A radial basis function method for the shallow water equations on a sphere

BY NATASHA FLYER1,* AND GRADY B. WRIGHT2

1Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, Boulder, CO 80305, USA
2Department of Mathematics, Boise State University, Boise, ID 83725, USA

The paper derives the first known numerical shallow water model on the sphere using radial basis function (RBF) spatial discretization, 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 a global steady-state flow with a compactly supported velocity field, while the second is an unsteady flow where features in the flow must be kept intact without dispersion. This behaviour is achieved by introducing forcing terms in the shallow water equations. Error and time stability studies are performed, both as the number of nodes are 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.

Keywords: radial basis functions; hyperbolic partial differential equations; spherical geometry

1. Introduction

Radial basis functions (RBFs) have the advantage of achieving spectral accuracy in multidimensions for arbitrary node layouts with extreme algorithmic simplicity. For the purposes of interpolating multidimensional 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. Kansa 1990; Hon & Mao 1998; Hon et al. 1999; Larsson & Fornberg 2003; Wright & Fornberg 2006; Driscoll & Heryundono 2007). 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 (Le ˆ Gia 2004, 2005), with the first application to a purely hyperbolic operator done in 2007 (Flyer & Wright 2007). 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 this 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 sphere and, in doing so, gives the first application of RBFs to a system of coupled nonlinear PDEs on a sphere; (b) demonstrates and analyses why RBFs can take unusually long time steps; and (c) shows that RBFs give high accuracy when compared with 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 with other high-order spectrally accurate methods.

The methodology is tested on two cases in which the 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. non-zero in a limited band region), admitting a solution that is represented as an infinite spherical harmonic (SH) 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 centre that is superimposed on a jet stream. An overview of the paper is as follows: §2 gives an introduction to RBFs; §3 provides a simple derivation of the shallow water equations for the sphere in Cartesian coordinates; §4 derives the discrete RBF formulation of the shallow water equations that is used in the test cases; §5 derives the linearized shallow water equations that will be used in the time stability analysis; §6 are the numerical studies; §7 gives timed benchmarks and §8 summarizes the paper with future prospects.

2. Introduction to RBFs

The motivation of the RBF methodology originated with Hardy (1971) asking the question, ‘Given a set of sparse scattered data, \( \{ f_j \}_{j=1}^N \), at the node locations \( \{ x_j \}_{j=1}^N \subset \mathbb{R}^d \), can an interpolant be constructed that adequately represents the unknown surface?’ It was first shown by Mairhuber (1956) that, in more than one dimension, interpolation by an expansion of basis functions, \( \{ \psi_j(x) \}_{j=1}^N \), \( x \in \mathbb{R}^d \), which 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 centre 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.

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 $r = \|x - x_j\|$ is the Euclidean norm, has a jump in the third derivative, at $x = x_j$, leading to the fourth-order convergence in one dimension, with the order of convergence increasing as the dimension increases (cf. Powell 1992). On the other hand, the evidence strongly suggests that infinitely smooth RBFs, such as $r^{-1}$, will lead to spectral convergence (Madych & Nelson 1992; Yoon 2001), as is demonstrated in this paper. Note that the infinitely smooth RBFs depend on a shape parameter $\varepsilon$. It was first shown by Driscoll & Fornberg (2002) that, in one dimension, in the limit of $\varepsilon \to 0$ (i.e. flat RBFs), the RBF methodology reproduces pseudospectral (PS) methods 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 & Piret (2007) showed that in the limit of $\varepsilon \to 0$, RBFs reproduce SH 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 $\varepsilon$ (Larsson & Fornberg 2003). In both cases, the shifted RBFs in the expansion approach the constant 1 as is clearly seen in the case for $\varepsilon = 0.1$ in figure 1, which leads to ill-conditioning. The Contour–Padé algorithm (Fornberg & Wright 2004) 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 $\varepsilon$ (even $\varepsilon = 0$). Furthermore, Fornberg & Piret (2007) 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 $\varepsilon$ is decreased right to 0. We will implement the RBF-QR to study how the error and time stability vary as a function of the shape parameter $\varepsilon$ 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 (2.1) 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 \(\{f_j\}_{j=1}^N\) at the node locations \(\{x_k\}_{k=1}^N\), the RBF interpolant \(s(x)\) to the data is defined by

\[
s(x) = \sum_{k=1}^{N} c_k \phi(||x - x_k||),
\]

where the expansion coefficients, \(\{c_k\}_{k=1}^N\), are found by enforcing the collocation conditions such that the residual is 0 at the data locations. This is equivalent to solving the symmetric linear system of equations

\[
\begin{bmatrix}
\phi(||x_1 - x_1||) & \phi(||x_1 - x_2||) & \ldots & \phi(||x_1 - x_N||) \\
\phi(||x_2 - x_1||) & \phi(||x_2 - x_2||) & \ldots & \phi(||x_2 - x_N||) \\
\vdots & \vdots & \ddots & \vdots \\
\phi(||x_N - x_1||) & \phi(||x_N - x_2||) & \ldots & \phi(||x_N - x_N||)
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_N
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
\vdots \\
f_N
\end{bmatrix},
\]

where \(A\) is the interpolation matrix. An example of RBF interpolation in two dimensions is illustrated in figure 2. For RBFs such as the Gaussian (GA), inverse multiquadric (MQ) and inverse quadratic, the matrix in (2.2) is positive definite regardless of the distinct node locations and the dimension. For details on the well-posedness of (2.2) see Cheney & Light (2000, chs 12–16).

Studies have shown that if the shape parameter, \(\varepsilon\), is kept fixed throughout the domain (as will be done in the current study—variable shape parameter is needed when implementing local mesh refinement; Wertz et al. 2006; Driscoll & Heryundono 2007; Fornberg & Zuev 2007; Flyer & Lehto submitted), best results are achieved with roughly evenly distributed nodes (Iske 2004). Since only a maximum of 20 nodes can be evenly distributed on a sphere, there are a
A multitude of algorithms to define ‘even’ distribution for larger numbers of nodes, such as equal partitioned area, convex hull approaches and electrostatic repulsion (Sherwood 2007). Although any of these will suffice, we have decided to use an electrostatic repulsion or minimum 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 similar to the inverse of the square root of the number of nodes $N$ (Womersley & Sloan 2003), i.e. $h \sim 1/\sqrt{N}$. Thus, they are similar to a uniform discretization of the unit square. In figure 3, the distribution for 1849 nodes on the unit sphere is displayed. For infinitely smooth RBFs, Jetter et al. (1999) show that, provided the underlying function being interpolated is sufficiently smooth, RBF interpolants converge in the $L^\infty$ norm such as $h^{-1/2} e^{-c/4h}$ for some constant $c>0$ that depends on the RBF. For the ME node sets, convergence will thus proceed as $N^{1/4} e^{-c\sqrt{N}/4}$.

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

As was discussed in Flyer & Wright (2007), 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 $(\lambda, \theta)$, where $\lambda$ and $\theta$ are the respective longitudinal and latitudinal variables, singularities will occur at the poles $(\theta = \pm(\pi/2))$ simply due to the coordinate system. Unlike orthogonal expansions on a latitude–longitude grid, we can apply the surface gradient operator $\nabla_s = \left((1/\cos \theta)(\partial/\partial \lambda)\lambda + (\partial/\partial \theta)\theta\right)$ 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 Flyer & Wright (2007) and Fornberg & Piret (2008)). 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 $\hat{\theta}$ 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 three-dimensional Cartesian coordinate system for a rotating fluid are as follows:

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

(3.1)

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

(3.2)

where $f$ is the Coriolis force; $\nabla = \partial_x \hat{i} + \partial_y \hat{j} + \partial_z \hat{k}$ and $\mathbf{u} = u \hat{i} + v \hat{j} + w \hat{k}$ are the velocity vectors; $h$ is the geopotential height; and $\mathbf{x} = \{x, y, z\}^T$ represents 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 a three-dimensional 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 $\mathbf{P}$ as,

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

(3.3)

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

Next, the gradient operator appearing in (3.1) must be constrained so that when it is applied to a scalar, it produces a vector that is tangent to the unit sphere. This is done by replacing all occurrences of $\nabla$ in (3.1) with the operator

$$\mathbf{P} \nabla = [\mathbf{p}_x \cdot \nabla, \mathbf{p}_y \cdot \nabla, \mathbf{p}_z \cdot \nabla],$$

since $\mathbf{P}$ projects arbitrary three-dimensional Cartesian vectors onto a plane tangent to the unit sphere. The divergence operator appearing in (3.2) 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 (r.h.s.) of (3.1), with the modified differential operators, onto the corresponding $\hat{i}$, $\hat{j}$ and $\hat{k}$ directions. For example, in the case of the $u$ momentum equation, this is done by taking the dot product of the modified r.h.s. of (3.1) 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

\[
\frac{\partial u}{\partial t} = -p_x \cdot \text{r.h.s.} \left[ (u \cdot P\nabla)u + f(x \times u) \cdot \hat{i} + g(p_x \cdot \nabla)h \right],
\]

\[
\frac{\partial v}{\partial t} = -p_y \cdot \text{r.h.s.} \left[ (u \cdot P\nabla)v + f(x \times u) \cdot \hat{j} + g(p_y \cdot \nabla)h \right],
\]

\[
\frac{\partial w}{\partial t} = -p_z \cdot \text{r.h.s.} \left[ (u \cdot P\nabla)w + f(x \times u) \cdot \hat{k} + g(p_z \cdot \nabla)h \right],
\]

r.h.s.

Note that the only spatial operator that needs to be discretized is the projected gradient, \(P\nabla\), and its components, \(p_x \cdot \nabla\), \(p_y \cdot \nabla\), \(p_z \cdot \nabla\).

4. The RBF method on the sphere

In §4a–c, we develop the discrete operators that are necessary to spatially discretize the shallow water equations using RBFs. In §4a, we define how to formulate the projected gradient operator in RBFs. In §4b, we do a step-by-step construction of the three needed differentiation matrices, \(D_N^r\), \(D_N^q\) and \(D_N^s\). Finally, §4c demonstrates the entire RBF formulation of the shallow water equations to be implemented computationally.

(a) Formulation of the projected gradient operator in RBFs

Let \(x = \{x, y, z\}\) and \(x_k = \{x_k, y_k, z_k\}\) be points on the unit sphere, then the Euclidean distance from \(x\) to \(x_k\) is

\[
r(x) = \|x - x_k\| = \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2} = \sqrt{2(1 - x^T x_k)}. \quad (4.1)
\]

It is important to note that the distances are not great circle arcs measured along the surface, but are the Euclidean distances 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, but only the scalar distances between the nodes and the locations at which the RBFs are centred (in our study the RBFs centres and the nodes coincide, but this need not be the case).

Let \(\phi_k(r(x))\) be an RBF centred at \(x_k\). Using the chain rule, the gradient of \(\phi_k(r(x))\) is given by

\[
\nabla \phi_k(r(x)) = \begin{bmatrix}
\frac{\partial}{\partial x} \phi_k(r(x)) \\
\frac{\partial}{\partial y} \phi_k(r(x)) \\
\frac{\partial}{\partial z} \phi_k(r(x))
\end{bmatrix} = \begin{bmatrix}
(x - x_k) \\
y - y_k \\
(z - z_k)
\end{bmatrix} \frac{\phi_i'(r(x))}{r(x)} = \left(\frac{x - x_k}{r(x)}\right) \frac{\phi_i'(r(x))}{r(x)}, \quad (4.2)
\]

where we have used \(^{\prime}\) to denote differentiation with respect to \(r\). The projected RBF gradient operator is then simply obtained by applying the projection matrix \(P=I-xx^T\) to (4.2)

\[
P\nabla\phi_k(r(x)) = P(x-x_k)\frac{\phi'_{k}(r(x))}{r(x)} = -Px_k\frac{\phi'_{k}(r(x))}{r(x)} = \begin{bmatrix} x\nabla x_k-x_k \\ y\nabla x_k-y_k \\ z\nabla x_k-z_k \end{bmatrix} \frac{\phi'_{k}(r(x))}{r(x)}.\]

Now, we have all the components necessary to build the action of \(P\nabla = [p_x \cdot \nabla, p_y \cdot \nabla, p_z \cdot \nabla]^T\) on an RBF representation of a geophysical quantity.

**(b) Constructing the RBF differentiation matrices**

Given a geophysical quantity \(f(x) = f(x, y, z)\) known at the node locations \(\{x_j\}_{j=1}^N\) on the surface of a unit sphere, we first represent \(f(x)\) as an RBF expansion

\[
f(x) = \sum_{k=1}^{N} c_k \phi_k(r(x)),\]

where the coefficients \(c_k\) are determined by collocation. Applying the projected gradient operator in the \(x\) direction to (4.3) and evaluating it at \(\{x_j\}_{j=1}^N\) gives

\[
\begin{align*}
[p_x \cdot \nabla f(x)] |_{x=x_j} &= \sum_{k=1}^{N} c_k [p_x \cdot \nabla \phi_k(r(x))] |_{x=x_j}, \quad j = 1, \ldots, N \\
&= \sum_{k=1}^{N} c_k \left[ x_j x_j^T x_k - x_k \right] \frac{\phi'_{k}(r(x_j))}{r(x_j)}, \quad j = 1, \ldots, N \\
&= B^x c = (B^x A^{-1}) f = D^x_N f,
\end{align*}
\]

where \(A\) is the interpolation matrix in (2.2) and we have used the property that \(c = A^{-1} f\). The differentiation matrix \(D^x_N\) is the discrete RBF approximation to the \(x\)-component of the projected gradient operator. Using the same concept, \(D^y_N\) and \(D^z_N\) are derived as follows:

\[
\begin{align*}
[p_y \cdot \nabla f(x)] |_{x=x_j} &= (B^y A^{-1}) f = D^y_N f, \\
[p_z \cdot \nabla f(x)] |_{x=x_j} &= (B^z A^{-1}) f = D^z_N f,
\end{align*}
\]

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

\[
B^y_{j,k} = [y_j x_j^T x_k - y_k] \frac{\phi'_{k}(r_k(x_j))}{r_k(x_j)}, \quad B^z_{j,k} = [z_j x_j^T x_k - z_k] \frac{\phi'_{k}(r_k(x_j))}{r_k(x_j)}.
\]

Note that all three differentiation operators (4.4)–(4.6) are well defined with no pole singularities.

**(c) The discretized RBF shallow water equations**

Let \(h\) be the geopotential height and \(\{u, v, w\}\) be the components of the velocity field \(u\) sampled at the node locations \(\{x_j\}_{j=1}^N = \{x, y, z\}\). Then, the r.h.s. of the Cartesian form of the shallow water equations in §3 can now be written in

discretized form r.h.s., \( D \) as

\[
\text{r.h.s., } D = \begin{bmatrix}
    u \cdot D_{x}^{y} u + v \cdot D_{x}^{y} u + w \cdot D_{x}^{y} u \\
    u \cdot D_{x}^{y} v + v \cdot D_{x}^{y} v + w \cdot D_{x}^{y} v + f \left( \frac{z}{u} - x \cdot w \right) + g \left( D_{x}^{y} h \right)
\end{bmatrix},
\]

where \( \circ \) denotes element by element multiplication of the vectors. The full discretized equations are given by

\[
\frac{\partial u}{\partial t} = -p_{x} \cdot \text{r.h.s., } D, \quad \frac{\partial v}{\partial t} = -p_{y} \cdot \text{r.h.s., } D, \quad \frac{\partial w}{\partial t} = -p_{z} \cdot \text{r.h.s., } D, \quad (4.7)
\]

\[
\frac{\partial h}{\partial t} = u \cdot D_{x}^{y} h + v \cdot D_{x}^{y} h + w \cdot D_{x}^{y} h + h \cdot \left( D_{x}^{y} u + D_{x}^{y} v + D_{x}^{y} w \right),
\]

where \( p_{x}, p_{y}, p_{z} \) are the vectors of \( p_{x}, p_{y}, p_{z} \) evaluated at the nodes \( \{ x_{j} \}_{j=1}^{N} \).

The method-of-lines 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 \( x \) and \( x_{j} \) as points on the unit sphere and divide \( D_{x}^{y}, D_{x}^{y} \) and \( D_{x}^{y} \) by \( a \).

5. Eigenvalue stability: the linearized equations

For each test case, we will perform an eigenvalue stability analysis in order to understand: (i) why the RBF method can take long time steps and (ii) the impact of \( \varepsilon \)-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 \( \tilde{u} \) and \( \tilde{h} \) are the approximate solutions to (3.4a) and (3.4b) and can be written as

\[
\tilde{u} = u_{0} + \delta u_{1} + O(\delta^{2}), \quad \tilde{h} = h_{0} + \delta h_{1} + O(\delta^{2}),
\]

where \( u_{0} \) and \( h_{0} \) are the initial conditions. Substituting this expansion into (3.4a) and (3.4b) and only considering perturbations of \( O(\delta) \) gives the linearized equations

\[
\frac{\partial u_{1}}{\partial t} = -p_{x} \cdot \begin{bmatrix}
    u_{1} \cdot P \nabla u_{0} + u_{0} \cdot P \nabla u_{1} + f(yw_{1} - zw_{1}) + gp_{x} \cdot \nabla h_{1}
\end{bmatrix}, \quad (5.1a)
\]

\[
\begin{bmatrix}
    u_{1} \cdot P \nabla v_{0} + u_{0} \cdot P \nabla v_{1} + f(zu_{1} - xw_{1}) + gp_{y} \cdot \nabla h_{1}
\end{bmatrix}, \quad (5.1b)
\]

\[
\begin{bmatrix}
    u_{1} \cdot P \nabla w_{0} + u_{0} \cdot P \nabla w_{1} + f(xw_{1} - yu_{1}) + gp_{z} \cdot \nabla h_{1}
\end{bmatrix}, \quad (5.1c)
\]

\[
\frac{\partial h_{1}}{\partial t} = -(u_{1} \cdot P \nabla h_{0} + u_{0} \cdot P \nabla h_{1} + h_{1} P \nabla \cdot u_{0} + h_{0} P \nabla \cdot u_{1}). \quad (5.1d)
\]

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 $\varepsilon$ ($\varepsilon$-refinement). We also perform a stability/eigenvalue study to explore how the time errors behave as a function of $\varepsilon$ and the time step. Also, how well linear stability analysis predicts the growth of unstable modes in the full RBF solution is explored. Finally, the performance of the RBF method is put into context with the following commonly used spectral models in the literature: a de-aliased SH method (Jakob-Chien et al. 1995), a double Fourier with a SH filter (DF/SHF; Spotz et al. 1998) and a spectral element (SE) method with filtering (Taylor et al. 1997). The RBF method does not need to use any spatial filtering to achieve the numerical accuracies reported.

(a) 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 Williamson et al. (1992), test case 3). For simplicity, the initial condition and solution for all time for the wind field in spherical coordinates is given by

$$u_{\text{sph}}(\theta) = u_0 \exp \left[ \frac{(\theta_e + \theta_b - 2\theta)^2}{\sigma^2(\theta_e - \theta)(\theta_e - \theta_b)} \right], \quad \theta_b < \theta < \theta_e,$$

where $\sigma = \sqrt{0.3}$, $\theta_b = -\pi/6$, $\theta_e = \pi/2$ and $u_0 = 2\pi a/12$ days. $u_{\text{sph}}(\theta)$ has a maximum at $\theta = (\theta_e + \theta_b)/2$, corresponding to a maximum wind positioned at $30^\circ$ N ($\pi/6$) and is 0 outside of $\theta_b < \theta < \theta_e$. The latitudinal component of the velocity field is 0. The analytic solution is then defined by a rotation of the coordinate system to tilt the wind field an angle $\alpha$ relative to the polar axis. For $\alpha = \pi/3$, the wind field is graphed in figure 4a as an orthographic projection centred at $60^\circ$ S and $0^\circ$ E. The geopotential height field is defined by numerically integrating to machine precision

$$\frac{\partial g h}{\partial \theta} = -(afu + u^2 \tan \theta),$$

where $f = 2\Omega \sin \theta$, $\Omega = 7.292 \times 10^{-5}$ s$^{-1}$ and $a = 6.37122 \times 10^6$ m. The geopotential height field is graphed in figure 4b. The test case is run for 5 days.

(i) Error study with regard to h-refinement

Figure 5 displays the error (exact minus numerical) in the RBF solution of the geopotential height field on an unrolled sphere for $N=1849$ (figure 5a) and $N=3136$ (figure 5b), respectively. The contour lines represent the exact solution in 100 m intervals, beginning at 3000 m. In figure 5a, all deviations in the solution of less than $10^{-4}$ m are shown in white, while in figure 5b 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 Flyer & Lehto (submitted).
Furthermore, note that there is little evidence of any dispersive wave trains in either plots, indicating that the r.h.s. of the PDEs in (4.7) are being adequately resolved.

Figure 6 shows the time traces of the relative $L^2$ error for the 5 day simulation for various values of $N$ and fixed $\varepsilon = 3.25$. Note that the error for all values of $N$ barely grows as a function of time during the simulation period. To determine 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 6b and see that indeed it is spectral. The figure also demonstrates that the $L^2$ error is relatively insensitive to the value of $\varepsilon$ used between 3.25 and 4. For larger nodes sets, if $\varepsilon$ goes below 3.25 ill-conditioning will set in. More on the sensitivity of the calculation to $\varepsilon$ will be addressed in §6a(ii) on $\varepsilon$-refinement.

Figure 7 shows the relative $L^1$, $L^2$ and $L^\infty$ norms of the error for the RBF method as a function of time for $N=3136$ and $\Delta t=10$ min. This spatial resolution was selected since it results in a relative $L^2$ error of $O(10^{-10})$, which is also the order of the lowest relative $L^2$ error reported by many of the other spectral methods (table 1). For this time step and node set, figure 8 shows the time traces as a function of $\varepsilon$. One can see that as $\varepsilon$ 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 (Fornberg & Piret 2007), which bypasses this ill-conditioning problem and allows for calculations in the limit as $\varepsilon \to 0$. Will we see the error drastically decrease?’

(ii) Error and time stability study with regard to $\varepsilon$-refinement

Our objective is to explore the accuracy and stability of the RBF shallow water method as the shape parameter $\varepsilon$ decreases to zero. Since the solution to this test case can be accurately resolved with a low-resolution model, as demonstrated in §6a(i), we fix the node set at $N = 484$ for the study. To compute the RBF differentiation matrices for this resolution in the low $\varepsilon$ regime (i.e. $0 \leq \varepsilon \leq 0.7$ for $N = 484$), we use the RBF-QR algorithm.

Figure 9 shows time traces for the 5 day run of the relative $\ell_2$ error for $\varepsilon$ varying from 0 to 0.7. Leapfrog with a Robert filter of $\gamma = 0.07$ and a time step of 18 min was used. Clearly, time instability sets in as $\varepsilon$ tends to zero, with the worse case being $\varepsilon = 0$ corresponding to SH 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 SH. The result is singular or close to singular SH 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 $\varepsilon$ is not in the $\varepsilon \to 0$ limit. However, one possibility as to why instability sets in as $\varepsilon$ becomes small is that we are not using a small enough time step. However, figure 10 easily throws that notion aside as we see that whether the time step is 2, 6 or 18 min, the blow-up of the $\ell_2$ 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 r.h.s. of the linearized system (5.1a)–(5.1d) to be contained within the stability domain of the time-stepping

\[ Z \]

\[ \text{(a)} \]

\[ \text{(b)} \]

Figure 6. (a) Relative $\ell_2$ error in the height field for the steady-state test case as a function of time and $N$. $\varepsilon = 3.25$ for all $N$. (b) Relative $\ell_2$ error in the height field for the steady-state test case at $t = 5$ days as a function $\sqrt{N}$ and $\varepsilon$. $\varepsilon = 4$, cross; $\varepsilon = 3.75$, circles; $\varepsilon = 3.5$, squares; $\varepsilon = 3.25$, diamonds.
scheme. Figure 11 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 $\varepsilon=0, 0.2$ and 0.7. The most important trend to note is that as $\varepsilon$ decreases the real part of the eigenvalues spread off the imaginary axis and into the right half plane both near the centre and at the ends of the spectrum. As was noted in figure 9, the worst case is for $\varepsilon=0$ when RBFs reproduce SH. However, if we look at the eigenvector associated with the eigenvalue having the largest real part in this case, as displayed in figure 12, we see such modes are spurious and not physical. As is well known in the SH

![Figure 7](image_url)  
Figure 7. Relative $\ell_1$, $\ell_2$ and $\ell_\infty$ errors in the height field for the steady-state test case as a function of time for $N=3136$, $\varepsilon=3.25$ and $\Delta t=10$ min.

Table 1. Comparison of commonly used spectral methods for steady-state flow with $\alpha=\pi/3$. (The number in parentheses in the RBF section corresponds to the square root of $N$, which is inversely proportional to the node spacing. The number in parentheses in the SH section corresponds 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 method. SH uses a semi-implicit time-stepping scheme denoted by the asterisks. For the SH 1849 case (Spotz et al. 1998), gives $\Delta t=3$ min when using a leapfrog scheme as is done in RBF and DF.)

<table>
<thead>
<tr>
<th>method</th>
<th>no. of nodes ($N$)</th>
<th>time step ($\Delta t$)</th>
<th>relative $\ell_2$ error in $h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF</td>
<td>784 (28)</td>
<td>20 min</td>
<td>$6.32\times10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>1849 (43)</td>
<td>12 min</td>
<td>$1.97\times10^{-8}$</td>
</tr>
<tr>
<td></td>
<td>3136 (56)</td>
<td>10 min</td>
<td>$3.65\times10^{-10}$</td>
</tr>
<tr>
<td></td>
<td>4096 (64)</td>
<td>8 min</td>
<td>$4.72\times10^{-11}$</td>
</tr>
<tr>
<td></td>
<td>5041 (71)</td>
<td>6 min</td>
<td>$6.88\times10^{-12}$</td>
</tr>
<tr>
<td>SH; Jakob-Chien et al. (1995)</td>
<td>8192 (1849)</td>
<td>20 min* (3)</td>
<td>$7\times10^{-10}$</td>
</tr>
<tr>
<td></td>
<td>18432 (4096)</td>
<td>15 min*</td>
<td>$2.5\times10^{-10}$</td>
</tr>
<tr>
<td>DF/SHF; Spotz et al. (1998)</td>
<td>2048</td>
<td>6 min</td>
<td>$2\times10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>8192</td>
<td>3 min</td>
<td>$4\times10^{-10}$</td>
</tr>
<tr>
<td></td>
<td>32768</td>
<td>90 s</td>
<td>$2\times10^{-13}$</td>
</tr>
<tr>
<td>SE; Taylor et al. (1997)</td>
<td>6144</td>
<td>90 s</td>
<td>$8\times10^{-7}$</td>
</tr>
<tr>
<td></td>
<td>24576</td>
<td>45 s</td>
<td>$1\times10^{-10}$</td>
</tr>
</tbody>
</table>
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 11 do not come into play for time integration periods in 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 is a good predictor, then the error in the RBF solution for a given $\varepsilon$ and $\Delta t$ should grow as

$$\text{error} \sim (1 + \text{Re}(\beta))^k,$$

where $\beta$ denotes the eigenvalue of the linearized RBF operator with the largest real part, scaled by $\Delta t$, which is not inside the time stability domain and $k$ is the number of time steps taken. In figure 13a–c, we plot this predicted growth rate together with the

---

**Figure 8.** Relative $l_2$ errors in the height field for the steady-state test case as a function of time and different values of $\varepsilon$.

**Figure 9.** Relative $l_2$ error in the height field for steady-state test case. The different curves correspond to different values of $\varepsilon$ 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.
error in the RBF solution for the height field for the same three values of $\varepsilon$ used in the previous figures ($\varepsilon = 0, 0.2, 0.7$). Recall that for the first two of these $\varepsilon$ values the solution went unstable well before $t = 5$ days, while for the last case the solution was stable and accurate. In figure 13a, we see the predicted growth matches the observed growth very nicely. In figures 13b,c, the linear stability theory actually predicts a slightly more elevated result than what is observed.

(iii) Discussion of spectral models and comparison

The methods presented in tables 1 and 2 are described below (Jakob-Chien et al. 1995; Taylor et al. 1997; Spotz et al. 1998).

(i) SH requires twice as many grid points as basis functions. Owing to a severe Courant–Friedrichs–Lewy (CFL) condition, a semi-implicit time stepping is commonly used (Jakob-Chien et al. 1995). 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 (Gottlieb & Orszag 1977). 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, where $M$ 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. Fast Fourier transforms (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.

(ii) 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
Figure 11. Spectrums of the RBF approximation operator of the r.h.s. of the linearized system in equations (5.1a) and (5.1b) for different values of $\varepsilon$. Each spectrum from (a) to (c) has been scaled by the time step $\Delta t=1080\text{ s} (18\text{ min})$, where $\xi=\Delta t\times$eigenvalues. The stability domain of time-stepping scheme, leapfrog scheme with a Robert filter of $\gamma=0.07$, is plotted as a grey line. (a) $\varepsilon=0$; (b) $\varepsilon=0.2$; and (c) $\varepsilon=0.7$. For all figures $N=484$.

Figure 12. The unit eigenvector for the height field corresponding the eigenvalue with the largest real part in figure 11a.
DF using 8192 grid points (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. Owing to FFTs, DF has only an operation count of $O(N \log N)$ per time step.

(iii) 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; Taylor et al. 1997) uses a tensor product of Legendre polynomials on each element. Owing 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 $P$ is the order of the polynomial expansion on each element.

(iv) 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 1–3 BLAS operations. Owing to close clustering of the eigenvalues of

---

Figure 13. Relative $\ell_2$ error in the RBF solution of the geopotential height for the steady-state test case together with the predicted growth rate (solid line) from equation (4.5). The value of $\varepsilon$ (cross) is as follows: (a) $\varepsilon = 0$; (b) $\varepsilon = 0.2$; and (c) $\varepsilon = 0.7$. For all figures $\Delta t = 1080$ s and $N = 484$. 

DF using 8192 grid points (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. Owing to FFTs, DF has only an operation count of $O(N \log N)$ per time step.

(iii) 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; Taylor et al. 1997) uses a tensor product of Legendre polynomials on each element. Owing 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 $P$ is the order of the polynomial expansion on each element.

(iv) 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 1–3 BLAS operations. Owing to close clustering of the eigenvalues of

---

the linearized RBF operator about the imaginary axis as seen in §6a(ii), RBFs can take comparatively much longer time steps (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 (Fornberg et al. 2008) and the fact that they only pick up information locally (Fornberg & Flyer 2005; Fornberg et al. in press) contributes to this phenomena. Furthermore, for a given spatial resolution, RBFs achieve a higher numerical accuracy (table 1). To implement global RBFs, as done in this paper, $O(N^2)$ operations are used. 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 $\epsilon=3.25$ (although any infinitely smooth RBFs can be used, but GA tends to be more sensitive to the value of $\epsilon$). Leapfrog 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 (6.1) so that it flows directly over the poles. For all $N$ in table 1, RBFs can easily take longer time steps with regard to stability but the time step was chosen so that temporal discretization errors matched spatial discretization errors, which is the optimal scenario. The RBF case is the only method that does not use a de-aliasing/spatial filter.

(b) Unsteady flow test case

The test case is a low-pressure system, initially centred at $(\lambda_0, \theta_0)=(0, \pi/4)$, superimposed on a jet stream that is symmetrical about the equator (see test case 4 in Williamson et al. (1992)). 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.

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 \( x_c(t) \) denote the centre of the translating low-pressure system in Cartesian coordinates at time \( t \). Then, \( x_c(t) \) is given by

\[
\begin{bmatrix}
  x_c(t) \\
  y_c(t) \\
  z_c(t)
\end{bmatrix} = \begin{bmatrix}
  a \cos \left( \frac{\lambda_0 + u_0}{a} t \right) \cos(\theta_0) \\
  a \sin \left( \frac{\lambda_0 + u_0}{a} t \right) \cos(\theta_0) \\
  a \sin(\theta_0)
\end{bmatrix},
\]

where \( a \) is the radius of the Earth and \((\lambda_0, \theta_0)\) is the initial position of the low-pressure system. Following the notation of Williamson et al. (1992), let \( \psi \) denote the stream function for the low-pressure system at time \( t \) without the superimposed jet stream

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

where \( \psi_0 \) and \( \sigma \) are constants defined below. With this definition, the velocity field in Cartesian coordinates is given by

\[
\hat{u} = \begin{bmatrix}
  u \\
  v \\
  w
\end{bmatrix} = \begin{bmatrix}
  -u_0 (2z)^{14} \left( 1 - z^2 \right)^{13/2} \\
  u_0 (2z)^{14} x \left( 1 - z^2 \right)^{13/2} \\
  0
\end{bmatrix} + \frac{2\sigma \psi(x, t)}{a(1 + x^T x_c(t))^2} [x \times x_c(t)].
\]

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

\[
g\hat{h} = g h_0 + \frac{2(2^{11} u_0)^2 (1 - z^2)^{14}}{35 102 025} p_1(z) + \frac{2^{15} a u_0 \Omega (1 - z^2)^{15/2}}{145 422 675} p_2(z) + f \psi(x, t),
\]

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

\[
p_1(z) = 20 058 300 z^{28} + 10 400 600 z^{26} + 5 200 300 z^{24} + 2 496 144 z^{22} + 1 144 066 z^{20} + 497 420 z^{18} + 203 490 z^{16} + 77 520 z^{14} + 27 132 z^{12} + 8568 z^{10} + 2380 z^8 + 560 z^6 + 105 z^4 + 14 z^2 + 1,
\]

\[
p_2(z) = 5 014 575 z^{12} + 2 600 150 z^{10} + 542 640 z^8 + 206 720 z^6 + 65 280 z^4 + 15 360 z^2 + 2048.
\]
The constants are specified as follows: \((\theta_0, \lambda_0) = (\pi/4, 0); \psi_0 = -0.03(gh_0/f_0); \)
\(\sigma = (12.74244)^2; \quad gh_0 = 10^5 \text{ m}^2 \text{ s}^{-2}; \quad \Omega = 7.292 \times 10^{-5} \text{ s}^{-1}; \quad f_0 = \sqrt{2}\Omega; \quad \) and the maximum velocity, \(u_0\) is 20 m s\(^{-1}\). Figure 14\(a,b\) display the initial \((t=0)\) velocity and height fields, respectively. The solution at any time \(t\) looks identical to these figures, but with the centre of the low-pressure system shifted to \(\lambda = (u_0/a)t\).

These values of \(\hat{u}\) and \(\hat{h}\) are not analytical solutions of the shallow water equations (3.4a) and (3.4b), but can be made analytical solutions by adding forcing terms \(F_u, F_v, F_w\) and \(F_h\) to the respective r.h.s.s of (3.4a) and (3.4b). The \(F_u\) forcing term for the \(u\) momentum equation (3.4a) is given by substituting \(\hat{u}\) and \(\hat{h}\) into the r.h.s. of (3.4a) and then subtracting this from the time derivative of \(\hat{u}\), i.e.

\[
F_u(x, t) = \frac{\partial \hat{u}}{\partial t} + \mathbf{p}_x \cdot \left[ \hat{u} \cdot \mathbf{P} \nabla \hat{u} + f(x \times \hat{u}) \cdot \hat{i} + g(p_x \cdot \nabla) \hat{h} \right] \\
= \frac{\partial \hat{u}}{\partial t} + \mathbf{p}_x \cdot \left[ \hat{u} \cdot \mathbf{P} \nabla \hat{v} + f(x \times \hat{u}) \cdot \hat{j} + g(p_y \cdot \nabla) \hat{h} \right] \\
= \frac{\partial \hat{u}}{\partial t} + \mathbf{p}_x \cdot \left[ \hat{u} \cdot \mathbf{P} \nabla \hat{w} + f(x \times \hat{u}) \cdot \hat{k} + g(p_z \cdot \nabla) \hat{h} \right]
\]

The forcing terms \(F_v, F_w\) and \(F_h\) are similarly computed through the respective substitution of the analytical solutions \(\hat{u}\) and \(\hat{h}\) into (3.4a) and (3.4b). As specified in Williamson et al. (1992), 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.

(ii) Error study with regard to \(h\)-refinement

Figure 15 displays the error (exact minus numerical) for the RBF solution of the height field for \(N=1849\) and 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 \(\ell_1, \ell_2\) and \(\ell_\infty\) error in the height field over the full 5 day integration period with a \(\Delta t=15\) min are given in figure 16. The trend in the growth of the \(\ell_1\) error observed in this figure is similar to that reported in the SE method (Taylor et al. 1997).

Figure 17a 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 time traces, we plot the relative $l_2$ error as a function of spatial resolution (i.e. $\sqrt{N}$) in figure 17b and see that the RBF method is converging spectrally to the true solution of the height field. In contrast to figure 6b, figure 17b shows that the relative $l_2$ error is more sensitive to the value of $\varepsilon$, decreasing almost an order of magnitude as $\varepsilon$ 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 $\varepsilon$ values in figure 18. As in the steady-state test case, we again ask the question, ‘Will the error decrease as $\varepsilon$ keeps decreasing?’. We will again be employing the RBF-QR algorithm (Fornberg & Piret 2008) to answer this question.

(iii) Error and time stability study with regard to $\varepsilon$-refinement

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

As in the previous test, we study time stability by analysing the spectrum of the RBF operator for the linearized system (5.1a)–(5.1d). 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 19a–d displays the stability domain for RK4 along with the spectrum of the RBF operator for the linearized equations for $\varepsilon = 0$, 0.2, 0.7 and 1.7, respectively. For the first three of these values of $\varepsilon$, 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 of the eigenvalues into the right half plane both near the centre and at the ends of the spectrum as $\varepsilon$ 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 why RBFs give high accuracy with much longer time steps when compared with 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 (Fornberg 1996, §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 20a–c the predicted growth according to (6.3) together with the $\ell_2$ error of the height field for $\varepsilon = 0.2$, $\varepsilon = 0.25$, $\varepsilon = 0.3$, $\varepsilon = 0.35$, and $\varepsilon = 0.4$. Figure 16. Relative $\ell_1$, $\ell_2$ and $\ell_\infty$ 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 $\Delta t = 15$ min.

Figure 16. Relative $\ell_1$, $\ell_2$ and $\ell_\infty$ 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 $\Delta t = 15$ min.

Figure 17. (a) Relative $\ell_2$ 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 $N$ values. (b) Relative $\ell_2$ 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 $\varepsilon$. $\varepsilon = 4$, cross; $\varepsilon = 3.75$, circle; $\varepsilon = 3.5$, squares; $\varepsilon = 3.25$, diamonds.
0.7 and 1.7. We do not include the results for $\varepsilon = 0$ since the solution blows up after only eight time steps with $\Delta t = 360$ s. Figure 20a can only be run for 100 time steps before the method goes completely unstable. All the plots show the eigenvalue analysis that predicts the growth relatively accurately. Note that in the plots from figure 20b,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.

(iv) 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 $\ell_2$ error is normalized with regard to the norm of the true height field without the mean term $h$ in (6.4). The larger RBF time steps in table 2 as opposed to table 1 are due to using the RK4, for which the stability domain extends further up the imaginary axis than the leapfrog scheme. In table 2, the time step was chosen at the break point where the temporal discretization error matched the spatial discretization error. For example, for the $N = 4096$ case, the relative $\ell_2$ error at the end of the 5 day run is graphed as a function of the time step in figure 21. The RBF method could easily have taken a larger timestep than 8 min (as reported in table 2), but temporal discretization errors would begin to dominate as shown by the large growth in the error after an 8 min 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 gave some timing benchmarks in table 3 for both test cases. Although RBF matrices are full and thus their inversion to calculate the three 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 $\ell_2$ error (or similarly as a function of $N$). All computations were performed in MATLAB v. 7.5 R2007b running on Dell PowerEdge 2950 Server with two 2.66 GHz Intel Xeon X5335 Quad Core processors. BLAS multithreading in MATLAB was enabled with a maximum of eight 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 at 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 as follows.

---

Figure 19. Spectrums of the RBF approximation operator of the r.h.s. of the linearized system in equations (5.1a) and (5.1b) for different values of $\varepsilon$. The spectrum of (a) has been scaled by the time step $\Delta t=360$ s (6 min), while the spectrums from (b) to (d) have been scaled by $\Delta t=1200$ s (20 min) with $\xi=\Delta tX$ eigenvalues. The stability domain of the RK4 time-stepping scheme has also been plotted as a grey line in each figure. (a) $\varepsilon=0$; (b) $\varepsilon=0.2$; (c) $\varepsilon=0.7$; and (d) $\varepsilon=1.7$. 

\[ \text{Proc. R. Soc. A (2009)} \]
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 $\ell_2$ error barely grew as function of time over the 5 day simulation period.

(i) $h$-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 $\ell_2$ error barely grew as function of time over the 5 day simulation period.

Figure 20. Relative $\ell_2$ error of the height field for the unsteady test case together with the predicted growth rate (solid line) from equation (6.3). $\Delta t=1200$ s (20 min) and $N=1849$. The value of $\epsilon$ (crosses) used is indicated below each figure. Figures (b) and (c) are run for 720 time steps, corresponding to 10 days of integration. (a) $\epsilon=7$; (b) $\epsilon=0.7$; and $\epsilon=1.7$.

Figure 21. Relative $\ell_2$ error in the height field at 5 days for the unsteady flow test case as a function of the time step for $\epsilon=3.25$ and $N=4096$. 

An RBF method on the sphere

Interestingly, in the limit as $\varepsilon \rightarrow 0$ (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 noted 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 PS 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.

For the steady flow test case, the RBF method achieves a relative $\ell_2$ error of $6.88 \times 10^{-12}$ with $N = 5041$ nodes and a time step of $6$ min 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 DF method when used with a SH filter with $N = 32\,768$ nodes and a time step of $90$ s (Spotz et al. 1998), also using a leapfrog scheme. For the unsteady test case, the RBF method achieved a relative $\ell_2$ error of $1.02 \times 10^{-8}$ with $N = 5041$ nodes and a time step of $6$ min using the standard RK4. The highest accuracy noted in the literature is $4 \times 10^{-5}$ for a SE method with $N = 24\,576$ nodes and a time step of $45$ s (Taylor et al. 1997).

The RBF node sets used in this study are for roughly uniform resolutions of approximately 500 km or greater on the surface of the Earth. However, to go to much higher resolutions (e.g. on the order of 10 km) and to do three-dimensional modelling 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 two or three dimensions, especially with regard to parallelization. In these cases, RBF methods, such as RBF finite differences, may give an alternative viable approach and are currently under development for a variety of applications.

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


Flyer, N. & Lehto, E. Submitted. Rotational transport on a sphere: local node refinement with radial basis functions.


