4-1-2009

A Radial Basis Function Method for the Shallow Water Equations on a Sphere

Natasha Flyer
National Center for Atmospheric Research Boulder, CO

Grady Wright
Boise State University

A RADIAL BASIS FUNCTION METHOD FOR THE SHALLOW WATER EQUATIONS ON A SPHERE

NATASHA FLYER AND GRADY B. WRIGHT

Abstract. The paper derives the first known numerical shallow water model on the sphere using radial basis function (RBF) spatial discretisation, a novel numerical methodology that does not require any grid or mesh. In order to perform a study with regard to its spatial and temporal errors, two nonlinear test cases with known analytical solutions are considered. The first is global steady-state flow with a compactly supported velocity field while the second is unsteady flow where features in the flow must be kept intact without dispersion. This behavior is achieved by introducing forcing terms in the shallow water equations. Error and time stability studies are performed both as the number of nodes is uniformly increased and the shape parameter of the RBF is varied, especially in the flat basis function limit. Results show that the RBF method is spectral, giving exceptionally high accuracy for low number of basis functions while being able to take unusually large time-steps. In order to put it in the context of other commonly used global spectral methods on a sphere, comparisons are given with respect to spherical harmonics, double Fourier series, and spectral element methods.

1. Introduction

RBFs have the advantage of achieving spectral accuracy in multi-dimensions for arbitrary node layouts with extreme algorithmic simplicity. For the purposes of interpolating multi-dimensional surfaces, the methodology has been around for nearly 40 years. However, it is only in the last 15 years that it has been applied to solving mixed partial differential equations (PDEs) containing parabolic and/or elliptic operators (cf. [23, 19, 18, 24, 34, 3]). Furthermore, with regard to spherical domains, it has only been considered for solving a *scalar* PDE for *strictly* linear operators in the last 5 years [14, 15], with the first application to a purely hyperbolic operator done in 2007 [5]. Given this perspective, the next obvious step would be to solve a system of coupled nonlinear PDEs on the sphere. Since the shallow water equations occur in a multitude of applications, they provide any ideal test bed to establish the viability of the RBF methodology in terms of numerical accuracy and stability in this setting.

Since no precursor in the RBF literature exists with regard to solving a system of coupled nonlinear PDEs on a sphere, it should be emphasized that the goal of the paper is to establish its performance and viability (both for larger values of the shape parameter and in the flat basis function limit). As a result, the paper: a) develops the first known RBF method for the shallow water equations on a

The National Center for Atmospheric Research is sponsored by the National Science Foundation. N. Flyer was supported by NSF grant ATM-0620100. G.B. Wright was supported by NSF grant ATM-0801309.The authors would like to thank Drs. Amik St-Cyr, Ram Nair, and Aime Fournier.

sphere and in doing so, gives the first application of RBFs to a system of coupled nonlinear PDEs on a sphere; b) demonstrates and analyzes why RBFs can take unusually long time steps; and c) shows that RBFs give high accuracy compared to the other spectrally accurate methods when the same degrees of freedom are used.

RBFs can be a competitive methodology, giving high numerical accuracy when compared to other high-order spectrally accurate methods.

The methodology is tested on two cases in which analytical solutions are known in order to be able to perform an accurate error analysis. The first test is a global steady-state flow where the initial (and for all time) velocity field is compactly supported (i.e. nonzero in a limited band region), admitting a solution that is represented as an infinite spherical harmonic expansion. The second test is more challenging as it represents unsteady flow, modelling nonlinear advection of the initial condition that must be kept intact over time with minimal dispersion. Physically, it can be described as a forced nonlinear system with a translating low pressure center that is superimposed on a jet stream. An overview of the paper is as follows: Section 2 gives an introduction to RBFs; Section 3 provides a simple derivation of the shallow water equations for the sphere in Cartesian coordinates; Section 4 derives the discrete RBF formulation of the shallow water equations that is used in the test cases; Section 5 derives the linearized shallow water equations that will be used in the time stability analysis; Section 6 are the numerical studies; Section 7 gives timing benchmarks and Section 8 summarizes the paper with future prospects.

2. Introduction to Radial Basis Functions

The motivation of the RBF methodology originated with R.L. Hardy [17] 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?'. It was first shown by Mairhuber [26] that, in more than one dimension, interpolation by an expansion of basis functions, $\{\psi_j(\mathbf{x})\}_{j=1}^N$, $\mathbf{x} \in \mathbb{R}^d$, that are independent of the node locations is not well-posed. That is, there exists an infinite number of node configurations that will yield a singular interpolation problem . Hardy bypassed this singularity problem with a novel approach in which the interpolant is constructed from linear combinations of a *single* basis function that is radially symmetric about its center and whose argument is *dependent* on the node locations. By giving up orthogonality, well-posedness of the interpolant and its derivatives for *any* set of distinct scattered nodes in *any* dimension is gained.

Commonly used RBFs are given in Figure 1, where $r = \|\mathbf{x}-\mathbf{x}_j\|$ is the Euclidean or ℓ_2 norm. The piecewise smooth RBFs feature a jump in some derivative at $\mathbf{x} = \mathbf{x}_j$ and thus can only lead to algebraic convergence. For instance, the radial cubic $|r|^3$ has a jump in the third derivative, leading to fourth order convergence in 1-D, with the order of convergence increasing as the dimension increases (c.f. [27]). On the other hand, the evidence strongly suggests that infinitely smooth RBFs will lead to spectral convergence [25, 35], as is demonstrated in this paper. Notice that the infinitely smooth RBFs depend on a shape parameter ε. It was first shown by Driscoll and Fornberg [2] that, in 1-D, 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

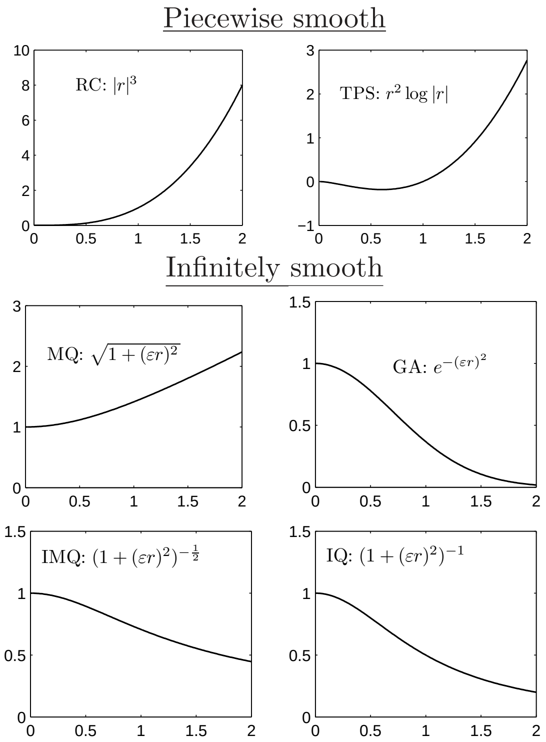


FIGURE 1. Commonly used RBFs $\phi(r)$: RC = radial cubic, TPS = thin plate spline, MQ = multiquadric, GA = Gaussian, IMQ = inverse multiquadric, IQ = inverse quadratic.

and Piret [10] 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.

The accuracy of an RBF approximation can be improved by increasing the number of terms in the expansion and/or decreasing the shape parameter ε [24]. In either case, the shifted RBFs in the expansion approach the constant one as is clearly seen in the case for $\varepsilon = 0.1$ in Figure 2, which leads to ill-conditioning. The Contour-Padé algorithm [12] can be used to bypass the RBF ill-conditioning mentioned above for the case of a fixed (relatively small) number of terms and increasingly small values of ε (even $\varepsilon = 0$). Furthermore, Fornberg and Piret [10] have recently developed an algorithm (RBF-QR) for bypassing the ill-conditioning for RBF interpolation on the surface of the sphere both as the number of terms is increased and ε is decreased right to zero. We will implement the RBF-QR to study how the error and time stability vary as a function of the shape parameter ε and what are the optimal choices for it.

A good way to introduce the RBF methodology is through interpolation, since the differentiation matrices can be obtained by applying the exact differential operator to the interpolant (1) and then evaluating it at the data locations. As mentioned above, RBFs approximate a function $f(x)$ sampled at some set of N distinct node locations by translates of a single radially symmetric function $\phi(r)$. For example, given the data values $\{f_k\}_{k=1}^N$ at the node locations $\{\mathbf{x}_k\}_{k=1}^N$, the RBF interpolant

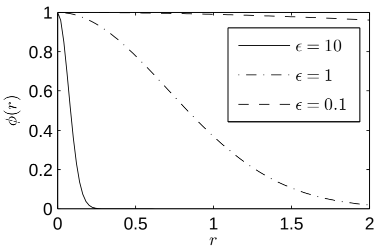


FIGURE 2. GA RBF for various values of the shape parameter ε

$s(\mathbf{x})$ to the data is defined by

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

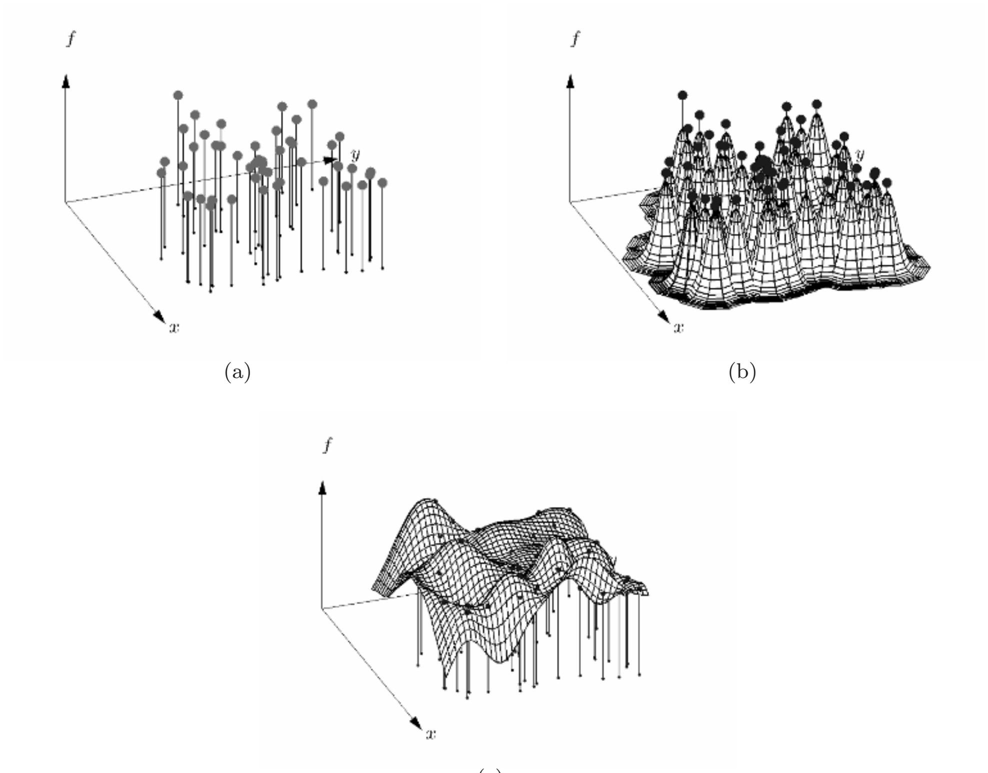
where the expansion coefficients, $\{c_k\}_{k=1}^{N}$, are found by enforcing the collocation conditions such that the residual is zero at the data locations. This is equivalent to solving the symmetric linear system of equations

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

where A is the interpolation matrix. An example of RBF interpolation in 2-D is illustrated in Figure 3. For RBFs such as the GA, IMQ, and IQ, the matrix in (2) is positive definite regardless of the distinct node locations and the dimension. For complete details on the well-posedness of (2) for all the RBFs listed in Figure 1, see for example [1, Ch. 12–16].

Studies have shown that if the shape parameter, ε, is kept fixed throughout the domain (as will be done in the current study—variable shape parameter is needed when implementing local mesh refinement [13, 3, 31, 4]) best results are achieved with roughly evenly distributed nodes [20]. 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, electrostatic repulsion [28]. Although any of these will suffice, we have decided to use an electrostatic 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.

The ME node sets have the property that h, a measure of the spacing of the nodes, decays approximately uniformly like the inverse of the square root of the number of nodes N [33], i.e. $h \sim \frac{1}{\sqrt{N}}$. Thus, they are similar to a uniform discretisation of the unit square. In Figure 4, the distribution for 1849 nodes on the unit sphere is displayed. For infinitely smooth RBFs, [22] shows that, provided the underlying function being interpolated is sufficiently smooth, RBF interpolants converge in the L^{∞} norm like $h^{-1/2}e^{-c/4h}$ for some constant $c > 0$ that depends on the RBF. For the ME node sets, convergence will thus proceed like $N^{1/4}e^{-c\sqrt{N}/4}$.



(a)

(b)

(c)

Figure 3. (a) Data values $\{f_j\}_{j=1}^{N}$, (b) the RBF collocation functions, (c) the resulting interpolant.

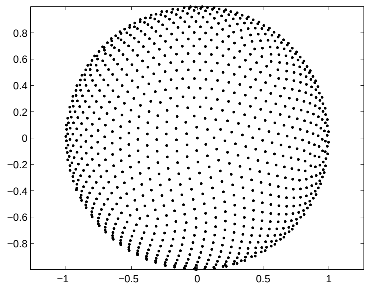


Figure 4. Minimum energy (ME) nodes on the sphere, $N = 1849$.

3. Cartesian form of the shallow water equations on the sphere

As was discussed in [5], the RBF differentiation matrix approximating a continuous operator is completely independent of the orientation and type of coordinate system in which the operator was originally posed. For example, the gradient operator has *no* singularities on the surface of the sphere. However, if we choose to represent it in spherical coordinates (λ, θ), where λ and θ are the respective longitudinal and latitudinal variables, singularities will occur at the poles $(\theta = \pm\frac{\pi}{2})$ simply due to the coordinate system. Unlike orthogonal expansions on a latitude-longitude grid, we can apply the surface gradient operator $\nabla_s = \frac{1}{\cos\theta}\frac{\partial}{\partial\lambda}\hat{\boldsymbol{\lambda}} + \frac{\partial}{\partial\theta}\hat{\boldsymbol{\theta}}$ to a RBF and there will be no evidence of the pole singularities as the operator, itself, is inherently smooth on the sphere. For PDEs, where the *scalar* variable is always acted on by a smooth spatial operator, one can easily pose the equation in spherical coordinates and then apply the RBF methodology (as was done in [5] and [11]). However, when solving the shallow water equations this is not the case. If a spherical coordinate system were to be used, the directional velocity *vector* components, u (latitudinal) and v (longitudinal) will inherently carry the pole singularities in their solution since the unit vector $\hat{\boldsymbol{\theta}}$ is discontinuous at the poles. As a result, the Cartesian form of the shallow water equations will be used since such a coordinate system has no singularities.

The full shallow water equations in a 3-D Cartesian coordinate system for a rotating fluid are

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - f(\mathbf{x}\times\mathbf{u}) - g\nabla h, \tag{3}$$

$$\frac{\partial h}{\partial t} = -\nabla\cdot(h\mathbf{u}), \tag{4}$$

where f is the Coriolis force, $\nabla = \partial_x\hat{\mathbf{i}} + \partial_y\hat{\mathbf{j}} + \partial_z\hat{\mathbf{k}}$, $\mathbf{u} = u\hat{\mathbf{i}} + v\hat{\mathbf{j}} + w\hat{\mathbf{k}}$ is the velocity vector, h is the geopotential height and $\mathbf{x} = \{x, y, z\}^T$ represents the is the position vector.

Next, we confine the flow to the surface of the unit sphere, which first requires constructing a linear operator, $\mathbf{P}$, for projecting vectors in 3-D Cartesian space onto a plane tangent to the sphere. The construction of $\mathbf{P}$ is quite simple since, if (x, y, z) is a point on the unit sphere, then the normal at (x, y, z) is just $\mathbf{x}$. Thus, if $\mathbf{u}$ is a vector at (x, y, z), then $\mathbf{x}\mathbf{x}^T\mathbf{u}$ gives the projection of $\mathbf{u}$ onto $\mathbf{x}$, and $\mathbf{u} - \mathbf{x}\mathbf{x}^T\mathbf{u}$ gives the projection of $\mathbf{u}$ onto the plane tangent to the sphere at (x, y, z). As a result, we define the projection operator $\mathbf{P}$:

$$\mathbf{P} = \mathbf{I} - \mathbf{x}\mathbf{x}^T = \begin{bmatrix} (1-x^2) & -xy & -xz \\ -xy & (1-y^2) & -yz \\ -xz & -yz & (1-z^2) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x^T \\ \mathbf{p}_y^T \\ \mathbf{p}_z^T \end{bmatrix}, \tag{5}$$

where $\mathbf{I}$ is the 3-by-3 identity matrix. $\mathbf{p}_x$, $\mathbf{p}_y$, $\mathbf{p}_z$ represent the projection operators in the x, y and z directions respectively.

Next, the gradient operator appearing in (3) must be constrained so that when it is applied to a scalar, it produces a vector which is tangent to the unit sphere. This is done by replacing all occurrences of ∇ in (3) with the operator

$$\mathbf{P}\nabla = \begin{bmatrix} \mathbf{p}_x\cdot\nabla \\ \mathbf{p}_y\cdot\nabla \\ \mathbf{p}_z\cdot\nabla \end{bmatrix},$$

since $\mathbf{P}$ projects arbitrary 3-D Cartesian vectors onto a plane tangent to the unit sphere. The divergence operator appearing in (4) must also be restricted so that it produces the surface divergence of a vector field. This is also accomplished with $\mathbf{P}$ by simply taking the dot product of the vector field with the projected gradient operator $\mathbf{P}\nabla$.

The final step in confining the motion to the unit sphere is to project the right hand side of (3), with the modified differential operators, onto the corresponding $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ directions. For example, in the case of the u momentum equation, this is done by taking the dot product of the modified right hand side of (3) with the $\mathbf{p}_x$ projection vector.

Putting all these pieces together, the final shallow water equations on the surface of the unit sphere in Cartesian coordinates is given by

$$\text{(6a)} \qquad \frac{\partial u}{\partial t} = -\,\mathbf{p}_{\mathbf{x}} \cdot \underbrace{\begin{bmatrix} (\mathbf{u}\cdot\mathbf{P}\nabla)u + f(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{i}} + g(\mathbf{p}_x\cdot\nabla)h \\ (\mathbf{u}\cdot\mathbf{P}\nabla)v + f(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{j}} + g(\mathbf{p}_y\cdot\nabla)h \\ (\mathbf{u}\cdot\mathbf{P}\nabla)w + f(\mathbf{x}\times\mathbf{u})\cdot\hat{\mathbf{k}} + g(\mathbf{p}_z\cdot\nabla)h \end{bmatrix}}_{\text{RHS}},$$

$$\text{(6b)} \qquad \frac{\partial v}{\partial t} = -\,\mathbf{p}_y \cdot \text{RHS},$$

$$\text{(6c)} \qquad \frac{\partial w}{\partial t} = -\,\mathbf{p}_z \cdot \text{RHS},$$

$$\text{(6d)} \qquad \frac{\partial h}{\partial t} = -\,(\mathbf{P}\nabla)\cdot(h\mathbf{u})\,.$$

Notice that the only spatial operator that needs to be discretized is the projected gradient, $\mathbf{P}\nabla$ and its components, $\mathbf{p}_x\cdot\nabla, \mathbf{p}_y\cdot\nabla, \mathbf{p}_z\cdot\nabla$.

4. The RBF method on the sphere

In the next 3 subsections, we develop the discrete operators that are necessary to spatially discretise the shallow water equations using RBFs. In 4.1, we define how to formulate the projected gradient operator in RBFs. In 4.2, we do a step-by-step construction of the 3 needed differentiation matrices, D_N^x, D_N^y, D_N^z. Lastly, 4.3 demonstrates the entire RBF formulation of the shallow water equations to be implemented computationally.

4.1. Formulation of the projected gradient operator in RBFs.

Let $\mathbf{x} = \{x, y, z\}$ and $\mathbf{x}_k = \{x_k, y_k, z_k\}_{k=1}^N$ be points on the unit sphere, then the Euclidean distance from $\mathbf{x}$ to $\mathbf{x}_k$ is

$$\text{(7)} \qquad r(\mathbf{x}) = \|\mathbf{x}-\mathbf{x}_k\| = \sqrt{(x-x_k)^2+(y-y_k)^2+(z-z_k)^2} = \sqrt{2(1-\mathbf{x}^T\mathbf{x}_k)}.$$

It is important to note that the distances are not great circle arcs measured along the surface but are the Euclidean distance measured straight through the sphere. The reason being is that RBFs do not “feel” the geometry of the domain in which they are applied nor the dimension, only the scalar distances between the nodes and the locations at which the RBFs are centered (in our study the RBFs centers and the nodes coincide, but this need not be the case).

Let $\phi_k(r(\mathbf{x}))$ be an RBF centered at $\mathbf{x}_k$. Using the chain rule, the gradient of $\phi_k(r(\mathbf{x}))$ is given by

$$\text{(8)} \qquad \nabla\phi_k(r(\mathbf{x})) = \begin{bmatrix} \frac{\partial}{\partial x}\phi_k(r(\mathbf{x})) \\ \frac{\partial}{\partial y}\phi_k(r(\mathbf{x})) \\ \frac{\partial}{\partial z}\phi_k(r(\mathbf{x})) \end{bmatrix} = \begin{bmatrix} (x-x_k) \\ (y-y_k) \\ (z-z_k) \end{bmatrix} \frac{\phi_k'(r(\mathbf{x}))}{r(\mathbf{x})} = (\mathbf{x}-\mathbf{x}_k)\frac{\phi_k'(r(\mathbf{x}))}{r(\mathbf{x})},$$

where we have used $'$ to denote differentiation with respect to r. The projected RBF gradient operator is then simply obtained by applying the projection matrix $\mathbf{P} = \mathbf{I} - \mathbf{x}\mathbf{x}^T$ to (8):

$$\mathbf{P}\nabla\phi_k(r(\mathbf{x})) = \mathbf{P}(\mathbf{x} - \mathbf{x}_k)\frac{\phi_k^{'}(r(\mathbf{x}))}{r(\mathbf{x})} = -\mathbf{P}\mathbf{x}_k\frac{\phi_k^{'}(r(\mathbf{x}))}{r(\mathbf{x})} = \begin{bmatrix} x\,\mathbf{x}^T\mathbf{x}_k - x_k \\ y\,\mathbf{x}^T\mathbf{x}_k - y_k \\ z\,\mathbf{x}^T\mathbf{x}_k - z_k \end{bmatrix}\frac{\phi_k^{'}(r(\mathbf{x}))}{r(\mathbf{x})}.$$

Now we have all the components necessary to build the action of $\mathbf{P}\nabla = [\mathbf{p}_x \cdot \nabla, \mathbf{p}_y \cdot \nabla, \mathbf{p}_z \cdot \nabla]^T$ on an RBF representation of a geophysical quantity.

4.2. **Constructing the RBF differentiation matrices.** Given a geophysical quantity $f(\mathbf{x}) = f(x, y, z)$ known at the node locations $\{\mathbf{x}_j\}_{j=1}^N$ on the surface of a unit sphere, we first represent $f(\mathbf{x})$ as an RBF expansion

$$f(\mathbf{x}) = \sum_{k=1}^{N} c_k\phi_k(r(\mathbf{x})), \tag{9}$$

where the coefficients c_k are determined by collocation. Applying the projected gradient operator in the x direction to (9) and evaluating it at $\{\mathbf{x}_j\}_{j=1}^N$ gives

$$\begin{aligned}
\left[\mathbf{p}_x \cdot \nabla f(\mathbf{x})\right]|_{\mathbf{x}=\mathbf{x}_j} &= \sum_{k=1}^{N} c_k \left[\mathbf{p}_x \cdot \nabla\phi_k(r(\mathbf{x}))\right]|_{\mathbf{x}=\mathbf{x}_j}, \qquad j = 1, \ldots, N \\
&= \sum_{k=1}^{N} c_k \underbrace{\left[x_j\,\mathbf{x}_j^T\mathbf{x}_k - x_k\right]\frac{\phi_k^{'}(r(\mathbf{x}_j))}{r(\mathbf{x}_j)}}_{B_{j,k}^x}, \quad j = 1, \ldots, N \\
&= B^x c \\
&= \left(B^x A^{-1}\right) f \\
&= D_N^x f, \qquad (10)
\end{aligned}$$

where A is the interpolation matrix in (2) and we have used the property that $c = A^{-1}f$. The differentiation matrix D_N^x is the discrete RBF approximation to the x-component of the projected gradient operator. Using the same concept, D_N^y and D_N^z are derived as follows,

$$\left[\mathbf{p}_y \cdot \nabla f(\mathbf{x})\right]|_{\mathbf{x}=\mathbf{x}_j} = \left(B^y A^{-1}\right) f = D_N^y f \tag{11}$$

$$\left[\mathbf{p}_z \cdot \nabla f(\mathbf{x})\right]|_{\mathbf{x}=\mathbf{x}_j} = \left(B^z A^{-1}\right) f = D_N^z f \tag{12}$$

where the entries in the matrices B^y and B^z are respectively given by

$$\begin{aligned}
B_{j,k}^y &= \left[y_j\,\mathbf{x}_j^T\mathbf{x}_k - y_k\right]\frac{\phi^{'}(r_k(\mathbf{x}_j))}{r_k(\mathbf{x}_j)}, \\
B_{j,k}^z &= \left[z_j\,\mathbf{x}_j^T\mathbf{x}_k - z_k\right]\frac{\phi^{'}(r_k(\mathbf{x}_j))}{r_k(\mathbf{x}_j)}.
\end{aligned}$$

Note that all three differentiation operators (10)-(12) are well-defined with no pole singularities.

4.3. **The discretized RBF shallow water equations.** Let $\underline{h}$ be the geopotential height and $\{\underline{u}, \underline{v}, \underline{w}\}$ be the components of the velocity field $\mathbf{u}$ sampled at the node locations $\{\mathbf{x}\}_{j=1}^{N} = \{\underline{x}, \underline{y}, \underline{z}\}$. Then, the right hand side (RHS) of the Cartesian form of the shallow water equations in Section 3 can now be written in discretized form RHS_D as

$$\text{RHS}_D = \begin{bmatrix} \begin{bmatrix} \underline{u} \circ D_N^x \underline{u} + \underline{v} \circ D_N^y \underline{u} + \underline{w} \circ D_N^z \underline{u} \\ \underline{u} \circ D_N^x \underline{v} + \underline{v} \circ D_N^y \underline{v} + \underline{w} \circ D_N^z \underline{v} \\ \underline{u} \circ D_N^x \underline{w} + \underline{v} \circ D_N^y \underline{w} + \underline{w} \circ D_N^z \underline{w} \end{bmatrix} + f \begin{bmatrix} \underline{y} \circ \underline{w} - \underline{z} \circ \underline{v} \\ \underline{z} \circ \underline{u} - \underline{x} \circ \underline{w} \\ \underline{x} \circ \underline{v} - \underline{y} \circ \underline{u} \end{bmatrix} + g \begin{bmatrix} D_N^x \\ D_N^y \\ D_N^z \end{bmatrix} \underline{h} \end{bmatrix}$$

where $\circ$ denotes element by element multiplication of the vectors. The full discretised equations are given by

$$\begin{aligned} (13) \qquad \frac{\partial \underline{u}}{\partial t} &= -\underline{p_x} \cdot \text{RHS}_D, \qquad \frac{\partial \underline{v}}{\partial t} = -\underline{p_y} \cdot \text{RHS}_D, \qquad \frac{\partial \underline{w}}{\partial t} = -\underline{p_z} \cdot \text{RHS}_D \\ (14) \qquad \frac{\partial \underline{h}}{\partial t} &= \underline{u} \circ D_N^x \underline{h} + \underline{v} \circ D_N^y \underline{h} + \underline{w} \circ D_N^z \underline{h} + \underline{h} \circ (D_N^x \underline{u} + D_N^y \underline{v} + D^z \underline{w}) \end{aligned}$$

where $\underline{p_x}, \underline{p_y}, \underline{p_z}$ are the vectors $\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z$ evaluated at the nodes $\{\mathbf{x}_j\}_{j=1}^{N}$. The method-of-lines (MOL) technique is used to advance the system in time. As a note, for flows on spheres of radius a, the modification to the method is simple - leave $\mathbf{x}$ and $\mathbf{x}_j$ as points on the unit sphere and divide D_N^x, D_N^y, and D_N^z by a.

5. Eigenvalue stability-the linearized equations

For each test case, we will perform an eigenvalue stability analysis in order to understand: 1) why the RBF method can take long time steps; and 2) the impact of ε-refinement on stability. However, in order to do so, we must linearize the shallow water equations about a given state. In all cases, we linearize about the initial condition, since this is the solution for all time for the steady state case and for unsteady flow the shape of the initial condition is advected intact.

Assume $\tilde{\mathbf{u}}$ and $\tilde{h}$ are the approximate solutions to (6) and can be written as

$$\begin{aligned} \tilde{\mathbf{u}} &= \mathbf{u}_0 + \delta \mathbf{u}_1 + O(\delta^2), \\ \tilde{h} &= h_0 + \delta h_1 + O(\delta^2), \end{aligned}$$

where $\mathbf{u}_0$ and h_0 are the initial conditions. Substituting this expansion into (6) and only considering perturbations of $O(\delta)$ gives the linearized equations

$$(15a) \qquad \frac{\partial u_1}{\partial t} = -\mathbf{p}_x \cdot \underbrace{\begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{P}\nabla u_0 + \mathbf{u}_0 \cdot \mathbf{P}\nabla u_1 + f(yw_1 - zv_1) + g\mathbf{p_x} \cdot \nabla h_1 \\ \mathbf{u}_1 \cdot \mathbf{P}\nabla v_0 + \mathbf{u}_0 \cdot \mathbf{P}\nabla v_1 + f(zu_1 - xw_1) + g\mathbf{p_y} \cdot \nabla h_1 \\ \mathbf{u}_1 \cdot \mathbf{P}\nabla w_0 + \mathbf{u}_0 \cdot \mathbf{P}\nabla w_1 + f(xv_1 - yu_1) + g\mathbf{p_z} \cdot \nabla h_1 \end{bmatrix}}_{\text{RHS}_1},$$

$$(15b) \qquad \frac{\partial v_1}{\partial t} = -\mathbf{p}_y \cdot \text{RHS}_1,$$

$$(15c) \qquad \frac{\partial w_1}{\partial t} = -\mathbf{p}_z \cdot \text{RHS}_1,$$

$$(15d) \qquad \frac{\partial h_1}{\partial t} = -(\mathbf{u}_1 \cdot \mathbf{P}\nabla h_0 + \mathbf{u}_0 \cdot \mathbf{P}\nabla h_1 + h_1 \mathbf{P}\nabla \cdot \mathbf{u}_0 + h_0 \mathbf{P}\nabla \cdot \mathbf{u}_1).$$

6. Numerical Studies

For each test case, we evaluate the convergence properties of the RBF method both with respect to uniformly increasing the number of nodes (h-refinement) and decreasing ε (ε-refinement). We also perform a stability/eigenvalue study to explore how the time errors behave as a function of ε and the time step. Also, how well linear stability analysis predicts the growth of unstable modes in the full RBF solution is explored. Lastly, the performance of the RBF method is put into context with the following commonly-used spectral models in the literature: a de-aliased spherical harmonic method (SH) [21], a double Fourier with a spherical harmonic filter (DF/SHF) [29], and a spectral element method (SE) with filtering [30]. The RBF method does not need to use any spatial filtering to achieve the numerical accuracies reported.

6.1. **Steady state test case: Compactly supported wind field.** In this test case, the flow field is nonlinear and compactly supported but still infinitely differentiable (see [32] test case 3). For simplicity, the initial condition and solution for all time for the wind field in spherical coordinates is given by

$$u_{sph}(\theta) = u_0 \exp[\frac{(\theta_e + \theta_b - 2\theta)^2}{\sigma^2(\theta_e - \theta)(\theta - \theta_b)}] \;\; , \; \theta_b < \theta < \theta_e, \tag{16}$$

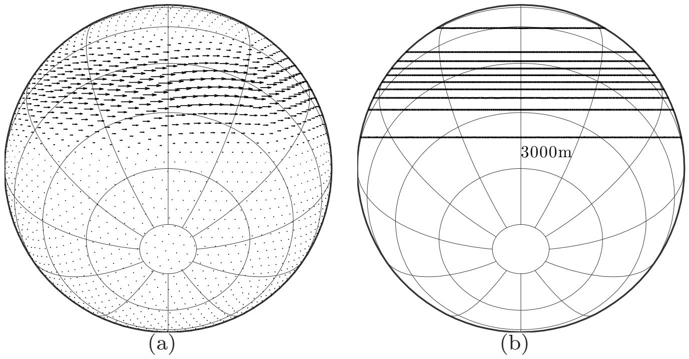
where $\sigma = \sqrt{0.3}, \theta_b = -\pi/6, \theta_e = \pi/2$ and $u_0 = \frac{2\pi a}{12\,days}$. $u_{sph}(\theta)$ has a maximum at $\theta = (\theta_e + \theta_b)/2$, corresponding to a maximum wind positioned at 30°N ($\pi/6$) and is zero outside of $\theta_b < \theta < \theta_e$. The latitudinal component of the velocity field is zero. The analytic solution is then defined by a rotation of the coordinate system to tilt the wind field an angle α relative to the the polar axis. For $\alpha = \pi/3$, the wind field is graphed in Figure 5(a) as an orthographic projection centered at 60°S and 0°E. The geopotential height field is defined by numerically integrating to machine precision

$$\frac{\partial gh}{\partial \theta} = -(afu + u^2 \tan\theta) \tag{17}$$

where $f = 2\Omega \sin\theta, \Omega = 7.292 \times 10^{-5}\text{s}^{-1}$ and $a = 6.37122 \times 10^6\text{m}$. The geopotential height field is graphed in Figure 5(b). The test case is run for 5 days.

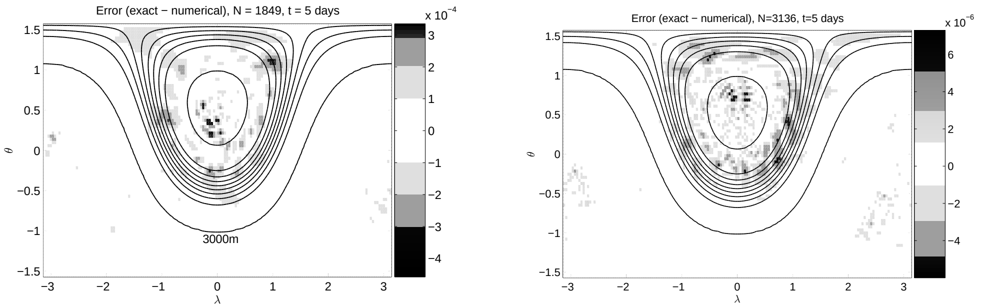
6.1.1. *Error study with regard to h-refinement.* Figure 6 displays the error (exact minus numerical) in the RBF solution of the geopotential height field on an unrolled sphere for (a) $N = 1849$ and (b) $N = 3136$, respectively. The contour lines represent the exact solution in 100m intervals, beginning at 3000m. In Figure 6a, all deviations in the solution of less than 10^{-4}m are shown in white while in Figure 6b deviations from the exact solution of less than 10^{-6}m are shown in white. As expected, both figures demonstrate that the dominant error is concentrated in the area with the steepest gradients. This test would be a prime candidate for local node refinement, as clustering the nodes in such areas would have a high likelihood of severely decreasing the error as has been demonstrated in [4]. Furthermore, notice that there is little evidence of any dispersive wave trains in either plots, indicating that the right hand side of the PDEs in (13) are being adequately resolved.

Figure 7 shows the time traces of the relative ℓ_2 error for the 5 day simulation for various values of N and fixed $\varepsilon = 3.25$. Notice that the error for all values of N barely grow as a function of time during the simulation period. To determine



(a) (b)

Figure 5. Analytical solution for the steady state test case in the rotated coordinate system. (a) The velocity field sampled at the $N = 3136$ ME nodes; (b) the contours of the height field from 2100 m to 3000 m in increments of 100 m. Both plots are orthographic projections centered at 60°S ($-\pi/3$) and 0°E, which is what the solution would look like in non-rotated coordinates.



(a) $N = 1849$ (b) $N = 3136$

Figure 6. The error (exact - numerical) for the height field from the steady state test case at $t = 5$ days on an "unrolled" sphere. (a) $N = 1849$ nodes, (b) $N = 3136$ nodes. Solid black lines correspond to contours of the exact solution plotted from 2100m to 3000m in increments of 100m. $\varepsilon = 3.25$ for all N.

.

spectral convergence from the time traces, we plot the relative ℓ_2 error as a function of spatial resolution (i.e. $\sqrt{N}$) in Figure 8 and see that indeed it is spectral. The figure also demonstrates that the ℓ_2 error is relatively insensitive to the value of ε used between 3.25 and 4. For larger nodes sets, if ε goes below 3.25 ill-conditioning will set in. More on the sensitivity of the calculation to ε will be addressed in the next section on ε-refinement.

Figure 9 shows the relative ℓ_1, ℓ_2 and ℓ_∞ norms of the error for the RBF method as a function of time for $N = 3136$ and $\Delta t = 10$ minutes. This spatial resolution was selected since it results in a relative ℓ_2 error of $O(10^{-10})$, which is also the

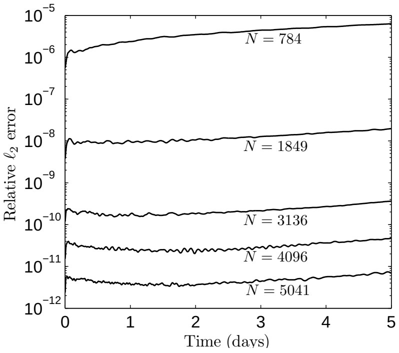


FIGURE 7. Relative ℓ_2 error in the height field for the steady state test case as a function of time and N. $\varepsilon = 3.25$ for all N.

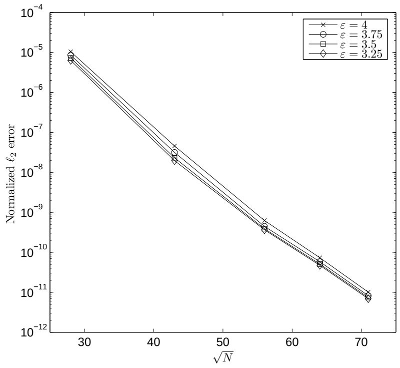


FIGURE 8. Relative ℓ_2 error in the height field for the steady state test case at $t = 5$ days as a function $\sqrt{N}$ and ε.

order of the lowest relative ℓ_2 error reported by many of the other spectral methods (see Table 1). For this time step and node set, Figure 10 shows the time traces as a function of ε. One can see that as ε decreases the error decreases. However, the rate at which the error decreases slows down as the value of $\varepsilon = 3.25$ is approached. One can then ask is this the effect of ill-conditioning as the shape of the RBFs become flat? The next logical question would then be, "What if we implemented the RBF-QR algorithm [10], which by-passes this ill-conditioning problem and allows for calculations in the limit as $\varepsilon \to 0$. Will we see the error drastically decrease?".

6.1.2. *Error and time stability study with regard to ε-refinement.* Our objective is to explore the accuracy and stability of the RBF shallow water method as the shape parameter ε decreases to zero. Since the solution to this test case can be accurately resolved with a low resolution model, as demonstrated in the previous section, we fix the node set at $N = 484$ for the study. To compute the RBF differentiation matrices for this resolution in the low ε regime (i.e. $0 \leq \varepsilon \leq 0.7$ for $N = 484$) we use the RBF-QR algorithm.

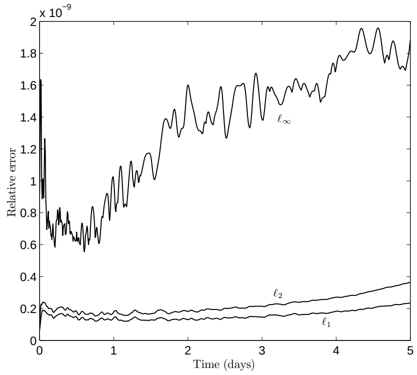


Figure 9. Relative ℓ_1, ℓ_2, and ℓ_∞ errors in the height field for the steady state test case as a function of time for $N = 3136$, $\varepsilon = 3.25$, $\Delta t = 10$ min.

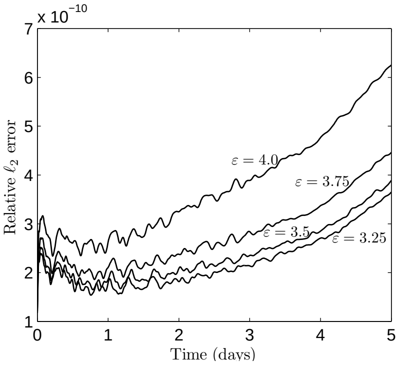


Figure 10. Relative ℓ_2 errors in the height field for the steady state test case as a function of time and different values of ε.

Figure 11 shows time traces for the 5 day run of the relative ℓ_2 error for ε varying from 0 to 0.7. Leap frog with a Robert filter of $\gamma = 0.07$ and a time step of 18 minutes was used. Clearly, time instability sets in as ε tends to zero, with the worse case being $\varepsilon = 0$ corresponding to spherical harmonics as previously mentioned. There are an infinite set of node layouts, such as a latitude-longitude grids or any perturbation to them, that fall on or close to the zeros of the spherical harmonics. The result is singular or close to singular spherical harmonic interpolation matrices. As a result, a least squares approach or over-sampling is needed in order to "pin down" a solution when using this basis. Thus, one conclusion is that the optimal ε is not in the $\varepsilon \to 0$ limit. However, one possibility as to why instability sets in as ε becomes small is that we are not using a small enough time step. However, Figure 12 easily throws that notion aside as we see that whether the time step is 2, 6, or 18 minutes the blow-up of the ℓ_2 error is identical.

A necessary condition for the RBF method to be stable for all time is for the spectrum of the RBF operator for the entire right hand side of the linearized system (15) to be contained within the stability domain of the time-stepping scheme.

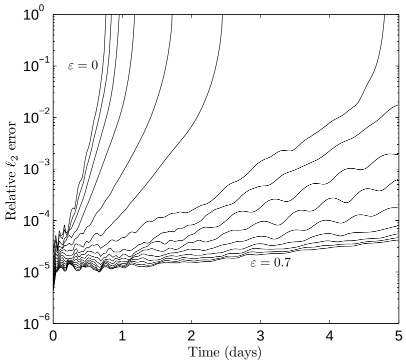


FIGURE 11. Relative ℓ_2 error in the height field for steady state test case. The different curves correspond to different values of ε used in the RBF solution, with the top curve corresponding to $\varepsilon = 0$ and the bottom curve to $\varepsilon = 0.7$ in increments of 0.05.

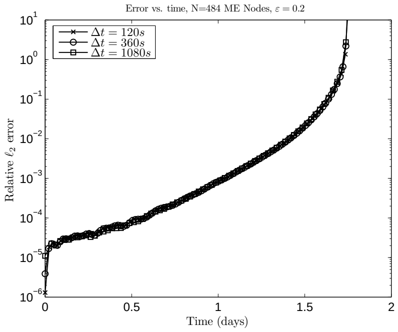
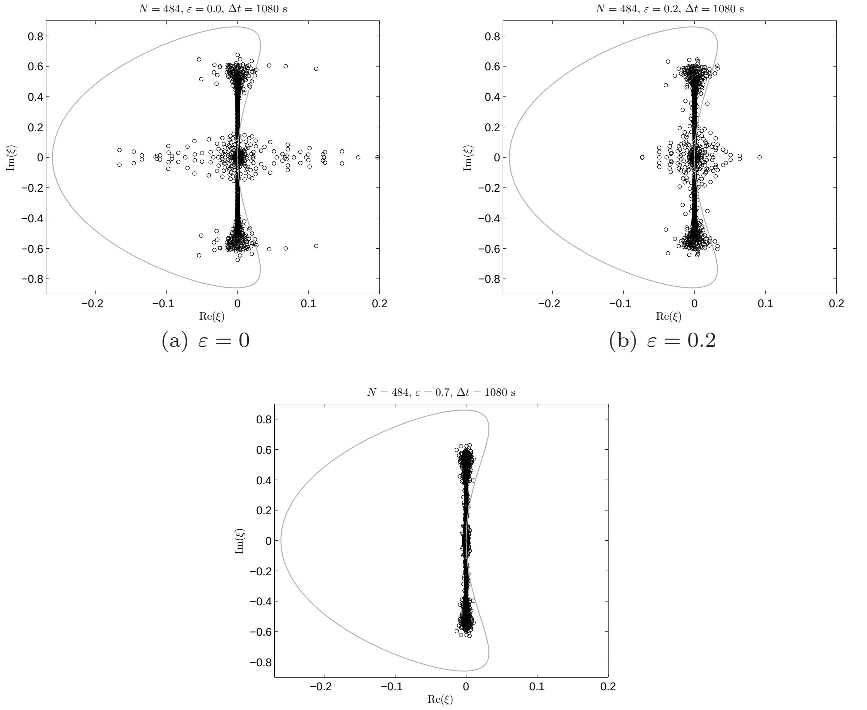


FIGURE 12. Relative ℓ_2 error in the height field for steady state test case. ε is fixed at 0.2 and the time-step is varied as indicated in the legend.

Figure 13 plots the stability domain for the leapfrog scheme with a Robert's filter of $\gamma = 0.07$ along with the spectrum of the RBF operator for the linearized equations for $\varepsilon = 0, 0.2, 0.7$. The most important trend to notice is that as ε decreases the real part of the eigenvalues spread off the imaginary axis and into the right half plane both near the center and at the ends of the spectrum. As was noticed in Figure 11, the worst case is for $\varepsilon = 0$ when RBFs reproduce spherical harmonics. However, if we look at the eigenvector associated with the eigenvalue having the largest real part in this case, as displayed in Figure 14, we see such modes are spurious and not physical. As is well known in the spherical harmonic literature, one needs to impose filters to take care of such eigenmodes. However, for the $N = 484$ case, the eigenmodes associated with the eigenvalues that are slightly off the imaginary axis in Figure 13(c) do not come into play for time integration periods on the order of a month.

We would also like to see how well the linear eigenvalue stability analysis predicts the growth of the unstable modes in the full RBF solution for this test case. If it



(a) $\varepsilon = 0$

(b) $\varepsilon = 0.2$

(c) $\varepsilon = 0.7$

FIGURE 13. Spectrums of the RBF approximation operator of the right hand side of the linearized system in equations (15a)-(15d) for different values of ε. Each spectrum in (a)–(c) has been scaled by the time step $\Delta t = 1080$ s (18 min), where $\xi = \Delta t \times \mathit{eigenvalues}$. The stability domain of time stepping scheme, leap frog scheme with a Robert filter of $\gamma = 0.07$, is plotted as light solid line.

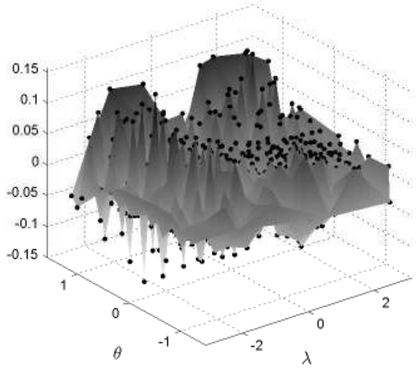
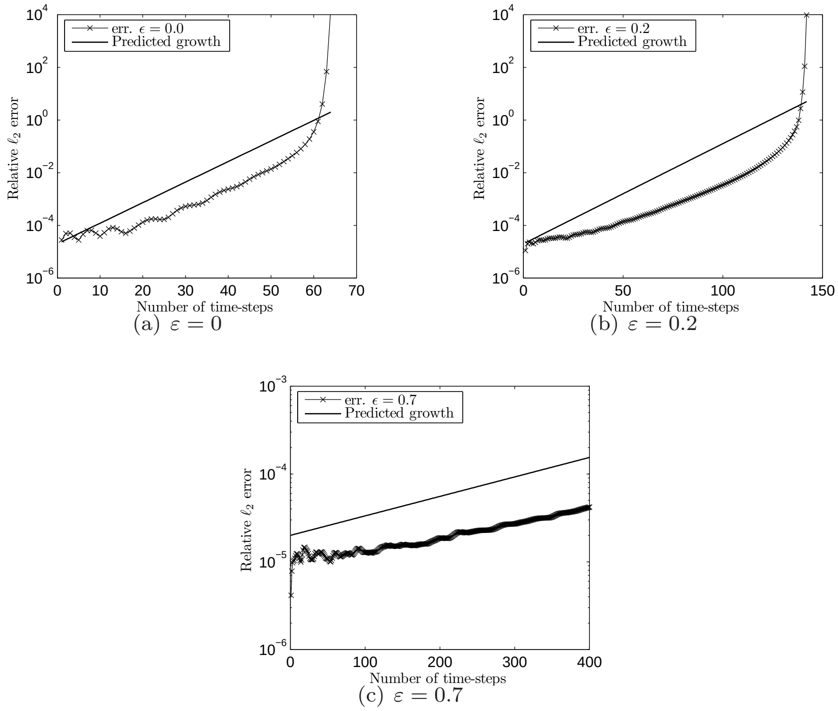


FIGURE 14. The unit eigenvector for the height field corresponding the eigenvalue with the largest real part in Figure 13(a).



(a) $\varepsilon = 0$

(b) $\varepsilon = 0.2$

(c) $\varepsilon = 0.7$

FIGURE 15. Relative ℓ_2 error in the RBF solution of the geopotential height for the steady state test case together with the predicted growth rate from equation 18. The value of ε used is given below each figure. For all plots, $\Delta t = 1080$ s and $N = 484$.

is a good predictor then the error in the RBF solution for a given ε and Δt should grow as

$$\text{Error} \sim (1 + \text{Re}(\beta))^k, \tag{18}$$

where β denotes the eigenvalue of the linearized RBF operator with the largest real part, scaled by Δt, that is not inside the time stability domain, and k is the number of time-steps taken. In Figure 15(a)-(c), we plot this predicted growth rate together with the error in the RBF solution for the height field for the same three value of ε used in the previous figures ($\varepsilon = 0$, 0.2, 0.7). Recall that for the first two of these ε values the solution went unstable well before $t = 5$ days, while for the last case the solution was stable and accurate. In Figure 15(a), we see the predicted growth matches the observed growth very nicely. In Figures 15(b) and (c), the linear stability theory actually predicts a slightly more elevated result than what is observed.

6.1.3. *Discussion of spectral models and comparison.* The methods presented in Tables 1 and 2 are described below [21] [29] [30].

(1) *Spherical Harmonics-SH* SH requires twice as many grid points as basis functions. Due to a severe CFL condition, a semi-implicit time-stepping is commonly used [21]. Also, SH are notorious for incorrectly increasing the

energy in low modes through aliasing. As a result, SH are never run without de-aliasing, normally using Orszag's 2/3 rule [16]. So, for example, a SH method that uses 1849 basis functions is in fact using 4096 basis functions with a grid of 8192 nodes. However, only 1849 bases are updated in time in spectral space due to de-aliasing. This leads to the forward SH transform (which is $O(M^3)$ operations, M is the number of associated Legendre roots in the latitudinal direction) being performed on 4096 bases using 8192 grid point values, and the reverse transform on 1849 spectral coefficients since the others are set to zero. FFTs can be used in the longitudinal direction, resulting in an operation count of $O(N^{3/2})$ per time step (N is the total number of grid points). Lastly, they involve high computational complexity.

(2) *Double Fourier-DF* Unlike SH, DF has a one-to-one correspondence between grid points and spectral coefficients. For comparison, a SH method that uses 1849 bases with the necessary 8192 grid points is approximately equivalent to a DF using 8192 grid points (see Table 1). DF for the same reasons as SH also requires the use of some type of de-aliasing filter. Neither SH nor DF have the option of local mesh refinement. Due to FFTs, DF has only an operation count of $O(N \log N)$ per time step.

(3) *Spectral Elements-SE* Grid generation is a large overhead cost for SE and include a variety of approaches such as icosahedral or the projection of a cube onto the surface of a sphere. SE allows for local node refinement but implementation is algorithmically highly complex. The method considered here (a cubed sphere [30]) uses a tensor product of Legendre polynomials on each element. Due to the clustering of Legendre grid points near the ends of each element boundary, the CFL-restriction is severe and increases as the order of polynomials used increases. SE models generally use some type of filtering or instability will set in. The operation count for SE is $O(kP^2)$, where k is the number of elements and P the order of the polynomial expansion on each element.

(4) *RBFs* RBFs are algorithmically very simple. The RBF method presented in this paper can be coded in less than 100 lines of MATLAB, using only level 1 to 3 BLAS operations. Due to close clustering of the eigenvalues of the linearized RBF operator about the imaginary axis as seen in the previous section, RBFs can take comparatively much longer time steps (see Table 1). Unlike the other methods, the RBF method can go without de-aliasing for longer time scales. Since aliasing is a spectral phenomena where all information is global, it is hypothesized that the locality properties of RBF coefficients [8] and the fact that they only pick up information locally [9] [7] contributes to this phenomena. Furthermore, for a given spatial resolution, RBFs achieve a higher numerical accuracy (see Table 1). To implement global RBFs, as done in this paper, is $O(N^2)$ operations. However, reducing this cost is discussed at the end of the paper and is currently under implementation.

For all RBF results listed in Table 1, MQ is used with $\varepsilon = 3.25$ (although any infinitely smooth RBFs can be used but GA tends to be more sensitive to the value of ε). Leap frog with a Robert's filter of $\gamma = 0.07$ is used to advance the model in time. All results are for $\alpha = \pi/3$, tilting the wind field in (16) so that it flows directly over the poles. For all N in Table 1, RBFs can *easily* take longer time steps

Method	Number of nodes (N)	Time-step (Δt)	Relative ℓ_2 error in h
RBF	784 (28)	20 minutes	6.32×10^{-6}
	1849 (43)	12 minutes	1.97×10^{-8}
	3136 (56)	10 minutes	3.65×10^{-10}
	4096 (64)	8 minutes	4.72×10^{-11}
	5041 (71)	6 minutes	6.88×10^{-12}
SH [21]	8192 (1849)	20 min.* (3)	7×10^{-10}
	18432 (4096)	15 minutes*	2.5×10^{-10}
DF/SHF [29]	2048	6 minutes	2×10^{-6}
	8192	3 minutes	4×10^{-10}
	32768	90 seconds	2×10^{-13}
SE [30]	6144	90 seconds	8×10^{-7}
	24576	45 seconds	1×10^{-10}

Table 1. Comparison of commonly used spectral methods for steady state flow with $\alpha = \pi/3$. The number in ( ) in the RBF section correspond to the square root of N, which is inversely proportional to the node spacing. The number in ( ) in the SH section correspond to the number of SH coefficients updated in time. RBF and DF use the same time-stepping scheme. SE uses a third-order Adams-Bashforth. SH uses a semi-implicit time-stepping scheme denoted by the *. For the SH 1849 case, [29] gives Δt=3 min. when using a leapfrog scheme as is done in RBF and DF.

with regard to stability but the time step was chosen so that temporal discretisation errors matched spatial discretisation errors, which is the optimal scenario. The RBF case is the only method that does not use a de-aliasing/spatial filter.

6.2. **Unsteady flow test case.** The test case is a low pressure system, initially centered at $(\lambda_0, \theta_0) = (0, \pi/4)$, superimposed on a jet stream that is symmetrical about the equator (see test case 4 in [32]). Forcing terms are added to the shallow water equations to constrain the motion of the system so that the initial condition is nonlinearly advected intact.

6.2.1. *Derivation of analytical solution in Cartesian form.* Since the Cartesian description of the solution does not appear to have been given before in the literature, we first discuss this topic.

Let $\mathbf{x}_{\mathrm{c}}(t)$ denote the center of the translating low pressure system in Cartesian coordinates at time t. Then, $\mathbf{x}_{\mathrm{c}}(t)$ is given by

$$\mathbf{x}_{\mathrm{c}}(t) = \begin{bmatrix} x_{\mathrm{c}}(t) \\ y_{\mathrm{c}}(t) \\ z_{\mathrm{c}}(t) \end{bmatrix} = \begin{bmatrix} a\cos\left(\lambda_0 + \dfrac{u_0}{a}t\right)\cos(\theta_0) \\ a\sin\left(\lambda_0 + \dfrac{u_0}{a}t\right)\cos(\theta_0) \\ a\sin(\theta_0) \end{bmatrix},$$

where a is the radius of the earth, and (λ_0, θ_0) is the initial position of the low pressure system. Following the notation of [32], let $\overline{\psi}$ denote the stream function for the low pressure system at time t without the superimposed jet stream:

$$\overline{\psi}(\mathbf{x},t) = \psi_0 \exp\left[-\sigma \frac{1-\mathbf{x}^T\mathbf{x}_c(t)}{1+\mathbf{x}^T\mathbf{x}_c(t)}\right],$$

where ψ_0 and σ are constants defined below. With this definition the velocity field in Cartesian coordinates is given by

$$\widetilde{\mathbf{u}} = \begin{bmatrix}\widetilde{u}\\ \widetilde{v}\\ \widetilde{w}\end{bmatrix} = \underbrace{\begin{bmatrix}-u_0(2z)^{14}y(1-z^2)^{13/2}\\ u_0(2z)^{14}x(1-z^2)^{13/2}\\ 0\end{bmatrix}}_{\overline{\mathbf{u}}} + \frac{2\sigma\overline{\psi}(\mathbf{x},t)}{a(1+\mathbf{x}^T\mathbf{x}_c(t))^2}[\mathbf{x}\times\mathbf{x}_c(t)].$$

The first term in this expression is the velocity field for the jet stream with maximum velocity u_0, while the second term is the velocity field for the low-pressure system. The geopotential height is given by

$$(19)\quad g\widetilde{h} = \underbrace{gh_0 + \frac{2(2^{11}u_0)^2(1-z^2)^{14}}{35102025}p_1(z) + \frac{2^{15}au_0\Omega(1-z^2)^{15/2}}{145422675}p_2(z)}_{g\overline{h}} + f\overline{\psi}(\mathbf{x},t),$$

where $f = 2\Omega z$ is the Coriolis force and

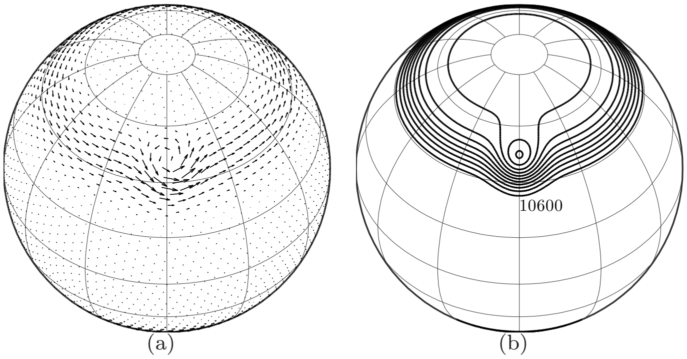
$$\begin{aligned}p_1(z) =& 20058300z^{28} + 10400600z^{26} + 5200300z^{24} + 2496144z^{22} + 1144066z^{20} +\\ & 497420z^{18} + 203490z^{16} + 77520z^{14} + 27132z^{12} + 8568z^{10} + 2380z^{8} +\\ & 560z^{6} + 105z^{4} + 14z^{2} + 1,\\ p_2(z) =& 5014575z^{12} + 2600150z^{10} + 542640z^{8} + 206720z^{6} + 65280z^{4} +\\ & 15360z^{2} + 2048.\end{aligned}$$

The constants are specified as follows: $(\theta_0, \lambda_0) = (\pi/4, 0)$, $\psi_0 = -0.03(gh_0/f_0)$, $\sigma = (12.74244)^2$, $gh_0 = 10^5$ m^2/s^2, $\Omega = 7.292 \times 10^{-5}$ s^{-1}, $f_0 = \sqrt{2}\Omega$, and the maximum velocity, u_0 is 20 m/s. Figure 16 (a) and (b) display the initial ($t = 0$) velocity and height fields, respectively. The solution at any time t looks identical to these figures, but with the center of the low shifted to $\lambda = (u_0/a)t$.

These values of $\widetilde{\mathbf{u}}$ and $\widetilde{h}$ are not analytical solutions of the shallow water equations (6a)–(6d), but can be made analytical solutions by adding forcing terms F_u, F_v, F_w, and F_h to the respective right hand sides of (6a)–(6d). The F_u forcing term for the u momentum equation (6a) is given by substituting $\widetilde{\mathbf{u}}$ and $\widetilde{h}$ into the right hand side of (6a) and then subtracting this from the time derivative of $\widetilde{u}$, i.e.

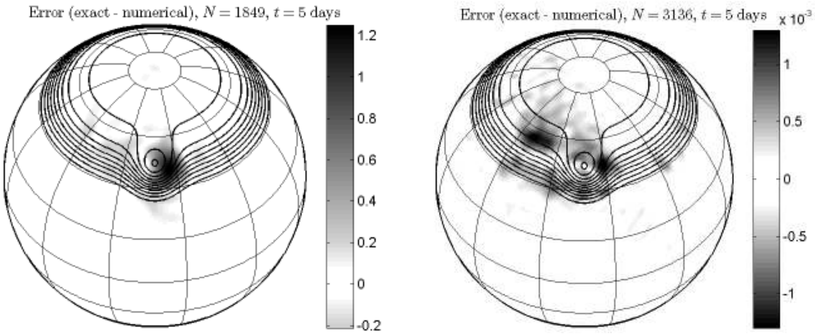
$$F_u(\mathbf{x},t) = \frac{\partial\widetilde{u}}{\partial t} + \mathbf{p}_x \cdot \begin{bmatrix}\widetilde{\mathbf{u}}\cdot\mathbf{P}\nabla\widetilde{u} + f(\mathbf{x}\times\widetilde{\mathbf{u}})\cdot\hat{\mathbf{i}} + g(\mathbf{p}_x\cdot\nabla)\widetilde{h}\\ \widetilde{\mathbf{u}}\cdot\mathbf{P}\nabla\widetilde{v} + f(\mathbf{x}\times\widetilde{\mathbf{u}})\cdot\hat{\mathbf{j}} + g(\mathbf{p}_y\cdot\nabla)\widetilde{h}\\ \widetilde{\mathbf{u}}\cdot\mathbf{P}\nabla\widetilde{w} + f(\mathbf{x}\times\widetilde{\mathbf{u}})\cdot\hat{\mathbf{j}} + g(\mathbf{p}_z\cdot\nabla)\widetilde{h}\end{bmatrix}$$

The forcing terms F_v, F_w, and F_h are similarly computed through the respective substitution of the analytical solutions $\widetilde{\mathbf{u}}$ and $\widetilde{h}$ into (6b)–(6d). As specified in [32], the complete forced shallow water system is to be simulated for 5 days. The standard fourth order Runge-Kutta (RK4) scheme is used to advance the RBF method in time. This was chosen over leapfrog with a Robert filter since the unsteady test case is much more sensitive to time truncation errors.



(a) (b)

FIGURE 16. Initial (a) velocity field and (b) height field with $N = 3136$ for the unsteady flow test case plotted as orthographic projections centered at 45°N and 0°E. The contours in (a) range from 10600 m to 10100 m in intervals of 50 m.



(a) $N = 1849$ (b) $N = 3136$

FIGURE 17. The error (exact - numerical) at $t = 5$ days for the height field from the unsteady flow test case. The results are for (a) $N = 1849$ and (b) $N = 3136$ nodes and $\varepsilon = 3.25$. Solid dark lines correspond to contours of the exact solution over the same values as Figure 16(b).

6.2.2. *Error study with regard to h-refinement.* Figure 17 displays the error (exact minus numerical) for the RBF solution of the height field for $N = 1849$ and $N = 3136$ nodes. It is clear that the largest errors are predominantly located where the gradients in the solution are the highest as would be expected. For the latter case ($N = 3136$), the time traces of the ℓ_1, ℓ_2, and ℓ_∞ error in the height field over the full 5 day integration period with a $\Delta t = 15$ minutes is given in Figure 18. The trend in the growth of the ℓ_1 error observed in this figure is similar to that reported in the SE method [30].

Figure 19 shows that the relative ℓ_2 error for various N barely grows as a function of time during the simulation period. To determine the convergence rate from these

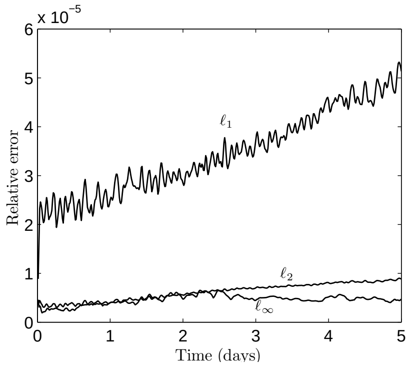


FIGURE 18. Relative ℓ_1, ℓ_2, and ℓ_∞ errors in the height field for the unsteady flow test as a function of time for the case of $N = 3136$, $\varepsilon = 3.25$, and $\Delta t = 15$minutes.

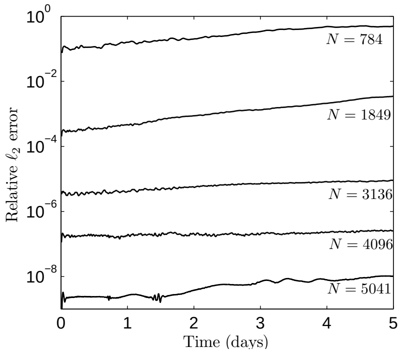


FIGURE 19. Relative ℓ_2 error in the approximate solution of the height field for the unsteady flow test case 3 as a function of time and different spatial resolutions. The shape parameter was fixed at $\varepsilon = 3.25$ for all N.

time traces, we plot the relative ℓ_2 error as a function of spatial resolution (i.e. $\sqrt{N}$) in Figure 20 and see that the RBF method is converging spectrally to the true solution of the height field. In contrast to Figure 8, Figure 20 shows that the relative ℓ_2 error is more sensitive to the value of ε, decreasing almost an order of magnitude as ε decreases from $\varepsilon = 4$ to 3.25 for $N = 5041$. Even for smaller values of N, such as $N = 3136$, we see this pattern as is illustrated by the time traces of the relative ℓ_2 error for varying ε values in Figure 21. As in the steady state test case, we again ask the question, "Will the error decrease as ε keeps decreasing?". We will again be employing the RBF-QR algorithm [11] to answer this question.

6.2.3. *Error and time stability study with regard to ε-refinement.* To study the accuracy and stability with respect to the shape parameter ε, we restrict our attention to the $N = 1849$ case. This is a larger node set than used in the previous ε-refinement study due to proper resolution of the translating low pressure system.

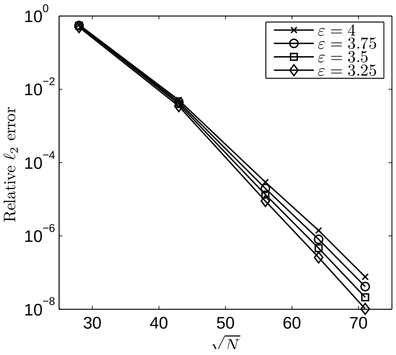


FIGURE 20. Relative ℓ_2 error in the approximate solution of the height field for the unsteady flow test case at $t = 5$ days for different spatial resolutions and values of the shape parameter ε.

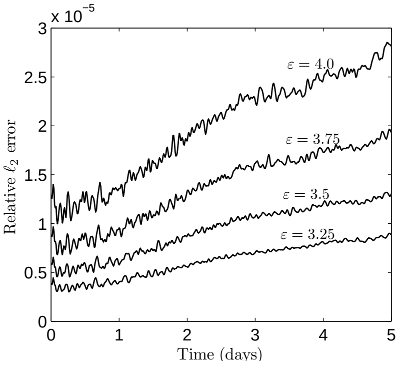
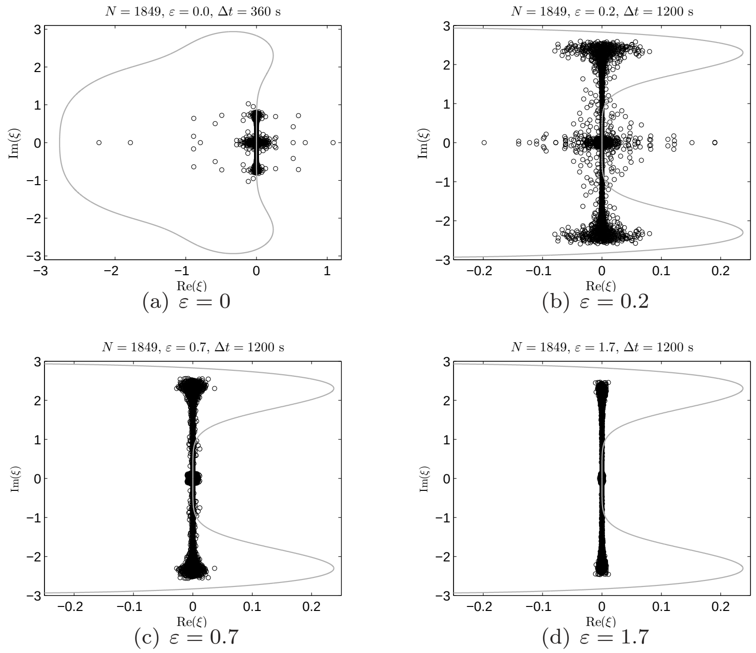


FIGURE 21. Relative ℓ_2 errors in the height field for the unsteady flow test case with $N = 3136$ as a function of time and different values of the shape parameter ε.

As in the previous test, we study time stability by analyzing the spectrum of the RBF operator for the linearized system (15). However, unlike the previous test case, the linearized system changes every time step since the analytical solution is not a steady state solution. In the results that follow, we will thus look at the spectrum for the case of $t = 0$. Although not demonstrated here, the spectrum for other values of t is very similar, which should be expected since the analytical solution at time t is just a translation of the $t = 0$ solution.

Figure 22(a)–(d) display the stability domain for RK4 along with the spectrum of the RBF operator for the linearized equations for $\varepsilon = 0, 0.2, 0.7, 1.7$, respectively. For the first three of these values of ε, the RBF-QR algorithm was used to compute the differentiation matrices, while for $\varepsilon = 1.7$ the direct approach of inverting the interpolation matrix A was used. Note that the $\varepsilon = 0$ spectrum has been scaled by $\Delta t = 360$ s, whereas the other spectrums have been scaled by $\Delta t = 1200$ s. This change was necessary since the eigenvalue with the largest negative real part for the $\varepsilon = 0$ case would not fit in the stability domain with $\Delta t = 1200$ s. We see a similar trend to the steady state test case, in that there is a terrible degradation



(a) $\varepsilon = 0$

(b) $\varepsilon = 0.2$

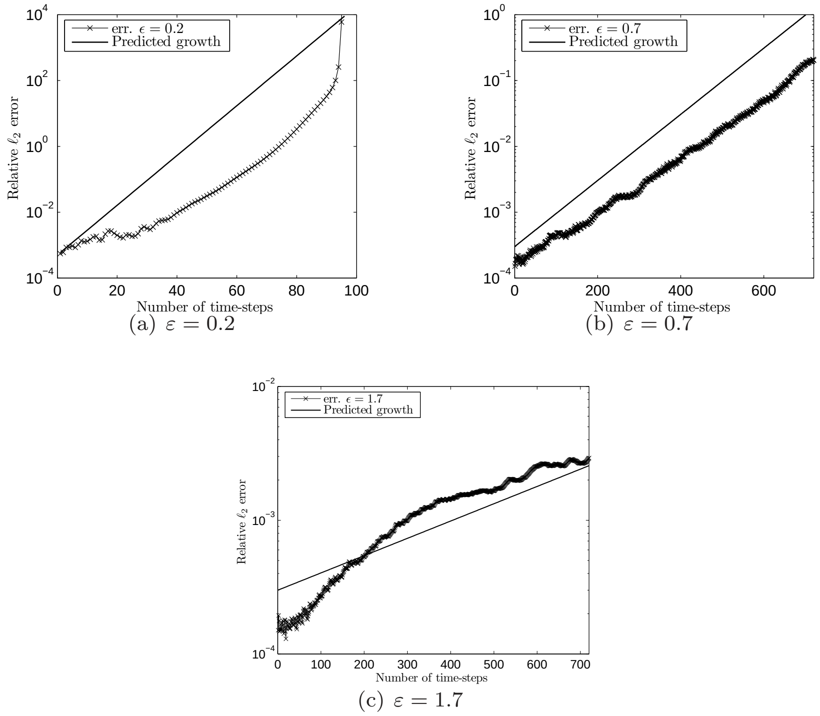
(c) $\varepsilon = 0.7$

(d) $\varepsilon = 1.7$

FIGURE 22. Spectrums of the RBF approximation operator of the right hand side of the linearized system in equations (15a)-(15d) for different values of ε. The spectrum of (a) has been scaled by the time step $\Delta t = 360$ s (6 min), while the spectrums in (b)–(d) have been scaled by $\Delta t = 1200$ s (20 min) with $\xi = \Delta t \times$ *eigenvalues*. The stability domain of the RK4 time stepping scheme, has also been plotted as light solid line in each figure.

of the eigenvalues into the right half plane both near the center and at the ends of the spectrum as ε is decreased. However, for the $\varepsilon = 1.7$ case the eigenvalues are tightly compacted around the imaginary axis, and do not spread far up the imaginary axis. This is the reason as to why RBFs give high accuracy with much longer time steps as compared to SE, which use orthogonal basis such as Legendre polynomials. Such bases have eigenvalues that are high up and low down in the left half plane, severely restricting the time steps [6, Section 4.4].

To see how well the linear eigenvalue theory predicts the error growth of the unstable modes in the model, we plot in Figure 23 (a)–(c) the predicted growth according to (18) together with the ℓ_2 error of the height field for $\varepsilon = 0.2$, $\varepsilon = 0.7$, and $\varepsilon = 1.7$. We do not include the results for $\varepsilon = 0$ since the solution blows up after only 8 time steps with $\Delta t = 360$s. Figure 23(a) can only be run for 100 time steps before the method goes completely unstable. All the plots show the eigenvalue analysis predicts the growth relatively accurately. Note that in the plots from (b) and (c), the x-axis ranges from 0 to 720 time steps, which is actually a *10 day integration period* instead of 5 days as specified in the test case. In the latter of these plots, we see the growth rate from 5 days (360 time steps) to 10 days is quite acceptable.



(a) $\varepsilon = 0.2$

(b) $\varepsilon = 0.7$

(c) $\varepsilon = 1.7$

FIGURE 23. Relative ℓ_2 error of the height field for the unsteady test case together with the predicted growth rate from equation 18. $\Delta t = 1200$ s (20 min) and $N = 1849$. The value of ε used is indicated below each figure. Figures (b) and (c) are run for 720 time steps, corresponding to 10 days of integration.

6.2.4. *Comparative results.* As with the previous test case, we compare the performance of the RBF method to several other high order methods in Table 2. From each study, the best results were chosen in terms of numerical accuracy. The ℓ_2 error is normalized with regard to the norm of the true height field without the mean term $\overline{h}$ in (19). The larger RBF time steps in Table 2 as opposed to Table 1 is due to using the RK4, for which the stability domain extends further up the imaginary axis than the leap frog scheme. In Table 2, the time step was chosen at the breakpoint where the temporal discretisation error matched the spatial discretisation error. For example, for the $N = 4096$ case the relative ℓ_2 error at the end of the 5 day run is graphed as a function of the time step in Figure 24. The RBF method could easily have taken a larger time step than 8 minutes (as reported in Table 2) but temporal discretisation errors would begin to dominate as shown by the large growth in the error after an 8 minute time step. Again, the RBF method uses no spatial filtering for the reported results.

7. Performance Benchmarks

In order to give the reader a feeling of the time requirements for running the RBF method we give some timing benchmarks in Table 3 for both test cases. Although

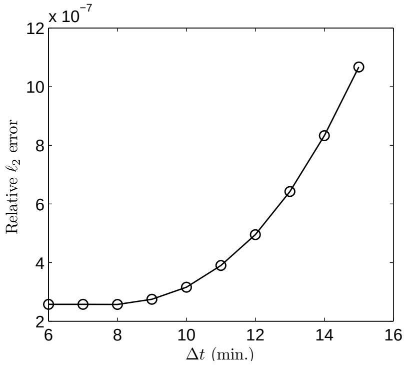


FIGURE 24. Relative ℓ_2 error in the height field at 5 days for the unsteady flow test case as a function of the time step for $\varepsilon = 3.25$ and $N = 4096$.

Method	Number of grid points	Time-step	Relative ℓ_2 error in h
RBF	784 (28)	40 minutes	4.88×10^{-1}
	1849 (43)	24 minutes	3.46×10^{-3}
	3136 (56)	15 minutes	8.83×10^{-6}
	4096 (64)	8 minutes	2.57×10^{-7}
	5041 (71)	6 minutes	1.02×10^{-8}
SH [21]	8192 (1849)	20 min.* (3)	2×10^{-3}
DF/SHF [29]	2048	6 minutes	3.9×10^{-1}
	8192	3 minutes	8.2×10^{-4}
	32768	90 seconds	4.0×10^{-4}
SE [30]	6144	90 seconds	6.5×10^{-3}
	24576	45 seconds	4×10^{-5}

TABLE 2. Comparison between commonly used spectral methods for the unsteady flow test case. The number in ( ) in the RBF section correspond to the square root of N, which is inversely proportional to the node spacing. The number in ( ) in the SH section correspond to the number of SH coefficients updated in time. RBF and DF use the same time-stepping scheme. SE uses a third-order Adams-Bashforth. SH uses a semi-implicit time-stepping scheme denoted by the *. For the SH 1849 case, [29] gives Δt=3 min. when using a leapfrog scheme as is done in RBF and DF.

RBF matrices are full and thus their inversion to calculate the 3 needed differentiation matrices is $O(N^3)$, this is a pre-processing step that is done only once. At every time step a matrix-vector multiply is needed, requiring $O(N^2)$ operations. Table 3 gives both the runtime per time-step as well as total runtime needed to achieve the given ℓ_2 error (or similarly as a function of N). All computations were performed in MATLAB Version 7.5 R2007b running on Dell PowerEdge 2950 Server

Test Case	N	Runtime per time-step	Total runtime	Time step	Relative ℓ_2 error in h
steady flow	784	0.0035 s	1.28 s	20 min.	6.32×10^{-6}
	1849	0.0191 s	11.5 s	12 min.	1.97×10^{-8}
	3136	0.0518 s	37.4 s	10 min.	3.65×10^{-10}
	4096	0.0885 s	80 s	8 min.	4.72×10^{-11}
	5041	0.131 s	157 s	6 min.	6.88×10^{-12}
unsteady flow	784	0.0299 s	5.5 s	40 min.	4.88×10^{-1}
	1849	0.110 s	33.3 s	24 min.	3.46×10^{-3}
	3136	0.247 s	118 s	15 min.	8.83×10^{-6}
	4096	0.409 s	368 s	8 min.	2.57×10^{-7}
	5041	0.600 s	1440 s	6 min.	1.02×10^{-8}

Table 3. Runtime results for the RBF method for a 5 day simulation of steady-state and unsteady forced flow. Time is given in seconds (s). See Section 7 for details.

with two 2.66GHz Intel Xeon X5335 quad-core processors. BLAS multi-threading in MATLAB was enabled with a maximum of 8 cores available.

The runtimes are greater for the unsteady flow test case than steady-state flow since (i) the RK4 method was used as opposed to a leapfrog scheme due to greater sensitivity in time truncation errors, and (ii) for unsteady flow, the forcing functions have to be evaluated each time step.

8. Summary and Future Prospects

The paper develops a stable spectrally convergent RBF method for solving the shallow water equations on a sphere and evaluates its convergence and time stability properties using test cases where analytical solutions are known so that an exact error study in time and space could be performed. Results are put into context with respect to those published in the literature for other commonly used spectral methods. The general findings with regard to the steady and unsteady flow tests considered are:

(1) h-refinement
For both test cases, *spectral convergence* is easily achieved. The error (exact-numerical) is concentrated in the region where the solution has the steepest gradients, outside of which little dispersion was seen. For all values of N tested, the ℓ_2 error barely grew as function of time over the 5 day simulation period.

(2) ε-refinement
Interestingly, in the limit as $\varepsilon \to 0$ (with $\varepsilon = 0$ being SH) time instability sets in. Performing a linearized eigenvalue stability analysis about the steady state solution of the RBF scheme, eigenvalues are noticed to spread off the imaginary axis and into the right half plane. However, away from this limit, eigenvalues are tightly clustered around the imaginary axis and near the origin in contrast to pseudospectral methods. This allows for *long*

time steps. In both test cases, the linear eigenvalue stability analysis of the RBF method proved to be a decent measure of error growth.

(3) Comparative study:
For the steady flow test case, the RBF method achieves a relative ℓ_2 error of 6.88×10^{-12} with $N = 5041$ nodes and a time step of 6 minutes using a leapfrog scheme with a Robert's filter of 0.07. The highest accuracy noted in the literature is $2(10^{-13})$ for a double Fourier method when used with a spherical harmonic filter with $N = 32768$ nodes and a time step of 90 seconds [29], also using a leapfrog scheme. For the unsteady test case, the RBF method achieved a relative ℓ_2 error of 1.02×10^{-8} with $N = 5041$ nodes and a time step of 6 minutes using the standard RK4. The highest accuracy noted in the literature is 4×10^{-5} for a spectral element method with $N = 24576$ nodes and a time step of 45 seconds [30].

The RBF node sets used in this study are for roughly uniform resolutions of approximately 500km or greater on the surface of the earth. However, to go to much higher resolutions (for example, on the order of 10km) and to do 3-D modeling requires many more nodes. Since global RBFs (as used in this paper) require full matrices, they are not a practical computing option for high resolution models in either 2- or 3-D, especially with regard to parallelization. In these cases, RBF methods, such as RBF finite differences, may give an alternate viable approach and are currently under development for a variety of applications.

References

1. E. W. Cheney and W. A. Light, *A course in approximation theory*, Brooks/Cole, New York, 2000.
2. T. A. Driscoll and B. Fornberg, *Interpolation in the limit of increasingly flat radial basis functions*, Comput. Math. Appl. **43** (2002), 413–422.
3. T. A. Driscoll and A. Heryundono, *Adaptive residual subsampling methods for radial basis function interpolation and collocation problems*, Comput. Math. Appl. **53** (2007), 927–939.
4. N. Flyer and E. Lehto, *Rotational transport on a sphere: Local node refinement with radial basis functions*, Theo. Comp. Fluid Dyn. **submitted** (2009).
5. N. Flyer and G. B. Wright, *Transport schemes on a sphere using radial basis functions*, J. Comp. Phys. **226** (2007), 1059–1084.
6. B. Fornberg, *A practical guide to pseudospectral methods*, Cambridge University Press, Cambridge, 1996.
7. B. Fornberg and N. Flyer, *Accuracy of radial basis function interpolation and derivative approximations on 1-*D *infinite grids*, Adv. Comput. Math. **23** (2005), 5–20.
8. B. Fornberg, N. Flyer, S. Hovde, and C. Piret, *Localization properties of rbf expansions for cardinal interpolation. i. equispaced nodes*, IMA J. Num. Analy. **28(1)** (2008), 121–142.
9. B. Fornberg, N. Flyer, and J.M. Russell, *Comparisons between pseudospectral and radial basis function derivative approximations*, IMA J. Num. Analy. **in press** (2009).
10. B. Fornberg and C. Piret, *A stable algorithm for flat radial basis functions on a sphere*, SIAM J. Sci. Comput. **200** (2007), 178–192.
11. ———, *On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere*, J. Comp. Phys. **227** (2008), 2758–2780.
12. B. Fornberg and G. Wright, *Stable computation of multiquadric interpolants for all values of the shape parameter*, Comput. Math. Appl. **48** (2004), 853–867.
13. B. Fornberg and J. Zuev, *The Runge phenomenon and spatially variable shape parameters in RBF interpolation*, Comput. Math. Appl. **54** (2007), 379–398.
14. Q. T. Lê Gia, *Galerkin approximation of elliptic* PDEs *on spheres*, J. Approx. Theory **130** (2004), 125–149.
15. ———, *Approximation of parabolic* PDEs *on spheres using spherical basis functions*, Adv. Comput. Math. **22** (2005), 377–397.

16. D. Gottlieb and S. A. Orszag, *Numerical analysis of spectral methods: Theory and applications*, SIAM, 1977.
17. R. L. Hardy, *Multiquadric equations of topograpy and other irregular surfaces*, J. Geophy. Res. **76** (1971), 1905–1915.
18. Y. C. Hon, K. F. Cheung, X. Z. Mao, and E. J. Kansa, *Multiquadric solution for shallow water equations*, ASCE J. Hydr. Engrg. **125** (1999), no. 5, 524–533.
19. Y. C. Hon and X. Z. Mao, *An efficient numerical scheme for Burgers' equation*, Appl. Math. Comput. **95** (1998), 37–50.
20. A. Iske, *Multiresolution methods in scattered data modelling*, Lecture Notes in Computational Science and Engineering, vol. 37, Springer-Verlag, Heidelberg, 2004.
21. R. Jakob-Chien, J. J. Hack, and D. L. Williamson, *Spectral transform solutions to the shallow water test set*, J. Comp. Phys. **119** (1995), 164–187.
22. Kurt Jetter, Joachim Stöckler, and Joseph D. Ward, *Error estimates for scattered data interpolation on spheres*, Math. Comput. **68** (1999), no. 226, 733–747.
23. E. J. Kansa, *Multiquadrics – a scattered data approximation scheme with applications to computational fluid-dynamics –* II*: Solutions to parabolic, hyperbolic and elliptic partial differential equations*, Comput. Math. Appl. **19** (1990), 147–161.
24. E. Larsson and B. Fornberg, *A numerical study of some radial basis function based solution methods for elliptic* PDEs, Comput. Math. Appl. **46** (2003), 891–902.
25. W. R. Madych and S. A. Nelson, *Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation*, J. Approx. Theory **70** (1992), 94–114.
26. John C. Mairhuber, *On Haar's theorem concerning Chebychev approximation problems having unique solutions*, Proc. Amer. Math. Soc. **7** (1956), 609–615.
27. M. J. D. Powell, *The theory of radial basis function approximation in 1990*, Advances in Numerical Analysis, Vol. II: Wavelets, Subdivision Algorithms and Radial Functions (W. Light, ed.), Oxford University Press, Oxford, UK, 1992, pp. 105–210.
28. Anton Sherwood, *How can I arrange N points evenly on a sphere?*, Website, 2007, http://www.ogre.nu/sphere.htm.
29. W. F. Spotz, M. A. Taylor, and P. N. Swarztrauber, *Fast shallow water equation solvers in latitude-longitude coordinates*, J. Comp. Phys. **145** (1998), 432–444.
30. M. Taylor, J. Tribbia, and M. Iskandarani, *The spectral element method for the shallow water equations on the sphere*, J. Comp. Phys. **130** (1997), 92–108.
31. J. Wertz, E. J. Kansa, and L. Ling, *The role of multiquadric shape parameters in solving elliptic partial differential equations*, Comput. Math. Appl. **51** (2006), 1335–1348.
32. D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, *A standard test set for numerical approximations to the shallow water equations in spherical geometry*, J. Comput. Phys. **102** (1992), 211–224.
33. R. S. Womersley and I. H. Sloan, *Interpolation and cubature on the sphere*, Website, 2003, http://web.maths.unsw.edu.au/∼rsw/Sphere/.
34. G. B. Wright and B. Fornberg, *Scattered node compact finite difference-type formulas generated from radial basis functions*, J. Comput. Phys. **212** (2006), 99–123.
35. J. Yoon, *Spectral approximation orders of radial basis function interpolation on the Sobolev space*, SIAM J. Math. Anal. **23** (2001), no. 4, 946–958.

Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, Boulder, CO 80305 USA

Department of Mathematics, Boise State University, Boise, ID 83725 USA
E-mail address: flyer@ucar.edu
E-mail address: wright@math.boisestate.edu