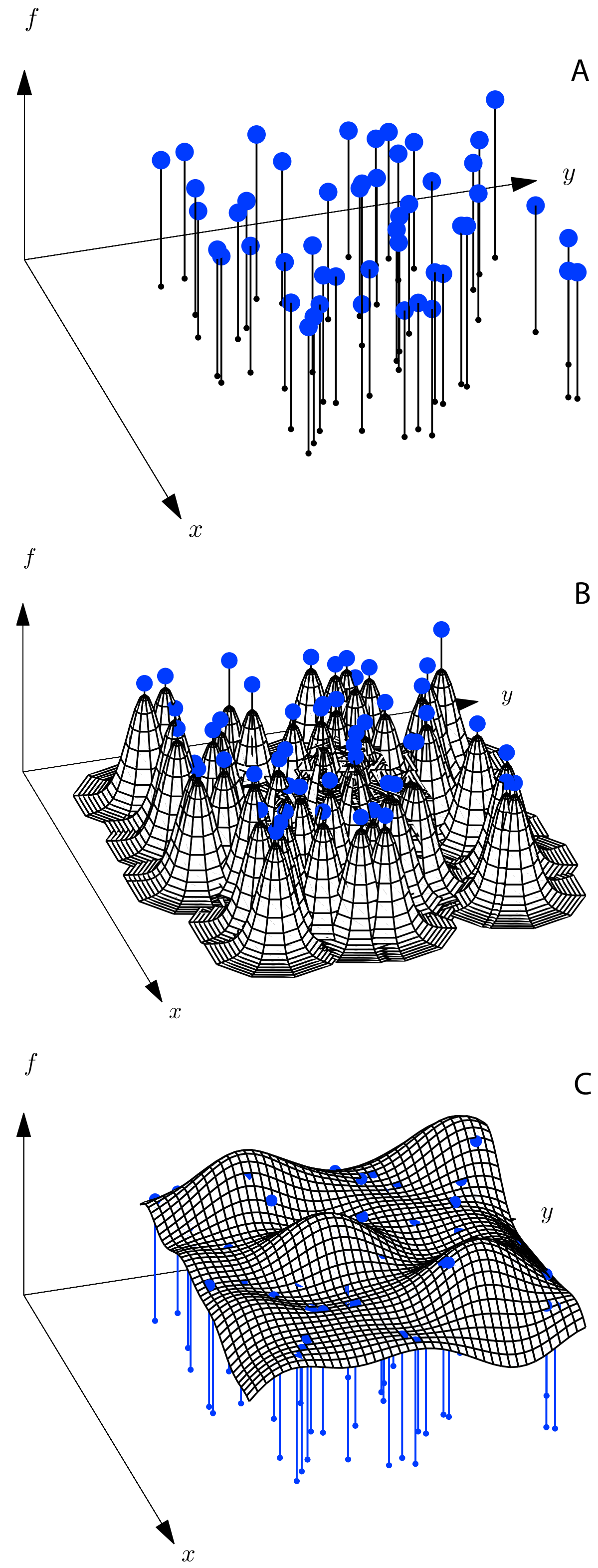

Figure 1. (a) Data values $\{f_j\}_{j=1}^{N}$, (b) the RBF collocation functions, and (c) the resulting RBF interpolant.

that it exists on a spherical manifold. It should be emphasized that the coordinate system is only used to identify the location of the nodes and not a representation of any grid or manifold (i.e., geometry) in n-dimensional space. Thus, if we choose to represent the node locations in spherical coordinates, a latitude-longitude grid is never used, but rather the nodes are placed as the user desires.

[10] Common RBFs are listed in Table 1. There are two distinct kinds, piecewise smooth and infinitely smooth. Piecewise smooth RBFs lead to algebraic convergence as they contain a jump in some derivative, e.g., $|d|^3$ jumps in the third derivative. Infinitely smooth RBFs lead to spectral convergence as they do not jump in any derivative and thus will be used in this paper. This latter group features a parameter ε which determines the shape of the RBF and plays an important role in both the conditioning and accuracy of RBF matrices [*Fornberg and Flyer*, 2005; *Buhmann*, 2003]. How the error of the solution varies as a function of the shape parameter ε for solving different classes of PDEs and what are the optimal choices for it has been studied by *Iske* [2004], *Wright and Fornberg* [2006], *Wertz et al.* [2006], *Driscoll and Heryundono* [2007], *Flyer and Wright* [2007], *Fasshauer and Zhang* [2007], *Fornberg and Zuev* [2007], *Fornberg and Piret* [2008], and *Flyer and Wright* [2009].

[11] The above studies have shown that best results are achieved with roughly evenly distributed nodes. Since only a maximum of 20 nodes can be evenly distributed on a sphere, there are a multitude of algorithms to define "even" distribution for larger numbers of nodes, such as equal partitioned area, convex hull approaches, Voronoi cells, electrostatic repulsion [*Hardin and Saff*, 2004]. Although any of these will suffice, we have decided to use an elec-

Table 1. Commonly Used RBFs

Abbreviation	Name	Definition
	Piecewise Smooth	
MN	monomial	$\lvert d\rvert^{2m+1}$
TPS	thin plate spline	$\lvert d\rvert^{2m} \ln \lvert d\rvert$
	Infinitely Smooth	
MQ[a]	multiquadric	$\sqrt{1+(\varepsilon d)^2}$
IMQ	inverse MQ	$\dfrac{1}{\sqrt{1+(\varepsilon d)^2}}$
IQ	inverse quadratic	$\dfrac{1}{1+(\varepsilon d)^2}$
GA	Gaussian	$e^{-(\varepsilon d)^2}$

[a] The MQ is used for all results in this study.

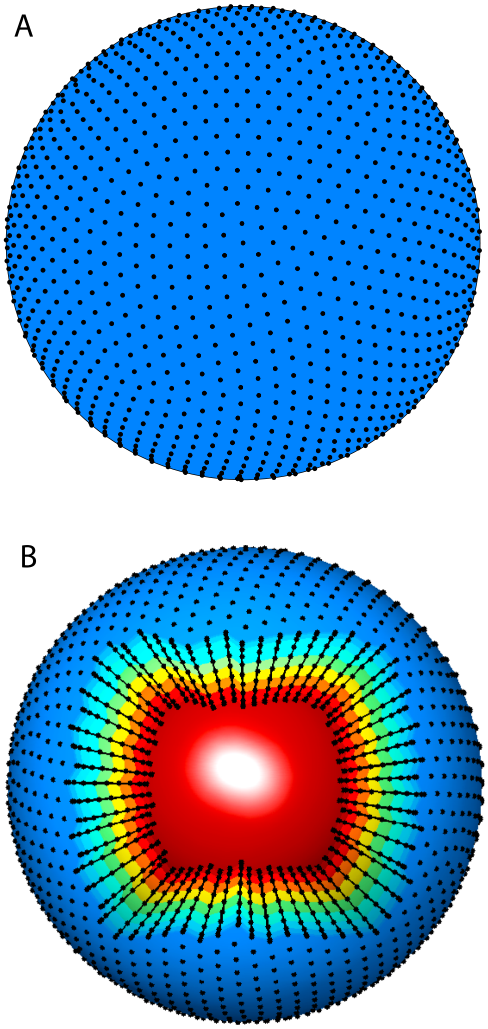

Figure 2. (a) RBF node layout on the surface of a sphere and (b) 3-D view of the discretization of the spherical shell used in the hybrid RBF-PS calculation. Blue is the outer boundary, and red is the inner boundary; black dots display the computational nodes, which are distributed in the radial direction along the extrema of the Chebyshev polynomials. Note that the spherical shell has been opened up in Figure 2b to show the detail.

trostatic repulsion or minimal energy (ME) approach since the nodes do not line up along any vertices or lines, emphasizing the arbitrary node layout and coordinate-free nature of a RBF methodology, as can be seen by the node layout in Figure 2a. A very important consequence of this is that although the PDEs of the physical model are posed and solved in spherical coordinates, there are no pole singularities.

4. Spatial Discretization

[12] To numerically solve (5)–(7) a "2 + 1" layering approach is used, where the lateral directions (θ, λ) are discretized separately from the radial direction. Using collocation, the approximate solution is calculated at the nodes shown in Figure 2b. We use $M + 2$ Chebyshev nodes in the radial direction (corresponding to M interior points and 2 boundary points) and N "scattered" nodes on each of the resulting M spherical surfaces. As shown in Figure 2b, this gives a tensor product structure between the radial and lateral directions, which allows the spatial operators to be computed in $O(M^2N) + O(MN^2)$ operations instead of $O(M^2N^2)$ as discussed below. While all radial derivatives are discretized using Chebyshev polynomials, differential operators in the latitudinal direction θ and longitudinal direction λ are approximated discretely on each spherical surface using RBFs. In sections 4.1 and 4.2, we will discuss how to discretize the lateral advection and surface Laplacian operators using RBFs, with a novel RBF formulation of the latter. The radial discretization by collocation with Chebyshev polynomials is standard and is therefore omitted (see, for example, *Fornberg* [1995], *Trefethen* [2000] or *Weideman and Reddy* [2000] for details).

4.1. RBF Discretization of the Lateral Advection Operator

[13] Given a velocity field tangent to the unit sphere that is a function of time and space, $\mathbf{u} = \{u_\theta(\theta, \lambda, t), u_\lambda(\theta, \lambda, t)\}$, the lateral advection operator is given by

$$\mathbf{u}\cdot\nabla = u_\theta \frac{\partial}{\partial\theta} + \frac{u_\lambda}{\cos\theta}\frac{\partial}{\partial\lambda}. \qquad (10)$$

which is singular at $\theta = \pm\frac{\pi}{2}$, the north and south poles, unless $\frac{\partial}{\partial\lambda}$ also vanishes there. We will next see that this is exactly what happens when the operator is applied to an RBF.

[14] Setting $\varepsilon = 1$ (for simplicity of notation), let $\phi_j(d) = \phi(\sqrt{2(1-\cos\theta\cos\theta_j\cos(\lambda-\lambda_j)-\sin\theta\sin\theta_j)})$ be an RBF centered at the node (θ_j, λ_j). Using the chain rule, the partial derivatives of the RBF $\phi_j(d)$ with respect to λ and θ are given by

$$\frac{\partial}{\partial\lambda}\phi_j(d) = \frac{\partial d}{\partial\lambda}\frac{\partial\phi}{\partial d} = \cos\theta\cos\theta_j\sin(\lambda-\lambda_j)\left(\frac{1}{d}\frac{\partial\phi_j}{\partial d}\right), \qquad (11)$$

$$\begin{aligned}\frac{\partial}{\partial\theta}\phi_j(d) &= \frac{\partial d}{\partial\theta}\frac{\partial\phi}{\partial d}\\ &= (\sin\theta\cos\theta_j\cos(\lambda-\lambda_j)-\cos\theta\sin\theta_j)\left(\frac{1}{d}\frac{\partial\phi_j}{\partial d}\right).\end{aligned} \qquad (12)$$

Inserting (11) and (12) into (10), we have

$$\begin{aligned}\mathbf{u}\cdot\nabla &= u_\theta(\cos\theta_j\sin\theta\cos(\lambda-\lambda_j)-\sin\theta_j\cos\theta)\left(\frac{1}{d}\frac{\partial\phi_j}{\partial d}\right)\\ &\quad + u_\lambda\cos\theta_j\sin(\lambda-\lambda_j)\left(\frac{1}{d}\frac{\partial\phi_j}{\partial d}\right). \qquad (13)\end{aligned}$$

Given that the velocities are smooth, notice that nowhere on the sphere is (13) singular.

[15] Now, we have all the components that are necessary to build the action of the advection operator on an RBF representation of the temperature field. We first represent $T(\lambda,\theta)$ as an RBF expansion given by

$$T(\theta,\lambda)=\sum_{j=1}^{N}c_j\phi_j(d(\theta,\lambda)). \qquad (14)$$

where c_j are the unknown expansion coefficients. We then apply the exact differential operator $\mathbf{u}\cdot\nabla$ to (14) and evaluate it at the node locations, $\{(\lambda_i,\theta_i)\}_{i=1}^{N}$, where $T(\lambda,\theta)$ is known. Note that because u_λ and u_θ are time dependent we will need to create two separate differentiation matrices, one to represent $\frac{1}{\cos\theta}\frac{\partial}{\partial\lambda}\phi_j(d)$ and another to represent $\frac{\partial}{\partial\theta}\phi_j(d)$, otherwise (13) could be written as a single differentiation matrix. Since they are created in the same way, we will only demonstrate how to formulate D_λ, the differentiation matrix for the longitudinal direction:

[16] 1. Take $\frac{1}{\cos\theta}\frac{\partial}{\partial\lambda}$ of (14):

$$\begin{aligned}\frac{1}{\cos\theta}\frac{\partial T(\lambda,\theta)}{\partial\lambda} &= \sum_{j=1}^{N}c_j\frac{1}{\cos\theta}\frac{\partial\phi_j(d)}{\partial\lambda}\\ &= \sum_{j=1}^{N}c_j\cos\theta_j\sin(\lambda-\lambda_j)\left(\frac{1}{d}\frac{\partial\phi_j}{\partial d}\right).\end{aligned}$$

[17] 2. Evaluate step 1 at the node locations:

$$\sum_{j=1}^{N}c_j\underbrace{\left(\cos\theta_j\sin(\lambda_i-\lambda_j)\frac{1}{d}\frac{\partial\phi_j}{\partial d}\right)\Bigg|_{(\lambda,\theta)=(\lambda_i,\theta_i)_{i=1}^{N}}}_{\text{Components of matrix } B_\lambda}=B_\lambda c,$$

where c contains the N unknown discrete expansion coefficients. If we evaluate the RBFs in (14) at the node locations $\{(\lambda_i,\theta_i)\}_{i=1}^{N}$ then we have the collocation problem

$$\underbrace{\begin{bmatrix}\phi(\|\mathbf{x}_1-\mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_1-\mathbf{x}_N\|)\\ \vdots & \ddots & \vdots\\ \phi(\|\mathbf{x}_N-\mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_N-\mathbf{x}_N\|)\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}c_1\\ \vdots\\ c_N\end{bmatrix}}_{c}=\underbrace{\begin{bmatrix}T_1\\ \vdots\\ T_N\end{bmatrix}}_{T}, \qquad (15)$$

where A is the RBF interpolation matrix for the node set. Thus, $c = A^{-1}T$ and substituting this into step 2 above gives $D_\lambda T = B_\lambda(A^{-1}T)$, in other words $D_\lambda = B_\lambda A^{-1}$. Put verbally, the differentiation matrices are obtained by applying the exact differential operator to the interpolant and then evaluating it at the data locations. Although the computation of D_λ and D_θ requires $O(N^3)$ operations, it is a preprocessing step that needs to be done only once.

4.2. A Novel RBF Surface Laplacian Formulation

[18] Since RBFs do not require the nodes on a spherical surface to have any directionality and since RBFs are not defined in terms of any surface-based coordinate system (as discussed in section 3), then for simplicity let us center an RBF at the north pole $\mathbf{x}_{\mathrm{np}} = (0, 0, 1)$ or $(\theta,\lambda) = (\pi/2, 0)$. The distance from this point to any point on the sphere is then given by

$$d(\mathbf{x})=\|\mathbf{x}-\mathbf{x}_{\mathrm{np}}\|=\sqrt{x^2+y^2+(z-1)^2}=\sqrt{2(1-\sin(\theta))}. \qquad (16)$$

Now, the surface Laplacian in spherical coordinates in given by

$$\Delta_s=\frac{\partial^2}{\partial\theta^2}-\tan\theta\frac{\partial}{\partial\theta}+\frac{1}{\cos^2\theta}\frac{\partial^2}{\partial\lambda^2}. \qquad (17)$$

However, for a radial function centered at the north pole there will be no λ dependence. So, (17) reduces to

$$\Delta_s=\frac{\partial^2}{\partial\theta^2}-\tan\theta\frac{\partial}{\partial\theta}. \qquad (18)$$

Applying (18) to an RBF $\phi(d)$ gives

$$\Delta_s\phi(d)=\frac{\partial^2 d}{\partial\theta^2}\frac{\partial\phi}{\partial d}+\left(\frac{\partial d}{\partial\theta}\right)^2\frac{\partial^2\phi}{\partial d^2}-\tan\theta\frac{\partial d}{\partial\theta}\frac{\partial\phi}{\partial d}. \qquad (19)$$

With the use of (16) and after some algebra, (19) reduces to

$$\Delta_s\phi(d)=\frac{1}{4}\left[(4-d^2)\frac{\partial^2\phi}{\partial d^2}+\frac{4-3d^2}{d}\frac{\partial\phi}{\partial d}\right]. \qquad (20)$$

Although we derived this formula by centering the RBF at the north pole, any node could have served as the north pole since an RBF is invariant to coordinate rotations. The beauty of (20) is that it expresses the action of the surface Laplacian on an RBF simply in terms of the distances between nodes without the coordinate system ever coming into play. The RBF surface Laplacian differentiation matrix is then defined as $L_s = B_s A^{-1}$, where B_s is now a matrix, evaluating (20) at

Table 2. Variables Composing the Influence Matrix Method With the PDEs and Boundary Conditions They Solve and If They Are Time Dependent[a]

Variable	PDE	Boundary Conditions at $r = R_\text{i}, R_\text{o}$	Time Dependent
Ω_h	$\Delta\Omega_h = Ra\ r\ T$	$\Omega_h\vert_{Ri,Ro} = 0$	Yes
Φ_h	$\Delta\Phi_h = \Omega_h$	$\Phi_h\vert_{Ri,Ro} = 0$	Yes
$\Omega_j^{R_\text{i}}$	$\Delta\Omega_j^{R_\text{i}} = 0$	$\Omega_j^{R_\text{i}}\vert_{Ro} = 0,\ \Omega_j^{R_\text{i}}\vert_{Ri} = \begin{cases} 1 & \text{if } (\theta,\lambda) = (\theta_j,\lambda_j), \\ 0 & \text{otherwise} \end{cases}$	No
$\Phi_j^{R_\text{i}}$	$\Delta\Phi_j^{R_\text{i}} = \Omega_j^{R_\text{i}}$	$\Phi_j^{R_\text{i}}\vert_{Ri,Ro} = 0$	No
$\Omega_j^{R_\text{o}}$	$\Delta\Omega_j^{R_\text{o}} = 0$	$\Omega_j^{R_\text{o}}\vert_{Ri} = 0,\ \Omega_j^{R_\text{o}}\vert_{Ro} = \begin{cases} 1 & \text{if } (\theta,\lambda) = (\theta_j,\lambda_j), \\ 0 & \text{otherwise} \end{cases}$	No
$\Phi_j^{R_\text{o}}$	$\Delta\Phi_j^{R_\text{o}} = \Omega_j^{R_\text{o}}$	$\Phi_j^{R_\text{o}}\vert_{Ri,Ro} = 0$	No
$\Omega_{bd,j}^{R_\text{i}}, \Omega_{bd,j}^{R_\text{o}}$	see (22)–(23)	not applicable	Yes

[a] That is, if they need to be solved at every time step.

$d = \sqrt{2(1-\cos\theta_i\cos\theta_i\cos(\lambda_i-\lambda_j)-\sin\theta_i\sin\theta_j)}$, $1 \le i, j \le N$, j indexing the RBF centers and i the node locations.

5. Momentum Equation Solver and Influence Matrix Method

[19] We cannot directly solve (5) and (6) since we have 4 boundary conditions on Φ, given by (8), and none on Ω. We therefore use the influence matrix method [*Peyret*, 2002] to find the unknown boundary values on Ω such that all 4 boundary conditions on Φ are satisfied. Since (5) and (6) are linear, the solution to each Poisson equation can be represented as a superposition of two solutions; the first, Ω_h and Φ_h, satisfies the *right-hand side* of the equations with *homogeneous* Dirichlet boundary conditions; the second, Ω_j and Φ_j, couples the unknown boundary values (abbreviated *bd*), $\Omega_{bd,j}^{R_\text{i}}$ and $\Omega_{bd,j}^{R_\text{o}}$, with $\Phi_j^{R_\text{i}}$ and $\Phi_j^{R_\text{o}}$ at each RBF collocation node, $\{\theta_j, \lambda_j\}_{j=1}^N$, on the inner ($R_\text{i}$) and outer ($R_\text{o}$) boundary spherical surfaces, respectively. In Table 2, the variables are defined in terms of the Poisson equations they solve with the overall solution written as

$$\Omega = \Omega_h + \sum_{j=1}^{N}\left[\Omega_{bd,j}^{R_\text{i}}\Omega_j^{R_\text{i}} + \Omega_{bd,j}^{R_\text{o}}\Omega_j^{R_\text{o}}\right] \quad \text{and}$$

$$\Phi = \Phi_h + \sum_{j=1}^{N}\left[\Omega_{bd,j}^{R_\text{i}}\Phi_j^{R_\text{i}} + \Omega_{bd,j}^{R_\text{i}}\Phi_j^{R_\text{o}}\right]. \tag{21}$$

The method is reminiscent of a Green's function type approach, but instead of expanding the source term of the PDEs in Dirac delta functions, we expand the unknown boundary conditions in this basis, solve Laplace's equation (as the right-hand side is taken care of by the solutions Ω_h and Φ_h) and superpose the solutions as is done by the summations in (21). Thus, for each boundary, we are building a table of N particular solutions, $\{\Omega_j^{R_\text{i}}\}_{j=1}^N$ and $\{\Omega_j^{R_\text{o}}\}_{j=1}^N$, whose boundary value is 1 at the jth boundary node and 0 at all others. These PDEs are solved N times, corresponding to the number of boundary nodes we have on each boundary surface. It is important to note that solving for $\Omega_j^{R_\text{i}}$ and $\Omega_j^{R_\text{o}}$ is a preprocessing step, since the equation is temperature independent and thus time independent. Also, once we solve for $\Phi_j^{R_\text{i}}$ and $\Phi_j^{R_\text{o}}$, $\Omega_j^{R_\text{i}}$ and $\Omega_j^{R_\text{o}}$ can be deleted as they are no longer needed for any computations. As discussed in Appendix A, the approximate solutions to the PDEs listed in Table 2 are computed in spectral space via a matrix diagonalization (or eigenvector decomposition) technique which requires $O(MN^2) + O(M^2N)$ operations per PDE and $O(M^2) + O(N^2)$ storage. This is significant savings over a direct solve of the equations, which would require $O(M^2N^2)$ operations and $O(M^2N^2)$ storage.

[20] Once Φ_h has been computed, the unknown coefficients $\Omega_{bd,j}^{R_\text{i}}$ and $\Omega_{bd,j}^{R_\text{o}}$ are determined by requiring the linear combination of Φ_h, $\Phi_j^{R_\text{i}}$, and $\Phi_j^{R_\text{o}}$ in (21) satisfy the boundary conditions (8). Since each of these variables satisfy the *homogeneous* Dirichlet boundary conditions by construction, the unknown coefficients are determined by the second Neumann-type boundary condition. Inserting the expression for Φ given in (21) into this boundary condition leads to the following set of linear equations which need to be enforced at each boundary node $j = 1,\ldots, N$:

$$\frac{\partial^2}{\partial r^2}\Phi_j^{R_\text{i}}\bigg|_{r=R_\text{i}}\Omega_{bd,j}^{R_\text{i}} + \frac{\partial^2}{\partial r^2}\Phi_j^{R_\text{o}}\bigg|_{r=R_\text{i}}\Omega_{bd,j}^{R_\text{o}} = -\frac{\partial^2}{\partial r^2}\Phi_h\bigg|_{r=R_\text{i}}, \tag{22}$$

$$\frac{\partial^2}{\partial r^2}\Phi_j^{R_\text{i}}\bigg|_{r=R_\text{o}}\Omega_{bd,j}^{R_\text{i}} + \frac{\partial^2}{\partial r^2}\Phi_j^{R_\text{o}}\bigg|_{r=R_\text{o}}\Omega_{bd,j}^{R_\text{o}} = -\frac{\partial^2}{\partial r^2}\Phi_h\bigg|_{r=R_\text{o}}. \tag{23}$$

The coefficient matrix that arises from this $2N \times 2N$ linear systems is called the influence matrix and can be precalculated, LU decomposed and stored as it is time independent. However, the solution to the linear system must be computed every time step as Φ_h is time dependent. Once $\Omega_{bd,j}^{R_i}$ and $\Omega_{bd,j}^{R_o}$ are found, Φ can be determined from the second equation in (21) and then the velocity field can be calculated according to (9).

[21] While the presentation above is the most straightforward way to describe the influence matrix technique for solving equations (5) and (6) subject to the boundary conditions (8), it is not the most computationally efficient given how the solutions to Ω_h and Φ_h are computed in the overall algorithm. In Appendix A, we discuss how the computation can be done in the spectral space of the discrete operators to reduce the computational cost. This description is the one used in the code.

6. Time Discretization

[22] The Chebyshev discretization of the radial component of the diffusion operator has a Courant-Friedrichs-Lewy (CFL) condition on the time step that is proportional to $O(1/M^4)$, which makes an explicit scheme implausible. As a result, we implement a semi-implicit time stepping scheme which treats this component implicitly and the remaining terms of the energy equation explicitly. We note that implicitly time stepping the entire diffusion term, that is also the RBF discretization of the Δ_s operator, would make no difference in terms of the overall CFL condition on the energy equation. This is because the CFL condition on this operator results in a time step restriction that scales like $O(1/N)$, with $N = O(1000)$ typically, and the time step restriction due to the Chebyshev discretization on the radial components of the nonlinear advection term scales as $O(1/M^2)$, with $M = O(10)$ typically.

[23] We separate the terms in the energy equation (7) as follows:

$$\frac{\partial T}{\partial t} = \underbrace{-\left(u_r \frac{\partial T}{\partial r} + u_\theta \frac{1}{r}\frac{\partial T}{\partial \theta} + u_\lambda \frac{1}{r \sin\theta}\frac{\partial T}{\partial \lambda}\right) + \frac{1}{r^2}\Delta_s T}_{f(T,t)} + \underbrace{\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial T}{\partial r}\right)}_{g(T,t)}. \tag{24}$$

Using a third-order Adams-Bashforth (AB3) method combined with a Crank-Nicolson (CN) method, (24) can be discretized by

$$T^{k+1} = T^k + \underbrace{\frac{\Delta t}{12}\left(23F^k - 16F^{k-1} + 5F^{k-2}\right)}_{\text{AB3}} + \underbrace{\frac{\Delta t}{2}\left(G^{k+1} + G^k\right)}_{\text{CN}}, \tag{25}$$

where all the terms are matrices of size $N \times M$, corresponding to the values in the interior of the spherical shell, and F^k and G^k are the respective approximations to $f(T, t)$ and $g(T, t)$ at the kth time step and are explicitly given by

$$\begin{aligned} F^k &= -\left(u_r^k \circ (T^k D_r) + u_\theta^k \circ (D_\theta T^k R^{-1}) + u_\lambda^k \circ (D_\lambda T^k R^{-1})\right) \\ &\quad + L_s T^k R^{-2} + B_f \\ G^k &= T^k L_r R^{-2} + B_g, \end{aligned} \tag{26}$$

where $\circ$ denotes element-wise matrix multiplication, and B_f and B_g contain the appropriate terms from the boundary conditions on T. The abbreviations for the differentiation matrices are given in Table 3. The matrices u_r^k, u_θ^k, u_λ^k are the approximations to the respective components of the velocity at the kth time step, while the diagonal matrix R contains the M interior Chebyshev nodes, defined by

$$R_{j,j} = \frac{1}{2}(R_i + R_o) + \frac{1}{2}(R_o - R_i)\cos\left(\frac{j}{M+1}\pi\right), \; j = 1, \ldots, M. \tag{27}$$

Equation (25) can be rewritten as

$$\begin{aligned} T^{k+1} &= \left[T^k + \frac{\Delta t}{12}\left(23F^k - 16F^{k-1} + 5F^{k-2}\right) + \frac{\Delta t}{2}G^k\right] \\ &\quad \cdot \left(I - \frac{\Delta t}{2} L_r R^{-2}\right)^{-1}. \end{aligned} \tag{28}$$

As a preprocessing step $(I - \frac{\Delta t}{2} L_r R^{-2})$ is LU decomposed and stored. The computational cost of computing (28) is then $O(M^2 N)$. However, since N is typically two orders of magnitude larger than M, the total cost per time step will be dominated by the solving the momentum equations (see Appendix A), calculating the velocity, and computing the values of F^k and G^k, all of which require $O(M\,N^2)$ operations. Exact details on each step of the algorithm are given in Appendix B.

7. Validation on the RBF-PS Method

[24] In this section, we first consider three benchmark studies for 3-D spherical shell models of mantle convection with constant viscosity and

Table 3. Notation for the Various Differentiation Matrices Used in the Time-Differencing Scheme

Matrix	Operator	Discretization	Dimension
D_λ	$\frac{1}{\cos\theta}\frac{\partial}{\partial\lambda}$	RBF	$N \times N$
D_θ	$\frac{\partial}{\partial\theta}$	RBF	$N \times N$
L_s	Δ_s	RBF	$N \times N$
D_r	$\frac{\partial}{\partial r}$	Chebyshev	$M \times M$
L_r	$\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right)$	Chebyshev	$M \times M$

report the first results in the literature from a purely spectral method run at $Ra = 10^6$. Although there are many numerical methods in the literature for mantle convection in spherical geometry [*Bercovici et al.*, 1989; *Zhang and Christensen*, 1993; *Ratcliff et al.*, 1996; *Richards et al.*, 2001; *Hernlund and Tackley*, 2003; *Yoshida and Kageyama*, 2004; *Harder and Hansen*, 2005; *Stemmer et al.*, 2006; *Choblet et al.*, 2007; *Kameyama et al.*, 2008; *Zhong et al.*, 2000, 2008], the obstacle we encountered is that there is not a set of standardized test cases for comparison with regard to *Ra* number and the viscosity profile. However, there are a number of published results for $Ra = 7000$ with constant viscosity and we compare our method with these. Above this *Ra*, there does not seem to be any consistency in the specifications of the physical model for testing the numerical methods published in the literature. Thus, for higher *Ra*, we have decided to use $Ra = 10^5$ results from the model for mantle convection, CitcomS, recently reported by *Zhong et al.* [2008] as a benchmark comparison. The only other study in the literature that gives isoviscous results for this *Ra* number is by *Ratcliff et al.* [1996], which is also included in our comparison.

7.1. *Ra* = 7000

[25] The two most common benchmarks for computational models of mantle convection in a spherical shell are the steady state tetrahedral and cubic test cases. For both of these benchmarks the fluid is treated as isoviscous and *Ra* is set to 7000. The initial condition for the temperature is specified as

$$T(r,\theta,\lambda) = \frac{R_i(r-R_o)}{r(R_i-R_o)} + 0.01Y_3^2(\theta,\lambda)\sin\left(\pi\frac{r-R_i}{R_o-R_i}\right) \quad (29)$$

for the tetrahedral test case and

$$T(r,\theta,\lambda) = \frac{R_i(r-R_o)}{r(R_i-R_o)} + 0.01\left[Y_4^0(\theta,\lambda)+\frac{5}{7}Y_4^4(\theta,\lambda)\right]\sin\left(\pi\frac{r-R_i}{R_o-R_i}\right) \quad (30)$$

for the cubic test case, where Y_ℓ^m denotes the normalized spherical harmonic of degree ℓ and order m. The first term in each of the initial conditions represents a purely conductive temperature profile, while the second terms are perturbations to this profile and determine the final steady state solution. The $\theta - \lambda$ temperature dependence of (29) and (30) on a spherical shell surface can be seen in Figures 3a and 3b, respectively.

[26] For this test case, two RBF-PS simulations are reported: (1) a higher-resolution case, in which $N = 1600$ nodes were used on each spherical surface (i.e., in the lateral directions) and 23 total Chebyshev nodes were used in the radial direction (i.e., $M = 21$ total interior nodes), giving a total of 36,800 nodes, and 2) a lower-resolution case of $N = 900$ and 19 nodes radially (17 interior nodes), giving a total of 17,100 nodes. A time step of 10^{-4} was used or 10,000 time steps were taken to reach steady state at the nondimensionalized time of $t = 1$, corresponding to roughly 58 times the age of the Earth. Figures 4a and 4b display the final RBF-PS steady state solutions for the tetrahedral and cubic test cases, respectively, in terms of the residual temperature $\delta T = T(r, \theta, \lambda) - \langle T(r)\rangle$, where $\langle\ \rangle$

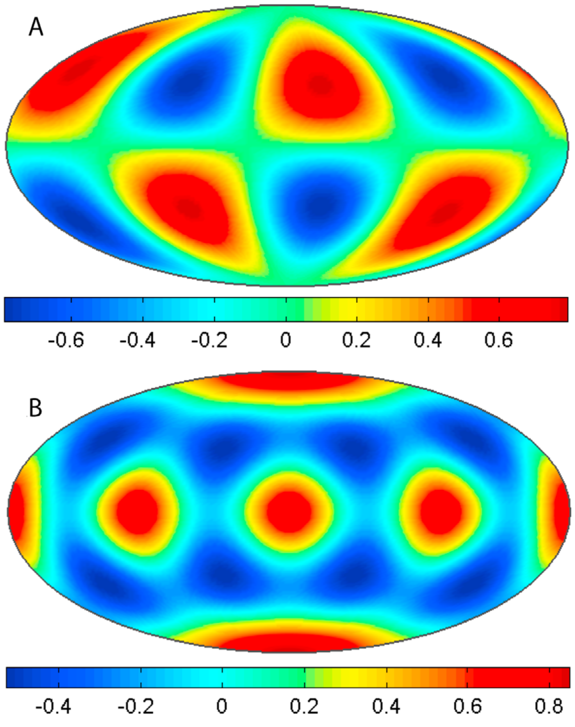

Figure 3. The $\theta - \lambda$ dependence of the initial condition for the (a) tetrahedral and (b) cubic mantle convection test cases.

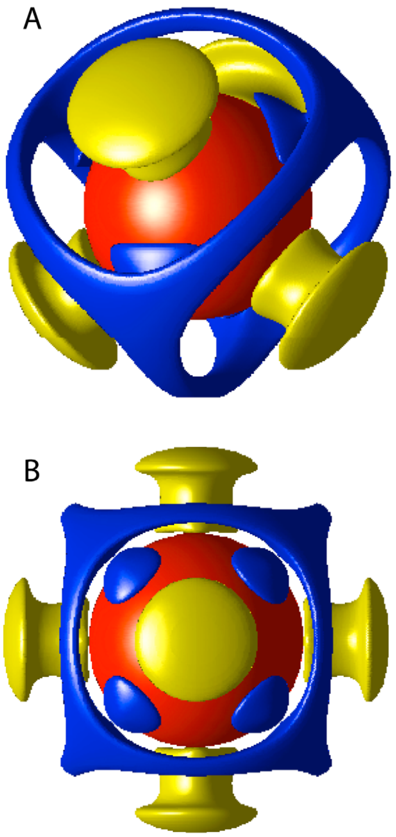

Figure 4. Steady state isosurfaces of the residual temperature, δT, at t = 1 for the isoviscous (a) tetrahedral and (b) cubic mantle convection test cases at Ra = 7000 computed with the RBF-PS model. Yellow corresponds to δT = 0.15 and denotes upwelling relative to the average temperature at each radial level, while blue corresponds to δT = −0.15 and denotes downwelling. The red solid sphere shows the inner boundary of the 3-D shell corresponding to the core.

denotes averaging over a spherical surface. Since no analytical solutions exist, validation is done via comparison to other published results in the literature with respect to scalar global quantities, such as Nusselt number at the inner and outer boundaries ($\mathrm{Nu_i}$ and $\mathrm{Nu_o}$), and the averaged root mean square velocity and temperature over the volume. Table 4 contains such a comparison for the RBF-PS method with respect to popular methods used in the mantle convection literature. The following observations can be can be made:

[27] 1. The only method that is spectral in at least one direction is the spherical harmonic–finite difference method of *Harder* [1998]. In the work by *Stemmer et al.* [2006], Harder's method was used with Romberg extrapolation to obtain the results to at least four digits of accuracy. With regard to almost all quantities for both test cases, the results of the RBF-PS method match exactly with Harder's extrapolated results.

[28] 2. The number of nodes (degrees of freedom) needed to accomplish the results in point 1 is an order of magnitude lower than what was used with the CitcomS model reported by *Zhong et al.* [2008], approximately one and a half orders of magnitude lower than either the finite volume method by *Stemmer et al.* [2006] or the method by *Harder* [1998] and three orders of magnitude less than the Yin-Yang, multigrid method by *Kameyama et al.* [2008]. It should be noted, however, that with exception to the *Stemmer et al.* [2006] and *Harder* [1998] results, a detailed convergence study was not performed for these methods to determine the minimal degrees of freedom needed to achieve their reported results.

[29] 3. For the scheme to conserve energy $\mathrm{Nu_i}$ = $\mathrm{Nu_o}$, notice that this is the case for both tests even with such a low number of nodes. This results from the spectral accuracy of the RBF-PS method, which by its shear high-order convergence, will inherently dissipate physical quantities less.

[30] 4. Even though we are using Chebyshev polynomials in the radial direction and an explicit time-stepping scheme for the advection term in the temperature equation, we still can take the same number of total time steps as *Zhong et al.* [2008] (i.e., 10,000).

[31] 5. Even if we decrease the number of nodes by over 50%, there is only very minor changes in the results. From our calculations, it seems that at least 17 interior nodes (19 total nodes) are needed in the radial direction to resolve the flow at Ra = 7000. Although not reported, we found that in the lateral direction the number of RBF nodes could be reduced by another 20% and the values reported changed only slightly (in the third decimal place).

7.2. Ra = 10^5

[32] For the isoviscous Ra = 10^5 case, there are only two published studies for comparison, the CitcomS study of *Zhong et al.* [2008] and *Ratcliff et al.* [1996]. Both use the cubic test case initial condition given by (30) and the model is integrated to approximately t = 0.3. Since Ra = 10^5 is a more convective regime, resulting in thinner plumes as seen in Figures 5a and 5b, larger resolution is needed. Thus, we use 43 Chebyshev nodes

Table 4. Comparison Between Computational Methods for the Isoviscous Tetrahedral and Cubic Mantle Convection Test Cases With Ra = 7000[a]

Model	Type	Nodes	$r \times (\theta \times \lambda)$	$\mathrm{Nu_o}$	$\mathrm{Nu_i}$	$\langle V_{\mathrm{rms}} \rangle$	$\langle T \rangle$
			Cubic Test Case, Ra = 7000				
Zhong et al. [2008]	FE	393216	32 × (12 × 32 × 32)	3.6254	3.6016	31.09	0.2176
Yoshida and Kageyama [2004]	FD	2122416	102 × (102 × 204)	3.5554	–	30.5197	–
Kameyama et al. [2008]	FD	12582912	128 × (2 × 128 × 384)	3.6083	–	31.0741	0.21639
Ratcliff et al. [1996]	FV	200000	40 × (50 × 100)	3.5806	–	30.87	–
Stemmer et al. [2006]	FV	663552	48 × (6 × 48 × 48)	3.5983	3.5984	31.0226	0.21594
Stemmer et al. [2006]	FV	Extrap.	Extrap.	3.6090	–	31.0709	0.21583
Harder [1998] and *Stemmer et al.* [2006]	SP-FD	552960	120 × (48 × 96)	3.6086	–	31.0765	0.21582
Harder [1998] and *Stemmer et al.* [2006]	SP-FD	Extrap.	Extrap.	3.6096	–	31.0821	0.21578
RBF-PS	SP	36800	23 × (1600)	3.6096	3.6096	31.0820	0.21577
RBF-PS	SP	17100	19 × (900)	3.6098	3.6098	31.0834	0.21579
			Tetrahedral Test Case, Ra = 7000				
Zhong et al. [2008]	FE	393216	32 × (12 × 32 × 32)	3.5126	3.4919	32.66	0.2171
Yoshida and Kageyama [2004]	FD	2122416	102 × (102 × 204)	3.4430	–	32.0481	–
Kameyama et al. [2008]	FD	12582912	128 × (2 × 128 × 384)	3.4945	–	32.6308	0.21597
Ratcliff et al. [1996]	FV	200000	40 × (50 × 100)	3.4423	–	32.19	–
Stemmer et al. [2006]	FV	663552	48 × (6 × 48 × 48)	3.4864	3.4864	32.5894	0.21564
Stemmer et al. [2006]	FV	Extrap.	Extrap.	3.4949	–	32.6234	0.21560
Harder [1998] and *Stemmer et al.* [2006]	SP-FD	552960	120 × (48 × 96)	3.4955	–	32.6375	0.21561
Harder [1998] and *Stemmer et al.* [2006]	SP-FD	Extrap.	Extrap.	3.4962	–	32.6424	0.21556
RBF-PS	SP	36800	23 × (1600)	3.4962	3.4962	32.6424	0.21556
RBF-PS	SP	17100	19 × (900)	3.4964	3.4963	32.6433	0.21557

[a] $\mathrm{Nu_o}$ and $\mathrm{Nu_i}$ denote the Nusselt number at the outer and inner spherical surfaces, respectively; $\langle V_{\mathrm{rms}} \rangle$ denotes the volume-averaged RMS velocity over the 3-D shell; and $\langle T \rangle$ denotes the mean temperature of the 3-D shell. Extrap. indicates that the results were obtained using Romberg extrapolation. Dashes indicate that numbers were not reported. Abbreviations are as follows: FE, finite element; FD, finite difference; FV, finite volume; SP-FD, hybrid spectral and finite difference; SP, purely spectral. For the RBF-PS method, the standard deviation of all the quantities from the last 1000 time steps was less than 5×10^{-5}, which is a standard measure for indicating the model has reached numerical steady state.

(41 interior nodes) in the radial direction and 4096 nodes on each spherical surface. Since the time step is purely restricted by the Chebyshev discretization, the increase in Chebyshev nodes results in a more severe CFL criterion that causes a necessary decrease in the time step. The time step for this case is 6×10^{-5} for stability or 50,000 time steps to reach $t = 0.3$ as opposed to the 35,000 time steps used by *Zhong et al.* [2008].

[33] Comparative results are given in Table 5. The following 5 points are notable:

[34] 1. The difference between $\mathrm{Nu_o}$ and $\mathrm{Nu_i}$ is 0.14%, showing that we are close to absolute conservation of energy.

[35] 2. The results of the RBF-PS method are much closer to that of *Zhong et al.* [2008] than to those of *Ratcliff et al.* [1996]. We attribute this difference, most likely, to the underresolution of the runs of the latter model.

[36] 3. The RBF-PS uses approximately an order of magnitude less degrees of freedom than the study by *Zhong et al.* [2008].

[37] 4. Once steady state has been reached, differences between our results and those of *Zhong et al.* [2008] are within 0.4% for $\mathrm{Nu_o}$ and $\mathrm{Nu_i}$, 0.2% for $\langle V_{\mathrm{rms}} \rangle$, and 0.9% for $\langle T \rangle$.

[38] 5. Even during the startup of the model, the curves for $\mathrm{Nu_o}$, $\langle V_{\mathrm{rms}} \rangle$, and $\langle T \rangle$ as a function of time, are almost indistinguishable from the results of *Zhong et al.* [2008], as seen in Figures 6a–6c.

7.3. *Ra* = 10^6

[39] Since no results from a purely spectral method have ever been reported in 3-D spherical geometry at $Ra = 10^6$, we have decided to include this result. The common practice at this and larger Ra is to start the simulation with an initial condition taken from a simulation run at a lower Ra. The primary

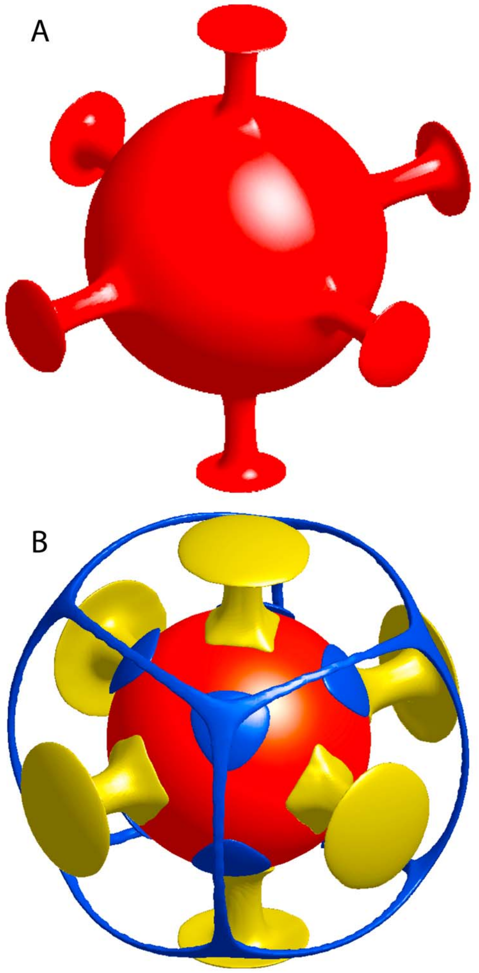

Figure 5. (a) Steady state isosurface temperature T = 0.5 at time t = 0.3. (b) Steady state isosurface of the residual temperature, $\delta T = T(r, \lambda, \theta) - \langle T(r) \rangle$, at t = 0.3 for the isoviscous cubic mantle convection test case at $Ra = 10^5$ computed with the RBF-PS model. The same color scheme as Figure 4 has been applied.

reason for this is to avoid the extremely high velocity values that occur at higher Ra during the initial redistribution of the temperature from a conductive profile to a convective profile, which severely restricts the time steps that can be taken. The initial condition used at $Ra = 10^6$ was taken from a simulation which was started at $Ra = 10^5$ with an initial condition consisting of two terms, a purely conductive temperature profile plus a small perturbation in the lateral direction that randomly combined all spherical harmonics up to degree 10. This latter term was multiplied by the same sine term in the radial direction as used in the previous cases. For the discretization, we used 81 Chebyshev nodes in the radial direction and 6561 nodes on each spherical surface, for a total of 531,441 nodes. The $Ra = 10^5$ simulation was run until the large spike in the radial velocity subsided, the Ra was then increased to 5×10^5 and the simulation was restarted with the $Ra = 10^5$ solution as the initial condition. This process was repeated once more and the Ra was then increased to 10^6 and the time was reset to 0. The $Ra = 10^6$ simulation results from $t = 0$ to $t = 0.08$ (approximately 4 and half times the age of the Earth), are displayed in Figure 7. Figure 7a displays the isosurfaces of the residual temperature at $t = 0.08$ and clearly shows the mantle in a purely convective regime. Figures 7b–7d show the time traces of $\langle T \rangle$, $\langle V_{\mathrm{rms}} \rangle$, and $\mathrm{Nu_o}$ and $\mathrm{Nu_i}$, respectively. Since this is a purely convective regime the choice of ending time is somewhat arbitrary. We chose to stop the simulation at $t = 0.08$ since by this time the average temperature had decreased to an acceptable level from its initial starting value and the influence of the initial condition had diminished.

8. Timing Results

[40] In this short section, runtime results are presented in Table 6 in order to give the reader a feel for how long it takes to run the code. All test cases were conducted on a workstation with one Intel i7 940 2.93 GHz processor, which is a quad core processor. The code was written in MATLAB and run under version 2009b with BLAS multithreading enabled. 8 GB of memory was also available for the calculations, although only a fraction of this was actually used. The results under “total runtime” in Table 6 include the preprocessing steps in

Table 5. Comparison Between Computational Methods for the Isoviscous Cubic Mantle Convection Test Case With $Ra = 10^5$[a]

Model	Type	Nodes	$r \times (\theta \times \lambda)$	$\mathrm{Nu_o}$	$\mathrm{Nu_i}$	$\langle V_{\mathrm{rms}} \rangle$	$\langle T \rangle$
Zhong et al. [2008]	FE	1,327,104	48 × (12 × 48 × 48)	7.8495 (0.0054)	7.7701 (0.001)	154.8 (0.04)	0.1728 (0.0002)
Ratcliff et al. [1996]	FV	200,000	40 × (50 × 100)	7.5669	–	157.5	–
RBF-PS	SP	176,128	43 × (4096)	7.8120	7.8005	154.490	0.17123

[a] Numbers in parentheses for *Zhong et al.* [2008] represent standard deviations. The standard deviation of the results from the last 1000 time steps for the RBF-PS model were all less than 5×10^{-5}.

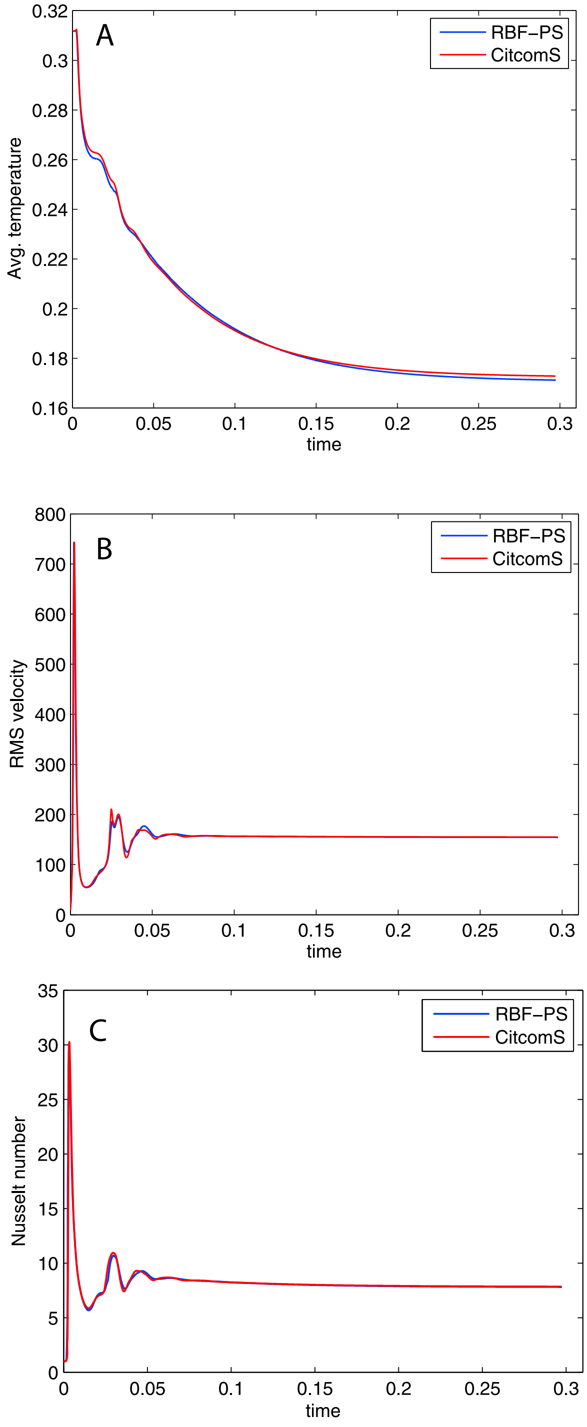

Figure 6. Time plots of the (a) $\langle T \rangle$, (b) $\langle V_{\text{rms}} \rangle$, and (c) Nu_o, comparing the results (obtained through Computational Infrastructure for Geodynamics (http://www.geodynamics.org/cig/workinggroups/mc/workarea/benchmark/3dconvention/)) of the CitcomS and RBF-PS for isoviscous mantle convection at $Ra = 10^5$.

Appendix B, such as setting up the differentiation matrices and diagonalizing them.

9. Extensions of the Method: High *Ra*, Variable Viscosity, and Local Node Refinement

[41] Each of these extensions requires different alterations of the method. We discuss them below in order of increasing complexity in terms of altering the method.

9.1. High *Ra*

[42] For simulations at high *Ra*, the resolution of the model must of course be increased further. The most expensive steps in the method are the elliptic solver and the time step due to the severe CFL restriction of Chebyshev discretization. With only moderate losses in accuracy, the time step restriction would be alleviated if sixth-order implicit (or compact) finite differences were used instead to discretize the radial direction. This would be a minor change to the implementation of the method. The advantage of a sixth-order implicit scheme over a sixth-order explicit scheme is that (1) the stencil size is almost 50% less for the same accuracy; (2) they have smaller error constants; (3) information only needs to be extrapolated to one point outside the boundary; and (4) most importantly, they have spectral-like resolution in the sense that they resolve higher wave numbers [*Lele*, 1992]. To date, the authors are not aware of a study that has employed such a scheme. With regard to the elliptic solver, global RBFs do not scale well as high *Ra* convection is reached (i.e., $Ra \sim O(10^9)$). However, RBF-based finite difference schemes (see section 9.3), already show exceptionally promising results on spherical surfaces, in terms computational scaling and accuracy (B. Fornberg and E. Lehto, A filter approach for stabilizing RBF-generated finite difference methods for convective PDEs, manuscript in preparation, 2010; N. Flyer et al., A radial basis function generated finite difference method for the unsteady shallow water equations on a sphere, manuscript in preparation, 2010).

9.2. Variable Viscosity

[43] To handle variable viscosity (depth and/or horizontal dependent), the PDEs in their primitive variables, i.e., (1) and (2), would need to be discretized. The main difficulty here is not in the

discretization of the variable coefficients terms that appear in (2), which can be handled straightforwardly by the method, but from solving the resulting coupled equations to ensure that the flow remains incompressible. This system will be of saddle point type for which many potential methods are available (see, for example, the review by *Benzi et al.* [2005]). The most promising approach for the RBF-PS system, currently being explored by authors, would use a block factorization of the discretized momentum and continuity equations, giving an upper block triangular system and resulting in a Schur complement problem. To avoid the excessive computational cost in solving this system directly at every time step, we would instead employ a preconditioned Krylov subspace method (see *Benzi et al.* [2005, section 10] for a discussion). A good choice for the preconditioner might be to use a coarse solution to the isoviscous problem, as presented by the method in section 5, since the velocities should be sufficiently smooth given that they are solutions to Poisson-type PDEs.

9.3. Local Refinement

[44] Local refinement will take the greatest alteration to the code. Once variables are separated, in this case (θ, λ) from r, the result is a tensor-like grid and local refinement becomes difficult. For RBFs, refinement is simple in the sense that the nodes can be easily clustered [*Flyer and Lehto*, 2010]. Thus, the next developmental step, which is currently in progress, is to use 3-D RBF-based finite difference stencils [*Wright and Fornberg*, 2006] for discretizing the equations. Here, all differentiation matrices are based on local high-order finite difference-type stencils that are generated by means of RBFs from nodes scattered in 3-D space, which results in exceptionally sparse matrices to solve.

10. Summary

[45] This paper develops the first spectral RBF method for 3-D spherical geometries in the math/science literature. Applications of this new hybrid

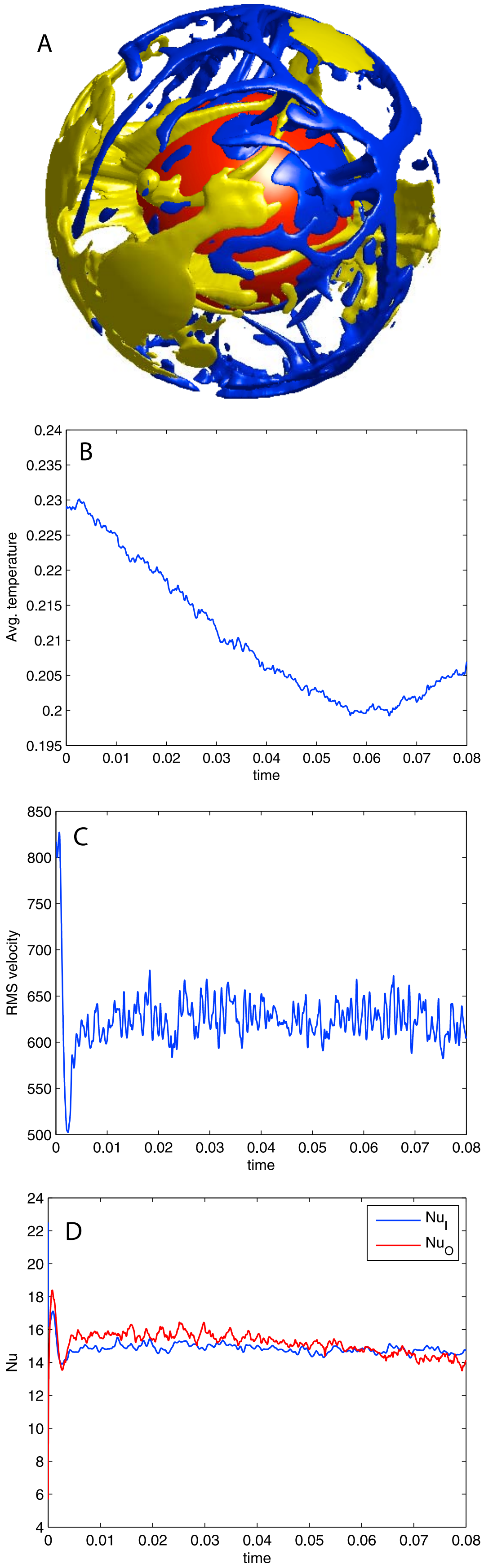

Figure 7. Results for $Ra = 10^6$ test case: (a) Residual temperature, δT, at $t = 0.08$, where yellow corresponds to $\delta T = 0.1$, blue corresponds to $\delta T = -0.1$, and the red solid sphere shows the inner boundary of the 3-D shell corresponding to the core. (b) Average temperature, $\langle T \rangle$, versus time. (c) Average RMS velocity, $\langle V_{\mathrm{rms}} \rangle$, versus time. (d) Inner and outer Nusselt numbers, $\mathrm{Nu_i}$ and $\mathrm{Nu_o}$, versus time.

Table 6. Runtime Results for the RBF-PS Method for the *Ra* = 7000 and *Ra* = 100,000 Cases on a Single 2.93 GHz Intel i7 940 Quad Core Processor

Test Case	Total Number of Nodes	Runtime per Time Step	Total Runtime	Total Time Steps
Ra = 7000	36,800	0.0516 s	8 min 16 s	10,000 (to t = 1)
Ra = 100,000	176,128	0.44 s	6 h 27 min	50,000 (to t = 0.3)

purely spectral methodology are given to the problem of thermal convection in the Earth's mantle, which is modeled by a Boussinesq fluid at infinite Prandtl number. To see whether this numerical technique warrants further investigation, the study limits itself to an isoviscous mantle. Two *Ra* number regimes are tested, the classical case in the literature of *Ra* = 7000 and the latest results from the CitcomS model [*Zhong et al.*, 2008] at *Ra* = 10^5. Also, a *Ra* = 10^6 simulation is run, as there are no results in the literature from a purely spectral model at this *Ra*. For the *Ra* = 7000 case, the method perfectly conserved energy, matched the extrapolated results of the only other partially spectral method (reported by *Stemmer et al.* [2006]) to four or 5 significant digits and used anywhere between 1 to 3 orders of magnitude less nodes than other methods. For the *Ra* = 10^5 case, the method almost perfectly conserved energy, and gave results that differed from CitcomS by 0.2% to 0.9% (depending on the scalar quantity being measured) yet required approximately an order of magnitude less nodes. All calculations were run on a workstation using a single quad core processor with *Ra* = 7000 case taking about 8 min and the *Ra* = 10^5 case taking about 6.5 h. Given this outcome, the methodology warrants more extensive testing. It will be altered to accommodate the PDEs in primitive form so that variable viscosity and thermochemical convection can be considered in future test runs.

[46] We conclude by noting that the spectral RBF method presented here may also be a potentially promising technique for simulating the geodynamo, which is well approximated by an isoviscous and Boussinesq fluid. For the geodynamo, the Prandtl number is finite and the (now time-dependent) momentum and energy equations are coupled to a time-dependent induction equation for the magnetic field [see, e.g., *Glatzmaier and Roberts*, 1995]. For the hybrid RBF method, a natural approach would be to decompose the velocity and magnetic fields into toroidal and poloidal potentials as is commonly done in other models [see, e.g., *Christensen et al.*, 2001; *Glatzmaier and Roberts*, 1995; *Oishi et al.*, 2007]. This decomposition results in a coupled set of time-dependent parabolic equations and a set of time-independent elliptic equations. For the former, a similar approach to the discretization for the energy equation discussed in section 6 could be used, while the latter could be solved using the technique discussed in Appendix A.

Appendix A: Solution of the Momentum Equations via Matrix Diagonalization

[47] The two coupled Poisson equations representing the momentum equations are solved through matrix diagonalization (i.e., spectral decomposition) of the RBF surface Laplacian and Chebyshev radial operators and the influence matrix method briefly discussed in section 5. In this appendix, we first show how the two coupled equations in rows 1 and 2 of Table 2 are solved for Ω_h and Φ_h. We then describe the influence matrix method in the spectral space of the discrete operators We conclude with details on how the full solution for Φ is obtained.

[48] If N is the number of nodes on a spherical surface and M is the number of interior Chebyshev nodes in the radial direction, then let $\Omega_h, \Phi_h \in \mathbb{R}^{N\times M}$ be the respective matrices for the unknown values of the two potentials at all the interior node points at time t. Also let $T \in \mathbb{R}^{N\times M}$ be the known values of the temperature at time t. Using the notation from Table 3 for the RBF differentiation matrix for the surface Laplacian and the Chebyshev differentiation matrix for the radial component of the 3-D Laplacian, the discrete form of the first two equations in Table 2 is written as

$$L_s\Omega_h + \Omega_h L_r = Ra\, T\, R, \tag{A1}$$

$$L_s\Phi_h + \Phi_h L_r = \Omega_h\, R^2, \tag{A2}$$

where R is the diagonal matrix given in (27). Now, letting $L_s = V_s\Lambda_s V_s^{-1}$ and $L_r = V_r\Lambda_r V_r^{-1}$ be the spectral decompositions of the operators L_s and L_r, respectively, (A1) and (A2) can be written, after some manipulations as

$$\Lambda_s(V_s^{-1}\Omega_h V_r) + (V_s^{-1}\Omega_h V_r)\Lambda_r = Ra\,(V_s^{-1} T\, R V_r),$$

$$\Lambda_s(V_s^{-1}\Phi_h V_r) + (V_s^{-1}\Phi_h V_r)\Lambda_r = (V_s^{-1}\Omega_h V_r)(V_r^{-1}R^2 V_r).$$

By defining $\widehat{\Omega}_h = V_s^{-1}\Omega_h V_r$, $\widehat{\Phi}_h = V_s^{-1}\Phi_h V_r$, $\widehat{T} = V_s^{-1}T\, R V_r$, and $\widetilde{R}^2 = V_r^{-1}R^2 V_r$, the above equations can be written as the *diagonal* system of equations

$$\Lambda_s\widehat{\Omega}_h + \widehat{\Omega}_h\Lambda_r = Ra\,\widehat{T}, \tag{A3}$$

$$\Lambda_s\widehat{\Phi}_h + \widehat{\Phi}_h\Lambda_r = \widehat{\Omega}_h\widetilde{R}^2. \tag{A4}$$

The solutions to these equations are given explicitly as

$$(\widehat{\Omega}_h)_{i,j} = Ra\frac{(\widehat{T})_{i,j}}{(\Lambda_s)_{i,i} + (\Lambda_r)_{j,j}} \quad \text{and} \quad (\widehat{\Phi}_h)_{i,j} = \frac{(\widehat{\Omega}_h\widetilde{R}^2)_{i,j}}{(\Lambda_s)_{i,i} + (\Lambda_r)_{j,j}}, \tag{A5}$$

for $i = 1,\ldots, N$, and $j = 1,\ldots, M$.

[49] The operators L_s and L_r are time independent so that their spectral decompositions can be computed as a preprocessing step. The solution to (A3) and (A4) thus requires $O(MN)$ operations per time step. However, the total cost in computing $\widehat{\Phi}_h$ per time step is dominated by the cost of computing $\widehat{T}$ and $\widehat{\Omega}_h\widetilde{R}^2$, which requires $O(MN^2)$ and $O(M^2N)$ operations, respectively. Since N is typically two orders of magnitude greater than M, the former computation will dominate everything.

[50] We also apply the influence matrix technique in spectral space as it allows some reduction in the storage and computational cost of computing the final value of Φ. The setup of this technique is similar to that described in section 5 in that we look for a superposition of solutions to the equations. However, by working in the spectral space of the lateral operator L_s, the lateral directions can be decoupled so that the superposition consists only of three $N \times M$ linear systems instead of N:

$$\overline{\Omega} = \overline{\Omega}_h + \overline{\Omega}_{bd}^{R_\mathrm{i}}\overline{\Omega}^{R_\mathrm{i}} + \overline{\Omega}_{bd}^{R_\mathrm{o}}\overline{\Omega}^{R_\mathrm{o}} \quad \text{and}$$
$$\overline{\Phi} = \overline{\Phi}_h + \overline{\Omega}_{bd}^{R_\mathrm{i}}\overline{\Phi}^{R_\mathrm{i}} + \overline{\Omega}_{bd}^{R_\mathrm{o}}\overline{\Phi}^{R_\mathrm{o}}, \tag{A6}$$

where bars indicate the variables are in the spectral space of the lateral operator L_s only, and $\overline{\Omega}_{bd}^{R_\mathrm{i}}$ and $\overline{\Omega}_{bd}^{R_\mathrm{o}}$ are $N \times N$ diagonal matrices with the unknown values for enforcing the boundary conditions. The values of $\overline{\Omega}_h$ and $\overline{\Phi}_h$, can be obtained by multiplying $\widehat{\Omega}_h$ and $\widehat{\Phi}_h$ (computed from (A3) and (A4)) on the right by V_r^{-1}. The remaining values are determined from the solution of N independent two-point boundary value systems in the radial direction which correspond to the transform of the two sets of coupled PDEs in rows 3–6 of Table 2 into the spectral space of L_s. The complete set of equations can be written in discrete form as

$$\begin{cases} \Lambda_s\overline{\Omega}^{R_\mathrm{i}} + \overline{\Omega}^{R_\mathrm{i}}L_r & = B^{R_\mathrm{i}}, \\ \Lambda_s\overline{\Phi}^{R_\mathrm{i}} + \overline{\Phi}^{R_\mathrm{i}}L_r & = \overline{\Omega}^{R_\mathrm{i}}\widetilde{R}^2, \end{cases} \quad \text{and}$$
$$\begin{cases} \Lambda_s\overline{\Omega}^{R_\mathrm{o}} + \overline{\Omega}^{R_\mathrm{o}}L_r & = B^{R_\mathrm{o}}, \\ \Lambda_s\overline{\Phi}^{R_\mathrm{o}} + \overline{\Phi}^{R_\mathrm{o}}L_r & = \overline{\Omega}^{R_\mathrm{o}}\widetilde{R}^2, \end{cases} \tag{A7}$$

where B^{R_i} and B^{R_o} contain the modifications from the boundary conditions $(\overline{\Omega}^{R_\mathrm{i}})_j = 1$ at $r = R_\mathrm{i}$ and $(\overline{\Omega}^{R_\mathrm{o}})_j = 1$ at $r = R_\mathrm{o}$ (the remaining boundary conditions on the variables are homogeneous). The solutions to both of these systems can be computed by a transformation into the spectral space of L_r as done in (A3) and (A4) and then solving a diagonal system. The total operation for the solution of $\overline{\Phi}^{R_\mathrm{i}}$ and $\overline{\Phi}^{R_\mathrm{o}}$ can then be computed in $O(M^2N)$ operations. However, this can be done as a preprocessing step as these values are time independent.

[51] To find the unknown values on the diagonals of $\overline{\Omega}_{bd}^{R_\mathrm{i}}$ and $\overline{\Omega}_{bd}^{R_\mathrm{o}}$ we apply the second set of boundary conditions in (8) to the right-hand side of the $\overline{\Phi}$ equation in (A6). The discrete set of equations that result are given by

$$\left(\overline{\Phi}^{R_\mathrm{i}}D_{rr}^{R_\mathrm{i}}\right)\overline{\Omega}_{bd}^{R_\mathrm{i}} + \left(\overline{\Phi}^{R_\mathrm{o}}D_{rr}^{R_\mathrm{i}}\right)\overline{\Omega}_{bd}^{R_\mathrm{o}} = -\overline{\Phi}_h D_{rr}^{R_\mathrm{i}}, \tag{A8}$$

$$\left(\overline{\Phi}^{R_\mathrm{i}}D_{rr}^{R_\mathrm{o}}\right)\overline{\Omega}_{bd}^{R_\mathrm{i}} + \left(\overline{\Phi}^{R_\mathrm{o}}D_{rr}^{R_\mathrm{o}}\right)\overline{\Omega}_{bd}^{R_\mathrm{o}} = -\overline{\Phi}_h D_{rr}^{R_\mathrm{o}}, \tag{A9}$$

where $D_{rr}^{R_\mathrm{i}}, D_{rr}^{R_\mathrm{o}} \in \mathbb{R}^{M\times 1}$ are the discrete Chebyshev second derivative operators in the radial direction at the inner and outer boundaries, respectively. Equations (A8) and (A9) correspond to N decoupled 2×2 linear system for finding the unknowns. Thus, all these systems can be solved in $O(N)$ operations. Furthermore, since $\overline{\Phi}^{R_\mathrm{i}}$ and $\overline{\Phi}^{R_\mathrm{o}}$ are time independent the values in parentheses on the left-hand side of these equations can be computed as a preprocessing step. The values on the right-hand side are time dependent and need to be computed every time step, which requires $O(MN)$ operations.

[52] Once $\overline{\Omega}_{bd}^{R_\mathrm{i}}$ and $\overline{\Omega}_{bd}^{R_\mathrm{o}}$ are determined, $\overline{\Phi}$ is computed according to (A6) and then transformed back into physical space by computing $\Phi = V_s\overline{\Phi}$. The computation of $\overline{\Phi}$ requires $O(MN)$ operations and the computation of Φ requires $O(MN^2)$ operations.

Combining this cost with the cost of computing $\widehat{\Phi}_h$ from (A3) and (A4), we see the total computational cost per time step for computing Φ will be $O(MN^2)$ operations.

Appendix B: Overview of RBF-PS Algorithm

B1. Setup (Preprocessing)

[53] 1. Discretize $\frac{\partial}{\partial\theta}$, $\frac{1}{\cos\theta}\frac{\partial}{\partial\lambda}$, and Δ_{s} using collocation with N RBFs. This will result in 3 $N \times N$ matrices, D_θ, D_λ, and L_s (see sections 4.1 and 4.2).

[54] 2. Discretize $\frac{\partial}{\partial r}, \frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right), \frac{\partial^2}{\partial r^2}\Big|_{r=R_{\mathrm{i}}}$, and $\frac{\partial^2}{\partial r^2}\Big|_{r=R_{\mathrm{o}}}$ using collocation with $M + 2$ Chebyshev polynomials. Since these are only applied on the interior of the shell, this will result in 2 $M \times M$ matrices D_r and L_r for the first two operators, and 2 $M \times 1$ vectors for the last two.

[55] 3. Form the $M \times M$ diagonal matrices R (see (27)) and R^2.

[56] 4. LU decompose the $M \times M$ matrix $(I - \Delta t/2L_rR^{-2})$ for the AB3-CN time stepping method (see section 6), where I is the $M \times M$ identity matrix.

[57] 5. Momentum equations:

i. Compute the spectral decomposition of L_s and L_r (see Appendix A). This results in the following matrices: 1 full $N \times N$ (V_s), 1 diagonal $N \times N$ (Λ_s), 1 full $M \times M$ (V_r), and 1 diagonal $M \times M$ (Λ_r).

ii. Compute the inverses of V_r and V_s.

iii. Compute the full $M \times M$ matrix $\widetilde{R}^2 = V_r^{-1}R^2V_r$.

iv. Solve (A7) for $\overline{\Phi}^{R_{\mathrm{i}}}$ and $\overline{\Phi}^{R_{\mathrm{o}}}$, which results in 2 $N \times M$ matrices.

v. Compute the influence matrix entries $\overline{\Phi}^{R_{\mathrm{i}}}D_{rr}^{R_{\mathrm{i}}}$, $\overline{\Phi}^{R_{\mathrm{i}}}D_{rr}^{R_{\mathrm{o}}}$, $\overline{\Phi}^{R_{\mathrm{o}}}D_{rr}^{R_{\mathrm{i}}}$, and $\overline{\Phi}^{R_{\mathrm{o}}}D_{rr}^{R_{\mathrm{o}}}$ from (A8) and (A8). This results in 4 vectors of length N.

B2. Execution

[58] 1. Momentum equations:

i. With the temperature T^k at the kth time step, calculate $\widehat{\Phi}_h$ (i.e., Φ_h in the spectral space of both the operators L_s and L_r) according to (A5).

ii. Calculate $\overline{\Phi}_h = \widehat{\Phi}_hV_r^{-1}$ (i.e., Φ_h in the spectral space of L_s only).

iii. Solve for the influence matrix system (A7) for the unknowns $\overline{\Omega}_{bd}^{R_{\mathrm{i}}}$ and $\overline{\Omega}_{bd}^{R_{\mathrm{o}}}$.

iv. Compute $\overline{\Phi}$ by updating $\overline{\Phi}_h$ with the influence matrix terms according to (A6).

v. Compute $\Phi = V_s\,\overline{\Phi}$ to find the poloidal potential in physical space.

vi. Compute the velocity field (9), which can be written

$$\mathbf{u} = (\underbrace{L_s\Phi R^{-1}}_{u_r}, \underbrace{D_\theta\Phi RD_rR^{-1}}_{u_\theta}, \underbrace{D_\lambda\Phi RD_rR^{-1}}_{u_\lambda}).$$

[59] 2. Compute the F_k and G_k in (26) with the velocity field from the previous step and the temperature T^k.

[60] 3. Solve (28) for the temperature at the next time step, T^{k+1}, and repeat step 1 with this new temperature.

Acknowledgments

[61] G. B. Wright was supported by NSF grants ATM-0801309 and DMS-0934581. N. Flyer was supported by NSF grants ATM-0620100 and DMS-0934317. N. Flyer's work was in part carried out while she was a Visiting Fellow at Oxford Centre for Collaborative Applied Mathematics (OCCAM) under support provided by award KUK-C1-013-04 to the University of Oxford, UK, by King Abdullah University of Science and Technology (KAUST). D. A. Yuen was supported by NSF grant DMS-0934564.

References

Backus, G. E. (1966), Potentials for tangent tensor fields on spheroids, *Arch. Ration. Mech. Anal.*, *22*, 210–252.

Baumgardner, J. R. (1985), Three dimensional treatment of convection flow in the Earth's mantle, *J. Stat. Phys.*, *39*(5–6), 501–511.

Benzi, M., G. H. Golub, and J. Liesen (2005), Numerical solution of saddle point problems, *Acta Numer.*, *14*, 1–137, doi:10.1017/S0962492904000212.

Bercovici, D., G. Schubert, G. A. Glatzmaier, and A. Zebib (1989), Three-dimensional thermal-convection in a spherical shell, *J. Fluid Mech.*, *206*, 75–104.

Buhmann, M. D. (2003), *Radial Basis Functions: Theory and Implementations*, Cambridge Univ. Press, Cambridge, U. K.

Chandrasekhar, S. (1961), *Hydrodynamic and Hydromagnetic Stability*, Dover, New York.

Choblet, G., O. Čadek, F. Couturier, and C. Dumoulin (2007), Oedipus: A new tool to study the dynamics of planetary interiors, *Geophys. J. Int.*, *170*, 9–30.

Christensen, U. R., et al. (2001), A numerical dynamo benchmark, *Phys. Earth Planet. Inter.*, *128*, 25–34.

Driscoll, T. A., and A. Heryundono (2007), Adaptive residual subsampling methods for radial basis function interpolation and collocation problems, *Comput. Math. Appl.*, *53*, 927–939.

Fasshauer, G. E. (2007), *Meshless Approximation Methods With MATLAB*, *Interdisciplinary Math. Sci.*, vol. 6, World Sci., Singapore.

Fasshauer, G. E., and J. G. Zhang (2007), On choosing "optimal" shape parameters for RBF approximation, *Numer. Algorithms*, *45*, 345–368.

Flyer, N., and E. Lehto (2010), Rotational transport on a sphere: Local node refinement with radial basis functions, *J. Comput. Phys.*, *229*, 1954–1969, doi:10.1016/j.jcp.2009.11.016.

Flyer, N., and G. B. Wright (2007), Transport schemes on a sphere using radial basis functions, *J. Comput. Phys.*, *226*, 1059–1084.

Flyer, N., and G. B. Wright (2009), A radial basis function method for the shallow water equations on a sphere, *Proc. R. Soc. A*, *465*, 1949–1976.

Fornberg, B. (1995), *A Practical Guide to Pseudospectral Methods*, Cambridge Univ. Press, Cambridge, U. K.

Fornberg, B., and N. Flyer (2005), Accuracy of radial basis function interpolation and derivative approximations on 1-D infinite grids, *Adv. Comput. Math.*, *23*, 5–20.

Fornberg, B., and C. Piret (2008), On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere, *J. Comput. Phys.*, *227*, 2758–2780.

Fornberg, B., and J. Zuev (2007), The Runge phenomenon and spatially variable shape parameters in RBF interpolation, *Comput. Math. Appl.*, *54*, 379–398.

Fornberg, B., T. A. Driscoll, G. Wright, and R. Charles (2002), Observations on the behavior of radial basis functions near boundaries, *Comput. Math. Appl.*, *43*, 473–490.

Fornberg, B., G. Wright, and E. Larsson (2004), Some observations regarding interpolants in the limit of flat radial basis functions, *Comput. Math. Appl.*, *47*, 37–55.

Fornberg, B., N. Flyer, and J. M. Russell (2010), Comparisons between pseudospectral and radial basis function derivative approximations, *IMA J. Numer. Anal.*, *30*(1), 149–172.

Glatzmaier, G. A., and P. H. Roberts (1995), A three-dimensional self-consistent computer simulation of a geomagnetic field reversal, *Nature*, *377*, 203–209.

Harder, H. (1998), Phase transitions and the three-dimensional planform of thermal convection in the Martian mantle, *J. Geophys. Res.*, *103*, 16,775–16,797.

Harder, H., and U. Hansen (2005), A finite-volume solution method for thermal convection and dynamo problems in spherical shells, *Geophys. J. Int.*, *161*, 522–532.

Hardin, D. P., and E. B. Saff (2004), Discretizing manifolds via minimum energy points, *Not. Am. Math. Soc.*, *51*, 1186–1194.

Hernlund, J. W., and P. J. Tackley (2003), Three-dimensional spherical shell convection at infinite Prandtl number using the 'cubed sphere' method, in *Computational Fluid and Solid Mechanics*, edited by K. Bathe, pp. 931–933, Elsevier, Oxford, U. K.

Hüttig, C., and K. Stemmer (2008), The spiral grid: A new approach to discretize the sphere and its application to mantle convection, *Geochem. Geophys. Geosyst.*, *9*, Q02018, doi:10.1029/2007GC001581.

Iske, A. (2004), *Multiresolution Methods in Scattered Data Modelling*, Springer, Heidelberg, Germany.

Kameyama, M. C., A. Kageyama, and T. Sato (2008), Multigrid-based simulation code for mantle convection in spherical shell using Yin-Yang grid, *Phys. Earth Planet. Inter.*, *171*, 19–32.

Lele, S. K. (1992), Compact finite difference schemes with spectral-like resolution, *J. Comput. Phys.*, *103*, 16–42.

Oishi, Y., A. Sakuraba, and Y. Hamano (2007), Numerical method for geodynamo simulations based on Fourier expansion in longitude and finite difference in meridional plane, *Phys. Earth Planet. Inter.*, *164*, 208–229.

Peyret, R. (2002), *Spectral Methods for Incompressible Viscous Flow*, Springer, New York.

Ratcliff, J. T., G. Schubert, and A. Zebib (1996), Steady tetrahedral and cubic patterns of spherical shell convection with temperature-dependent viscosity, *J. Geophys. Res.*, *101*, 473–484.

Richards, M. A., W.-S. Yang, J. R. Baumgardner, and H.-P. Bunge (2001), Role of a low-viscosity zone in stabilizing plate tectonics: Implications for comparative terrestrial planetology, *Geochem. Geophys. Geosyst.*, *2*, 1026, doi:10.1029/2000GC000115.

Stemmer, K., H. Harder, and U. Hansen (2006), A new method to simulate convection with strongly temperature-dependent and pressure-dependent viscosity in spherical shell, *Phys. Earth Planet. Inter.*, *157*, 223–249.

Trefethen, L. N. (2000), *Spectral Methods in MATLAB*, Soc. for Ind. and Appl. Math., Philadelphia, Pa.

Weideman, J. A., and S. C. Reddy (2000), A MATLAB differentiation matrix suite, *ACM Trans. Math. Software*, *26*, 465–519.

Wertz, J., E. J. Kansa, and L. Ling (2006), The role of multiquadric shape parameters in solving elliptic partial differential equations, *Comput. Math. Appl.*, *51*, 1335–1348.

Wright, G. B., and B. Fornberg (2006), Scattered node compact finite difference-type formulas generated from radial basis functions, *J. Comput. Phys.*, *212*, 99–123.

Yoshida, M., and A. Kageyama (2004), Application of the Yin-Yang grid to a thermal convection of a Boussinesq fluid with infinite Prandtl number in a three-dimensional spherical shell, *Geophys. Res. Lett.*, *31*, L12609, doi:10.1029/2004GL019970.

Zhang, S., and U. Christensen (1993), On effects of lateral viscosity variations on geoid and surface velocities induced by density anamolies in the mantle, *Geophys. J. Int.*, *114*, 531–547.

Zhong, S., M. T. Zuber, L. Moresi, and M. Gurnis (2000), Role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection, *J. Geophys. Res.*, *105*, 11,083–11,082.

Zhong, S., A. McNamara, E. Tan, L. Moresi, and M. Gurnis (2008), A benchmark study on mantle convection in a 3-D spherical shell using CitcomS, *Geochem. Geophys. Geosyst.*, *9*, Q10017, doi:10.1029/2008GC002048.