

# Rotational transport on a sphere: Local node refinement with radial basis functions <sup>☆</sup>

Natasha Flyer<sup>a,\*</sup>, Erik Lehto<sup>b</sup>

<sup>a</sup> Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, Boulder, CO 80305, USA

<sup>b</sup> Division of Scientific Computing, Uppsala University, Sweden

## ARTICLE INFO

### Article history:

Received 18 March 2009

Received in revised form 9 November 2009

Accepted 10 November 2009

Available online 16 November 2009

### Keywords:

Radial basis functions

Local node refinement

Hyberbolic PDEs

## ABSTRACT

This paper develops an algorithm for radial basis function (RBF) local node refinement and implements it for vortex roll-up and transport on a sphere. A heuristic based on an electrostatic repulsion type principle is used to re-distribute the nodes, clustering in areas where higher resolution is needed. It is then important to have a scheme that varies the shape of the RBFs over the domain so as to counteract the effects of Runge phenomena where the nodes are sparse. The roll-up of two diametrically opposed moving vortices are studied. The performance differences between near-uniform and refined nodes are addressed in terms of convergence, time stability, and computational cost. RBF results are put into context by comparison with published results for methods such as finite volume and discontinuous Galerkin.

© 2009 Elsevier Inc. All rights reserved.

## 1. Introduction

Although applications of radial basis functions (RBFs) have bloomed in recent years, using RBFs to solve evolutionary partial differential equations (PDEs) is a young research field. The strength of the method is in its ability to achieve spectral or high-order accuracy for scattered node layouts while being able to node refine in areas where increased resolution is needed. Although, this latter quality of local node refinement seems to naturally extend from the method being meshless and thus being able to place the points where needed, few papers have addressed this issue and the numerical complications that arise in doing so [6,14,16]. It is the aim of this paper to develop a meshless algorithm for RBF local node refinement on the sphere.

Since physical phenomena in fluid dynamics often require variable resolution depending on the formation of flow features, it would be advantageous to have a method that would naturally refine according to the physics. Currently used methods that allow for local mesh refinement, such as finite volume or elements, discontinuous Galerkin, and spectral elements, are linked to underlying grids that introduce artificial boundaries necessary to perform the numerics. In contrast, since RBFs are *not* linked to any surface-based coordinate system (i.e. grid or mesh), the placement of the nodes and how they are refined will physically reflect the features of the flow (and not resemble boxes, triangles, etc.). However, one can not simply ‘clump’ where needed without taking into account the Runge phenomenon, ill-conditioning, and adverse effects on the smoothness of the solution. The question then arises, ‘How does one node refine?’. Since even without boundaries these issues arise, we will begin with defining a methodology for local node refinement in periodic domains, such as the surface of a sphere.

<sup>☆</sup> The National Center for Atmospheric Research is sponsored by the National Science Foundation. The work was supported by NSF Grant ATM-0620100.

\* Corresponding author. Address: Institute for Mathematics Applied to Geosciences, National Center for Atmospheric Research, Boulder, CO 80305, USA. Tel.: +1 303 497 1292.

E-mail addresses: [flyer@ucar.edu](mailto:flyer@ucar.edu) (N. Flyer), [erik.lehto@it.uu.se](mailto:erik.lehto@it.uu.se) (E. Lehto).

An overview of the paper is as follows: Section 2 gives an introduction to RBFs via interpolation for constant and spatially variable shape RBFs; Section 3 discusses node distributions and convergence of RBF interpolants on the sphere; Section 4 defines the local node refinement scheme; Section 5 discusses the conditioning of the RBF interpolation matrix for spatially variable shape RBFs; Section 6 derives the discrete RBF advection operator; Sections 7 and 8 test the method on the roll-up of two diametrically opposed moving vortices, which is a simple model for cyclogenesis in the atmosphere. In the first test, the vortices are stationary as they roll-up; in the second test, they traverse the sphere at an arbitrary angle, combining linear advection and deformational flow. The performance of the local node refinement method is analyzed both in terms of a convergence and eigenvalue stability study with comparisons to other currently used methods in the literature, such as finite volume and discontinuous Galerkin. Section 9 summarizes the paper with future prospects.

## 2. An introduction to RBFs via interpolation

A good way to introduce the RBF method for PDEs is through interpolation since at each time step (in an explicit scheme) the exact spatial derivative operator is applied to the RBF interpolant to arrive at the derivative of the function at the node points. An RBF interpolant approximates a set of scattered data,  $\{f_j\}_{j=1}^N$ , sampled at some set of  $N$  distinct node locations,  $\mathbf{x} \in \mathbb{R}^d$ , by translates of a single radially symmetric function  $\phi(r)$ , where  $r$  is the Euclidean distance between where the RBF is centered,  $\mathbf{x}_j$ , and where it is evaluated  $\mathbf{x}$ , i.e.  $r = \|\mathbf{x} - \mathbf{x}_j\|_2$  (in future notation we drop the subscript 2). Examples of some RBFs are given in Fig. 1(a). Piecewise smooth RBFs feature a jump in some derivative at the location where they are centered and thus can only lead to algebraic convergence. For instance, the radial cubic  $r^3$  has a jump in the third derivative at  $\mathbf{x}_j$ , leading to fourth order convergence in 1-D, with the order of convergence increasing as the dimension increases (c.f. [23]). On the other hand, infinitely smooth RBFs, such as  $\sqrt{1 + (\epsilon r)^2}$  lead to spectral convergence of the interpolant as is noted in [19,35]. As a result, we will be using them in this study. They also have a parameter  $\epsilon$  that controls the shape of the RBF, which will play a crucial role in the node refinement scheme. The next two subsections contrast the differences in the interpolation problem between keeping the shape parameter  $\epsilon$  fixed throughout the domain and spatially varying it as will be needed for local node refinement throughout this study [6,32,14].

### 2.1. Constant $\epsilon$

Given the nodes  $\{\mathbf{x}_j\}_{j=1}^N$  and corresponding scalar function values  $\{f_j\}_{j=1}^N$ , the RBF interpolant  $s(\mathbf{x})$  to the data is defined by

$$s(\mathbf{x}) = \sum_{j=1}^N c_j \phi(\epsilon \|\mathbf{x} - \mathbf{x}_j\|), \tag{1}$$

where the expansion coefficients,  $\{c_j\}_{j=1}^N$ , 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, if the shape parameter,  $\epsilon$ , is kept fixed. For example, setting  $\epsilon = 1$  for simplicity of notation,

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

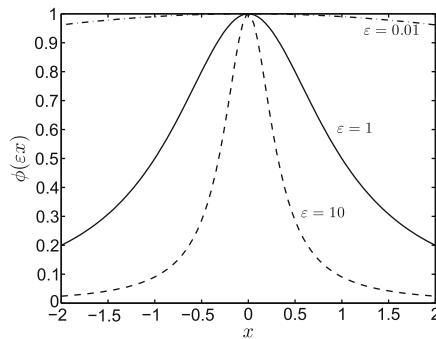
#### Piecewise Smooth

Monomials	$r^{2m+1}$
Thin Plate Splines	$r \log r$

#### Infinitely Smooth

Multiquadric	$\sqrt{1 + (\epsilon r)^2}$
Inverse Multiquadric	$\frac{1}{\sqrt{1 + (\epsilon r)^2}}$
Inverse Quadratic	$\frac{1}{1 + (\epsilon r)^2}$
Gaussian	$e^{-(\epsilon r)^2}$

(a)



(b)

Fig. 1. (a) Commonly used RBFs. (b) The varying shape of an IQ RBF, centered at  $x = 0$ , as a function of  $\epsilon$ .

where  $A$  is known as the RBF interpolation matrix. The non-singularity of RBF interpolation matrices for any distinct scattered node layout can be traced back to the theorems of Schoenberg [28,29]. For RBFs such as the Gaussians ( $e^{-(\varepsilon r)^2}$ ), inverse quadratics ( $1/(1 + (\varepsilon r)^2)$ ), and inverse multiquadrics ( $1/\sqrt{1 + (\varepsilon r)^2}$ ),  $A$  is positive definite regardless of dimension and node locations. For multiquadric RBFs ( $\sqrt{1 + (\varepsilon r)^2}$ ),  $A$  is conditionally positive definite and non-singularity of the matrix was not proven until 1986 [20]. For complete details on the well-posedness of (2) see [4, Chapters 12–16]. In addition, studies have shown that if the shape parameter,  $\varepsilon$ , is kept fixed throughout the domain best results are achieved with roughly evenly distributed nodes [16]. Section 3 gives a brief description of the numerous ways to distribute nodes quasi-uniformly on a sphere and the convergence rates for interpolation that are associated with such distributions. In general, for an in-depth study of the analytical properties of RBFs for spatially fixed  $\varepsilon$ , see [31,2,10,24].

With regard to the conditioning of the  $A$  matrix, the choice of  $\varepsilon$  is crucial to the accuracy of the method. One can immediately see from Fig. 1(b) that as  $\varepsilon \rightarrow 0$ , the translated RBFs become indistinguishable from one another, leading to ill-conditioning of the  $A$  matrix. In fact, Fornberg and Zuev [14] show that the condition number of  $A$  asymptotically scales as  $O(1/(\varepsilon^{2\sqrt{n-1}}))$  in the limit as  $\varepsilon \rightarrow 0$  for both constant and variable  $\varepsilon$  on the sphere (in the latter case  $\varepsilon$  refers to a constant by which the vector that contains the values of  $\varepsilon$  across the domain are uniformly scaled by). As a result, there has been some research with regard to the optimal choice of  $\varepsilon$  based on a technique known as cross-validation that is implemented extensively in the parameter optimization literature [5] and has been introduced to the RBF literature [26,3,7]. However, how to optimally choose  $\varepsilon$ , especially in the context of solving time-dependent PDEs (as is needed here) is still area of wide open research. To date, none of the above mentioned studies consider the  $\varepsilon \rightarrow 0$  limit (Fornberg and Piret [11] have developed algorithms to bypass the ill-conditioning of  $A$ ), where accuracy seems to greatly increase for solid body rotation on the sphere, as is shown in [12]. However, Flyer and Wright [9], using the algorithm of [11], show that for the nonlinear shallow water equations on the sphere, the accuracy degrades in this limit yet small  $\varepsilon$  is beneficial. This phenomena is yet to be understood. As to the authors' knowledge, there are no studies that relate how to optimally choose  $\varepsilon$  with respect to the eigenvalue stability of the differentiation matrix, which depends on  $A$ .

## 2.2. Spatially variable $\varepsilon$

In the case when  $\varepsilon$  varies from one RBF center to the next over the spatial domain, the RBF interpolant is no longer an expansion in terms of translates of a single basis function but of different basis functions. In other words, given the nodes  $\{\mathbf{x}_j\}_{j=1}^N$  and corresponding scalar data values  $\{f_j\}_{j=1}^N$ , the RBF interpolant  $s(\mathbf{x})$  to the data is defined by  $s(\mathbf{x}) = \sum_{j=1}^N a_j \phi_j(\varepsilon_j \|\mathbf{x} - \mathbf{x}_j\|)$ , where  $\{a_j\}_{j=1}^N$  is again determined by enforcing that the residual is zero at the collocation nodes. However, now the basis functions  $\phi_j$  are distinctly different from one another, varying in shape according to  $\varepsilon_j$ . This lead to a more linearly independent basis and thus better conditioning of the interpolation matrix  $A$ , as will be demonstrated in Section 5.

For spatially variable  $\varepsilon$ , theory severely lags the small amount of numerical results there are in the literature [18,6,32,14,27] with regard to accuracy and convergence and none to speak of with respect to the time stability of the RBF method. The standard proofs for non-singularity of the RBF interpolation matrix  $A$  no longer apply and those theoretical results that do exist, such as [1], depend on small perturbations to the  $A$  matrix that do not apply in this study. One of the main reasons for the lack in theory is that once  $\varepsilon$  varies, the  $A$  matrix is no longer symmetric and loses its structure. However, when clustering nodes,  $\varepsilon$  needs to vary in order to suppress Runge phenomenon [14] and will be discussed in Section 4.

Without its structure, the  $A$  matrix can no longer be guaranteed non-singular. However, Fornberg and Zuev [14] (Figs. 5.2 and 5.3) show that no matter what the node layout is or how  $\varepsilon$  is varied across the domain, the eigenvalues decay in an exceptionally regular pattern with respect to small  $\varepsilon$  (recalling that here  $\varepsilon$  refers to a constant by which the vector that contains the values of  $\varepsilon$  across the domain are uniformly scaled by). If the singularity of the  $A$  matrix were that sensitive to spatially varying  $\varepsilon$ , then we would not expect to see such regular distinct patterns. The locally clustered node sets demonstrated in this study result in a spatially variable  $\varepsilon$  of roughly  $O(1)$  and only vary the condition number of  $A$  by at most two orders of magnitude, as will be shown in Section 5.

In the next section, we define the mesh norm that is often used to prove the given error bound for interpolation on the sphere for quasi-uniform nodes with constant  $\varepsilon$ . Although we are clustering nodes, we start our refinement algorithms with the node sets for which these bounds are valid. Although technically not applicable to the refined node set or the RBF method-of-lines approximation, we do see similar error bounds in our numerical studies with respect to achieving spectral convergence as is noted in Sections 7 and 8.

## 3. Distributing nodes on a sphere and convergence of RBF interpolants

When using a meshless method such as the RBF method, the difficult task of creating a grid or mesh on an arbitrary domain is replaced by a simpler task of scattering nodes. Both near-uniform nodes and nodes clustered to reflect the physics of the problem are considered here.

### 3.1. Nearly uniform node distributions

For the surface of the sphere ( $\mathbb{S}^2$ ) there is no evenly distributed node set with more than 20 points. However, there are many different ways of creating near-uniform node distribution. Examples include distribution according to the golden ratio, equal partitioned area, convex hull approaches, Voronoi cells, etc. [30].

Near-uniform node sets can also be created by numerical optimization of various objective functions. Numerous node sets for the surface of the sphere are available online [34,30]. For the numerical results in the following sections, these types are chosen:

**Minimum Energy (ME) nodes:** ME nodes correspond to the minimum of the potential energy for electrostatic repulsion of point charges scattered on the surface of the sphere, hence the name.

**Maximum Determinant (MD) nodes:** These nodes can be attained by maximizing the determinant of the spherical harmonic interpolation matrix [33].

Since there is essentially no difference in the results between the two node sets (as seen in Table 2), ME nodes were used in discussions. A plot of these node sets with 1849 nodes is shown in Fig. 2. Note that the MD nodes have a more irregular layout than the ME nodes, which exhibit an almost hexagonal pattern.

Assuming  $\{\mathbf{x}_j\}_{j=1}^N$  are  $N$  nodes on the unit sphere, both ME and MD node distributions provide a quasi-uniform distribution on the sphere. ME node set maximizes the minimum distance between nodes according to the measure

$$h = \max_{\mathbf{x} \in \mathbb{S}^2} \min_{1 \leq i \leq N} \text{dist}(\mathbf{x}, \mathbf{x}_i), \tag{3}$$

where  $\text{dist}$  is the geodesic distance from  $\mathbf{x}$  to  $\mathbf{x}_i$ . This quantity is referred to as the mesh norm [34,17] and, geometrically, it represents the radius of the largest cap that covers the area between any subset of nodes on the sphere. The ME node sets have the property that  $h$  decays approximately uniformly like the inverse of the square root of the number of nodes  $N$ , i.e.

$$h \sim \frac{1}{\sqrt{N}}.$$

Thus, they are similar to a uniform discretization of the unit square.

The mesh norm is also of practical importance since it appears in many proofs of error bounds for RBF interpolation on the sphere (e.g. [17,15]). Indeed, in the context of infinitely smooth RBFs, it is shown in [17] that, provided the underlying function being interpolated is sufficiently smooth, RBF interpolants converge (in the  $L^\infty$  norm) like  $h^{-1/2} e^{-c/4h}$ , i.e. at an exponential rate, for some constant  $c > 0$  that depends on the RBF. For the ME node sets, convergence will thus proceed like  $N^{1/4} e^{-c\sqrt{N}/4}$ . In the experiments that follow we will demonstrate that this error bound seems to also hold for the RBF method-of-lines approximation of the two test cases.

### 4. Local node refinement scheme

When refining nodes, various criteria can be used. For instance, nodes can be refined in a region where an error measure is the largest or where the gradient of a quantity such as the magnitude of velocity or vorticity is greatest. In either case, nodes can not simply be clustered together in an arbitrary manner. The transition from sparse nodes to where the clustering occurs in the domain must be smooth. If there is an abrupt change in node density, then as a feature in the flow advects over the nodes, wave dispersion will occur due to an abrupt change in the wavelengths that can be supported. As a result, we will first look at ways to smoothly cluster the nodes. Secondly, decreasing the node density in one area of the domain can cause Runge phenomena to occur since there are not enough nodes to properly “pin down” the solution there [14]. Thus, to counteract

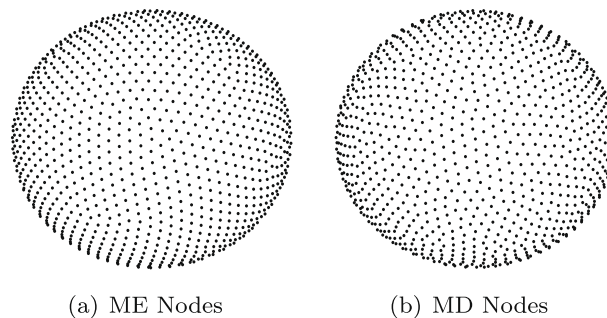


Fig. 2. Near-uniform node sets with  $N = 1849$  nodes. Displayed are the nodes on the northern hemisphere.

this, the shape of the RBFs need to vary over the domain. We will implement a heuristic that was discussed for 1-D interpolation in [14].

#### 4.1. Heuristic for smooth node clustering

The node refinement scheme should reflect the physics of the PDE(s), as mentioned above. The question then becomes how to refine in order to achieve a smooth node distribution. A quite simple way that has proven to be effective in our studies is to simulate electrostatic repulsion on the surface of the sphere to a low order (so that it is computationally cheap yet achieves the purpose). If every node is assigned the same charge in a random distribution, then the result of our code will be a roughly approximate ME node set. However, by applying different charges to the nodes through a charge distribution function, the node density over the domain will vary. We let nodes move until force equilibrium is reached with respect to a set tolerance, such as  $\|\Delta \mathbf{x}\|_\infty < 10^{-4}$  ( $\Delta \mathbf{x}$  is defined as  $\max_{1 \leq j \leq N} \|\mathbf{x}_j^{\text{new}} - \mathbf{x}_j^{\text{old}}\|_2^2$ , where we have distinguished between the new and old positions of the nodes in the iterative algorithm). Where the nodes have high charges, the node density will be low and vice versa. With this method, the node set will also have smooth transitions between areas of different node densities.

In our test cases, the analytical solution is known so that we can do an exact error study. As a result, the angular velocity at which the vortices roll-up is known (albeit, this is usually not the case – yet, it provides a good example of how to do node refinement with RBFs). Since the vortex will form fine features, requiring high resolution, where the angular wind velocity is high, it is sensible to use this velocity to assign charges for the generation of the node set. Other charge distributions for node refinement, such as a Gaussian charge distribution, were evaluated but did not perform as well in comparison. The angular wind velocity for vortices centered at the poles is given by

$$\omega(\theta) = \begin{cases} v_0 \frac{3\sqrt{3}}{2\rho} \operatorname{sech}^2(\rho) \tanh(\rho) & \text{if } \rho \neq 0, \\ 0 & \text{if } \rho = 0, \end{cases} \quad (4)$$

where  $\theta$  is the latitude and  $\rho = \rho_0 \cos \theta$  with  $\rho_0 = 3$ . For each test,  $\omega(\theta)$  is scaled differently in order to correspond to the literature. For the stationary case,  $v_0 = 1$  and for the translating vortex test case  $v_0 = 2\pi/12$  (however, for the repel algorithm  $v_0 = 2\pi/12$ ). In order to achieve a high node density where the angular velocity is high, the charge distribution for the electrostatic simulation was given by

$$q(\theta) = \frac{1}{0.1 + c \cdot \omega(\theta)}, \quad (5)$$

where  $c$  is a parameter that controls the amount of node clustering in the polar regions. The small constant with value 0.1 in the denominator was added to avoid the singularities at  $\theta = -\frac{\pi}{2}, \frac{\pi}{2}$ . Node distributions obtained for a few values of this parameter are shown in Fig. 3. In Fig. 4, a node set of 900 nodes with parameter value  $c = 10$  is plotted together with the known analytic solution of the first test case, the roll-up of a stationary vortex, at time  $t = 6$ .

#### 4.2. Heuristic for varying $\varepsilon$

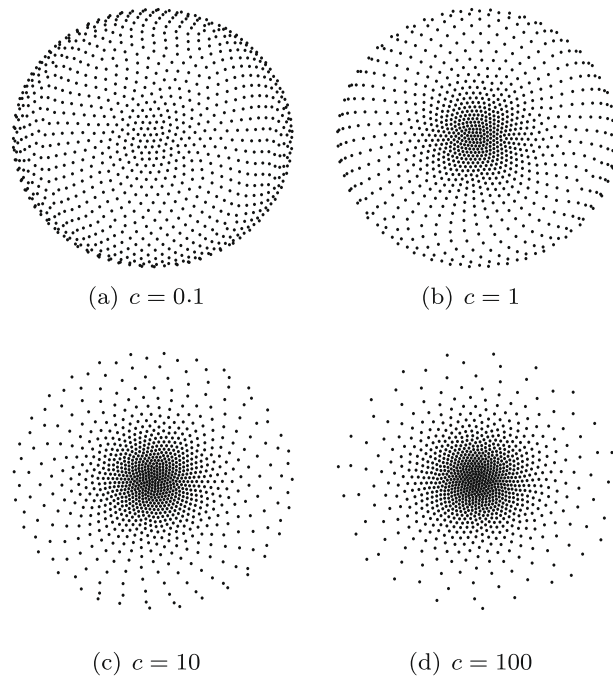
When clustering nodes, the shape parameter of the RBF must vary across the domain in order to avoid Runge phenomena in areas more sparsely populated by nodes, as mentioned above. This oscillatory error phenomenon, commonly associated with high-order polynomial interpolation on uniform grids, was studied in relation to RBF interpolation by Fornberg and Zuev [14]. For a 1-D interpolation problem, they showed that scaling  $\varepsilon$  according to the inverse of the Euclidean ( $\ell_2$  norm) distance to the nearest neighbor node gives nearly optimal results with respect to the interpolation error. This rule was used throughout this study with excellent results obtained. The nearest neighbor rule for an RBF  $\phi_j$  centered at  $(\lambda_j, \theta_j)$  was implemented by letting

$$\varepsilon_j = \varepsilon_{\min} \left( \frac{\max_j d_{j,\min}}{d_{j,\min}} \right), \quad (6)$$

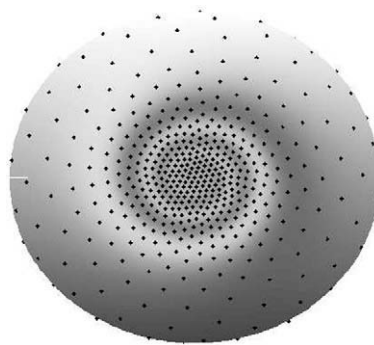
where  $d_{j,\min}$  is the Euclidean distance between the node  $(\lambda_j, \theta_j)$  and its closest neighbor node and  $\varepsilon_{\min}$  is a scaling parameter. For the refined node sets acquired from the electrostatic repulsion, the shape parameter will roughly range from  $\varepsilon_{\min}$  to  $10\varepsilon_{\min}$  and the optimal value for the scaling parameter was typically  $\varepsilon_{\min} \sim \mathcal{O}(1)$ . Shown in Fig. 5 is a comparison between the angular velocity and the shape parameter, showing how the value of  $\varepsilon$  varies according to the nodal density, which in turn relates to the angular velocity through the node refinement strategy. The charge distribution  $\{q(\theta_j)\}_{j=1}^N$  that we place on the nodes determines the final separation between the nodes  $\{x\}_{j=1}^N$  and  $\{\varepsilon_j\}_{j=1}^N$  is defined in terms of the inverse of those distances. Therefore,  $\varepsilon_j$  becomes proportional to  $1/q(\theta_j)$ .

### 5. The conditioning of the interpolation matrix $A$ for spatially variable $\varepsilon$

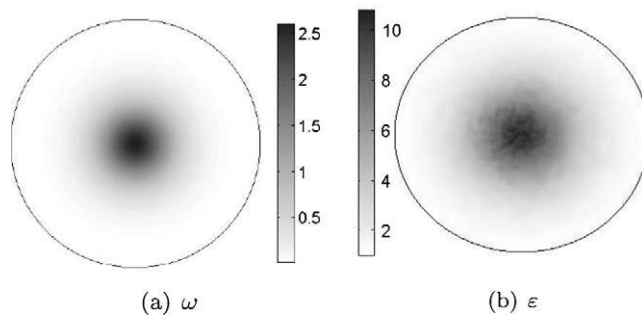
Since the differentiation matrices depend on inverting  $A$  (as will be shown in the next section), we investigate how the conditioning of  $A$  varies with both the clustering  $c$  and the number of nodes,  $N$ , for both ME nodes and refined nodes (i.e.



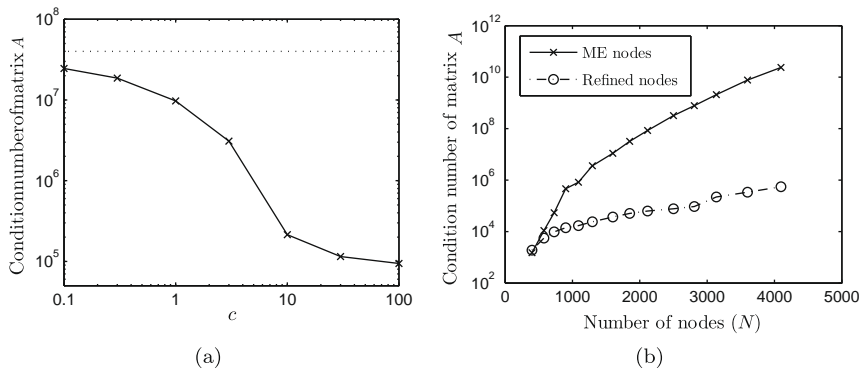
**Fig. 3.** Refined node sets of 1849 nodes obtained from electrostatic repulsion for different values of the parameter  $c$  with the charge distribution in (5). Nodes in northern hemisphere shown.



**Fig. 4.** The analytic solution to the stationary vortex test case at time  $t = 6$  displayed for the northern hemisphere. Dots represents the node locations for a refined node set of 900 nodes. The value  $c = 10$  was used.



**Fig. 5.** A comparison of the angular velocity and the value of the shape parameter according to the nearest neighbor rule.



**Fig. 6.** The condition number of the RBF interpolation matrix  $A$  as a function of (a) the clustering of the nodes,  $c$ ; dotted line indicates ME node set; and (b) the total number of nodes  $N$ . For subplot (a),  $N = 1849$  and  $\varepsilon = 3.3$  for the ME nodes, and for the clustered nodes  $\varepsilon_{\min}$  ranged from 1.5 for  $c = 100$  to 3 for  $c = 0.1$ .

spatially variable  $\varepsilon$ ). Fig. 6(a) shows that the conditioning of the  $A$  gets better as we cluster the nodes, that is for values away from small  $c$ . This behavior is expected since the basis functions become more linearly independent, i.e. they look different from one another and are no longer translates of an identical function. On the other end of the figure, as  $c$  becomes large, many of the nodes are located in the clustered region, resulting in a set of basis functions that are highly peaked and do not vary as widely across the domain. Thus, the conditioning of the  $A$  matrix is barely improved. The condition number of  $A$  for the ME node set is shown by the dotted line in Fig. 6(a) and is approximately two orders of magnitude larger than for the refined node sets used in the paper. Fig. 6(b) shows that the condition number of  $A$  for the ME node set grows much more rapidly as function of the number of nodes  $N$  than the refined (clustered) node set. Normally, larger  $N$  is detrimental for ill-conditioning. However, where the nodes are denser  $\varepsilon$  is made larger, which is better for conditioning. These two effects offset each other such that the result in Fig. 6(b) is observed. That is, from  $N = 400$  to  $N = 4096$ , the condition only grows by approximately two orders of magnitude.

**6. Deriving the RBF discrete advection operator**

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

$$r(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_k\| = \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2} \tag{7}$$

$$= \sqrt{2(1 - \cos \theta \cos \theta_k \cos(\lambda - \lambda_k) - \sin \theta \sin \theta_k)}, \tag{8}$$

where  $\theta$  and  $\lambda$  represent latitude and longitude, respectively. 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 RBF are centered.

For a RBF  $\phi_k = \phi(\varepsilon_k \|\mathbf{x} - \mathbf{x}_k\|)$  centered at  $\mathbf{x}_k$  with shape parameter  $\varepsilon_k$ , the partial derivatives with respect to  $\lambda$  and  $\theta$  can be obtained by using the chain rule and are given by

$$\frac{\partial \phi_k}{\partial \lambda} = \cos \theta \cos \theta_k \sin(\lambda - \lambda_k) \left( \frac{1}{r} \frac{\partial \phi_k}{\partial r} \right), \tag{9}$$

$$\frac{\partial \phi_k}{\partial \theta} = (\sin \theta \cos \theta_k \cos(\lambda - \lambda_k) - \cos \theta \sin \theta_k) \left( \frac{1}{r} \frac{\partial \phi_k}{\partial r} \right). \tag{10}$$

Applying the partial derivative with respect to  $\lambda$  to the RBF interpolant in (1) of the geopotential height field  $h$  and evaluating at the nodes gives

$$\left( \frac{1}{\cos \theta} \frac{\partial h}{\partial \lambda} \right) \Big|_{\{\theta_j, \lambda_j\}_{j=1}^N} = \sum_{k=1}^N c_k \underbrace{\cos \theta_k \sin(\lambda_j - \lambda_k) \left( \frac{1}{r} \frac{d\phi_k}{dr} \right)}_{B_{j,k}^\lambda} \Big|_{\{\theta_j, \lambda_j\}_{j=1}^N} = \sum_{k=1}^N c_k B_{j,k}^\lambda, \quad j = 1, \dots, N \tag{11}$$

$$D^i \vec{h} = (B^i A^{-1}) \vec{h},$$

where we have used the fact that  $c = A^{-1}h$ .  $D^0$  is derived in the same manner. Note that the singularities in the spherical coordinate system are no longer present in the RBF formulation. The advection equation for the geopotential height field, which is the core of our test cases, can be written in semi-discrete form as



$$\frac{\partial \vec{h}}{\partial t} + U(\alpha, \theta, \lambda, t) D^\lambda \vec{h} + V(\alpha, \theta, \lambda, t) D^\theta \vec{h} = 0, \tag{12}$$

where  $\vec{h}$  is the height field sampled at the nodes  $\{\theta_j, \lambda_j\}_{j=1}^N$ . The diagonal matrices  $U$  and  $V$  correspond to the time-dependent wind vectors also sampled at  $\{\theta_j, \lambda_j\}_{j=1}^N$  and  $\alpha$  is the degree of rotation of the wind field with respect to the polar axis. The classical fourth order Runge–Kutta (RK4) scheme was used to advance the PDE in time with the inverse multiquadric RBF,  $1/\sqrt{1 + (\epsilon r)^2}$ , being used.

**7. Numerical test case 1: stationary vortex roll-up**

In this test case, vortices form at the north and south pole of a coordinate system rotated an angle  $\alpha$  with respect to the original polar axis. Being a test of deformational flow with increasingly stronger gradients over time, it is a simple model for the observed evolution of cold and warm frontal zones (for complete details on the test case see [22]). However, since node layouts do not lie along any directional lines or vertices, as seen in Fig. 2, the position of the north pole is completely arbitrary. Thus, the PDE can be immediately implemented in a rotated  $(\lambda', \theta')$  system without needing to do any transformations, where  $(\lambda', \theta')$  is given by

$$\lambda'_j = \arctan \left( \frac{\sin(\lambda_j - \lambda_p)}{\sin \lambda_p \cos(\lambda_j - \lambda_p) - \cos \theta_p \tan \theta_j} \right),$$

$$\theta'_j = \arcsin (\sin \theta_j \sin \theta_p + \cos \theta \cos \theta_p \cos(\lambda_j - \lambda_p))$$

with  $(\theta_p, \lambda_p)$  being the north pole of the rotated coordinate system. In this rotated coordinate system, the wind velocities are

$$u' = \omega(\theta') \cos \theta', \tag{13}$$

$$v' = 0, \tag{14}$$

where  $\omega$  is given in (4) with the advection equation (12) reducing to

$$\frac{\partial \vec{h}}{\partial t} + U(\theta') D^{\lambda'} \vec{h} = 0. \tag{15}$$

The analytic solution at time  $t$  is given by

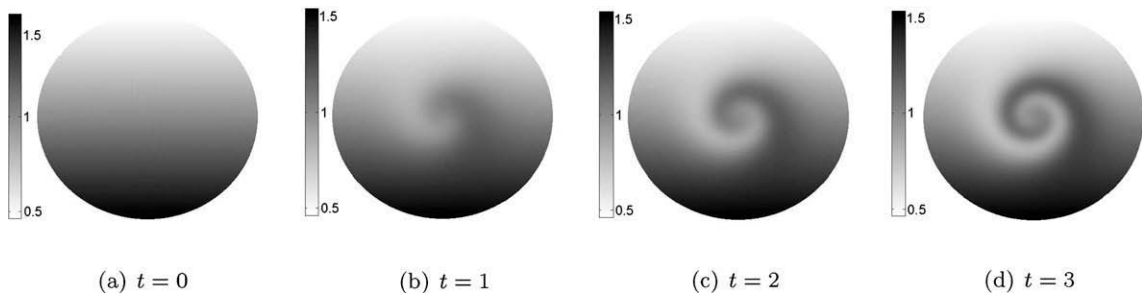
$$h(\lambda', \theta', t) = 1 - \tanh \left( \frac{\rho}{\gamma} \sin(\lambda' - \omega(\theta')t) \right). \tag{16}$$

$\gamma$  controls the steepness of the solution and is chosen to be 5. The total simulation time was chosen to be  $t = 3$  for comparison with results in the literature. The analytic solution (although there is absolutely no difference to the eye with regard to the numerical solution) at various times is given in Fig. 7

As the stationary vortex test case was implemented with the RBF method by Flyer and Wright [8], only a short section is devoted to this test case, with the main focus on a comparison of the error between near-uniform nodes and refined nodes. For the moving vortex test case, stability issues and other aspects of the numerical results are discussed in some detail.

*7.1. Numerical results: near-uniform nodes versus refined nodes*

Since it would be costly to optimize the parameter  $c$  in Eq. (5) to achieve the lowest possible error, a number of runs were performed and a comparison of the error for a few different values of this parameter is given in Fig. 8. As the difference between the node sets at the optimal value of  $\epsilon_{\min}$  is rather small, no further optimization was attempted and the following numerical results for this test case were achieved with  $c = 3$ . In cases where the analytical solution is not explicitly known,



**Fig. 7.** The analytical solution at different times  $t$  to the stationary vortex test case as viewed from the north pole.



a measure of the flow, such as the gradient of the vorticity, would need to be calculated and then the minimum wavelength scales that need to be resolved could be determined. From the fact that spectral methods need about two nodes to resolve a wavelength, the amount clustering could be estimated.

In [8], results for this test with ME nodes were given. To achieve a comparable error estimate when using node refinement, the solution for the refined case is interpolated to a 6400 ME node set and measured against the analytical solution at these points. In both ME and refined cases, we would expect the error to grow in time. This is true even in the latter case since we are not adaptively refining, that is adding more nodes in order to resolve the finer features of the solution as they appear. However, the growth rate of the error for the refined case should be smaller as we have higher resolution in areas where it is needed. This is indeed the case as seen in Fig. 9. At the end of the run, Fig. 10 shows the magnitude of the error in the numerical solution with and without node refinement (clustering) for  $N = 3136$ . With node refinement, not only have we gained a factor of 15 in accuracy but no longer does the error occur in the center of the vortex (as is the case in Fig. 10(a)) but in the outer regions where the gradients in the solution are much softer and thus the node density is sparser.

As shown in Table 1, the node refinement improves the accuracy by approximately one to two orders of magnitude in the  $\ell_\infty$  norm, a factor of 20–30 in  $\ell_2$ , and an order of magnitude in the  $\ell_1$  norm. For a given number of nodes, the time step for refined nodes is smaller due to the more restrictive local CFL condition. In all cases, it has decreased by roughly a factor of 3. Since the node spacing for near-uniform nodes is proportional to  $\frac{1}{\sqrt{N}}$ , the error should decay as  $\mathcal{O}(e^{-\sqrt{N}})$  to achieve spectral accuracy. This might not be so evident by looking at the numbers in Table 1, so we plot the  $\ell_2$  error as a function of  $\sqrt{N}$  in Fig. 11 for both the ME and refined cases.

When doing refinement with any type of numerical method, one expects a reduction in computational cost for a given accuracy. RBFs are no exception. For example, to achieve an  $\ell_2$  error of about  $1 \cdot 10^{-5}$ , 4096 nodes are needed in the near-uniform case compared to 1849 nodes with node refinement. At  $\mathcal{O}(N^2)$  arithmetic operations per time step, this results in computational savings by a factor of 5. Furthermore, the memory cost per full matrix is also reduced by a factor of 5.

### 8. Numerical test 2: translating vortex roll-up

For this test case, the vortices move along the equator of the rotated coordinate system as they form. It is a simple cyclogenesis model for any type of moving vortex roll-up in atmospheric dynamics, such as hurricanes and tropical cyclones (for complete details on the test case see [21]). The wind field is a combination of linear advection and the stationary vortex wind

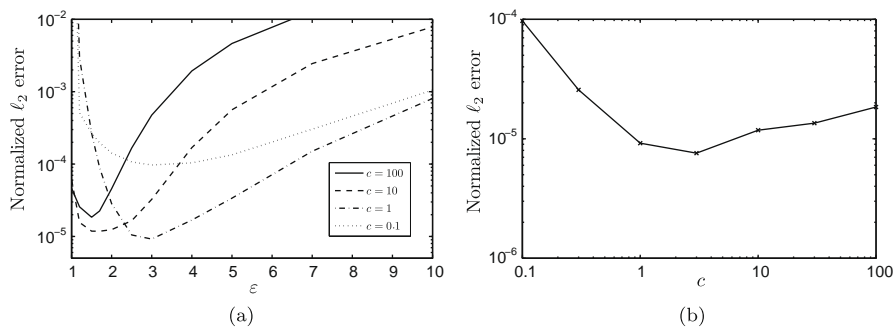


Fig. 8. (a) The normalized  $\ell_2$  error as a function of  $\epsilon_{\min}$  for different values of the parameter  $c$  at time  $t = 3$  and  $N = 1849$ ; (b) the error for the optimal  $\epsilon_{\min}$  is plotted as a function of the parameter  $c$  at time  $t = 3$  and  $N = 1849$ .

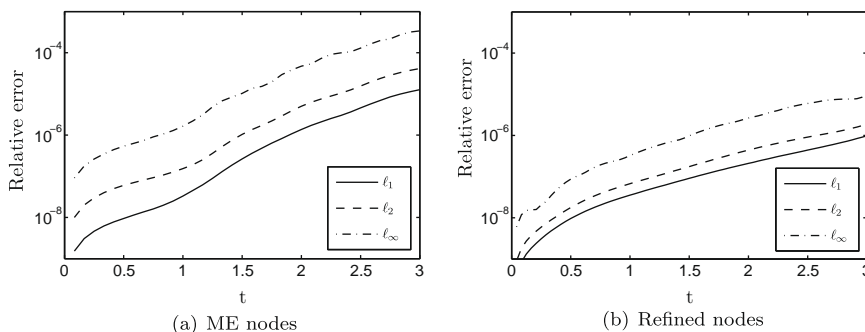
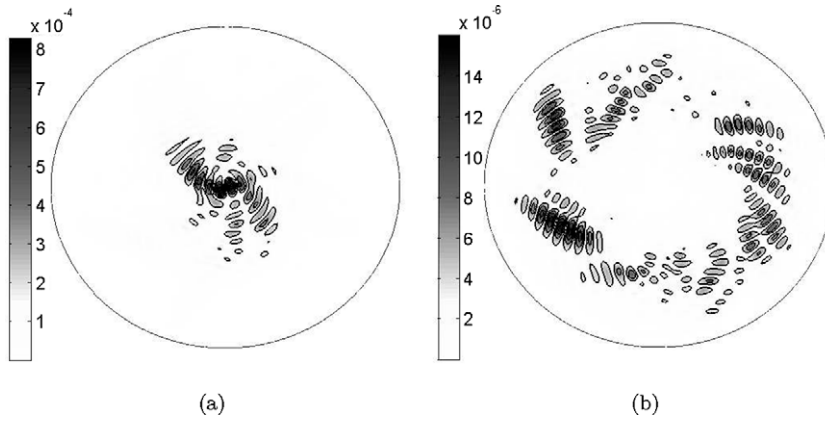


Fig. 9. Time traces of the normalized error for the stationary vortex test case using  $N = 3136$  nodes.

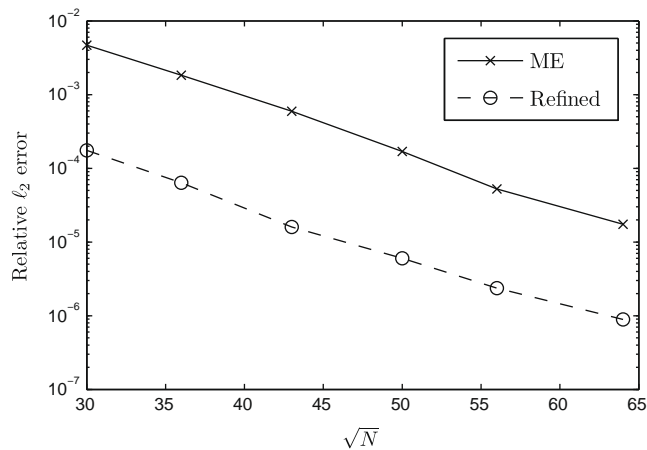


**Fig. 10.** The magnitude of the error at  $t = 3$  for (a) ME nodes and (b) refined (clustered) nodes for  $N = 3136$ . Notice that the scales are different on the two figures, with the maximum absolute error being approximately  $8 \cdot (10^{-4})$  for (a) and  $1.5 \cdot (10^{-5})$  for (b).

**Table 1**

Comparison of the error for different node distributions for the stationary vortex test case. For the refined node sets,  $\varepsilon$  refers to  $\varepsilon_{\min}$ .

Node set	N	$\varepsilon$	$\Delta t$	Normalized error		
				$\ell_1$	$\ell_2$	$\ell_\infty$
Refined	900	2.5	1/6	$9.2 \cdot 10^{-5}$	$1.8 \cdot 10^{-4}$	$9.0 \cdot 10^{-4}$
	1296		1/8	$3.4 \cdot 10^{-5}$	$6.4 \cdot 10^{-5}$	$5.0 \cdot 10^{-4}$
	1849		1/8	$9.3 \cdot 10^{-6}$	$1.6 \cdot 10^{-5}$	$7.8 \cdot 10^{-5}$
	2500		1/10	$3.3 \cdot 10^{-6}$	$6.0 \cdot 10^{-6}$	$3.7 \cdot 10^{-5}$
	3136		1/12	$1.3 \cdot 10^{-6}$	$2.4 \cdot 10^{-6}$	$1.7 \cdot 10^{-5}$
	4096		1/16	$5.1 \cdot 10^{-7}$	$8.9 \cdot 10^{-7}$	$5.8 \cdot 10^{-6}$
ME	900	3	1/2	$1.3 \cdot 10^{-3}$	$4.7 \cdot 10^{-3}$	$4.7 \cdot 10^{-2}$
	1296		1/2	$5.8 \cdot 10^{-4}$	$1.8 \cdot 10^{-3}$	$1.7 \cdot 10^{-2}$
	1849		1/4	$1.6 \cdot 10^{-4}$	$6.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-2}$
	2500		1/4	$4.8 \cdot 10^{-5}$	$1.7 \cdot 10^{-4}$	$1.9 \cdot 10^{-3}$
	3136		1/6	$1.5 \cdot 10^{-5}$	$5.2 \cdot 10^{-5}$	$6.9 \cdot 10^{-4}$
	4096		1/6	$5.3 \cdot 10^{-6}$	$1.7 \cdot 10^{-5}$	$1.7 \cdot 10^{-4}$



**Fig. 11.** The normalized  $\ell_2$  error as a function of  $\sqrt{N}$  for both ME and refined node distributions for the stationary test case.

given in (13) and (14). As the vortex wind velocity depends on the position of the vortex centers, the wind field will be time-dependent.

Again, since the RBF method is independent of which coordinate system the PDE is expressed in, it is beneficial to implement the PDE in the rotated system and all the following equations are given in this coordinate system. The time-independent linear advection wind is given by



where  $\omega(\theta')$  is obtained by rotating the coordinate system to the vortex center and then applying (4). This gives the total wind field

$$\mathbf{u}'(t) = \mathbf{u}'_s + \mathbf{u}'_r(t), \quad (21)$$

$$\mathbf{v}'(t) = \mathbf{v}'_s + \mathbf{v}'_r(t). \quad (22)$$

For the moving vortex test case, the analytic solution is given by (16) as for the stationary case. Since the vortices are no longer stationary, a series of rotations must be performed to calculate the analytic solution at a given point. For a point with coordinates  $(\lambda', \theta')$  in the rotated coordinate system, the analytic solution is obtained by first rotating the coordinate system to a position  $(\lambda'_s, \theta'_s) = (\lambda' - u_0 t, \theta')$ , then determining the corresponding coordinates in the regular spherical coordinate system  $(\lambda_s, \theta_s)$  and finally applying a rotation such that the north pole of this system is at the current vortex center  $(\lambda_c, \theta_c)$ .

When applying node refinement, the node set must also rotate as the vortices translate across the sphere. This does not affect the differentiation matrices, but the solution must be interpolated to the new node locations at every rotation of the node set. The total computational cost of  $\mathcal{O}(N^2)$  operations per time step is still the same. For the numerical results presented here, the node set was rotated at every time step. RK4 was used to advance the system in time with inverse quadratic RBFs,  $1/(1 + (\epsilon r)^2)$ , being used simply for variety. To compare with results in the literature, this case was run to  $t = 12$ . As a result, a value of  $c = 10$  was used for the refinement in order to capture sharp gradients that occurred in the solution at later times.

Shown in Fig. 12 is a contour plot of the solution and the magnitude of the error at different times with a refined node set of 3136 nodes.

### 8.1. Convergence study

Due to a smooth solution for all time, we expect the RBF method to be spectrally accurate, i.e. the error should decrease exponentially with decreasing node spacing. Shown in Fig. 13 is a plot of the normalized  $\ell_2$  error as a function of  $\sqrt{N}$ . In this log-linear plot, the RBF method with ME nodes appears to be converging as expected. The refined node set improves the accuracy by a factor of 20 in accuracy whilst giving the same convergence rate.

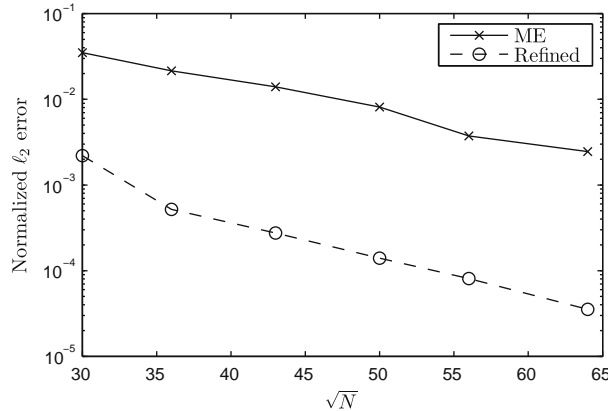
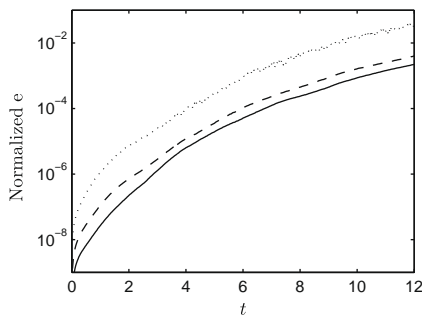


Fig. 13. The normalized  $\ell_2$  error as a function of  $\sqrt{N}$  at time  $t = 12$  days for the moving vortex test case.

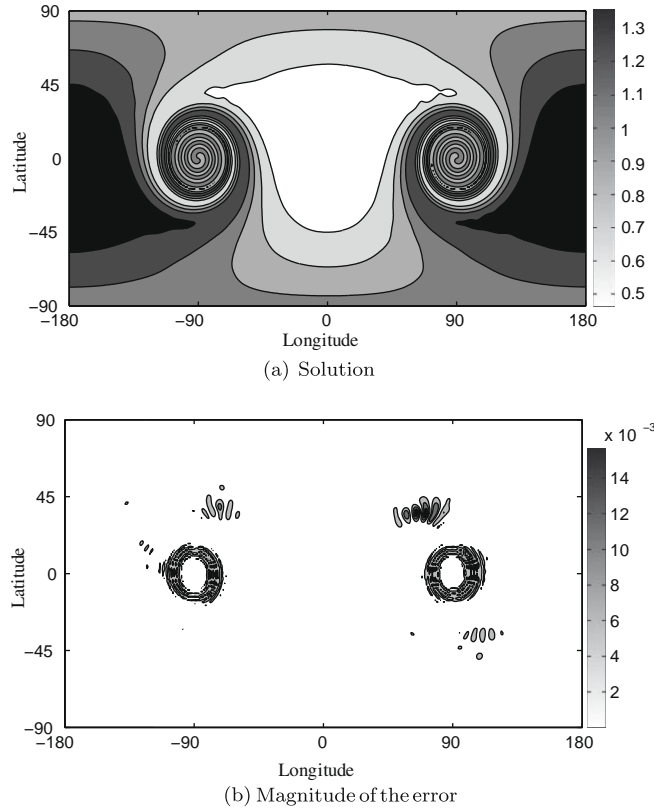


Time traces of the error are shown in Fig. 14. Since  $c = 10$  here (Fig. 3(c)), we will have a more restrictive CFL condition than in the previous test case. As a result, the time step of  $\Delta t = 18$  minutes was chosen in order that spatial errors dominate. Refining the nodes gains roughly a factor of 20 in accuracy throughout the simulation in all norms reported.

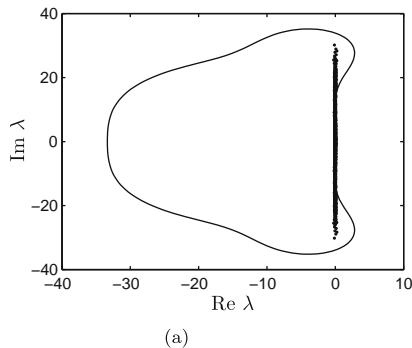
To show an extreme case of vortex roll-up, we run the simulation for 24 days. Fig. 15 shows the solution and the magnitude of the error at time  $t = 24$  days with a refined node set of  $N = 3136$  nodes. Even after this long time integration and with the solution displaying extremely fine features, the  $\ell_\infty$  error is still on the order of  $10^{-2}$ . Also, note that there is little to no dispersion in the solution as seen by Fig. 15(b).

## 8.2. Time stability study

Classic eigenvalue stability analysis does not theoretically apply to these non-normal differentiation matrices and the time-dependent wind velocities add further complication to the stability assessment. In practice however, the magnitude



**Fig. 15.** The numerical solution and the magnitude of the error at time  $t = 24$  days for the moving vortex test case using  $N = 3136$  refined nodes.



of the largest eigenvalue does not change over time and the eigenvalue spectrum gives a very good indication on the time step necessary to obtain stability. Fig. 16 shows the eigenvalues of the full differentiation matrix at time  $t = 12$  days together with the stability region for RK4. For any hyperbolic problem, the absolute ideal situation is that the eigenvalues of the discretized spatial operator lie on the imaginary axis. As seen, the eigenvalues lie tightly clustered along the imaginary axis. While the eigenvalues with positive real parts correspond to growing eigenmodes, this appears to be of little practical concern as the method is stable for very long time integrations as was seen in Fig. 15.

The error versus the time step is shown in Fig. 17 and the results are in agreement with the eigenvalues shown in Fig. 16. Note that although a time step of  $\Delta t = 45$  minutes is stable with the refined node set, temporal errors will dominate for time steps larger than  $\Delta t = 20$  min, the location in the figure where the refined error begins to grow.

### 8.3. Comparative results

As this is a recent test case, the only results found in the literature are presented in [21,25] and displayed in Table 2. In [21], finite volume (FV) on a latitude–longitude grid (lat–lon) was run also with adaptive mesh refinement (AMR), the base level being  $N = 2592 (5^\circ \times 5^\circ)$  with three adaptive refinement levels. Each method uses a different grid structure with the discontinuous Galerkin (DG) having been run sub-optimally with respect to the CFL limit (personal communication with Ram Nair). In general, the RBF method with near-uniform nodes, as given by both ME and MD node sets, achieves a compa-

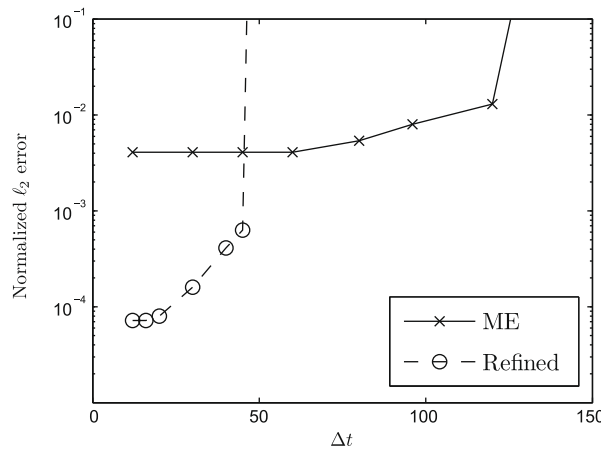


Fig. 17. The normalized  $\ell_2$  error as a function of the time step.

Table 2

Results using RBF, DG, and FV for the moving vortex test case run for 12 days. The FV AMR has a base level of  $5^\circ \times 5^\circ$ , corresponding to  $N = 2592$  if a point is centered in each control volume.

Method	$N$	$\Delta t$ (min)	Normalized error	
			$\ell_1$	$\ell_2$
RBF, refined	900	60	$4.0 \cdot 10^{-3}$	$5.4 \cdot 10^{-3}$
	1849	30	$2.6 \cdot 10^{-4}$	$3.9 \cdot 10^{-4}$
	3136	18	$4.6 \cdot 10^{-5}$	$7.2 \cdot 10^{-5}$
RBF, ME	900	180	$1.6 \cdot 10^{-2}$	$2.8 \cdot 10^{-2}$
	1849	120	$6.5 \cdot 10^{-3}$	$1.5 \cdot 10^{-2}$
	3136	60	$2.0 \cdot 10^{-3}$	$4.1 \cdot 10^{-3}$
RBF, MD	900	180	$1.6 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$
	1849	120	$6.9 \cdot 10^{-3}$	$1.5 \cdot 10^{-2}$
	3136	60	$2.9 \cdot 10^{-3}$	$4.7 \cdot 10^{-3}$
DG	9600	6	$2.1 \cdot 10^{-3}$	$7.1 \cdot 10^{-3}$
FV (lat–lon)	AMR (2592-base, 3 levels)	Variable	$1.4 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$
	165,888 ( $0.625^\circ \times 0.625^\circ$ )	10	$5 \cdot 10^{-4}$	$2.0 \cdot 10^{-3}$
FV (cubed sphere)	38,400	30	$5 \cdot 10^{-3}$	$2 \cdot 10^{-3}$

rable accuracy with FV, both on a cubed sphere and latitude–longitude grid, and with DG methods but with lower  $N$  and larger time steps on the order of hours. When adopting node refinement, the RBF method is more accurate to any previously reported result, with the  $\ell_1$  and  $\ell_2$  errors on the order of  $10^{-5}$ .

8.4. Performance benchmarks

To assess the performance of the implementation, a number of runs were performed and the runtime was measured. See Figs. 18 and 19 for a comparison of runtime between refined nodes and ME nodes. In Fig. 18 the runtime per time step is shown and it appears to grow quadratically with the number of nodes. Since the RBF method requires  $\mathcal{O}(N^2)$  operations per time step, this result is expected. When using refined nodes, the computational cost is just slightly higher since an extra matrix–vector multiply is needed to interpolate the solution to the new rotated node locations at each time step. Basically, for almost the same computational cost we have a vast increase in accuracy.

The total runtime as a function of the normalized  $\ell_2$  error is plotted in Fig. 19. This figure shows the large benefits of using node refinement. Reaching an  $\ell_2$  error of  $4 \cdot 10^{-3}$  requires a runtime of about 12 s with refined nodes compared to 3 min with near-uniform nodes. Another important performance aspect is memory usage, which due to the full differentiation matrices grows rapidly with increasing number of nodes. The largest ME node set used, with  $N = 3136$  nodes, gives an  $\ell_2$  error of  $4 \cdot 10^{-3}$ . In comparison, less than 1000 refined nodes are needed to achieve a comparable error, which corresponds to a reduction of the memory cost for each full matrix by a factor of 6. The lowest error reached with refined nodes is also almost two orders of magnitude lower than for the largest ME node set.

The performance was measured running MATLAB version 7.1 on a laptop with a 1.83 GHz dual-core processor and 2 GB of memory available. This version of MATLAB does not use multi-threading. Thus, only one processor core is used.

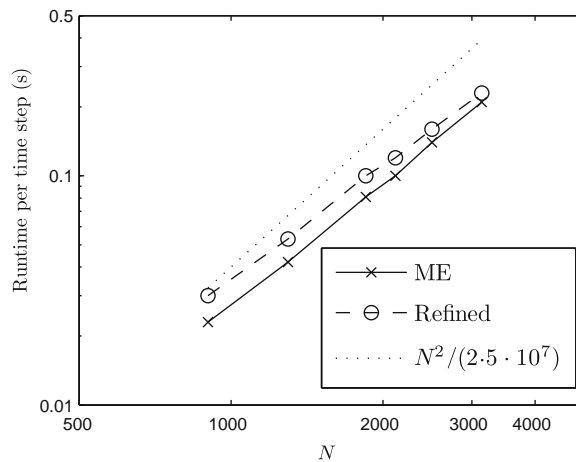


Fig. 18. The runtime per time step as a function of the number of nodes.

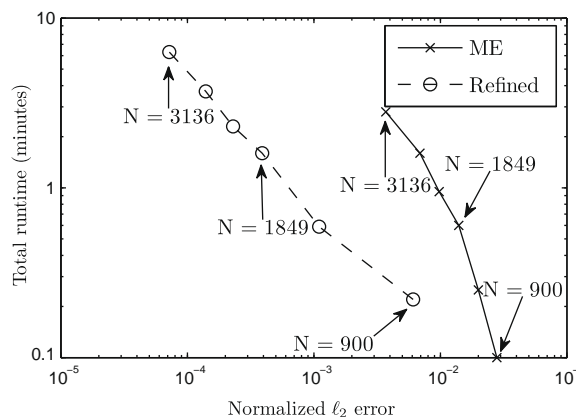


Fig. 19. The total runtime as a function of the normalized  $\ell_2$  error at 12 days.



## 9. Conclusion

A heuristic was developed for performing local node refinement with RBFs by: (1) running an “electrostatic repulsion” type algorithm where the charge density distribution for the nodes was (a) smooth and (b) reflected some physical property of the problem; and (2) the shape parameter of the RBFs was varied across the domain according to the inverse of the Euclidean ( $\ell_2$ ) distance to the nearest neighbor node. Two test cases were addressed, stationary and moving vortex roll-up. In the former case, node refinement increased the accuracy by as much as two orders of magnitude and for the latter case by a factor of 20. It had been known from previous studies [8,9] that RBFs can take unusually long time steps for hyperbolic problems due to the nature of the eigenvalue spectrum of the discretized spatial operator. It was shown here that this property also holds when local node refinement was implemented, although, due to the clustering of nodes leading to a more restrictive CFL condition, the time steps are shorter in comparison. Furthermore, as would be expected of any refinement scheme for a numerical method, there was a large savings in computational cost and memory. For the same error the refined scheme was 15 times faster with a 6 times reduction in memory cost.

The algorithm presented here can easily be used for adaptive node refinement. However, since the differentiation matrices would need to be recalculated every time the node distribution is re-arranged (most likely every couple of time steps) a more cost effective scheme would be highly desirable. Using the refinement algorithm presented in this paper, the authors are currently exploring faster methods to calculate differentiation matrices such as localized RBF stencils [13] and iterative methods.

## Acknowledgments

The authors would like to thank the reviewers for very helpful comments which have significantly improved the quality of the paper.

## References

- [1] M. Bozzini, L. Lenarduzzi, M. Rossini, R. Schaback, Interpolation by basis functions of different scales and shapes, *CALCOLO* 41 (2004) 77–87.
- [2] M.D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge Monographs on Applied and Computational Mathematics, vol. 12, Cambridge University Press, Cambridge, 2003.
- [3] R.E. Wyatt, C.J. Trahan, Radial basis function interpolation in the quantum trajectory method: optimization of the multi-quadric shape parameter, *JCP* 185 (2003) 27–49.
- [4] E.W. Cheney, W.A. Light, *A Course in Approximation Theory*, Brooks/Cole, New York, 2000.
- [5] V. Cherkassky, F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, John Wiley, New York, 1998.
- [6] T.A. Driscoll, A. Heryundono, Adaptive residual subsampling methods for radial basis function interpolation and collocation problems, *Comput. Math. Appl.* 53 (2007) 927–939.
- [7] G.E. Fasshauer, J.G. Zhang, On choosing “optimal” shape parameters for RBF approximation, *Numer. Algorithms* 45 (2007) 345–368.
- [8] N. Flyer, G.B. Wright, Transport schemes on a sphere using radial basis functions, *J. Comput. Phys.* 226 (2007) 1059–1084.
- [9] N. Flyer, G.B. Wright, A simple radial basis function method for the shallow water equations on a sphere, *P. Roy. Soc. A-Math. Phys.* 465 (2106) (2009) 1949–1976.
- [10] B. Fornberg, N. Flyer, S. Hodve, C. Piret, Locality properties of radial basis function expansion coefficients for equispaced interpolation, *IMA J. Numer. Anal.* 28 (1) (2008) 121–142.
- [11] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, *SIAM J. Sci. Comput.* 200 (2007) 178–192.
- [12] B. Fornberg, C. Piret, On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere, *J. Comput. Phys.* 227 (2008) 2758–2780.
- [13] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, *Comput. Math. Appl.* 48 (2004) 853–867.
- [14] B. Fornberg, J. Zuev, The Runge phenomenon and spatially variable shape parameters in RBF interpolation, *Comput. Math. Appl.* 54 (2007) 379–398.
- [15] S. Hubbert, T. Morton,  $l^p$ -error estimates for radial basis function interpolation on the sphere, *J. Approx. Theory* 129 (2004) 58–77.
- [16] A. Iske, *Multiresolution Methods in Scattered Data Modelling*, Lecture Notes in Computational Science and Engineering, vol. 37, Springer-Verlag, Heidelberg, 2004.
- [17] Kurt Jetter, Joachim Stöckler, Joseph D. Ward, Error estimates for scattered data interpolation on spheres, *Math. Comput.* 68 (226) (1999) 733–747.
- [18] E.J. Kansa, R. Carlson, Improved accuracy of multiquadric interpolation using variable shape parameters, *Comput. Math. Appl.* 24 (1992) 99–120.
- [19] W.R. Madych, S.A. Nelson, Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation, *J. Approx. Theory* 70 (1992) 94–114.
- [20] C.A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constr. Approx.* 2 (1986) 11–22.
- [21] R. Nair, C. Jablonowski, Moving vortices on the sphere: a test case for horizontal advection problems, *Mon. Weather Rev.* 136 (2008) 699–711.
- [22] R.D. Nair, S.J. Thomas, R.D. Loft, A discontinuous Galerkin transport scheme on the cubed-sphere, *Mon. Weather Rev.* 133 (2005) 814828.
- [23] M.J.D. Powell, The theory of radial basis function approximation in 1990, in: W. Light (Ed.), *Advances in Numerical Analysis, Wavelets, Subdivision Algorithms and Radial Functions*, vol. II, Oxford University Press, Oxford, UK, 1992, pp. 105–210.
- [24] M.J.D. Powell, Radial basis function methods for interpolation to functions of many variables, DAMTP Report NA11, University of Cambridge, 2001.
- [25] W.M. Putman, S.-J. Lin, Finite-volume transport on various cubed-sphere grids, *J. Comput. Phys.* 227 (1) (2007) 55–78.
- [26] S. Rippa, An algorithm for selecting a good value for the parameter  $c$  in radial basis function interpolation, *Adv. Comput. Math.* 11 (1999) 193–210.
- [27] S.A. Sarra, D. Sturgill, A random variable shape parameter strategy for radial basis function approximation methods, *Eng. Anal. Bound. Elem.* 33 (2009) 1239–1245.
- [28] I.J. Schoenberg, Metric spaces and completely monotone functions, *Ann. Math.* 39 (1938) 811–841.
- [29] I.J. Schoenberg, Positive definite functions on spheres, *Duke Math. J.* 9 (1942) 96–108.
- [30] Anton Sherwood, How can I arrange  $N$  points evenly on a sphere? Website, 2007, <<http://www.ogre.nu/sphere.htm>>.
- [31] H. Wendland, *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, vol. 17, Cambridge University Press, Cambridge, 2005.
- [32] J. Wertz, E.J. Kansa, L. Ling, The role of multiquadric shape parameters in solving elliptic partial differential equations, *Comput. Math. Appl.* 51 (2006) 1335–1348.
- [33] R.S. Womersley, I.H. Sloan, How good can polynomial interpolation on the sphere be?, *Adv. Comput. Math.* 23 (2001) 195–226.
- [34] R.S. Womersley, I.H. Sloan, Interpolation and cubature on the sphere, Website, 2003, <<http://web.maths.unsw.edu.au/rsw/Sphere/>>.
- [35] J. Yoon, Spectral approximation orders of radial basis function interpolation on the Sobolev space, *SIAM J. Math. Anal.* 23 (4) (2001) 946–958.