Handbook of Geomathematics
DOI 10.1007/978-3-642-27793-1_61-1

Radial Basis Function-Generated Finite Differences: A Mesh-Free Method for Computational Geosciences

Natasha Flyer[a]*, Grady B. Wright[b] and Bengt Fornberg[c]
[a]Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, Boulder, CO, USA
[b]Department of Mathematics, Boise State University, Boise, ID, USA
[c]Department of Applied Mathematics, University of Colorado, Boulder, CO, USA

Abstract

Radial basis function-generated finite differences (RBF-FD) is a mesh-free method for numerically solving partial differential equations that emerged in the last decade and have shown rapid growth in the last few years. From a practical standpoint, RBF-FD sprouted out of global RBF methods, which have shown exceptional numerical qualities in terms of accuracy and time stability for numerically solving PDEs, but are not practical when scaled to very large problem sizes because of their computational cost and memory requirements. RBF-FD bypass these issues by using local approximations for derivatives instead of global ones. Matrices in the RBF-FD methodology go from being completely full to 99 % empty. Of course, the sacrifice is the exchange of spectral accuracy from the global RBF methods for high-order algebraic convergence of RBF-FD, assuming smooth data. However, since natural processes are almost never infinitely differentiable, little is lost and much gained in terms of memory and runtime. This chapter provides a survey of a group of topics relevant to using RBF-FD for a variety of problems that arise in the geosciences. Particular emphasis is given to problems in spherical geometries, both on surfaces and within a volume. Applications discussed include nonlinear shallow water equations on a sphere, reaction–diffusion equations, global electric circuit, and mantle convection in a spherical shell. The results from the last three of these applications are new and have not been presented before for RBF-FD.

1 Introduction to the Concept of Radial Basis Functions (RBFs)

The motivation for the RBF method originated with R.L. Hardy asking the question, "Given a set of sparse *scattered* data, $\{f_j\}_{j=1}^N$, at the node locations $\{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{R}^d$, can an interpolant be constructed that adequately represents the unknown surface?" (Hardy 1971). It was first shown by Mairhuber (1956) that, in more than one dimension, interpolation is not well posed when using an expansion of basis functions, $\{\psi_j(\mathbf{x})\}_{j=1}^N$, $\mathbf{x} \in \mathbb{R}^{d\geq 2}$, that are independent of the node locations. That is, there exist an infinite number of node configurations that will yield a singular interpolation problem. Hardy bypassed this singularity problem by constructing the interpolant from a basis function set consisting of translates of a single radially symmetric function with one centered at each data location. By giving up orthogonality, well posedness of the interpolant and its derivatives for *any* set of distinct scattered nodes in *any* dimension is guaranteed. Although unconditional non-singularity of the interpolation problem was known early in some special cases (Bochner 1933;

*E-mail: flyer@ucar.edu

DOI 10.1007/978-3-642-27793-1_61-1

Table 1 Some common choices for radial functions

Type of basis function	Radial function $\phi(r)$
Piecewise smooth RBFs	
Generalized Duchon spline (GDS)	$r^{2m}\log r,\ m \in \mathbb{N}$
	$r^{2m},\ m > 0$ and $m \notin \mathbb{N}$
Matern	$\frac{2^{1-m}}{\Gamma(m)} r^m K_m(r), m > 0$ (Bessel K-function)
Compact support ("Wendland")	Special cases: e^{-r}, $e^{-r}(1+r)$ for $m = \frac{1}{2}, \frac{3}{2}$. $(1-r)_+^m p(r)$, p certain polynomials, $m \in N$
Infinitely smooth RBFs	
Gaussian (GA)	$e^{-(\varepsilon r)^2}$
Multiquadric (MQ)	$\sqrt{1+(\varepsilon r)^2}$
Inverse multiquadric (IMQ)	$1/\sqrt{1+(\varepsilon r)^2}$
Inverse quadratic (IQ)	$1/(1+(\varepsilon r)^2)$
Sech (SH)	$\operatorname{sech} \varepsilon r$
Bessel (BE) ($d = 1, 2, \ldots$)	$J_{d/2-1}(\varepsilon r)/(\varepsilon r)^{d/2-1}$

Schoenberg 1938), the proof in 1986 of guaranteed non-singularity for multiquadric (MQ) RBFs (Micchelli 1986) accelerated the further development and acceptance of RBFs. Pioneering work by M.J.D. Powell and his collaborators at University of Cambridge played also a major role in the early history of RBFs (Powell 1992).

Piecewise smooth RBFs feature a jump in some derivative and thus can only lead to algebraic convergence. For example, the radial cubic $|r|^3$, where $r = \|\mathbf{x} - \mathbf{x}_j\|$ is the Euclidean norm, has a jump in the third derivative at $\mathbf{x} = \mathbf{x}_j$, leading to fourth-order convergence in 1D, with the order of convergence increasing as the dimension increases (cf. Powell 1992). On the other hand, interpolating with infinitely smooth RBFs, such as $\sqrt{1+(\varepsilon r)^2}$, $\exp\left(-(\varepsilon r)^2\right)$, and $1/\left(1+(\varepsilon r)^2\right)$, will lead to spectral convergence (Madych and Nelson 1992; Yoon 2001). Table 1 shows commonly used RBFs, noting that infinitely smooth RBFs depend on a shape parameter ε. It was first shown by Driscoll and Fornberg (2002) that, in 1D, in the limit of $\varepsilon \to 0$ (i.e., flat RBFs) the RBF methodology reproduces pseudospectral methods (PS) if the nodes are accordingly placed (i.e., equispaced nodes for Fourier methods, Gauss–Chebyshev nodes for Chebyshev methods, etc.). Similarly, on the surface of a sphere, Fornberg and Piret (2007) showed that, in the limit of $\varepsilon \to 0$, RBFs reproduce spherical harmonics in the sense that they span an equivalent space for any scattered node set. Later in the chapter, algorithms of how to compute in the near-flat RBF limit will be discussed.

2 RBF-Generated Finite Difference (RBF-FD) Approximations

Before delving into differentiation, interpolation is discussed as it can be considered a zeroth-order differentiation. RBF interpolation is based on a linear combination of translates of a single radially symmetric function, $\phi(\|\mathbf{x} - \mathbf{x}_k\|)$, that collocates the data $\{f_k\}_{k=1}^N$ at the nodes $\{\mathbf{x}_k\}_{k=1}^N$, $\mathbf{x}_k \in \mathbb{R}^d$, and is given by

$$s(\mathbf{x}) = \sum_{k=1}^{N} c_k \phi(\|\mathbf{x} - \mathbf{x}_k\|), \tag{1}$$

DOI 10.1007/978-3-642-27793-1_61-1

where $\|\cdot\|$ denotes the ℓ_2 norm. The expansion coefficients, c_k, can be found by inverting the matrix, A, in (2):

$$\underbrace{\begin{bmatrix} \phi(||\mathbf{x}_1-\mathbf{x}_1||) & \phi(||\mathbf{x}_1-\mathbf{x}_2||) & \cdots & \phi(||\mathbf{x}_1-\mathbf{x}_N||) \\ \phi(||\mathbf{x}_2-\mathbf{x}_1||) & \phi(||\mathbf{x}_2-\mathbf{x}_2||) & \cdots & \phi(||\mathbf{x}_2-\mathbf{x}_N||) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(||\mathbf{x}_N-\mathbf{x}_1||) & \phi(||\mathbf{x}_N-\mathbf{x}_2||) & \cdots & \phi(||\mathbf{x}_N-\mathbf{x}_N||) \end{bmatrix}}_{\text{Interpolation matrix A}} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}. \tag{2}$$

Since RBFs only depend on a scalar distance (defined by the ℓ_2 norm), the form of (2) is independent of *coordinate system*, *dimension*, and *domain geometry*. As an example, even if node locations on a sphere are given in spherical coordinates, no such grids need to be used. Distances are measured straight through the sphere and not along great arcs. That is, the argument of the RBF centered at, (x_1, y_1, z_1) and evaluated at (x_2, y_2, z_2) is

$$\begin{aligned} r &= \sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2} = \sqrt{2\left(1-\mathbf{x}_2^T\mathbf{x}_1\right)} \\ &= \sqrt{2\left(1-\cos\theta_2\cos\theta_1\cos(\lambda_2-\lambda_1)-\sin\theta_2\sin\theta_1\right)}. \end{aligned}$$

where $\theta_{1,2}$, $\lambda_{1,2}$ are the respective latitude and longitude of the points.

The RBF differentiation matrix (DM), D_N, is derived by applying the analytic derivative operator L of the RBF interpolant and evaluating it at the desired node locations. As an example on a sphere, to approximate L at $(\lambda, \theta) = (\lambda_i, \theta_i)$

$$\begin{aligned} Ls(\lambda_i,\theta_i) &= \sum_{k=1}^{N} c_k \underbrace{[L\phi_k(r)]|_{(\lambda,\theta)=(\lambda_i,\theta_i)}}_{\text{Components of } B} \qquad (i=1,\ldots,N) \\ &= B\boldsymbol{c} \\ &= \left(BA^{-1}\right)\mathbf{f} \\ &= D_N\mathbf{f}, \end{aligned} \tag{3}$$

where $\mathbf{f}$ contains the N discrete data values at the node locations and $\boldsymbol{c}$ contains the N discrete expansion coefficients and is formally given by $\boldsymbol{c} = A^{-1}\mathbf{f}$, where A^{-1} is the inverse of the RBF interpolation matrix defined in (2).

The above derivation in (3) is for global RBF DM. That is, to calculate the discretized derivative operator at one node, all the nodes in the domain are used. This is analogous to a spectral collocation method. Thus, for smooth data and using infinitely smooth RBF, spectral convergence can be expected. Deriving RBF-FD DMs is very similar, except that for each stencil, only n nodes out of the N nodes in the domain are used. Similar to regular finite differences, as the stencil size n increases so does the rate of algebraic convergence. Now, the differential operator L is approximated at the node $\mathbf{x}_c$ by a linear combination of the function values, u_k, at the neighboring $n-1$ node locations, $\mathbf{x}_k$. In other words $\sum_{k=1}^{n} a_k u_k = Lu_c$. The differentiation weights, a_k, are calculated by enforcing that this linear combination should be exact for RBFs, $\{\phi(\|\mathbf{x}-\mathbf{x}_k\|)\}_{k=1}^{n}$, centered at each of the node locations $\{\mathbf{x}_k\}_{k=1}^{n}$ (classical FD would enforce that it be exact for

DOI 10.1007/978-3-642-27793-1_61-1

polynomials instead). It has also been shown through experience and studies (Fornberg et al. 2002; Wright and Fornberg 2006; Fornberg and Lehto 2011) that better accuracy is gained by the interpolant being able to reproduce a constant. Hence, the constraint $\sum_{k=1}^{n} a_k = L1|_{\mathbf{x}=\mathbf{x}_c} = 0$ is added. Combining these constraints, the solution for the RBF-FD weights can be determined from the following linear system:

$$\begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_1 - \mathbf{x}_n\|) & 1 \\ \vdots & & \ddots & \vdots & \\ \phi(\|\mathbf{x}_n - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_n - \mathbf{x}_2\|) & \cdots & \phi(\|\mathbf{x}_n - \mathbf{x}_n\|) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ a_{n+1} \end{bmatrix} = \begin{bmatrix} L\phi(\|\mathbf{x} - \mathbf{x}_1\|)|_{\mathbf{x}=\mathbf{x}_c} \\ \vdots \\ L\phi(\|\mathbf{x} - \mathbf{x}_n\|)|_{\mathbf{x}=\mathbf{x}_c} \\ 0 \end{bmatrix}, \tag{4}$$

where a_{n+1} is not actually used in RBF-FD approximation after the system is solved. Solving this system once gives one row of the RBF-FD DM. For a total of N nodes on the sphere, there will be N linear systems of size $(n+1)\times(n+1)$ to solve, resulting in a preprocessing cost of $\mathrm{O}(n^3 N)$. Each subsequent matrix-vector multiplication will then cost only $O(nN)$ operations. In high-resolution computations, $N \gg n$, and thus, the cost to time step the RBF-FD method is $O(nN)$. This results in a significant speedup from global RBFs that require $\mathrm{O}(N^3)$ operations to create the DMs and $\mathrm{O}(N^2)$ to time step. In general, a kd-tree algorithm is used to find the $n-1$ nearest neighbors when calculating the differentiation weights for the stencils. MATLAB conveniently has a kd-tree algorithm called "knnsearch" in its statistical toolbox. RBF-FD matrices are very sparse, generally about 99 % empty. Thus, in order to reduce the bandwidth of the matrix as well as index the entries effectively in memory, a Cuthill–McKee algorithm is applied to the RBF-FD DM via the MATLAB statement "symrcm."

2.1 Limit of Flat Basis Functions

With ε available as a free parameter, an obvious numerical test is to see how the choice of it influences the accuracy that is obtained. A typical experiment is shown in Fig. 1. The error is observed to be rapidly decreasing with ε when the calculation suddenly breaks down due to the ill conditioning of *RBF-Direct* – evaluation of (2) followed by (1). This may suggest that a tradeoff will be required between accuracy and numerical conditioning (described as an *uncertainty principle* in Schaback 1995). It was however realized shortly afterward that the RBF interpolation problem itself actually does not become ill conditioned in the flat basis function limit (Driscoll and Fornberg 2002; Fornberg et al. 2004). Instead, it is only the RBF-Direct algorithm that then becomes an ill-conditioned numerical procedure for a problem that remains well conditioned. Several well-conditioned numerical algorithms were subsequently developed. Using these, it was soon found that RBF errors often become extremely low before leveling off or bouncing back up, cf. Fig. 2. Sometimes, the most accurate ε-range can be reached already with RBF-Direct, while at other times it requires a stable algorithm (or more costly extended precision arithmetic).

If the nodes are latticelike, it can happen that the RBF interpolant diverges as $\varepsilon \to 0$ (although never in the GA case Schaback 2005). For node sets with some irregularity, the interpolant will in the flat limit take the form of a multivariate polynomial (Driscoll and Fornberg 2002; Fornberg et al. 2004). One reason that ε small often is better than $\varepsilon \to 0$ is that, with RBF interpolants converging to polynomials, the boundary accuracy often deteriorates (the polynomial Runge phenomenon).

DOI 10.1007/978-3-642-27793-1_61-1

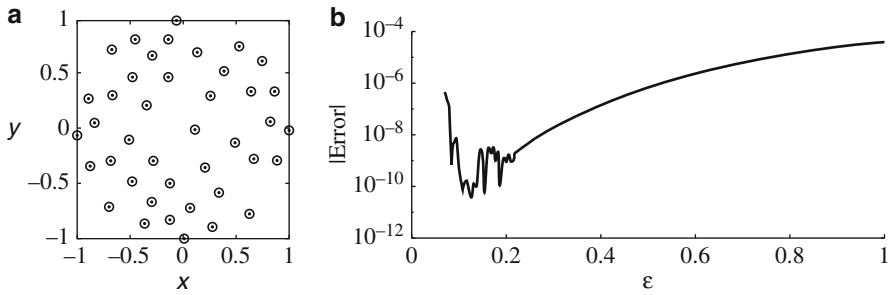

Fig. 1 (**a**) A set of 41 scattered nodes in the unit circle. (**b**) The error in max norm when the test function $f(x, y) = 59/\left(67 + \left(x + \frac{1}{7}\right)^2 + \left(y - \frac{1}{11}\right)^2\right)$ is interpolated using these nodes, displayed as a function of the shape parameter ε (Reproduced from Fornberg and Wright 2004)

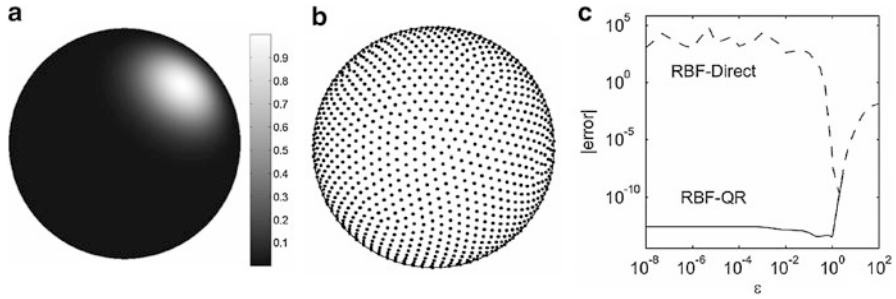

Fig. 2 (**a**) Test function $f(x) = e^{-7\left(x+\frac{1}{2}\right)^2-8\left(y+\frac{1}{2}\right)^2-9\left(z-\frac{1}{\sqrt{2}}\right)^2}$, (**b**) $n = 1{,}849$ ME (minimal energy) nodes, (**c**) Interpolation errors (in max norm) when using RBF-Direct vs. using the stable RBF-QR algorithm

2.2 The Ill Conditioning of the A-Matrix

Sideways translates of near-flat basis functions all look the same, and it is intuitively obvious that they must form a very ill-conditioned base to expand in. Just how bad it is can readily be quantified (Fornberg and Zuev 2007). For example, for scattered nodes in 2D, the eigenvalues of the A-matrix form distinct groups, following the specific pattern

$$\{O(1)\},\ \left\{O\left(\varepsilon^2\right), O\left(\varepsilon^2\right)\},\ \left\{O\left(\varepsilon^4\right), O\left(\varepsilon^4\right), O\left(\varepsilon^4\right)\},\ \left\{O\left(\varepsilon^6\right), O\left(\varepsilon^6\right), O\left(\varepsilon^6\right), O\left(\varepsilon^6\right)\},\ \ldots \tag{5}$$

until the last eigenvalue is reached (causing the last group to possibly contain fewer eigenvalues than the general pattern would suggest). Different choices of scattered node locations or of RBF types (IQ, MQ, or GA) make no difference in this regard; lattice-based nodes or Bessel-type RBFs form exceptions. More concisely, the eigenvalue pattern above can be written as

$$1, 2, 3, 4, 5, 6, \ldots, \tag{6}$$

indicating how many eigenvalues there are of orders ε^0, ε^2, ε^4, ε^6, ε^8, ε^{10}, etc. Table 2 shows some more of such sequences. Given these patterns, one can immediately calculate the orders of both $\mathrm{cond}(A)$ and $\det(A) = \prod_{k=1}^{n} \lambda_k$ as functions of n. For the examples in Figs. 1 and 2, $\mathrm{cond}(A)$ becomes equal to $O\left(\varepsilon^{-16}\right)$ and $O\left(\varepsilon^{-84}\right)$, respectively.

DOI 10.1007/978-3-642-27793-1_61-1

Table 2 Numbers of eigenvalues of different sizes (powers of ε) for different geometries

	Power of ε =								
Geometry	0	2	4	6	8	10	12	14	...
1D nonperiodic	1	1	1	1	1	1	1	1	...
1D on circle periphery	1	2	2	2	2	2	2	2	...
2D nonperiodic	1	2	3	4	5	6	7	8	...
2D on spherical surface	1	3	5	7	9	11	13	15	...
3D nonperiodic	1	3	6	10	15	21	28	36	...

2.3 Overview of Some Computational Options

In cases when RBF-Direct is too ill conditioned, the most straightforward approach is to resort to extended precision arithmetic. The only drawback is that the cost usually becomes excessive. Given the results quoted in the previous section, one can readily determine in advance just how many digits of precision would be needed as function of n and ε in various geometrical settings.

Some types of preconditionings and SVD enhancements have been suggested for the RBF-Direct approach. While preconditioning can speed up certain iterative procedures, cf. Chapter 34 in Fasshauer (2007), they do not address the issue that significant information becomes lost already when the coefficient matrix A is formed (with all its entries virtually the same when ε is small). Recovery of such missing information is challenging or impossible.

Stable algorithms use computational steps that remain numerically well conditioned all the way into the $\varepsilon \to 0$ limit (and therefore require only standard double precision arithmetic, no matter how small ε is). So far, two main classes of stable algorithms have been proposed. The first realizations of these were denoted *Contour-Padé* (Fornberg and Wright 2004) and *RBF-QR* (Fornberg and Piret 2007), respectively. Related to the latter is the recent *RBF-GA* algorithm (Fornberg et al. 2013).

2.3.1 The Contour-Padé Algorithm

Although ε is a real-valued quantity, nothing stops us from considering it also for complex values. Focusing on the GA case, it can be shown that the interpolant $s(\mathbf{x}, \varepsilon)$, for any fixed evaluation point $\mathbf{x}$, then becomes a meromorphic function of ε (i.e., with poles as its only singularities across the finite complex ε-plane). Furthermore, it is known that $s(\mathbf{x}, 0)$ is finite even as $\varepsilon \to 0$. The origin $\varepsilon = 0$ must therefore be a removable singularity of $s(\mathbf{x}, \varepsilon)$, i.e., there is no actual singularity in the flat basis function limit as far as the RBF interpolant is concerned. The Contour-Padé algorithm is based on Cauchy's integral theorem, allowing the evaluation of an analytic function at a point (such as $\varepsilon = 0$) using an integration path that does not need to come close to it. The path can thus follow such a large circle in the ε-plane that RBF-Direct can safely be used along it.

2.3.2 The RBF-QR Algorithm

As noted above, translates of near-flat RBFs form a basis that is ill suited for immediate numerical use. This naturally raises the question whether the underlying approximation *space* also is bad, or if all conditioning issues can be resolved by just finding an alternate good basis in exactly the same space. This latter case turns out to hold true, leading to the follow-up issue of how one can carry out the basis conversion by analytic means also in scattered node cases, i.e., so that no numerical

DOI 10.1007/978-3-642-27793-1_61-1

cancelations will arise in the process. The RBF-QR method offers a systematic approach for this, first implemented for nodes on the surface of a sphere (Fornberg and Piret 2007) and more recently for node sets in 1D, 2D, and 3D (however then limited to GA RBFs Fornberg et al. 2011).

2.3.3 The RBF-GA Algorithm

The RBF-QR algorithm involves extensive manipulations of Taylor expansions. While these can be truncated as needed, the RBF-GA algorithm utilizes that exact remainders can be expressed as incomplete gamma functions. This leads to a stable algorithm that is free from both infinite expansions and inexact truncations. It applies to GA RBFs in any number of dimensions and is presently the fastest stable option available (at around 10 times the cost of RBF-Direct). Although it may be slightly less accurate than RBF-QR in some cases (such as for large latticelike node sets), it is nevertheless well suited for generating RBF-FD approximations.

2.4 Time Stabilization: Hyperviscosity

Stability issues for RBF-FD emanate from the fact that the natural intrinsic irregularity of the RBF-FD stencils causes eigenvalues of the DM to scatter into the right half of the complex plane. This becomes a hurdle to the RBF-FD method when (1) solving naturally dissipation-free PDEs, such that even a very mild numerical scatter of the eigenvalues into the right half complex plane can cause severe instability, and (2) using large RBF-FD stencils, since as the stencil size increases so does the scatter of eigenvalues. This latter point is even an issue for systems with dissipation, in which case the scatter might be too large for the natural dissipation to control.

Stabilization of the RBF-FD method is achieved by applying a hyperviscosity filter, which is a high-order Laplacian $\Delta^k, k \in \mathbb{N} > 1$, to the right-hand side (RHS) of the system of PDEs being solved. Since the stencils for the hyperviscosity are same size as those used in discretizing the spatial operators and are simply added to the RHS, there is no additional cost per time step (Fornberg and Lehto 2011; Flyer et al. 2012). If GA RBFs $\left(\phi(r) = e^{-(\varepsilon r)^2}\right)$ are used for creating the hyperviscosity stencils, the values for $\Delta^k \phi(r)$ are available explicitly, thanks to the relation

$$\Delta^k \phi(r) = \varepsilon^{2k} p_k(r) \phi(r). \tag{7}$$

Here, k is the order of the Laplacian, and $p_k(r)$ are multiples of generalized Laguerre polynomials that are generated recursively (see Section 3.2, Fornberg and Lehto 2011). A 2D Laplacian operator is assumed when working on the surface of the sphere since a local stencil can be viewed as planar.

Applying hyperviscosity leaves the physically relevant eigenvalues largely intact but shifts all the spurious ones of the PDE system to the left half of the complex plane. This shift is controlled by k, the order of the Laplacian, and a scaling parameter γ_c, defined by

$$H = \gamma \Delta^k = \gamma_c N^{-k} \Delta^k.$$

where N is the total number of nodes in the domain. Given a choice of N, n, and ε, it was found experimentally that γ_c ranging from $O(1)$ to $O\left(10^{-2}\right)$ provides stability with good accuracy for PDEs with convective operators (Flyer et al. 2012). In general, the larger the stencil size, the higher the order of the Laplacian. This is attributed to the fact that as the stencil size n increases,

Handbook of Geomathematics
DOI 10.1007/978-3-642-27793-1_61-1

the accuracy of the RBF-FD method increases; that is, a wider range of physical modes represented accurately. As a result, the hyperviscosity operator should preserve as much of that range as possible; as k increases, more physical modes are preserved correctly.

3 Implicit (Compact) RBF-FD Approximations

Since a derivative is a "local" property of a function, there is something intuitively contradictory about enhancing the order of a FD approximation by invoking data located increasingly far away. When the task is to solve a PDE (rather than just to approximate an operator), *compact approximations* offer a different opportunity for improving the order of accuracy. For finite differences, the concept has a long history (Fox 1947; Collatz 1960) with several more recent enhancements available (such as to nonlinear PDEs in 2D and 3D, etc., Gupta 1991; Lele 1992; Li et al. 1995; Zhai et al. 2013).

Before considering compact approximations in scattered node RBF-FD cases, the basic idea is illustrated in the case of approximating $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$ on a 2D lattice, with spacing h in each direction. The most obvious FD approximation can be written as

$$\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} u/h^2 = \Delta u + O\left(h^2\right) . \tag{8}$$

Using only a 3×3 stencil size, it is impossible to find weights that improve the accuracy above second order. Extending the stencil to 5 weights in both directions permits fourth order, but causes problems when solving the PDE $\Delta u = f$:

- The center weight becomes smaller in magnitude than the sum of magnitudes of the remaining weights, i.e., *diagonal dominance* is lost. This damages the convergence rate of many iterative schemes, and it also opens up the possibility of system singularities.
- Wider stencils need more boundary information than what is readily available.

Another way to approximate Δu is given by

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} u/\left(6h^2\right) = \Delta u + O(h^2) , \tag{9}$$

with no immediately obvious advantage over (8). However, Taylor expansions will reveal that, for solving $\Delta u = f$,

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} u/\left(6h^2\right) = \begin{bmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{bmatrix} f/12 + O(h^4),$$

and for solving $\Delta u = 0$,

DOI 10.1007/978-3-642-27793-1_61-1

$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} u/\left(6h^2\right) = 0 + O(h^6)\,.$$

The latter two approximations suffer neither of the problems noted above but achieve nevertheless significantly improved levels of accuracy.

The weights a_i in "regular" RBF-FD approximations of a linear operator L at a node $\underline{x}_1$ are obtained by requiring that $Lu|_{\underline{x}=\underline{x}_1} = \sum_{i=1}^{n} a_i u_i$ is exact for all the RBFs $\phi\left(||\underline{x} - \underline{x}_i||\right)$, leading to an $n \times n$ linear system for the weights (system represented in (4) without column and row of ones). For compact RBF-FD approximations, first described in Wright and Fornberg (2006), we again consider a RBF-FD stencil centered at $\underline{x}_1$, with further nodes $\underline{x}_2, \ldots, \underline{x}_n$, and then repeat $m \leq n-1$ of the latter ones: $\underline{x}_{\sigma_1}, \ldots, \underline{x}_{\sigma_m}$. Wishing to solve $Lu = f$, we extend the basis function set to also include $L\phi\left(||\underline{x} - \underline{x}_{\sigma_j}||\right)$ and require $Lu|_{\underline{x}=\underline{x}_1} = \sum_{i=1}^{n} a_i u|_{\underline{x}=\underline{x}_i} + \sum_{j=1}^{m} b_j Lu|_{\underline{x}=\underline{x}_{\sigma_j}}$ to hold for them all. This leads to a $(n+m) \times (n+m)$ linear system for the weights a_i and b_j. With $Lu|_{\underline{x}=\underline{x}_1} = f_1, u|_{\underline{x}=\underline{x}_i} = u_i$ and $Lu|_{\underline{x}=\underline{x}_{\sigma_j}} = f_{\sigma_j}$, we have thus arrived at the desired compact RBF-FD formula (sometimes denoted RBF-HFD for being based on a Hermite-type interpolation). The usual enhancement of also enforcing exact results for $u(\underline{x}) \equiv 1$ together with the constraint $\sum_{i=1}^{n} \lambda_i = 0$ is again beneficial for accuracy and is readily incorporated. The resulting $(n+m+1) \times (n+m+1)$ linear system will again generally be positive definite.

The advantages noted above for compact formulas in the FD case carry directly over to the scattered node RBF-FD case. Several test examples and further discussions are provided in Wright and Fornberg (2006).

4 Applications on the Surface of Spheres and Spheroids

4.1 Cartesian Form of Surface Differential Operators

Spherical coordinates are well known to suffer from the "pole problem." When expressing PDEs in this coordinate system, this problem is exacerbated by the fact that directional velocity *vector* components in the lateral direction u (latitudinal) and v (longitudinal) will inherently carry pole singularities in their solution since the unit vectors $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\lambda}}$ are singular at the poles. Similar issues occur when representing more general surfaces in surface-based (or intrinsic) coordinate systems, such as oblate or prolate spheroidal coordinates. If, however, a Cartesian coordinate system is used, these singularities can be completely avoided. This section describes how to express common surface differential operators in Cartesian form. The discussion below is for general two-dimensional surfaces embedded in three-dimensional space, with some specific comments on the sphere and spheroids toward the end. These Cartesian-based operators are completely natural to apply an RBF expansion since the radial functions are free of any surface-based coordinate systems.

Let $\mathbf{x} = (x, y, z)$ be a point on the target surface and $\mathbf{n} = (n_x, n_y, n_z)$ denote the unit normal vector to the surface at $\mathbf{x}$. If $\mathbf{u} = (u, v, w)$ is a vector expressed in Cartesian coordinates and is placed at (x, y, z), then $\mathbf{n}\mathbf{n}^T\mathbf{u}$ gives the projection of $\mathbf{u}$ onto the normal to the surface, and $\mathbf{u} - \mathbf{n}\mathbf{n}^T\mathbf{u}$ gives the projection of $\mathbf{u}$ onto the plane tangent to the surface at (x, y, z). This projection operation can be expressed using the following projection matrix:

DOI 10.1007/978-3-642-27793-1_61-1

Table 3 Common surface differential operators expressed in Cartesian coordinates. The projection matrix $\mathbf{P}$ is defined in (10) and ∇ is the standard Cartesian-based gradient, i.e., $\nabla = \partial_x \hat{\mathbf{i}} + \partial_y \hat{\mathbf{j}} + \partial_z \hat{\mathbf{k}}$

Surface differential operator	Expression in Cartesian coordinates
Gradient of a scalar h	$\mathbf{P}\nabla h = \begin{bmatrix} (\mathbf{p}_x \cdot \nabla)h \\ (\mathbf{p}_y \cdot \nabla)h \\ (\mathbf{p}_z \cdot \nabla)h \end{bmatrix}$
Divergence of a vector $\mathbf{u} = (u, v, w)$	$(\mathbf{P}\nabla) \cdot \mathbf{u} = (\mathbf{p}_x \cdot \nabla)u + (\mathbf{p}_y \cdot \nabla)v + (\mathbf{p}_z \cdot \nabla)w$
Curl of a vector $\mathbf{u} = (u, v, w)$	$(\mathbf{P}\nabla) \times \mathbf{u} = \begin{bmatrix} (\mathbf{p}_y \cdot \nabla)w - (\mathbf{p}_z \cdot \nabla)v \\ (\mathbf{p}_z \cdot \nabla)u - (\mathbf{p}_x \cdot \nabla)w \\ (\mathbf{p}_x \cdot \nabla)v - (\mathbf{p}_y \cdot \nabla)u \end{bmatrix}$
Laplace-Beltrami of scalar h	$(\mathbf{P}\nabla) \cdot (\mathbf{P}\nabla h) = (\mathbf{p}_x \cdot \nabla)(\mathbf{p}_x \cdot \nabla)h + (\mathbf{p}_y \cdot \nabla)(\mathbf{p}_y \cdot \nabla)h + (\mathbf{p}_z \cdot \nabla)(\mathbf{p}_z \cdot \nabla)h$

$$\mathbf{P} = \mathbf{I} - \mathbf{n}\mathbf{n}^T = \begin{bmatrix} (1 - n_x n_x) & -n_x n_y & -n_x n_z \\ -n_x n_y & (1 - n_y n_y) & -n_y n_z \\ -n_x n_z & -n_y n_z & (1 - n_z n_z) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x^T \\ \mathbf{p}_y^T \\ \mathbf{p}_z^T \end{bmatrix}, \tag{10}$$

where $\mathbf{I}$ is the 3-by-3 identity matrix. Here $\mathbf{p}_x$, $\mathbf{p}_y$, and $\mathbf{p}_z$ represent the projection operators in the x, y, and z directions, respectively. This projection operator can be combined with the standard Cartesian-based gradient operator ∇ to define a variety of surface differential operators in Cartesian coordinates. Table 3 summarizes these results.

In the case of the unit sphere, the normal vector $\mathbf{n}$ at a point $\mathbf{x} = (x, y, z)$ on the sphere is just $\mathbf{x}$. In this case, the projection operator simplifies to

$$\mathbf{P} = \mathbf{I} - \mathbf{x}\mathbf{x}^T = \begin{bmatrix} (1 - x^2) & -xy & -xz \\ -xy & (1 - y^2) & -yz \\ -xz & -yz & (1 - z^2) \end{bmatrix}. \tag{11}$$

For a spheroid defined by

$$\frac{x^2 + y^2}{a^2} + \frac{z^2}{c^2} = 1, \tag{12}$$

the unit normal vectors are given by

$$\mathbf{n} = (n_x, n_y, n_z) = \left(\frac{2x}{a^2 \xi}, \frac{2y}{a^2 \xi}, \frac{2z}{c^2 \xi} \right), \tag{13}$$

where $\xi = \sqrt{(x^2 + y^2)/a^4 + z^2/c^4}$.

4.2 Nonlinear Shallow Water Equations on a Sphere

The shallow water equations describe the nonlinear flow of an incompressible fluid. They represent the 2D atmospheric flow conditions in a single hydrostatic atmospheric layer and are therefore considered an idealized test bed for the horizontal dynamics (known as the dynamical core) of all

DOI 10.1007/978-3-642-27793-1_61-1

3D climate model developments. The equations address the majority of the modeling challenges associated with the temporal and horizontal discretization techniques in spherical geometry.

Using the notation from Table 3, the shallow water equations on the surface of the unit sphere in Cartesian coordinates are given by

$$\frac{\partial u}{\partial t} = -\,\mathbf{p}_{\mathbf{x}} \cdot \underbrace{\begin{bmatrix} (\mathbf{u}\cdot\mathbf{P}\nabla)u + f(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{i}} + g(\mathbf{p}_x\cdot\nabla)h \\ (\mathbf{u}\cdot\mathbf{P}\nabla)v + f(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{j}} + g(\mathbf{p}_y\cdot\nabla)h \\ (\mathbf{u}\cdot\mathbf{P}\nabla)w + f(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{k}} + g(\mathbf{p}_z\cdot\nabla)h \end{bmatrix}}_{\text{RHS}}, \tag{14a}$$

$$\frac{\partial v}{\partial t} = -\,\mathbf{p}_y \cdot \text{RHS}, \tag{14b}$$

$$\frac{\partial w}{\partial t} = -\,\mathbf{p}_z \cdot \text{RHS}, \tag{14c}$$

$$\frac{\partial h}{\partial t} = -\,(\mathbf{P}\nabla)\cdot(h\mathbf{u})\,. \tag{14d}$$

where $\mathbf{P}$ is the projection operator, given in (11), and confines the flow to the sphere, f is the Coriolis force, $\mathbf{u} = u\hat{\mathbf{i}} + v\hat{\mathbf{j}} + w\hat{\mathbf{k}}$ is the velocity vector, $\mathbf{x}$ represents the position vector, and h is the geopotential height. Notice that the only spatial operator that needs to be discretized are the components of the surface gradient $(\mathbf{p}_x\cdot\nabla)$, $(\mathbf{p}_y\cdot\nabla)$, $(\mathbf{p}_z\cdot\nabla)$. These operators are used as L in (4) to generate RBF-FD approximations in the numerical test cases that follow.

4.2.1 Flow Over an Isolated Mountain

This test case describes flow over a single isolated mountain, which is achieved by adding a forcing term h_{mtn} to the right-hand side of the geopotential height h equation in (14d). The standard undifferentiable C^0 mountain is given by

$$h_{\text{mtn}} = h_{\text{max}}(1 - d/R) \tag{15}$$

where $h_{\text{max}} = 2{,}000\,\text{m}$, $R = \pi/9$, and $d^2 = \min\left[R^2, (\lambda-\lambda_c)^2 + (\theta-\theta_c)^2\right]$ with ($\theta_c = 30°\text{N}$, $\lambda_c = -90°\text{W}$) being the center of the mountain. To differentiate between errors due to a non-smooth forcing, which causes Gibbs phenomena in any high-order method, and those inherent in the RBF-FD method, the results for convergence and accuracy are compared against test runs that use an exceptionally steep C^∞ Gaussian profile given by

$$h(\lambda,\theta) = h_{\text{max}} e^{-\left(2.8\frac{d}{R}\right)^2}, \tag{16}$$

DOI 10.1007/978-3-642-27793-1_61-1

where $h_{\max}$, d, and R are the same as for the cone mountain. The initial conditions are given by

$$h = h_0 - \frac{1}{g}\left(a\Omega u_0 + \frac{u_0^2}{2}\right) z^2, \qquad \mathbf{u} = u_0\{-y, x, 0\}, \tag{17}$$

where $h_0 = 5{,}400\,\text{m}$ (mean reference height), $g = 9.80616\,\text{m}$, $u_0 = 20\,\text{m/s}$, $a = 6{,}371{,}220\,\text{m}$ (mean radius of the earth), and $\Omega = 7.292(10)^{-5}\,\text{s}^{-1}$ (rotation rate of the earth). The simulation is run for 15 days using the standard RK4 time-stepping scheme.

The left column of Fig. 3 shows the profile of the conical mountain, the solution in h at day 15 and the magnitude of the error between the RBF-FD solution for $N = 25{,}600$, $n = 31$, and the discontinuous Galerkin (DG) reference solution. The same holds for the right column but with the Gaussian mountain profile. With the solutions looking identically the same, the key difference to notice is that even though the C^∞ Gaussian mountain is slightly steeper than the C^0 mountain, no Gibbs phenomena is observed in the former. With the C^0 mountain, there are high-frequency waves emanating throughout the domain (i.e., Gibb's phenomena), illustrating the sensitivity of high-order methods to non-smooth forcing. Another consequence is that the accuracy of the RBF-FD method does not indefinitely increase with stencil size n, as shown in the bottom left panel of Fig. 3. After $n = 31$, stencil size has no bearing on accuracy when a non-smooth mountain forcing is present. Hence, this is the stencil size chosen for this test case. In contrast, the bottom right panel of Fig. 3 demonstrates how the accuracy of the RBF-FD method for a C^∞ solution does increase as n increases, that is, as the derivative approximations become more global. However, even with a smooth forcing, the rate of convergence is not much greater than for the cone case, since both the Gaussian and cone mountains are so steep, leading to underresolution even for very large nodes sets. To overcome this, adaptive node refinement in the area of the mountain needs to be used, as was done in St-Cyr et al. (2008).

Next the performance of the RBF-FD method is compared not only against itself but also against a discontinuous Galerkin (DG) and spherical harmonic (SH) models.

1. DG: The DG model (Blaise and St-Cyr 2012) is in 3D Cartesian coordinates, with flows being tangentially constrained to the sphere by adding a Lagrange multiplier to the system of equations. The simulations used as references herein have been performed on a cubed sphere grid made up of 6,144 elements. Each element contains 12×12 Legendre quadrature nodes to represent the solution, which results in a total of 884,736 degrees of freedom and an average resolution around 26 km. For computing these reference solutions, no dissipation mechanism was found to be needed. However, for the run time versus error computations in Fig. 5, the two-dimensional exponential filter described in Hesthaven et al. (2007) was applied.
2. SH: The SH model from the DWD (Deutscher Wetterdienst, German National Weather Service, see http://icon.enes.org/) is an updated derivative of the NCAR spectral transform model. It is implemented with de-aliasing, using Orszag's 2/3 rule (Gottlieb and Orszag 1977) and has become the standard reference solution in the community. For the flow over a mountain test, it has a spectral truncation of T426, that is, it uses 182,329 spherical harmonic bases.
3. RBF-FD: A high-resolution RBF-FD model is also used as a reference, based on $N = 163{,}824$ icosahedral-type nodes on the sphere, representing a 60 km resolution. It uses a stencil size of $n = 31$.

DOI 10.1007/978-3-642-27793-1_61-1

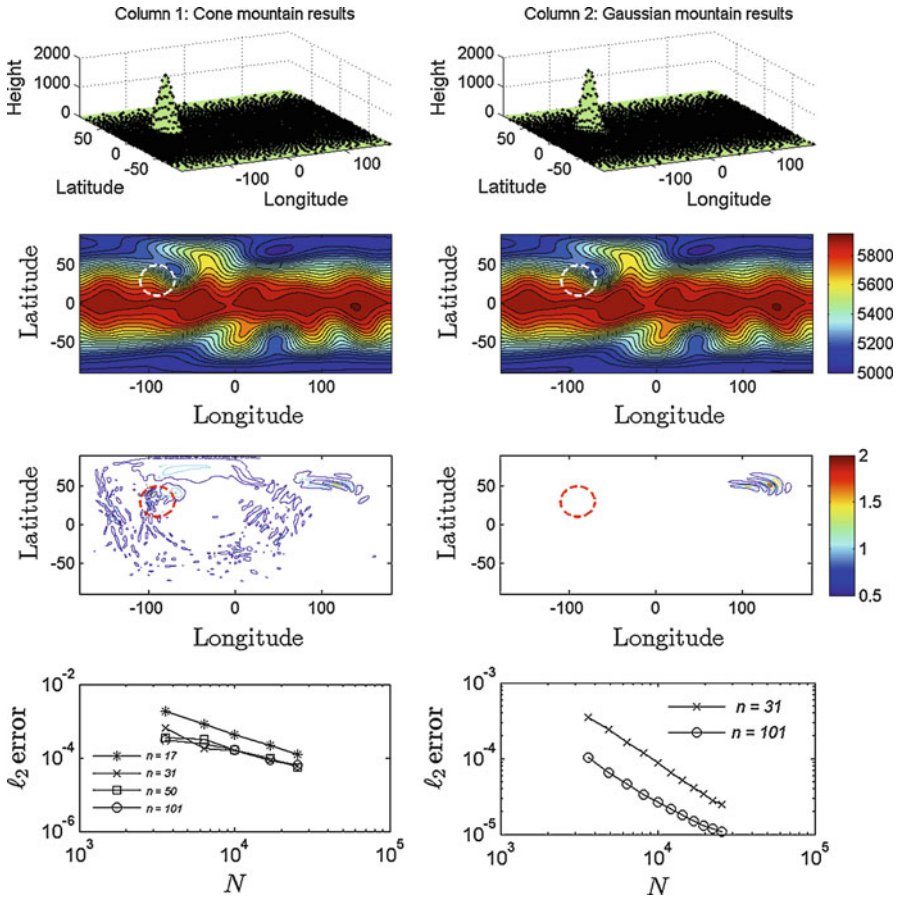

Fig. 3 *Left column* – cone mountain results: (1) profile of mountain; (2) RBF-FD solution for h at day 15, $N = 25{,}600$, and $n = 31$ with contour intervals at 50 m; (3) magnitude in the error between the RBF-FD solution and DG reference solution; and (4) ℓ_2 error as function of the resolution N for varying stencil sizes. *Right column*, same as *left* but for the Gaussian mountain forcing. *Dashed circle* in all plots is the base of the mountain

The left panel of Fig. 4 shows that the normalized ℓ_2 error is an order of magnitude larger when the DWD-SH reference solution is used, as opposed to DG or the RBF-FD ($N = 163{,}842$) reference solutions. Furthermore, when these latter reference solutions are used, the normalized ℓ_2 errors are almost identical (notice the ◯ overlay the □). This same trend is also seen in the right panel of Fig. 4 with global RBFs, a different approach than both RBF-FD and DG that does not require hyperviscosity. Given that DG, RBF-FD, and global RBFs are vastly different numerical methods, this strongly indicates that the DWD-SH T426 spectral simulation is providing a less accurate solution. This is further supported by the few articles that do report ℓ_2 errors for this test case (Taylor et al. 1997; Spotz et al. 1998; St-Cyr et al. 2008), all of which use either the NCAR or DWD-SH reference solution and obtain errors on the order of 10^{-4}, an order of magnitude larger than that obtained by DG or RBF-FD.

Next, time benchmarking is considered with comparison against the DG model. Benchmarking was done on a MacBook Pro laptop with an Intel i7 2.2 GHz quad-core processor, using only a single core and 8 GB of memory. The RBF-FD code was written in MATLAB and the DG code in C++. The RBF-FD reference solution of $N = 163{,}842$ and $n = 31$ (i.e., 60 km resolution) was used for calculating the ℓ_2 error versus runtime (i.e., wall-clock time) for both methods in Fig. 5. The RBF-FD resolutions with corresponding time steps is given in Table 4. The RBF-FD method

DOI 10.1007/978-3-642-27793-1_61-1

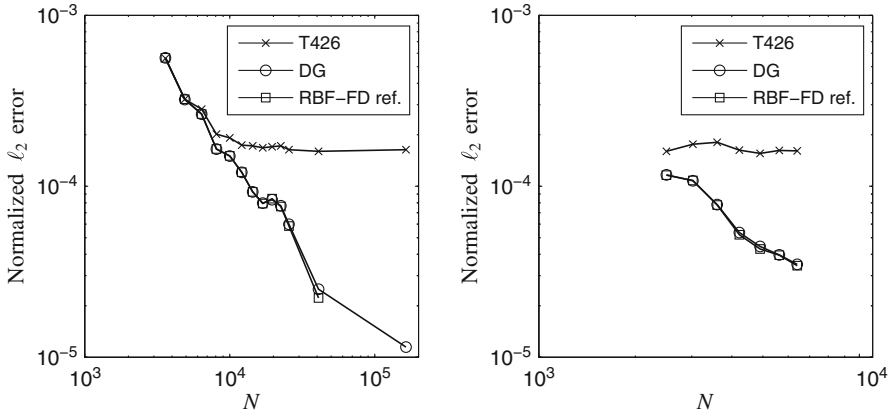

Fig. 4 The normalized ℓ_2 error in the height field h as a function of N for flow over a conical mountain at day 15 for RBF-FD (*left panel*) and global RBFs (*right panel*). The different markers correspond to different reference solutions

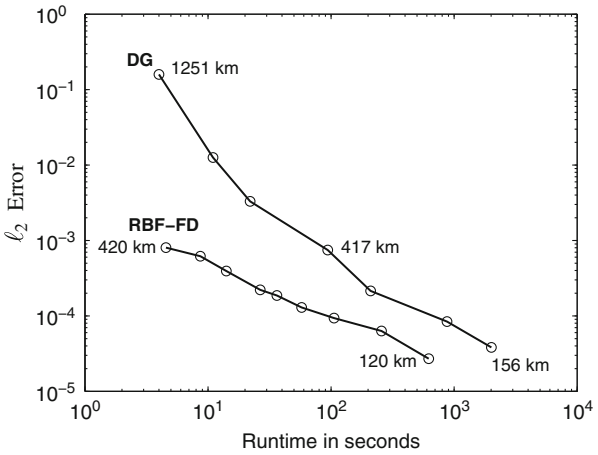

Fig. 5 The error as a function of runtime (defined by wall-clock time) for the flow over the cone mountain test case

Table 4 Time steps used for the cone mountain case with respective spatial resolutions for the maximal determinant (MD) nodes based on the mesh norm, $\max_{\mathbf{x}\in\mathbb{S}^2} \min_{1\le i\le N} \operatorname{dist}(\mathbf{x}, \mathbf{x}_i)$ (see Womersley and Sloan 2003/2007 for discussion and tabulation)

N	Resolution (km)	Node type	Δt (min)
4,096	550	MD	20
6,400	420		15
12,100	330		12
25,600	220		5
40,962	120	ICO	3
163,842	60		1

DOI 10.1007/978-3-642-27793-1_61-1

was computationally faster than the DG method, from about an order and a half of magnitude for coarser resolutions to four times faster for the finest resolutions.

4.2.2 Evolution of a Highly Nonlinear Wave

This test case describes the evolution of a highly nonlinear wave with rapid energy transfer from large to small scales over a short time period. First, high-frequency gravity waves propagate around the sphere followed by complex vortex formations with sharp gradients. The details of how to set up the test can be found in Galewsky et al. (2004) and Flyer et al. (2012). The background flow is only a function of latitude, represented by an exponential profile that is zero everywhere except in the latitudinal band $\pi/7 \leq \theta \leq 5\pi/14$. To generate the instability, the height (pressure) field is perturbed by Gaussians, in longitude and latitude, multiplied by a cosine to force the perturbation to go to zero at the poles $\theta = \pm\pi/2$. The test case is run for 6 days.

There are two main concerns with this test case: (a) how well the sharp gradients are resolved and (b) the effect of Gibbs phenomena. For *short* time integration periods, as here, these two numerical issues become a balancing act. As both the solution N and order of the method n are increased, the gradients are resolved better. However, as with classical FD, the higher the order of the method, i.e., larger n, the more prominent the Gibbs phenomena. Here with $N = 25{,}000$, $n = 101$ would correspond approximately to a ninth-order method. Notice in Fig. 6, when an $n = 101$ stencil size is used, the contour lines are more jagged with evidence of of Gibbs phenomena. Yet, the gradients and vortices are much better defined than for $n = 31$, roughly analogous to a fifth–sixth-order method. It this latter case, the features appear smoothed out and underdeveloped. Increasing the resolution N and keeping $n = 31$ fixed gives the best results as seen in Fig. 7 for $N = 163{,}842$ (60 km or $0.54^\circ \times 0.54^\circ$). The solution is extremely similar to that given by the high-order DG solution with a resolution of 39 km or $0.35^\circ \times 0.35^\circ$. Due to its high accuracy in approximating derivatives, the RBF-FD method is able to produce the basic wave pattern structure even at very coarse resolutions such as $N = 4{,}096$ ($5^\circ \times 5^\circ$), displayed in Fig. 8. This is not the case with the DG method (a slightly higher resolution of $3^\circ \times 3^\circ$ is displayed in the third panel of Fig. 8), a spectral element method (SE), or a finite volume method (FV). At such coarse resolutions, the DG and SE (St-Cyr et al. 2008) methods instead produce features of the grid such as an artificial wavenumber-four pattern for the cubed sphere. The FV method, notorious for being dissipative, shows no spatial structures at a resolution of $5^\circ \times 5^\circ$ in Fig. 8.

4.3 Reaction-Diffusion Equations on Spheroids

While the sphere plays a prominent role in several applications from geoscience, it is often a physical idealization. In some cases, spheroidal domains (oblate or prolate) may be more physically relevant. Unfortunately, many numerical techniques that work well for the sphere can be considerably more difficult in spheroidal domains (e.g., going from spherical harmonics to oblate/prolate spheroidal harmonics). Since RBFs are free of any surface-based coordinate system, they do not suffer from these complications. This fundamental feature is illustrated in this section by considering the problem of simulating certain reaction-diffusion equations on the sphere, an oblate spheroid, and a prolate spheroid using the RBF-FD method. These types of equations and these domains have a number of applications in biology and chemistry to model such things as diffusion of chemicals on biological cells or membranes, pattern formations in biology, and nonlinear chemical oscillators in excitable media. While the discussion below is for the surface of

DOI 10.1007/978-3-642-27793-1_61-1

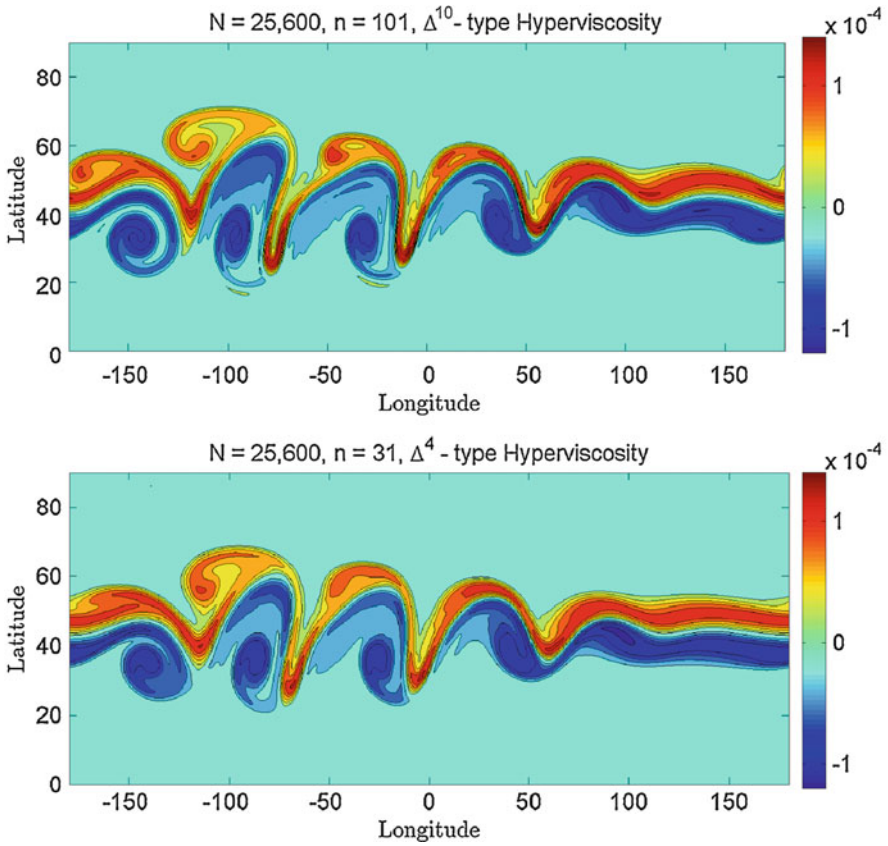

Fig. 6 The relative vorticity at day 6 for the evolution of the highly unstable wave case as function of resolution N and stencil size n. Contour interval is $2 \cdot 10^{-5}\,\mathrm{s}^{-1}$

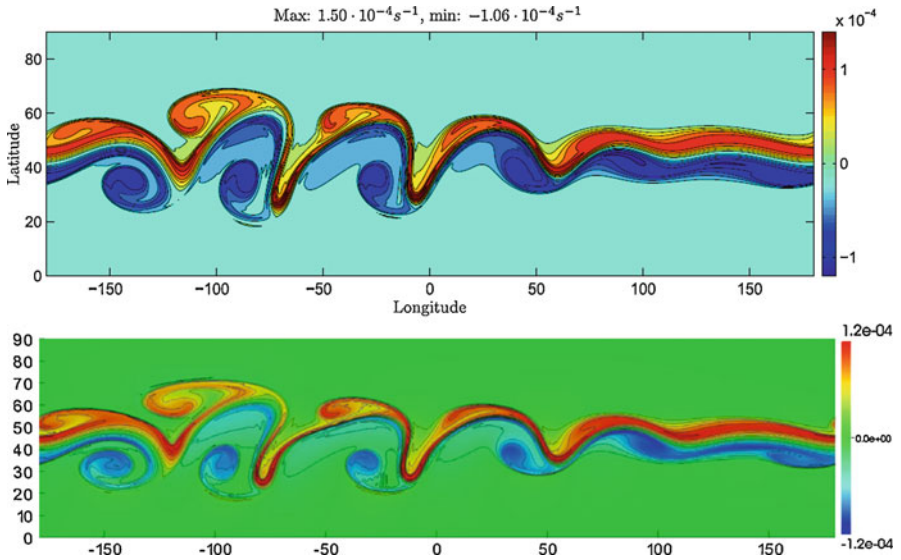

Fig. 7 *Top*: RBF-FD solution for $N = 163{,}842$ and $n = 31$. *Bottom*: DG solution at 39 km resolution

Handbook of Geomathematics
DOI 10.1007/978-3-642-27793-1_61-1

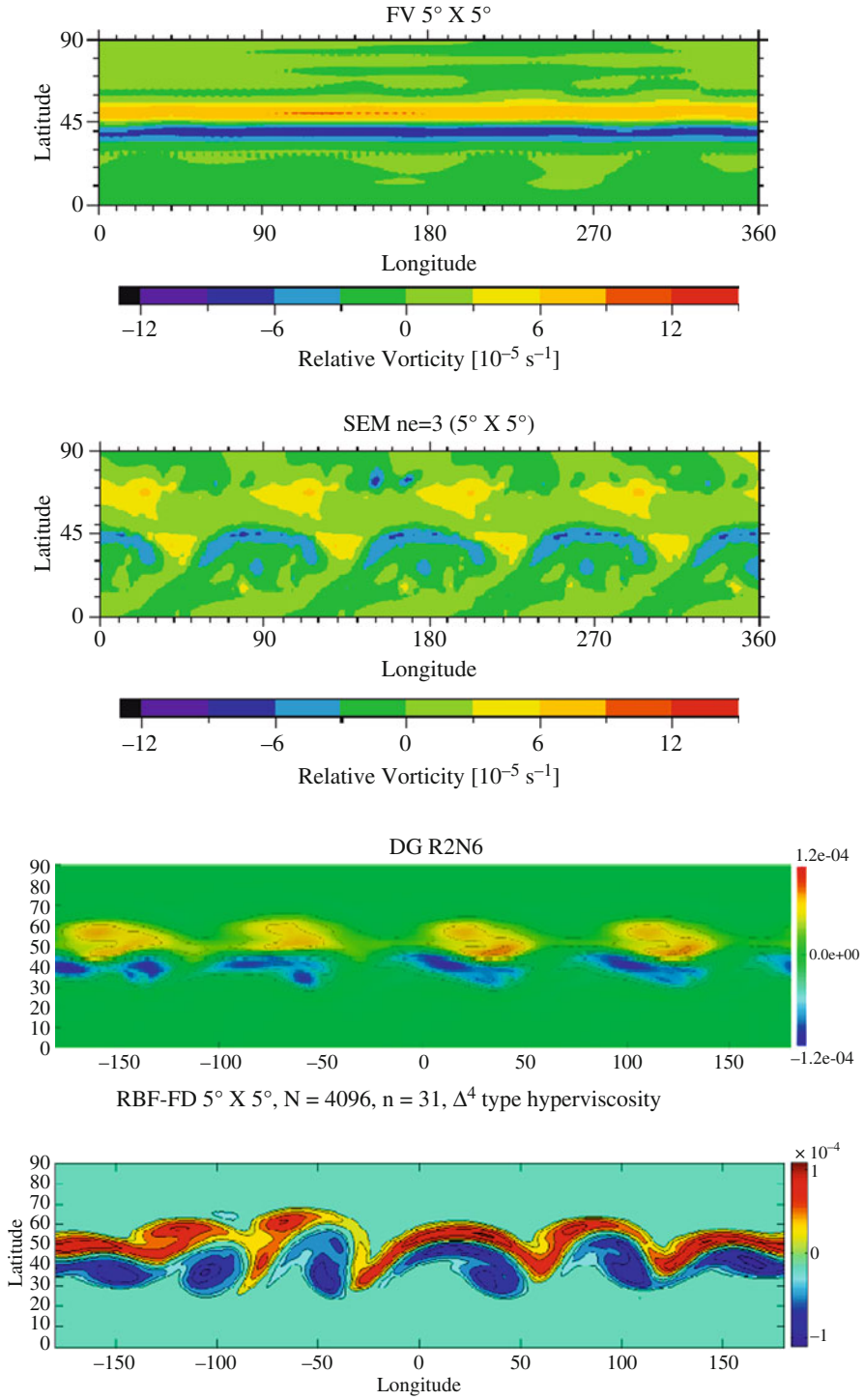

Fig. 8 The relative vorticity at day 6 for the evolution of the highly unstable wave case with resolution of $5° \times 5°$ for a spectral element model (SEM), finite volume model (FV) (St-Cyr et al. 2008), and the RBF-FD model. The DG model is approximately a resolution of $3° \times 3°$. R2 corresponds to 16 elements per face of the cubed sphere for a total of 96 elements, while N6 corresponds using $6 \times 6 = 36$ Legendre quadrature nodes. In the SEM model, $ne = 3$ corresponds to $3 \times 3 = 9$ elements per face of the cubed sphere. In each element 8×8 Gauss–Legendre–Lobatto nodes are used

DOI 10.1007/978-3-642-27793-1_61-1

a spheroid defined by (12), it may be possible to generalize it to more general (closed) surfaces, as has been done for the global RBF method in Piret (2012) and Fuselier and Wright (2013).

In the case of two species u and v, the prototypical form of reaction–diffusion equations on a spheroid (or more general surface) can be written as

$$\begin{aligned} \frac{\partial u}{\partial t} &= \delta_u \Delta_{\mathcal{S}} u + f_u(t, u, v), \\ \frac{\partial v}{\partial t} &= \delta_v \Delta_{\mathcal{S}} v + f_v(t, u, v), \end{aligned} \tag{18}$$

where $\delta_u, \delta_v \geq 0$, f_u, f_v are typically nonlinear scalar functions describing reactions of the species and $\Delta_{\mathcal{S}}$ is the Laplace–Beltrami operator for the spheroid (or surface). As mentioned above, this operator can be difficult to treat numerically if using surface-based coordinate systems (i.e., oblate/prolate spheroidal coordinates) as they will induce an artificial coordinate singularity (e.g., at the poles). As in the case of the shallow water wave equations, Cartesian coordinates will thus be used to avoid these singularities. For (18), this means expressing $\Delta_{\mathcal{S}}$ using the formulation given in the last row of Table 3.

In the case of a sphere, the direct discretization of $\Delta_{\mathcal{S}}$ using RBFs is incredibly simple (as first discussed in Wright et al. 2010 and employed below in Sect. 5.2). Unfortunately when switching to a more general spheroid, this simplicity is lost. The discretization approach taken in Fuselier and Wright (2013) is therefore adopted here. In this approach, N "scattered" nodes are first distributed on the surface of the spheroid. Each of the components of the surface gradient listed in the top row of Table 3 is then approximated with the RBF-FD method using the projection matrix $\mathbf{P}$ defined by the unit normal vectors in (13). Finally, the DM for the surface Laplacian is constructed like the continuous formulation listed in the last row of Table 3 but using the RBF-FD DMs for each component of the surface gradient instead of the continuous ones (see Fuselier and Wright 2013 for full details).

Based on numerical experiments, one small modification to the RBF-FD method described in Sect. 2 is suggested for the above procedure on the spheroid. Instead of computing the distance from node $\mathbf{x}_j$ to $\mathbf{x}_k$ using the standard Euclidean distance, a weighted Euclidean distance defined as

$$\|\mathbf{x}_j - \mathbf{x}_k\|_{\mathcal{S}}^2 = \frac{(x_j - x_k)^2 + (y_j - y_k)^2}{a^2} + \frac{(z_j - z_k)^2}{c^2}$$

should be used, where a and c define the parameters of the spheroid in (12). Although the results are not presented here due to space limitations, this was observed to give more accurate approximations of the surface Laplacian. It should be noted that this modification does not alter the non-singularity results of the linear systems involved in computing the RBF-FD formulas discussed in Sect. 2.

4.3.1 Turing Patterns

Since Turing's classical paper Turing (1952) that suggested how certain nonlinear models of reaction and diffusion can lead to stable, heterogeneous pattern formations, there has been an

DOI 10.1007/978-3-642-27793-1_61-1

explosion of research in reaction–diffusion-type models for various kinds of morphogenesis. The following is one such model that has been studied on spherical domains (Varea et al. 1999):

$$\begin{aligned}\frac{\partial u}{\partial t} &= \delta_u \Delta_{\mathcal{S}} u + \alpha u\left(1-\tau_1 v^2\right) + v(1-\tau_2 u),\\ \frac{\partial v}{\partial t} &= \delta_v \Delta_{\mathcal{S}} v + \beta v\left(1+\frac{\alpha\tau_1}{\beta} uv\right) + u(\gamma+\tau_2 v).\end{aligned} \tag{19}$$

Here u and v are morphogens with u the "activator" and v is the "inhibitor." If $\alpha = -\gamma$ then $(u, v) = (0, 0)$ is a unique equilibrium point of this system. By changing the diffusivity rates of u and v, an instability can form that leads to different pattern formations. The cubic coupling parameter τ_1 favors the formation of stripes, while the quadratic coupling parameter τ_2 favors the formation spots. The spot pattern formations are more robust than stripes and take far less time to reach "steady state."

The RBF-FD method was applied to numerically solving (19) on three surfaces: an oblate spheroid defined by (12) with $a = 1$ and $c = 0.5$, a prolate spheroid with $a = 1$ and $c = 1.5$, and the unit sphere (i.e., $a = c = 1$). The surfaces were discretized using a radial projection of a set of N spherical MD nodes to the surface. In all experiments, $N = 16{,}384$ and $n = 31$ for the RBF-FD approximations of the surface Laplacian. The equations were advanced in time using the third-order, semi-implicit, backward differentiation formulae (SBDF3) method (Ascher et al. 1995). This scheme treats the diffusion terms implicitly and the (nonlinear) reaction terms explicitly. The implicit systems were solved using the (unpreconditioned) BiCGSTAB iterative method (van der Vorst 1992) with a tolerance of 10^{-8} on the relative residual. A time step of $\Delta t = 0.05$ was used, which was near the largest that could be used and still maintain time stability due to the non linear reaction terms. Throughout the simulations, the largest number of BiCGSTAB iterations required to solve the implicit systems was 5. Similar to the experiments in Varea et al. (1999), we set the initial values of u and v to random values between -0.5 and 0.5 in a thin strip around the "equator" of each surface and $u = v = 0$ elsewhere.

Figure 9 shows the results from the simulations using parameters that lead to both spot and stripe patterns. These parameter values were motivated by those used in Varea et al. (1999). For the spot patterns, the parameters $\alpha = 0.899$, $\beta = -0.91$, $\gamma = -0.899$, $\tau_1 = 0.02$, and $\tau_2 = 0.2$ were used for all the surfaces. The diffusion values were set as $\delta_u = 0.516\delta_v$, with $\delta_v = 2.25 \cdot 10^{-3}$ for the oblate spheroid, for $\delta_v = 3 \cdot 10^{-3}$ for the sphere, and $\delta_v = 4.5 \cdot 10^{-3}$ for the prolate spheroid. For the stripe patterns, $\tau_1 = 3.5$ and $\tau_2 = 0$, with α, β, and γ, set the same as the spots for all the surfaces. The diffusion values were set as $\delta_u = 0.516\delta_v$, with $\delta_v = 1.575 \cdot 10^{-3}$ for the oblate spheroid, for $\delta_v = 2.1 \cdot 10^{-3}$ for the sphere, and $\delta_v = 3.15 \cdot 10^{-3}$ for the prolate spheroid.

While the initial conditions used in these simulations are random, the resulting spot and stripe patterns observed in Fig. 9 are qualitatively similar to those obtained from the global RBF method from Fuselier and Wright (2013). This new RBF-FD method is significantly more computationally efficient since iterative methods can be used to successfully solve the implicit linear systems from the time-stepping scheme as opposed to direct methods used in Fuselier and Wright (2013). This also allows much higher resolutions to be used in the simulations.

DOI 10.1007/978-3-642-27793-1_61-1

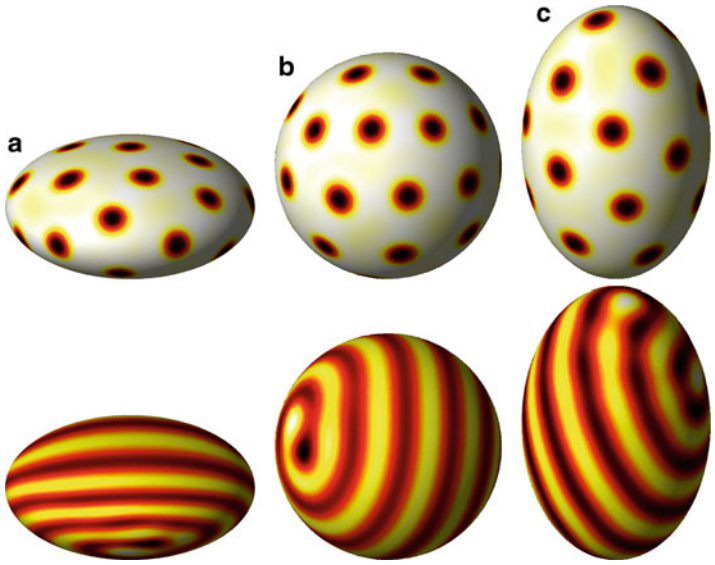

Fig. 9 Turing spot and stripe patterns computed from the model (19) on an (**a**) oblate spheroid, (**b**) sphere, and (**c**) prolate spheroid. The pseudocolor plots are for the activator u once steady state is reached. In all plots, *black* corresponds to a high concentration of u, and *white* to a low concentration. For all simulations, $N = 16{,}384$ and $n = 31$, with a time step of $\Delta t = 0.05$. All simulations were run until a steady-state solution was reached, which was about 800 time steps for the spot patterns and 6,000 for the strip patterns

5 3D Applications in a Spherical Shell

5.1 Global Electric Circuit

Electrical linkages within the atmosphere are often discussed in terms of a "global electric circuit" (GEC). The GEC extends from the Earth's surface to the base of the ionosphere defined to be 90 km altitude. The basic idea, first postulated by Wilson (1920, 1929), is that thunderclouds and other highly electrified clouds produce an upward current ($\sim$1,000–2,000 A) that maintains the ionosphere at a quasi-static potential on the order 240 ± 40 kV with respect to the ground. A downward return current density of 1–10 $p\mathrm{Am}^{-2}$ is distributed over the rest of the globe in the so-called "fair weather regions" (no cloud activity).

The quasi-static coupled atmospheric and ionospheric components of the GEC can be modeled as an elliptical Poisson-type PDE

$$\nabla \cdot (\sigma \nabla \Phi) = J_s(\vec{x}), \quad \vec{x} \in \Omega, \tag{20}$$

with *suitable boundary conditions*. Here, Φ is the electrostatic potential, Ω is the atmosphere below the ionosphere, and σ is the *atmospheric conductivity*. The right-hand side, J_s, is a representation of the *current sources* (e.g., thunderstorms) in the domain. The conductivity of the atmosphere is σ, and to a good first approximation, it can be considered simply as a function of altitude. Below, it is even further simplified by taking it to be a constant, thus resulting in Poisson's equation. This allows for a simple direct comparison between explicit RBF-FD stencil implementation to an implicit (compact) RBF-FD one in 3D spherical geometry, with the focus being on accuracy and convergence of a Krylov iterative solver (BiCGSTAB van der Vorst 1992 in this case).

DOI 10.1007/978-3-642-27793-1_61-1

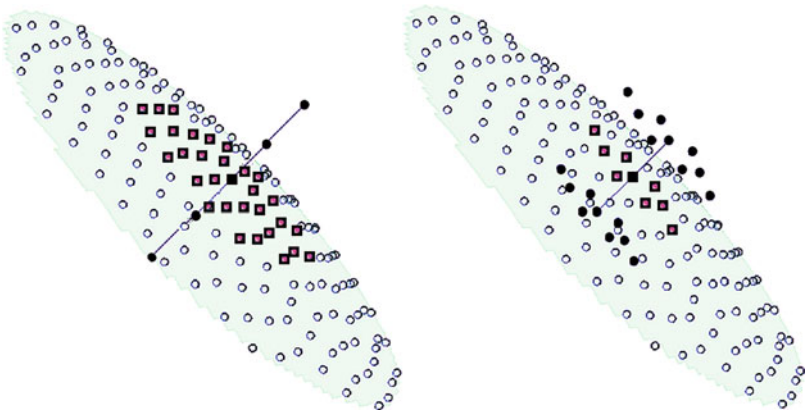

Fig. 10 Schematic picture of the two types of stencils. *Left panel*: explicit stencil. *Right panel*: implicit stencil

5.1.1 Poisson's Equation

Consider Poisson's equation with homogeneous Dirichlet conditions, given by

$$\Delta u = f, \qquad \mathbf{x} \in \Omega, \tag{21}$$

$$u = 0, \qquad \mathbf{x} \in \partial\Omega, \tag{22}$$

where Ω is a spherical shell with inner radius $R_{\mathrm{i}} = 2$ and outer radius $R_{\mathrm{o}} = 4$. The right-hand side is chosen such that the exact solution is given by

$$u_{\mathrm{exact}}(r, \theta, \varphi) = \sin\left(2\pi \frac{r - r_i}{r_0 - r_i}\right) Y_{9,3}(\theta, \varphi), \tag{23}$$

where $Y_{l,m}(\theta, \varphi)$ denotes a spherical harmonic of degree l and order m.

Two discretizations are considered, based on RBF-FD. The first is an explicit scheme, where the angular terms of the Laplacian in (21) are implemented using 31-point RBF-FD stencils on scattered nodes and the radial terms are discretized using 5-point RBF-FD stencils on equidistant nodes. The second scheme is implicit, based on the methodology in Wright and Fornberg (2006) and discussed in Sect. 3. For the angular terms in the Laplacian, $\{n = 9, m = 6\}$ node stencils are applied, while $\{n = 3, m = 2\}$ nodes are used in the radial discretization. Maximum determinant nodes (MD), see Womersley and Sloan (2003/2007), are used in the angular discretization for both stencil types. A typical stencil of each kind is shown in Fig. 10. The sparsity pattern of the Laplacian DM for each method, following a reverse Cuthill–McKee reordering of the entries, is shown in Fig. 11. The horizontal resolution is given by N_H, and the radial one by N_R. The total number of nodes is denoted $N = N_H \cdot N_R$.

The linear system arising from the discretization is solved using BiCGSTAB with zero-fill modified incomplete LU factorization as a preconditioner. A residual tolerance of 10^{-8} is used in the iterative solver. The subplots in Fig. 12 show the relative ℓ_2 error and the number of iterations required, respectively, as a function of the total number of nodes. While both methods achieve fourth-order convergence, the implicit method is more efficient in terms of memory and number of iterations required to converge.

Table 5 gives the exact specifications of the six node sets used for the results in Fig. 12.

DOI 10.1007/978-3-642-27793-1_61-1

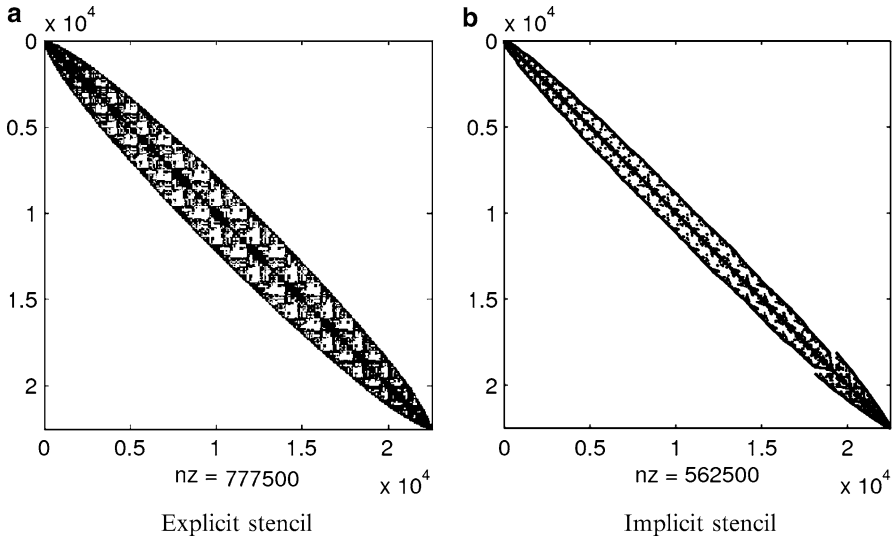

Fig. 11 The sparsity pattern for $N_H = 2{,}500$, $N_R = 11$. (**a**) Explicit stencil. (**b**) Implicit stencil

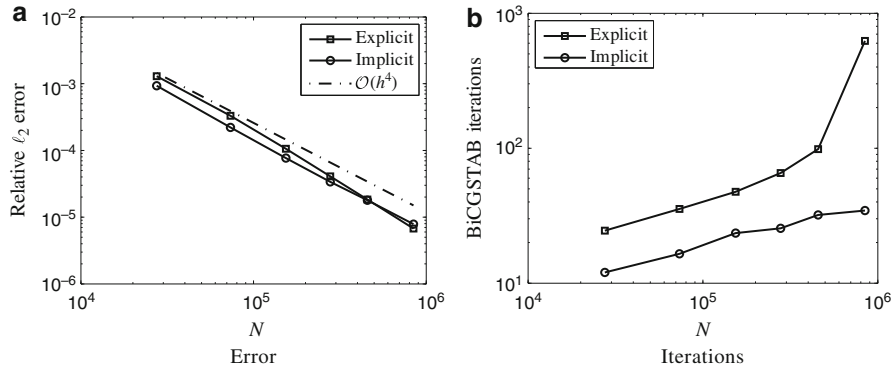

Fig. 12 The error and the number of iterations as a function of N. (**a**) Error. (**b**) Iterations

5.2 Mantle Convection

Thermal convection of an incompressible, Boussinesq fluid at infinite Prandtl number in a spherical shell that is heated from below is discussed here. This can be considered as a simplified version of mantle convection with constant viscosity. The dynamics of the fluid are governed by the Rayleigh number, 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. Since the fluid is incompressible and isoviscous, the velocity can be expressed solely in terms of a poloidal potential, $\mathbf{u} = \nabla \times \nabla \times ((\Phi r)\hat{\mathbf{r}})$ (see, e.g., Chandrasekhar 1981; Backus 1966). Exploiting this relationship, the governing equations for a shell of inner radius R_i and outer radius R_o can be written as

$$\Delta_{\mathcal{S}}\Omega + \frac{\partial}{\partial r}\left(r^2\frac{\partial \Omega}{\partial r}\right) = \mathrm{Ra}\, r\, T, \tag{24}$$

$$\Delta_{\mathcal{S}}\Phi + \frac{\partial}{\partial r}\left(r^2\frac{\partial \Phi}{\partial r}\right) = r^2\Omega, \tag{25}$$

DOI 10.1007/978-3-642-27793-1_61-1

Table 5 The specifications for the results shown in Fig. 12

Stencil type	N_R	N_H	ε_R	ε_H	ℓ_2 error	Iterations
Explicit	11	2,500	0.5	5	0.00129	24.5
	15	4,900			0.000330	35.5
	19	8,100			0.000106	47.5
	23	12,100			$4.09 \cdot 10^{-5}$	65.5
	27	16,900			$1.84 \cdot 10^{-5}$	98.5
	33	25,600			$6.73 \cdot 10^{-6}$	624.5
Implicit	11	2,500	0.5	4	0.000930	12
	15	4,900			0.000221	16.5
	19	8,100			$7.68 \cdot 10^{-5}$	23.5
	23	12,100			$3.39 \cdot 10^{-5}$	25.5
	27	16,900			$1.79 \cdot 10^{-5}$	32
	33	25,600			$7.87 \cdot 10^{-6}$	34.5

$$\frac{\partial T}{\partial t} = -\left(u_r \frac{\partial T}{\partial r} + u_\theta \frac{1}{r}\frac{\partial T}{\partial \theta} + u_\lambda \frac{1}{r\cos\theta}\frac{\partial T}{\partial \lambda}\right) + \frac{1}{r^2}\Delta_{\mathcal{S}} T + \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial T}{\partial r}\right), \tag{26}$$

where $R_\mathrm{i} \le r \le R_\mathrm{o}$, $-\pi/2 \le \theta \le \pi/2$, $-\pi < \lambda \le \pi$ are the standard spherical coordinates, $\mathbf{u} = (u_r, u_\theta, u_\lambda)$ is the velocity field in spherical coordinates, T is temperature, and $\Delta_{\mathcal{S}}$ is the surface Laplacian operator. The boundary conditions on the velocity of the fluid at the inner and outer surfaces of the spherical shell are assumed to be impermeable and shear stress-free, which translates into

$$\Phi|_{r=R_\mathrm{i},R_\mathrm{o}} = 0 \qquad \text{and} \qquad \left.\frac{\partial^2 \Phi}{\partial r^2}\right|_{r=R_\mathrm{i},R_\mathrm{o}} = 0. \tag{27}$$

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 (24)–(26) have been nondimensionalized with the length scale chosen as the thickness of the shell, $\Delta R = R_\mathrm{o} - R_\mathrm{i}$; the timescale 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. In the experiments below, the inner and outer radii of the shell are set to $R_\mathrm{o} = 20/9$ and $R_\mathrm{i} = 11/9$, which give approximately the same ratio of the inner and outer radii of the Earth's mantle (i.e., 0.55).

5.2.1 Overview of the Numerical Approach

To numerically solve (24)–(26), an operator-splitting method similar to Wright et al. (2010) is used in space where the lateral directions (θ, λ) are discretized separately from the radial direction (see Fig. 13 for an illustration). M Chebyshev nodes are used in the radial direction, and N "scattered" nodes are used on each of the resulting M spherical surfaces, giving a total of MN nodes in the 3D spherical shell. The maximal determinant (MD) nodes (Womersley and Sloan 2003/2007) are again used for the spherical surfaces. All radial derivatives are discretized using collocation with

DOI 10.1007/978-3-642-27793-1_61-1

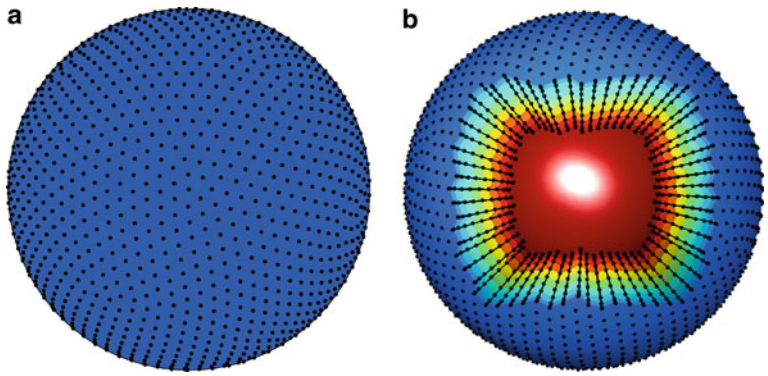

Fig. 13 (**a**) Node layout for the RBF-FD approximations on the surface of a sphere; (**b**) 3D view of the discretization of the spherical shell used in the hybrid RBF-FD/Chebyshev calculation. *Blue* is the outer boundary and *red* is the inner boundary, and *black circles* display the computational nodes, which are distributed in the radial direction along the extrema of the Chebyshev polynomials. The spherical shell has been opened up in (**b**) to show the detail of the radial discretization

Chebyshev polynomials (see, for example, Fornberg 1996; Trefethen 2000 for details), while all differential operators in the latitudinal direction θ and longitudinal direction λ are approximated on each spherical surface using n-node RBF-FD stencils.

The complete numerical algorithm is nearly identical to the one in Wright et al. (2010), with the only difference being that RBF-FD is used instead of global RBFs. The reader is therefore referred to Appendix B of Wright et al. (2010) for a detailed description. A general overview is given by the following steps:

1. Discretize $\dfrac{\partial}{\partial\theta}$, $\dfrac{1}{\cos\theta}\dfrac{\partial}{\partial\lambda}$, and $\Delta_{\mathcal{S}}$ at each of the N MD nodes using n point RBF-FD formula. As discussed in Wright et al. (2010), the coordinate singularities associated with the latter of these two operators is harmlessly removed when applied to an RBF expansion.
2. Discretize $\dfrac{\partial}{\partial r}$, $\dfrac{\partial}{\partial r}\left(r^2\dfrac{\partial}{\partial r}\right)$, $\left.\dfrac{\partial^2}{\partial r^2}\right|_{r=R_{\mathrm{i}}}$, and $\left.\dfrac{\partial^2}{\partial r^2}\right|_{r=R_{\mathrm{o}}}$ using collocation with M Chebyshev polynomials.
3. With the given temperature initial condition, solve the Poisson equations (24) and (25) for Ω and Φ. This procedure is complicated by the fact that all four boundary conditions on these equations are only specified on Φ. To handle this, the influence matrix (Peyret 2002) method is employed to find the unknown boundary values on Ω such that all four boundary conditions on Φ are satisfied (see Section 5 of Wright et al. 2010).
4. Compute the velocity field: $\mathbf{u} = \nabla\times\nabla\times[(\Phi r)\hat{\mathbf{r}}] = \left(\dfrac{1}{r}\Delta_{\mathcal{S}}\Phi, \dfrac{1}{r}\dfrac{\partial^2}{\partial r\partial\theta}(\Phi r), \dfrac{1}{r\cos\theta}\dfrac{\partial^2}{\partial r\partial\lambda}(\Phi r)\right)$.
5. Update the energy equation (26) to the next time step $t + \Delta t$ using the semi-implicit third-order Adams–Bashforth (AB3) and second-order trapezoidal rule (or Crank–Nicolson) method. In this case, all terms are treated explicitly with AB3, except the radial component of the diffusion term which is treated implicitly using trapezoidal rule. Return to step 3 with an updated temperature profile.

DOI 10.1007/978-3-642-27793-1_61-1

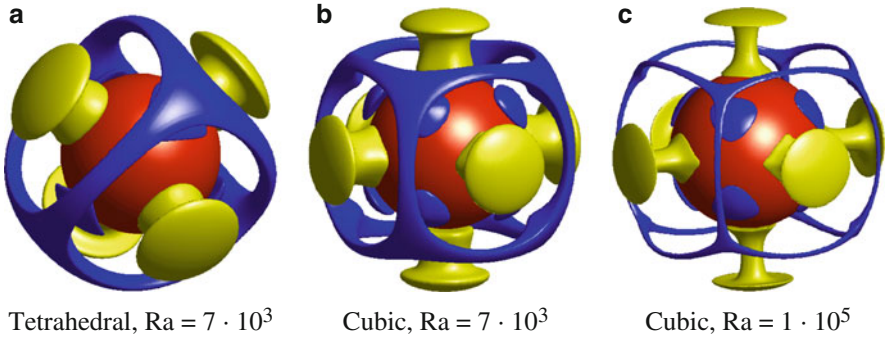

Fig. 14 Steady-state isosurfaces of the residual temperature, δT, for the three isoviscous benchmark test cases using the RBF-FD model. Here the residual temperature $\delta T = T(r, \theta, \lambda) - \langle T(r) \rangle$ is visualized, where $\langle\ \rangle$ denotes averaging over a spherical surface. *Yellow* corresponds to $\delta T = 0.15$ and denotes upwelling relative to the average temperature at each radial level, while *blue* corresponds to $\delta T = -0.15$ and denotes downwelling. The *red solid sphere* shows the inner boundary of the 3D shell corresponding to the core. (**a**)Tetrahedral, Ra $= 7 \cdot 10^3$. (**b**) Cubic, Ra $= 7 \cdot 10^3$. (**c**) Cubic, Ra $= 1 \cdot 10^5$

5.2.2 Validation on Community Benchmark Problems

The two most common benchmarks for computational models of mantle convection in a spherical shell are the isoviscous steady-state tetrahedral and cubic test cases. These names correspond to the steady-state upwelling plumes that result from these tests at the faces of regular tetrahedron and cube, respectively. The initial condition for the temperature is specified as

$$T(r, \theta, \lambda) = \frac{R_i(r - R_o)}{r(R_i - R_o)} + 0.01 Y_3^2(\theta, \lambda) \sin\left(\pi \frac{r - R_i}{R_o - R_i}\right) \tag{28}$$

for the tetrahedral test case and

$$T(r, \theta, \lambda) = \frac{R_i(r - R_o)}{r(R_i - R_o)} + 0.01 \left[Y_4^0(\theta, \lambda) + \frac{5}{7} Y_4^4(\theta, \lambda)\right] \sin\left(\pi \frac{r - R_i}{R_o - R_i}\right) \tag{29}$$

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.

Both the tetrahedral and cubic tests were first run at Ra $=$ 7,000, with $N = 2{,}601$ nodes on each spherical shell and $M = 23$ Chebyshev nodes in the radial direction giving a total of 59,823 nodes. Each differential operator on the spherical surfaces was approximated using $n = 50$ node RBF-FD formulas. A time step of 10^{-4} was used to reach steady state at the nondimensionalized time of $t = 1$, corresponding to roughly 58 times the age of the Earth. The final RBF-FD steady-state solutions for the Tetrahedral and Cubic test cases are displayed in Fig. 14a, b, respectively. As 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. Such a comparison for the RBF-FD method with respect to established methods (including the global RBF-PS method Wright et al. 2010) is given in Table 6. For the RBF-FD and RBF-PS

DOI 10.1007/978-3-642-27793-1_61-1

Table 6 Comparison between computational methods for the isoviscous tetrahedral and cubic mantle convection test cases with Ra = 7,000. $\mathrm{Nu_o}$ and $\mathrm{Nu_i}$ denote the respective Nusselt number at the outer and inner spherical surfaces, $\langle V_{\mathrm{rms}} \rangle$ the volume-averaged RMS velocity over the 3D shell and $\langle T \rangle$ the mean temperature of the 3D shell

Model	Type	Nodes	$r \times (\theta \times \lambda)$	$\mathrm{Nu_o}$	$\mathrm{Nu_i}$	$\langle V_{\mathrm{rms}} \rangle$	$\langle T \rangle$
Tetrahedral test case, Ra = 7,000							
Zhong et al. (2008)	FE	393,216	$32 \times (12 \times 32 \times 32)$	3.5126	3.4919	32.66	0.2171
Yoshida and Kageyama (2004)	FD	2,122,416	$102 \times (102 \times 204)$	3.4430	—	32.0481	—
Kameyama et al. (2008)	FD	12,582,912	$128 \times (2 \times 128 \times 384)$	3.4945	—	32.6308	0.21597
Ratcliff et al. (1996)	FV	200,000	$40 \times (50 \times 100)$	3.4423	—	32.19	—
Stemmer et al. (2006)	FV	663,552	$48 \times (6 \times 48 \times 48)$	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	552,960	$120 \times (48 \times 96)$	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 (Wright et al. 2010)	SP	36,800	$23 \times (1{,}600)$	3.4962	3.4962	32.6424	0.21556
RBF-FD, $n = 50$	SP-FD	59,823	$23 \times (2{,}601)$	3.4962	3.4962	32.6425	0.21556
Cubic test case, Ra = 7,000							
Zhong et al. (2008)	FE	393,216	$32 \times (12 \times 32 \times 32)$	3.6254	3.6016	31.09	0.2176
Yoshida and Kageyama (2004)	FD	2,122,416	$102 \times (102 \times 204)$	3.5554	—	30.5197	—
Kameyama et al. (2008)	FD	12,582,912	$128 \times (2 \times 128 \times 384)$	3.6083	—	31.0741	0.21639
Ratcliff et al. (1996)	FV	200,000	$40 \times (50 \times 100)$	3.5806	—	30.87	—
Stemmer et al. (2006)	FV	663,552	$48 \times (6 \times 48 \times 48)$	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	552,960	$120 \times (48 \times 96)$	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 (Wright et al. 2010)	SP	36,800	$23 \times (1{,}600)$	3.6096	3.6096	31.0820	0.21577
RBF-FD, $n = 50$	SP-FD	59,823	$23 \times (2{,}601)$	3.6095	3.6096	31.0819	0.21577

Extrap. indicates that the results were obtained using Romberg extrapolation
Solid lines indicate numbers were not reported
Abbreviations: *FE* finite element, *FD* finite difference, *FV* finite volume, *SP-FD* hybrid spectral and finite difference, *SP* purely spectral

methods, the standard deviation of all the quantities from the last 1,000 time steps was less than $5 \cdot 10^{-5}$, which is a standard measure for indicating the model has reached numerical steady state.

While the Ra = 7,000 tests above are the de facto benchmarks, there are also some results in the literature also for cubic test at $\mathrm{Ra} = 10^5$. This Rayleigh number results in a more convective regime, with thinner plumes, so that higher resolutions are needed to properly capture the solution. For this test, $N = 6{,}561$ nodes were used on each spherical shell and $M = 43$ Chebyshev nodes in the radial direction. The spherical surface differential operators were again approximated using

DOI 10.1007/978-3-642-27793-1_61-1

Table 7 Comparison between computational methods for the isoviscous Cubic mantle convection test case with $\mathrm{Ra} = 10^5$

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	$48 \times (12 \times 48 \times 48)$	7.8495	7.7701	154.8	0.1728
RBF-PS (Wright et al. 2010)	SP	176,128	$43 \times (4{,}096)$	7.8120	7.8005	154.49	0.17123
RBF-FD, $n = 50$	SP-FD	282,123	$43 \times (6{,}561)$	7.8072	7.8030	154.43	0.17069

$n = 50$. Increasing M necessitates a decrease in the time step to maintain time stability since the Chebyshev nodes cluster quadratically at the boundaries. A time step of $\Delta = 6 \cdot 10^{-6}$ was used to reach a final time of $t = 0.35$, at which numerical steady state was reached (using the same criteria as the $\mathrm{Ra} = 7{,}000$ cases). The final RBF-FD steady-state solution is displayed in Fig. 14c, and a comparison with two other methods is given in Table 7.

The following observations can be made regarding the results for these benchmarks:

1. For the $\mathrm{Ra} = 7{,}000$ test case, the results from the RBF-PS method (Wright et al. 2010) and Harder's extrapolated results from a spherical harmonic-finite difference method (Harder 1998) are expected to be correct to four digits. The hybrid RBF-FD method was able to match these values to four digits for both tests. Furthermore, only in three values does the RBF-FD method not match the global RBF-PS method to all digits.
2. The number of nodes (degrees of freedom) needed to accomplish the $\mathrm{Ra} = 7{,}000$ and $\mathrm{Ra} = 10^5$ results for the RBF-FD method are significantly lower (over an order of magnitude in all but one case) than any of the other methods that use a hybrid spectral-FD discretization or a full finite difference, finite volume, or finite element method.
3. The Nusselt number is nondimensional, measuring the ratio of convective to conductive heat transfer across a boundary. Thus, if there are no sources or sinks in the domain, energy should be conserved and $\mathrm{Nu_i} = \mathrm{Nu_o}$. This is exhibited quite clearly in the $\mathrm{Ra} = 7{,}000$ cases for the RBF-FD method (like the RBF-PS), and to a slightly lesser extent in the $\mathrm{Ra} = 10^5$ (with a difference of only 0.05 %). This suggests the method will inherently dissipate physical quantities less.
4. While the RBF-FD method requires a higher resolution (larger N) than the global RBF-PS method to achieve similar results, it has a significant advantage in terms of computational cost and memory storage, as spatial discretizations require a small fraction of the number of terms required by the global method.

6 Future Directions and Concluding Remarks

This chapter provides a survey of a group of topics relevant to using RBF-FD, a meshless method, for a variety of problems that arise in the geosciences, with particular emphasis on spherical geometries, both on surfaces and within a volume. RBF-FD is a method that has only begun to grow in the last few years. From a practical standpoint, RBF-FD sprouted out of global RBFs, which showed exceptional numerical qualities in terms of accuracy and time stability for solving PDEs but reduction in computational cost needed to be addressed if RBFs were to be effective when scaled to very large problem sizes. With the RBF-FD methodology, matrices go from full to becoming 99 % empty. Of course, the sacrifice is the exchange of spectral accuracy for high-order

DOI 10.1007/978-3-642-27793-1_61-1

algebraic convergence, assuming smooth data. However, since natural processes are almost never infinitely differentiable, little is lost and much gained in terms of memory and runtime.

Although the future of RBF-FD is bright on the horizon, many topics still need to be addressed. These include but are definitely not limited to improved reliability, effectiveness, and scalability of RBF-FD implementations on novel computer architectures (Bollig et al. 2012), dynamic adaptive node refinement (Driscoll and Heryudono 2007), application of preconditioners and iterative solvers for elliptic PDEs, stability analysis in the presence of boundary conditions for hyperbolic problems, and the treatment of discontinuities in the domain.

References

Ascher UM, Ruuth SJ, Wetton BTR (1995) Implicit-explicit methods for time-dependent partial differential equations. SIAM J Numer Anal 32:797–823

Backus GE (1966) Potentials for tangent tensor fields on spheroids. Arch Ration Mech Anal 22:210–252

Blaise S, St-Cyr A (2012) A dynamic *hp*-adaptive discontinuous Galerkin method for shallow water flows on the sphere with application to a global tsunami simulation. Mon Wea Rev 140(3):978–996

Bochner S (1933) Monotine Functionen, Stieltjes Integrale und Harmonische Analyse. Math Ann 108:378–410

Bollig E, Flyer N, Erlebacher G (2012) Solution to PDEs using radial basis function finite-differences (RBF-FD) on multiple GPUs. J Comput Phys 231:7133–7151

Chandrasekhar S (1981) Hydrodynamic and hydromagnetic stability. Dover, New York

Collatz L (1960) The numerical treatment of differential equations. Springer, Berlin

Driscoll TA, Fornberg B (2002) Interpolation in the limit of increasingly flat radial basis functions. Comput Math Appl 43:413–422

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

Fasshauer GE (2007) Meshfree approximation methods with MATLAB. Interdisciplinary mathematical sciences, vol 6. World Scientific, Singapore

Flyer N, Lehto E, Blaise S, Wright GB, St-Cyr A (2012) A guide to RBF-generated finite differences for nonlinear transport: shallow water simulations on a sphere. J Comput Phys 231:4078–4095

Fornberg B (1996) A practical guide to pseudospectral methods. Cambridge University Press, Cambridge

Fornberg B, Lehto E (2011) Stabilization of RBF-generated finite difference methods for convective PDEs. J Comput Phys 230:2270–2285

Fornberg B, Piret C (2007) A stable algorithm for flat radial basis functions on a sphere. SIAM J Sci Comput 30:60–80

Fornberg B, Wright G (2004) Stable computation of multiquadric interpolants for all values of the shape parameter. Comput Math Appl 48:853–867

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

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

DOI 10.1007/978-3-642-27793-1_61-1

Fornberg B, Wright G, Larsson E (2004) Some observations regarding interpolants in the limit of flat radial basis functions. Comput Math Appl 47:37–55
Fornberg B, Larsson E, Flyer N (2011) Stable computations with Gaussian radial basis functions. SIAM J Sci Comput 33(2):869–892
Fornberg B, Lehto E, Powell C (2013) Stable calculation of Gaussian-based RBF-FD stencils. Comput Math Appl 65:627–637
Fox L (1947) Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations. Proc R Soc A 190:31–59
Fuselier EJ, Wright GB (2013) A high-order kernel method for diffusion and reaction-diffusion equations on surfaces. J Sci Comput 1–31. doi:10.1007/s10915-013-9688-x, http://dx.doi.org/10.1007/s10915-013-9688-x
Galewsky J, Scott RK, Polvani LM (2004) An initial-value problem for testing numerical models of the global shallow-water equations. Tellus 56A:429–440
Gottlieb D, Orszag SA (1977) Numerical analysis of spectral methods. SIAM, Philadelphia
Gupta MM (1991) High accuracy solutions of incompressible Navier-Stokes equations. J Comput Phys 93:343–359
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
Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. J Geophys Res 76:1905–1915
Hesthaven JS, Gottlieb S, Gottlieb D (2007) Spectral methods for time-dependent problems. Cambridge University Press, Cambridge
Kameyama MC, Kageyama A, Sato T (2008) Multigrid-based simulation code for mantle convection in spherical shell using Yin-Yang grid. Phys Earth Planet Inter 171:19–32
Lele SK (1992) Compact finite difference schemes with spectral-like resolution. J Comput Phys 103:16–42
Li M, Tang T, Fornberg B (1995) A compact fourth-order finite difference scheme for the steady incompressible Navier-Stokes equations. Int J Numer Methods Fluids 20:1137–1151
Madych WR, Nelson SA (1992) Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. J Approx Theory 70:94–114
Mairhuber JC (1956) On Haar's theorem concerning Chebyshev approximation problems having unique solutions. Proc Am Math Soc 7(4):609–615
Micchelli CA (1986) Interpolation of scattered data: distance matrices and conditionally positive definite functions. Constr Approx 2:11–22
Peyret R (2002) Spectral methods for incompressible viscous flow. Springer, New York
Piret C (2012) The orthogonal gradients method: a radial basis functions method for solving partial differential equations on arbitrary surfaces. J Comput Phys 231:4662–4675
Powell MJD (1992) The theory of radial basis function approximation in 1990. In: Light W (ed) Advances in numerical analysis, vol II: wavelets, subdivision algorithms and radial functions. Oxford University Press, Oxford, pp 105–210
Ratcliff JT, Schubert G, Zebib A (1996) Steady tetrahedral and cubic patterns of spherical shell convection with temperature-dependent viscosity. J Geophys Res 101:473–484
Schaback R (1995) Error estimates and condition numbers for radial basis function interpolants. Adv Comput Math 3:251–264
Schaback R (2005) Multivariate interpolation by polynomials and radial basis functions. Constr Approx 21:293–317

DOI 10.1007/978-3-642-27793-1_61-1

Schoenberg IJ (1938) Metric spaces and completely monotone functions. Ann Math 39:811–841
Spotz WF, Taylor MA, Swarztrauber PN (1998) Fast shallow water equation solvers in latitude-longitude coordinates. J Comput Phys 145:432–444
St-Cyr A, Jablonowski C, Dennis JM, Tufo HM, Thomas SJ (2008) A comparison of two shallow-water models with nonconforming adaptive grids. Mon Weather Rev 136:1898–1922
Stemmer K, Harder H, Hansen U (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
Taylor M, Tribbia J, Iskandarani M (1997) The spectral element method for the shallow water equations on the sphere. J Comput Phys 130:92–108
Trefethen LN (2000) Spectral methods in MATLAB. SIAM, Philadelphia
Turing A (1952) The chemical basis of morphogenesis. Philos Trans R Soc B 237:37–52
van der Vorst H (1992) BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. SIAM J Sci Stat Comput 13(2):631–644. doi:10.1137/0913035
Varea C, Aragon J, Barrio R (1999) Turing patterns on a sphere. Phys Rev E 60:4588–4592
Wilson CTR (1920) Investigations on lightning discharges and on the electric field of thunderstorms. Philos Trans R Soc Lond A 221:73–115
Wilson CTR (1929) Some thundercloud problems. J Frankl Inst 208:1–12
Womersley RS, Sloan IH (2003/2007) Interpolation and cubature on the sphere. Website, http://web.maths.unsw.edu.au/~rsw/Sphere/
Wright GB, Fornberg B (2006) Scattered node compact finite difference-type formulas generated from radial basis functions. J Comput Phys 212:99–123
Wright GB, Flyer N, Yuen DA (2010) A hybrid radial basis function – pseudospectral method for thermal convection in a 3D spherical shell. Geophys Geochem Geosyst 11(7):Q07,003
Yoon J (2001) Spectral approximation orders of radial basis function interpolation on the Sobolev space. SIAM J Math Anal 33(4):946–958
Yoshida M, Kageyama A (2004) Application of the Ying-Yang grid to a thermal convection of a Boussinesq fluid with infinite Prandtl number in a three-dimensional spherical shell. Geophys Res Lett 31:L12,609
Zhai S, Feng X, He Y (2013) A family of fourth-order and sixth-order compact difference schemes for the three-dimensional Poisson equation. J Sci Comput 54:97–120
Zhong S, McNamara A, Tan E, Moresi L, Gurnis M (2008) A benchmark study on mantle convection in a 3-D spherical shell using CitcomS. Geochem Geophys Geosyst 9:Q10,017