RBF-Vortex Methods for the Barotropic Vorticity Equation on a Sphere

Jianping Xiao[1], Lei Wang[2], and John P. Boyd[3]

[1]Department of Atmospheric, Oceanic and Space Science, University of Michigan,
2455 Hayward Avenue, Ann Arbor MI 48109-2143

[2]Department of Mathematics, University of Wisconsin at Milwaukee,
wang256@uwm.edu

[3]Department of Atmospheric, Oceanic and Space Science, University of Michigan,
2455 Hayward Avenue, Ann Arbor MI 48109-2143, jpboyd@umich.edu

February 6, 2015

Post-print archive. Published in the *Journal of Comutaational Physics*, vol. 285, 15 March issue, pgs. 208–225 (2015).

Abstract

Vortex blob methods approximate a flow as a sum of many small vortices of Gaussian shape, and adaptively move the vortex centers with the current. Gaussian radial basis functions (RBFs) do exactly the same. However, RBFs solve an exact interpolation problem — expensive but accurate — while vortex methods sacrifice accuracy through *quasi-interpolation* for the absence of a matrix inversion. We show that vortex-RBF algorithms with spectral accuracy are stable for flows on the sphere. The version in Eulerian coordinates is fast; the fully-Lagrangian variant is much slower for a given basis size N, but is highly adaptive for advection-dominated flows. Both versions are excellent for small-to-medium-N problems — N up to 10,000, say, where N is the number of RBF grid points/vortex blobs. Neither is good for large N problems because the cost of the Eulerian model scales as N^2 per timestep while the Lagrangian vortex-RBF method scales as N^3. The slow-Lagrangian scheme is unique among vortex methods in being genuinely (like its Eulerian sibling) a spectrally-accurate method: for laminar flows the error falls exponentially fast with N.

1 Introduction

Large-scale flows in the atmosphere and ocean, and often flows in engineering, too, are dominated by vorticity, which measures the local rotation of the fluid. The physics is conservation of vorticity. However, the conserved vorticity can be wound up into a complex topology of spiral filaments, disk-like vortex patches of same-signed vorticity (monopoles), translating pairs of such patches ("dipoles" or "modons") and so on. In the absence of wave propagation, blobs of fluid are advected by the flow so as to follow this topology. Vortex methods exploit this inherent adaptivity by replaced a fixed grid (that is, a grid in Eulerian coordinates) by a collection of vortex blobs that move with the flow. Vorticity conservation is automatic because each blob conserves its vorticity as it moves. The blobs are a grid in Lagrangian coordinates that discretizes a narrow shear layer or vorticity filament by placing computational elements (vortex blobs) only where there is actually vorticity, dynamically and automatically adapting to the flow even as vortex filaments wind up into tighter and tighter spirals. Cottet and Koumoutsakos' book is a good introduction to vortex methods [23].

It sounds almost too grid to be true. Indeed, difficulties have rendered vortex methods less popular than finite element and spectral methods. Nevertheless, given the dominance of vorticity — the *control* of every aspect of the dynamics by vorticity — and the complicated topology of vorticity, vorticity methods deserve further study.

Radial basis functions (RBFs) are a type of pseudospectral method in which the basis functions are translates and dilations of a simple shape like a Gaussian. Modern vortex modelers commonly use blobs with a Gaussian distribution of vorticity also. In a very real sense, vortex modelers were using RBFs a decade or two before geophysicists and applied mathematicians discovered them.

However, as explained below, a spectrally-accurate Lagrangian coordinate RBF method requires forming, factoring and solving a dense matrix problem at each step. This is so expensive that vortex modelers have traditionally cheated by using quasi-interpolation. Unfortunately, this degrades the spectral method to mediocre finite order accuracy.

In this work, we show that on modern workstations, spectrally-accurate RBF-vortex methods are perfectly feasible for small and medium-sized problems. However, the Eulerian coordinates model, though not a Lagrangian vortex method, scales much better to large N where N is the number of blobs/grid points than Lagrangian models. Vortex methods are very good tools, but only when used selectively.

We illustrate the Eulerian/Lagrangian and interpolation/quasi-interpolation issues by solving the barotropic vorticiy equation on a sphere.

The earliest numerical weather forecasts solved the barotropic vorticity equation on a limited domain [20]. Numerical studies of this partial differential equation on the sphere span a half century from [38] through [25, 52, 66] to the present. The barotropic vorticity equation is actually a system of two partial differential equations: a prognostic equation (for the time rate of change of the vorticity) and a diagnostic elliptic partial differential equation which is the Poisson equation. ("Barotropic means that the flow is independent of height: this is a sensible first approximation for large-scale atmospheric flows although all operational weather forecasting models are fully three-dimensional.)

In Sec. 2, we describe the equations of motion. The following section reviews the history of vortex methods and quasi-interpolation. After a brief introduction to RBFs, the following pair of sections outline the Eulerian coordinate (fixed grid) and Lagrangian coordinate (vortex method) models. In Sec. 8, we introduce two new exact nonlinear Rossby solutions: rotated Legendre polynomials and rotated sectoral harmonics. We then discuss the role of wave physics versus advection and some illuminating numerical examples.

2 Governing equations

The barotropic vorticity equation on the sphere is conservation of the absolute vorticity following the motion,

$$\frac{D\zeta_{abs}}{Dt} = 0 \qquad \text{[Vorticity Eq.]}$$

where the absolute vorticity is

$$\zeta_{abs} \equiv \zeta + 2\Omega\cos(\theta) \qquad \text{[Absolute Vorticity]} \tag{1}$$

and the total, particle-following derivative is

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \frac{u}{a\sin(\theta)}\frac{\partial}{\partial \lambda} + \frac{v}{a}\frac{\partial}{\partial \theta} \tag{2}$$

where a is the radius of the earth, Ω $(= 2\pi/86400s)$ is the angular frequency of the earth's rotation, λ is longitude and θ is colatitude. The velocities are given in terms of the stream function ψ as

$$v \equiv a\,\frac{D\theta}{Dt} = -\frac{1}{\sin\theta}\frac{\partial \psi}{\partial \lambda} \tag{3}$$

$$u = a\,\sin(\theta)\,\frac{D\lambda}{Dt} = \frac{\partial \psi}{\partial \theta} \tag{4}$$

Note that v is the southward velocity, which is the opposite of the usual meteorological convention. The stream function is determined from the vorticity by solving the Poisson equation.

$$\nabla^2\,\psi = \zeta, \qquad \nabla^2\psi = \frac{1}{a^2}\frac{1}{\sin^2(\theta)}\frac{\partial^2\psi}{\partial\lambda^2} + \frac{1}{a^2}\frac{\partial^2\psi}{\partial\theta^2} + \cot(\theta)\frac{1}{a^2}\frac{\partial\psi}{\partial\theta} \tag{5}$$

With our convention, the stream function has its usual meteorological meaning: the latitudinal derivative of ψ gives the *northward* velocity whereas its negative gives the southward velocity employed here.

The vorticity equation can be rewritten as

$$\frac{\partial\zeta}{\partial t} = -\frac{\partial\psi}{\partial\theta}\frac{1}{a\sin(\theta)}\frac{\partial\zeta}{\partial\lambda} + \frac{1}{\sin(\theta)}\frac{\partial\psi}{\partial\lambda}\frac{1}{a}\frac{\partial\zeta}{\partial\theta} - \frac{2\Omega}{a}\frac{\partial\psi}{\partial\lambda} \qquad \text{[Vorticity Eq.]}$$

In the rest of the article, we shall nondimensionalize so that the length scale is a, the earth's radius, 6.36×10^6. Similarly, we nondimensionalize time so that the earth's vorticity, 2Ω, is one. Thus one day is 12.56 nondimensional time units and a unit velocity is equivalent to a speed of 924 m/s.

3 Point vortices, Gaussian blobs and quasi-interpolation

The application of vortex methods to a model whose physics is simply the conservation of vorticity is rather obvious. The earliest numerical simulations of vortex-dominated flow were performed using so-called "point vortices" by Louis Rosenhead in 1931 [61]. A point vortex, also called a "line vortex" or "free vortex", is an exact solution to inviscid, two-dimensional flow in which the vorticity is a Dirac delta function. That is to say, the vorticity is everywhere zero except at a single point where the vorticity is infinite. The delta function is the limit of a Gaussian function of $\vec{x}/\epsilon$ such that integral of the function remains finite as $\epsilon \rightarrow 0$. Half a century earlier, Lord Rayleigh idealized a thin shear layer as a vortex sheet and successfully analyzed the early stages of what is now called the "Kelvin-Helmholtz instability". Rosenhead was able to extend Rayleigh's analysis into the nonlinear regime by replacing the vortex sheet by a line of N point vortices and numerically solving the $2N$ ordinary differential equations in time that describe the positions of the point vortices as they move under their mutual interaction. Although his only tools were paper and pencils plus an adding machine, he was able to show, in agreement with laboratory experiments, that the shear layer rolled up into a spatially periodic train of large vortices, each one with a core composed of a spiriform vorticity filament interlaced with ambient irrotational fluid.

However, point vortices have the obvious disadvantage that a smooth, continuous vorticity is approximated [badly!] by a sum of Dirac delta functions. By 1980, it was common to replace point vortices by so-called "vortex blobs" [21,22]. The most common choice was the Gaussian blob for which the vorticity field is

$$\phi \equiv \left(-\frac{\alpha^2}{h^2} \left\{ (x - x_0)^2 + (y - y_0)^2 \right\} \right) \tag{6}$$

where α is a grid-independent constant, h is the average spacing between blobs, and the center of the blob is the point (x_0, y_0). This is identical in form to the Gaussian radial basis functions employed later. For a smooth vorticity distribution, accuracy requires that the blobs *overlap.* The approximate vorticity field is then (and only then) free of spurious irrotational voids between the vortex peaks. However, the blob overlap has lamentable numerical consequences.

The vorticity equation embodies conservation following the motion for individual *points*, that is, for vortices of infinitesimally small volume. The vorticity of a single Gaussian blob says nothing directly about the strength of the vorticity at a point because this is the sum of the N blobs that overlap at that point where N is the total numbers of blobs. The vorticity of the sum of blobs is conserved, but vorticity of an individual blob is not conserved, but instead varies as the blobs are moved around by the flow. The amplitudes of the blobs must thus be determined by solving an interpolation problem which imposes N conditions that the sum of the vortex blobs at each of N points match the vorticity imposed by conservation at that point. The cost of factoring the

Vandermonde (interpolation) matrix by a direct method is $O([2/3N^3)$ floating point operations.

Consequently, vortex modelers have retreated to "quasi-interpolation". Any set of basis functions can be rearranged into the corresponding "cardinal basis" which satisfies the "cardinal conditions"

$$C_j(x_k) = \left\{ \begin{array}{ll} 1, & j = k \\ 0, & j \neq k \end{array} \right| . \tag{7}$$

The coefficients of the cardinal function series for the interpolant are simply the samples of the function being approximated, $f(x)$, that is, the interpolant is

$$f_N(x) = \sum_{j=1}^{N} f(x_j)\, C_j(x) \tag{8}$$

Quasi-interpolation uses a representation of the same form except that the basis functions satisfy the cardinal conditions only *approximately.* If the centers of the vortex blob basis functions are allowed to move with the flow, the conserved potential vorticity at that moving point is taken as the coefficient of that basis function in the series of vortex blobs that sum to approximate the overall vorticity field. The approximation is only a "quasi-interpolant" because the j-th vortex blob overlaps with its neighbors and in general is nonzero at all of the other blob centers. Indeed, because the blobs overlap, the approximation would seem to be a *zeroth order* method. If the basis functions are scaled as we shall do here by choosing the blobs to be functions of $((\alpha/h)||x - x_j||)$ where α is independent of the average distance between blob centers, then the overlap is independent of h in the limit that $h \rightarrow 0$, and thus cannot be eliminated merely by increasing resolution.

Fortunately, second order accuracy can be restored by redefining the basis function as

$$\varphi_j \equiv \frac{\phi_j(x)}{\sum_{k=1}^{N} \phi_k(x_j)} \tag{9}$$

Then

$$|f(x) - \sum_{j=1}^{N} f_N(x_j)\, \varphi_j(x)| \sim O(1/N^2) \tag{10}$$

Hald and del Prete showed in 1978 that errors could be reduced to $O(h^4)$ by replacing blobs that decay monotonically with distance from their centers by non-monotonic blobs with wisely-chosen zeros that reduce vortex-vortex overlap at the gridpoints. Four years later, Beale and Majda published a theory of arbitrary order quasi-interpolation in one, two or three dimensions [4]. The books by Fasshauer [27] and by Maz'ya and Schmidt (Chapter 1) [55] are good treatments of quasi-interpolation.

Fig. 1 shows that quasi-interpolation is much less accurate than interpolation, which has a geometric rate of convergence for smooth functions with Gaussian basis functions, that is, the error falls as $\exp(-\mu N)$ for some $\mu > 0$ as $N \to \infty$. Almost thirty years ago, J. Thomas Beale asked the obvious question: Was the extra *accuracy* of interpolation worth the extra *cost* of matrix-solving? (He thus applied Gaussian RBF interpolation to solve partial differential equations even before the pioneering work of Edward Kansa [44].) His results were ambiguous. Accuracy was greatly improved for small times, but this did not prevent underresolution failure at large times as the flow developed ever finer and finer spatial scales, and his work never appeared in a journal.

One of the initial goals of this work is to revisit the interpolation/quasi-interpolation question. However, maximum performance for interpolation needs complicated summation schemes for particle interactions (tree codes, fast multipole methods) and indirect matrix-solvers (preconditioned iterations, GPU acceleration [9]). These complications have made interpolation versus quasi-interpolation a complex issue postponed to future articles.

There is an extensive literature on *point* vortex methods on the sphere, but not vortex blobs. Long after point vortices has been abandoned in other areas of fluid mechanics in favor of overlapping vortex blobs of finite size [4, 5, 23, 49, 69], point vortices have continued to be used for flows on a sphere [8, 26, 34, 43, 45–48, 51, 58, 63].

However, most of this work is in the mathematics literature. Because a sum of point vortices is a poor approximation to a smooth, continuous vorticity field, point vortex models have not been taken seriously in geophysical fluid dynamics as a *numerical* method. However, point vortices have been used extensively for *theoretical* purposes on the sphere and various approximations to the sphere [40, 41, 56, 60].

Part of the reason for the "pointy-ness" of vortex methods, that is, the persistence of point vortex schemes, on the sphere was the unavailability of a solution to the Poisson equation with a blob inhomegeneity except for the special case of a point vortex. Recently, Boyd and Zhou [17] filled this gap by discovering an analytical solution for the

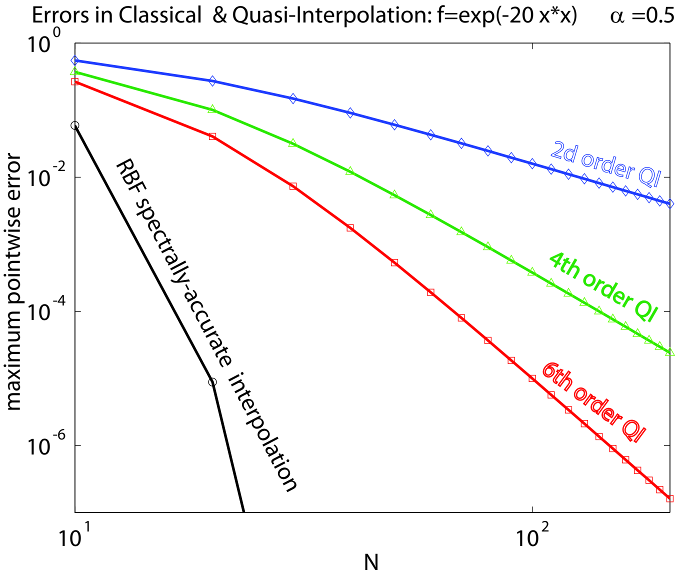

Figure 1: Spectrally-accurate RBF interpolation of a typical function, $f(x) = \exp(-20x^2)$ using RBFs (shape parameter $\alpha = 1/2$) versus quasi-interpolation of different orders as marked. The error is plotted versus N, the number of interpolation points. Both axes are logarithmic so that power laws (error proportional to $1/N^k$ where here k is 2, 4 or 6) appear linear.

Poisson equation when forced by a single Gaussian, which we may dub the "P-function", $\mathcal{P}$. The Poisson equation can then be solved for a general inhomogeneous term by expanding the forcing (vorticity) as a series of radial basis functions; the Poisson solution for the stream function is then a sum of translated copies of $\mathcal{P}$, each multiplied by the RBF coefficient of the vorticity series. This advance has made it feasible to extend vortex methods on the sphere to Gaussian blobs instead of point vortices. However, we shall not discuss it here because Poisson-solutions-for-Gaussian -forcing reduce computer time only when the necessary expansion of the vorticity into an RBF series is performed by quasi-interpolation.

Natasha Flyer, Bengt Fornberg, Grady Wright and collaborators have applied RBFs as a global basis for flows on a sphere or in spherical coordinates in a series of papers [28, 29, 31, 57, 78] all employing an Eulerian coordinate system. We shall make comparisons where appropriate, but because of the rapid growth in cost with the number of degrees of freedom N, Flyer and Fornberg have switched to of RBF-FD [9, 30, 32, 33]. This is identical to the differential quadrature scheme of Bellman, Kashef and Casti [6, 83] except that RBFs, instead of polynomials, are fitted through a few points to approximate the derivative near the center of the point cluster.

We go in a different direction, unrepentantly global, and very much interested in connections with classical vortex methods.

Although adaptive and Lagrangian grids are major themes, disorder needs an orderly starting point. Our was always a grid derived from the spherical icosahedron (Fig. 2). Such grids were first used by Vestine (1963) *et al.* [72] and independently introduced with more details into meteorology five years later by Williamson [76] and Sadourny [62]. Details of the construction are omitted here.

4 Radial Basis Functions

4.1 Radial Basis Functions (RBFs): General Background

In any number of dimensions d, an RBF approximation is of the form [18, 27, 42, 75]

$$f(\vec{x}) \approx \sum_{j=1}^{N} \lambda_j \, \phi\left(||\vec{x} - \vec{x}_j||_2 \right) \qquad\qquad \vec{x} \in R^d \tag{11}$$

for some function $\phi(r)$ and some set of N points $\vec{x}_j$ which for simplicity we assume coincide with the interpolation points. The coefficients λ_j are found by solving a matrix problem

$$\vec{\vec{G}}\,\vec{\lambda} = \vec{f} \tag{12}$$

where $\vec{\lambda}$ is a vector holding the RBF coefficients, the elements of $\vec{f}$ are $f(\vec{x}_k)$, and the elements of the interpolation matrix are

$$G_{jk} \equiv \phi(\vec{x}_j - \vec{x}_k) \tag{13}$$

Although many types of $\phi(r)$ have been used, we will restrict ourselves to Gaussians in Cartesian coordinates,

$$\phi(r) \equiv \exp(-\epsilon^2 r^2) \qquad \text{[Gaussian RBF]} \tag{14}$$

where $\epsilon > 0$ is a user-choosable constant, the "absolute inverse width". We will use this definition in spherical coordinates, too, but when all points are confined to the surface of a sphere, nondimensionalized to a sphere of unit radius, the Gaussian RBF can written without approximation as a function that depends only on longitude and colatitude as described in the next subsection.

4.2 Distance on the Sphere

There are two popular ways to measure distance on the sphere. The first is the *Euclidean* distance, which is the distance between two points in ordinary three-dimensional space; the Euclidean distance is the length of a line segment connecting the two points which passes *through* the *interior* of the sphere. The second distance metric is the *geodesic* distance, which is the length of the shortest arc along the *surface* of the sphere which connects the two points.

In the RBF literature, the Euclidean distance is more common [30] and is the definition employed here:

$$\begin{aligned} r_E &= ||\vec{x} - \vec{x}_j|| && (15) \\ &= \sqrt{(\vec{x} - \vec{x}_j)^T \bullet (\vec{x} - \vec{x}_j)} && (16) \\ &= \sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2} && (17) \\ &= \sqrt{2\,(1 - \sin(\theta)\sin(\theta_j)\cos(\lambda - \lambda_j) - \cos(\theta)\cos(\theta_j)\,)} && (18) \end{aligned}$$

Using the Eulerian distance, the series and the j-th Gaussian RBF can be written as

$$f^{RBF}(\lambda, \theta) \approx \sum_{j=1}^{N} \lambda_j \, \phi_j(\lambda, \theta) \tag{19}$$

$$\phi_j(\lambda, \theta) = \exp\left\{-\epsilon^2\, 2\,(1 - \sin(\theta)\sin(\theta_j)\cos(\lambda - \lambda_j) - \cos(\theta)\cos(\theta_j)\,)\right\} \tag{20}$$

4.3 Coordinate Rotation

The following theorem will be exploited repeatedly in later sections.

Theorem 4.1 (Coordinate Rotation)**.** *Suppose $f(\lambda, \theta)$ is a radially symmetric function on the sphere in the sense that*

$$f(\lambda, \theta; \lambda_j, \theta_j) = f(||\vec{x} - \vec{x}_j||) \tag{21}$$

for point x_j; thus if the coordinate system is rotated to make $\vec{x}_j$ the new north pole, f

is independent of longitude and may be written

$$f(\lambda, \theta; \lambda_j, \theta_j) = f'(\xi) \tag{22}$$

for some function f' that depends only on the new colatitude ξ. Then in terms of the original coordinate system,

$$f(\lambda, \theta; \lambda_j, \theta_j) = f'(\xi(\lambda, \theta; \lambda_j, \theta_j)) \tag{23}$$

where

$$\cos(\xi) = \cos(\theta)\,\cos(\theta_j) + \sin(\theta)\sin(\theta_j)\,\cos(\lambda - \lambda_j) \tag{24}$$

Proof: elementary consequence of the classical formulas for rotation of spherical coordinates.

5 An RBF model in Eulerian coordinates

RBFs can be applied to solve the barotropic vorticity equation even when the interpolation points are fixed in time. Such an approach is usually called an "Eulerian" approach because the Eulerian coordinate system for fluids is one in which the equations track the dynamics at fixed spatial locations. The vortex method is a "Lagrangian" coordinate approach in the sense that the dynamics is tracked at points that move with the flow. (Note that latitude and longitude will be used to describe the flow in both Eulerian and Lagrangian coordinates.)

To build the Eulerian model, it is convenient to define the Green's operator for the Poisson operator by

$$\nabla^2 \psi = \zeta \qquad \leftrightarrow \qquad \psi \equiv \mathcal{G}\zeta \qquad \text{[Green's operator]}$$

We can then formally write the barotropic vorticity model as a single equation for the relative vorticity as

$$\frac{d\zeta}{\partial t} = -\frac{\partial(\mathcal{G}\zeta)}{\partial\theta}\frac{1}{\sin(\theta)}\frac{\partial\zeta}{\partial\lambda} + \frac{1}{\sin\theta}\frac{\partial(\mathcal{G}\zeta)}{\partial\lambda}\frac{\partial\zeta}{\partial\theta} - 2\Omega\cot(\theta)\frac{\partial(\mathcal{G}\zeta)}{\partial\lambda} \qquad \text{[Vorticity Eq.]} \tag{25}$$

In discrete form, the Green's operator is a matrix $\vec{\vec{G}}$ which is the *inverse* of the matrix $\vec{\vec{P}}$ which discretizes the Poisson equation with elements P_{ij}. To evaluate this efficiently, we use the coordinate trick. Let us specialize to $\theta_j = 0$, i. e., the vector $\vec{x}_j$ is the north pole; λ_j is then irrelevant. Let ξ be colatitude. Then

$$\nabla^2 \phi_0(\xi;\epsilon) = 4\epsilon^2 \left\{ \epsilon^2 - \cos(\xi) - \epsilon^2 \cos^2(\xi) \right\} \exp(-2\epsilon^2(1-\cos(\xi)) \tag{26}$$

where $\phi_0 = \exp(-2\epsilon^2(1-\cos(\theta)))$. The coordinate rotation theorem then implies

$$P_{ij} \quad = \quad 4\epsilon^2 \left\{ \epsilon^2 - \Xi_{ij} - \epsilon^2 \Xi_{ij}^2 \right\} \exp(-2\epsilon^2(1-\Xi_{ij})) \tag{27}$$

where

$$\Xi_{ij} = \cos(\theta_i)\,\cos(\theta_j) + \sin(\theta_i)\sin(\theta_j)\,\cos(\lambda_i - \lambda_j) \tag{28}$$

Let $\vec{a}_\zeta$ and $\vec{a}_\psi$ denote the RBF coefficients of ζ and ψ, respectively, and let $\vec{\zeta}$ and $\vec{\psi}$ denote the values of these variables at the interpolation points. Then

$$\vec{\vec{P}}\,\vec{a}_\psi = \vec{\zeta} \qquad \leftrightarrow \qquad \vec{a}_\psi = \vec{\vec{P}}^{-1}\vec{\zeta} = \vec{\vec{G}}\vec{\zeta} \tag{29}$$

To convert from RBF coefficients to the grid point values of the same function, we use a matrix whose elements are ϕ_j, where j is the *column* index, evaluated at the point λ_i, θ_i where i is the row index:

$$D_{ij}^0 \equiv \phi_j(\lambda_i, \theta_i) = \exp\left\{ -\epsilon^2\, 2\,(1 - \sin(\theta_i)\sin(\theta_j)\cos(\lambda_i - \lambda_j) - \cos(\theta_i)\cos(\theta_j)\,)\right\} \tag{30}$$

To convert from grid point values to RBF coefficients, we use the inverse of $\vec{\vec{D}}_0$ which we shall denote by $\vec{\vec{Q}}$: thus

$$\vec{\zeta} = \vec{\vec{D}}^0 \vec{a}_\zeta \qquad \vec{a}_\zeta = \vec{\vec{Q}}\vec{\zeta} \tag{31}$$

To evaluate the derivatives,

$$\begin{aligned} D_{ij}^\lambda \quad &\equiv \quad \frac{\partial \phi_j}{\partial \lambda}(\lambda_i, \theta_i) = \exp\left\{ -\epsilon^2\, 2\,(1 - \sin(\theta_i)\sin(\theta_j)\cos(\lambda_i - \lambda_j) - \cos(\theta_i)\cos(\theta_j)\,)\right\} \\ & \qquad \times(-2\epsilon^2)\sin(\theta_i)\sin(\theta_j)\sin(\lambda_i - \lambda_j) \\ &= \quad -2\epsilon^2 \sin(\theta_i)\sin(\theta_j)\sin(\lambda_i - \lambda_j)\, D_{ij}^0 \end{aligned} \tag{32}$$

$$\begin{aligned} D^{\theta}_{ij} &\equiv \frac{\partial \phi_j}{\partial \theta}(\lambda_i, \theta_i) = \exp\left\{-\epsilon^2 \, 2\,(1 - \sin(\theta_i)\sin(\theta_j)\cos(\lambda_i - \lambda_j) - \cos(\theta_i)\cos(\theta_j))\right\} \\ &\quad \times(-2\epsilon^2)\ \{-\cos(\theta_i)\sin(\theta_j)\cos(\lambda_i - \lambda_j) + \sin(\theta_i)\cos(\theta_j)\} \\ &= D^0_{ij} 2\epsilon^2 \{\cos(\theta_i)\sin(\theta_j)\cos(\lambda_i - \lambda_j) - \sin(\theta_i)\cos(\theta_j)\} \end{aligned} \tag{33}$$

It follows that, using an arrow to indicate that the derivative of the continuous function ζ is discretized by the vector to the right of the arrow,

$$\frac{\partial \zeta}{\partial \lambda} \rightarrow \vec{\vec{D}}^{\lambda} \vec{a}_{\zeta} = \vec{\vec{D}}^{\lambda} \, (\vec{\vec{D}}^{0})^{-1} \vec{\zeta} \tag{34}$$

$$\frac{\partial \zeta}{\partial \theta} \rightarrow \vec{\vec{D}}^{\theta} \vec{a}_{\zeta} = \vec{\vec{D}}^{\theta} \, (\vec{\vec{D}}^{0})^{-1} \vec{\zeta} \tag{35}$$

To compute derivatives of the stream function, $\psi = \mathcal{G}\zeta$, recall that $a_{\psi} = \vec{\vec{G}}\vec{\zeta}$ where $\vec{\vec{G}}$ is the inverse of the matrix $\vec{\vec{P}}$ defined earlier. To get the grid point values of the derivatives of the stream function, we can therefore write

$$\frac{\partial \psi}{\partial \lambda} \rightarrow \frac{\partial \mathcal{G}\zeta}{\partial \lambda} = \vec{\vec{D}}^{\lambda} \vec{\vec{G}} \vec{\zeta} \tag{36}$$

$$\frac{\partial \psi}{\partial \theta} \rightarrow \frac{\partial \mathcal{G}\zeta}{\partial \theta} = \vec{\vec{D}}^{\theta} \vec{\vec{G}} \vec{\zeta} \tag{37}$$

We use $\odot$ to denote the "Hadamard product", which is the element-wise multiplication of two vectors, that is, $\vec{a} \odot \vec{b}$ is a vector whose components are $\{a_1 b_1, a_2 b_2 \ldots a_N b_N\}$. (The familiar Matlab notation for this operation is Matlab ".$*$.)

The discrete velocities are

$$\vec{u} = \vec{\vec{D}}^{\theta} \vec{\vec{G}} \vec{\zeta} \tag{38}$$

$$\vec{v} = -\frac{1}{\sin(\vec{\theta})} \odot \vec{\vec{D}}^{\lambda} \vec{\vec{G}} \vec{\zeta} \tag{39}$$

The spatially-discretized vorticity equation in Eulerian coordinates is

$$\frac{d\vec{\zeta}}{\partial t} = -\vec{u} \odot \frac{1}{\sin(\vec{\theta})} \odot (\vec{\vec{D}}^{\lambda} \, \vec{\vec{Q}} \vec{\zeta}) - \vec{v} \odot \vec{\vec{D}}^{\theta} \, \vec{\vec{Q}} \vec{\zeta} - (\vec{\vec{D}}^{\lambda} \, \vec{\vec{G}} \vec{\zeta}) \qquad \text{[Vorticity Eq.]} \tag{40}$$

where $1/\sin(\vec{\theta})$ is used a shorthand for the diagonal matrix whose elements are $1/\sin(\theta_i)$.

6 The Lagrangian Vortex/RBF Algorithm

Note that this algorithm does not exploit the analytical Poisson solution of Boyd and Zhou [17].

Preprocessing steps:

1. generate an initial grid: (λ_j, θ_j)
2. specify the initial relative vorticity $\vec{\zeta})_j = \zeta(\lambda_j, \theta_j, 0)$
3. generate the initial (and always conserved) absolute vorticity $\vec{\zeta}_{abs} = \vec{\zeta} + \cos(\vec{\theta})$

Define the vectors of length $2N$,

$$\vec{\Lambda} = \left(\begin{array}{ccc} \vec{\lambda} & , & \vec{\theta} \end{array} \right) \tag{41}$$

$$\vec{V} = \left(\begin{array}{c} \vec{u}./\sin(\vec{\theta}) \\ \vec{v} \end{array} \right) \tag{42}$$

In the time loop, we apply a Runge-Kutta or Adams-Bashforth method to solve the system of $2N$ first ODEs in time,

$$\frac{d\vec{\Lambda}}{dt} = \vec{V} \tag{43}$$

To calculate the velocities, the subroutine that computes the right-hand side of the ODE system must do the following:

1. split $\vec{\Lambda}$ into two smaller parts: $\vec{\lambda} = \vec{\Lambda}(1:N), \vec{\theta} = \vec{\Lambda}(N+1:2N)$
2. evaluate the relative vorticity $\vec{\zeta} = \vec{\zeta}_{abs} - \cos(\vec{\theta})$
3. compute the vector whose elements are $1/\sin(\theta_j)$
4. Generate matrices:
   $\vec{\vec{P}}$: RBF discretization of the Poisson equation such that $\vec{\vec{P}} a_{\psi} = \vec{\zeta}$
   $\vec{\vec{D}}^{\lambda}$: λ derivative at grid points from RBF coefficients so that $\partial\zeta/\partial\lambda$
   at the grid points is given by $\vec{\vec{D}}^{\lambda} \vec{a}_{\zeta}$
   $\vec{\vec{D}}^{\theta}$: θ derivative at grid points from RBF coefficients so that $\partial\zeta/\partial\theta$
   at the grid points is given by $\vec{\vec{D}}^{\theta} \vec{a}_{\zeta}$
5. Compute the stream function by solving $\vec{\vec{P}} \vec{\psi} = \vec{\zeta}$ [$O(N^3)$ ops., [rate-dtermining step]
6. Evaluate the velocities by $\vec{u} = \vec{\vec{D}}^{\theta} \vec{\psi}$ and $\vec{v} = -(1/\sin(\vec{\theta})) \odot \vec{\vec{D}}^{\lambda} \vec{\psi}$
7. Combine $D\vec{\lambda}/Dt = \vec{u}./\sin(\vec{\theta})$ and $\vec{v}$ into the single vector $\vec{V}$

Post-processing and graphs:

1. Compute the RBF interpolation matrix $\vec{\vec{D}}_0$, whose inverse transforms grid point values to RBF coefficients.
2. Solve $\vec{\vec{D}}_0 \vec{a}_\zeta = \vec{\zeta}$.
3. Sum RBF series on a "graphing grid" $(\lambda_j^g, \theta_j^g)$ which is unrelated to the interpolation grid.
4. plot graphs

RBF methods are so accurate that it is often possible to obtain acceptable answers with just a handful of points. Graph software, alas, employs piecewise linear interpolation or other low order schemes. To make a good graph, it is therefore necessary to supply the graphing code with a very fine array. It is therefore common to make graphs by evaluating the RBF series on a fine grid unrelated to the original interpolation grid.

7 Choosing RBF Width

In applications of Gaussian RBFs in other contexts, the absolute RBF width ϵ is irrelevant; what matters is the *relative* inverse width parameter α, defined by

$$\alpha \equiv h\epsilon \tag{44}$$

where h is some measure of the local grid spacing. On the Sadourny icosahedral grid, we will define h to be the shortest distance between a grid point and its nearest neighbor.

Fig. 3 shows that except for the outlier of the coarsest grid, $N = 12$, the optimum choice of the relative inverse width parameter is roughly $\alpha = 1/3$, independent of N.

8 New Flocks of Rossby-Haurwitz Waves

8.1 Flocks by Rotation

As usually defined, a Rossby-Haurwitz wave is an exact nonlinear solution of the barotropic vorticity equation whose stream function and vorticity are both proportional to a single spherical harmonic Y_n^m. The wave translates steadily at a constant phase speed, $c = -1/(n(n+1)$ in our nondimensional units or $c = -n(n+1)2\Omega/a$

in dimensional form. Here we apply five classical theorems to generate a new class of explicit test cases for the barotropic vorticity equation.

Theorem 8.1. *1. If the vorticity is proportional to the stream function, that is, the ratio ζ/ψ is a constant, then the Jacobian operator is zero and the nonlinear term in the barotropic vorticity equation vanishes.*

2. The spherical harmonic Y_n^m is an eigenfunction of the Laplacian operator on the surface of a sphere with an eigenvalue that depends only on the subscript n

$$\nabla Y_n^m = -n(n+1)Y_n^m \tag{45}$$

3. It follows from the first two propositions and the Rossby-Haurwitz solution that any linear combination of the $2n+1$ spherical harmonics of the same subscript n will translate at the steady westward phase speed $c = -1/(n(n+1)$. We shall dub such a conglomeration of Rossby-Haurwitz solutions translating in synchrony like migrating birds a "Rossby-Haurwitz flock".

4. When the spherical coordinate system is rotated, a spherical harmonic (in the old coordinates) is exactly given by a sum of the $2n+1$ spherical harmonics of the same subscript n.

These theorems have been known for decades (Chapter 18 of [12]). Barrett [3] constructed and applied such rotated Rossby-Haurwitz waves more than half a century ago. Two decades later, Tribbia and Verkley used rotated spherical harmonics to construct nonlinear modons on the sphere [70, 71]

To generate new Rossby-Haurwitz flocks by rotating spherical harmonics and Legendre polynomials, we need the following.

Theorem 8.2. *If spherical coordinates are transformed by rotating through an angle of $-\Xi$ about the y-axis, then*

$$\theta' = \arccos\left(\sin(\mathit{\Xi})\sin(\theta)\cos(\lambda) + \cos(\mathit{\Xi})\cos(\theta)\right) \tag{46}$$

$$\lambda' = \arcsin\left\{\frac{\sin(\lambda)\,\sin(\theta)}{\sin(\arccos\left(\sin(\mathit{\Xi})\sin(\theta)\cos(\lambda) + \cos(\mathit{\Xi})\cos(\theta)\right))}\right\} \tag{47}$$

If a function $f(\lambda,\theta)$ is thus transformed, then its form after rotation is

$$g(\lambda,\theta) = f(\lambda'(\lambda,\theta), \theta'(\lambda,\theta)) \tag{48}$$

8.2 Rotated Legendre Polynomials

Legendre polynomials in the cosine of colatitude are the axisymmetric spherical harmonics, that is, $P_n = \text{constant} Y_n^0$. Sums of spherical harmonics with a common subscript n, that is, a flock of Rossby-Haurwitz waves of different zonal wavenumbers but traveling at a common speed, can be generated by rotating the Legendre polynomial $P_n(\cos(\theta)$. Such a flock will have no time dependence except for a steady translation.

Rotating coordinates so that (λ_p, θ_p), instead of $\theta = 0$, is the north pole, the Legendre polynomial $P_n(\cos(\xi))$ becomes

$$\mathcal{L} = P_n(\cos(\theta)\ \cos(\theta_p) + \sin(\theta)\sin(\theta_p)\ \cos(\lambda - \lambda_p)) \tag{49}$$

For example,

$$P_2(\mu) = (3/2)\mu^2 - (1/2) \tag{50}$$

with $c = -1/6$ with our usual nondimensionalization gives the traveling wave

$$\psi(\lambda - ct, \theta) = (3/2)\left\{\cos(\theta)\ \cos(\theta_p) + \sin(\theta)\sin(\theta_p)\ \cos(\lambda - \lambda_p)\right\}^2 - (1/2) \tag{51}$$

for arbitrary (λ_p, θ_p).

Similarly,

$$P_3(\mu) = (5/2)\mu^3 - (3/2)\mu \tag{52}$$

with $c = -1/12$ gives the traveling wave

$$\begin{aligned}\psi(\lambda - ct, \theta) \quad &= \quad (5/2)\left\{\cos(\theta)\ \cos(\theta_p) + \sin(\theta)\sin(\theta_p)\ \cos(\lambda - \lambda_p)\right\}^3 \\ &\quad -(3/2)\left\{\cos(\theta)\ \cos(\theta_p) + \sin(\theta)\sin(\theta_p)\ \cos(\lambda - \lambda_p)\right\}\end{aligned}$$

for arbitrary (λ_p, θ_p).

8.3 Rotated Sectoral Harmonics

The lowest spherical harmonic for a given zonal wavenumber m is Y_m^m, which is given the special name of a "sectoral harmonic". For large m, the sectoral harmonic is approximated by a Gaussian in colatitude, multiplied by the usual sinusoidal factor in longitude as illustrated in Fig. 18.8 of [12]. If this is rotated, we obtain the more general family

$$\begin{aligned} \psi(\lambda,\theta) &= \left[1 - \{\cos(\theta)\,\cos(\Xi) + \sin(\theta)\sin(\Xi)\,\cos(\lambda)\}^2\right]^{m/2} \qquad (53) \\ &\times \cos\left(m \arcsin\left\{\frac{\sin(\lambda)\,\sin(\theta)}{\sin(\arccos\left(\sin(\mathit{\Xi})\sin(\theta)\cos(\lambda) + \cos(\mathit{\Xi})\cos(\theta)\right)}\right\}\right) \end{aligned}$$

where m is an integer for arbitrary constant Ξ. This translates steadily at $c = -1/(m(m+1))$.

9 Waves and Adaptation

9.1 Inviscid Burgers Equation Example

Lagrangian coordinate models are automatically and dynamically adaptive when the flow is dominated by advection. Unfortunately, waves can diminish or even complete destroy this property. We shall not offer numerical case studies because the issues can be demonstrated with a model even simpler than the barotropic vorticity equation: the inviscid Burgers equation with a linear wave propagation term:

$$u_t + u u_x = -c u_x \qquad \text{[Modified Burgers' equation]} \qquad (54)$$

where c is a constant. To apply the equivalent of a vortex method, move each grid point x_j with the velocity u, thus implicitly incorporating the effects of advection. Denoting the timestep by τ and using a superscript n to denote the time level, we employed

$$u_j^{(n+1)} = u_j^{(n-1)} + 2\tau(-c u_{x,j}^{(n)}) \qquad (55)$$

$$x_j^{(n+1)} = x_j^{(n)} + (1/2)\tau(u_j^{(n+1)} + u_j^{(n)}) \qquad (56)$$

Note that in a Lagrangian approach, the index j labels moving blobs of vorticity, not a fixed spatial location. The crucial point is not the mechanics of the scheme, but the consequences of different choices of the linear phase speed c.

When $c = 0$, the velocity u is conserved following the motion and the grid automatically and dynamically adapts to concentrate grid points right where they are needed. When $c \neq 0$, however, the developing front does not move only by advection, but moves at $u(x,t) + c$. There is no systematic trend in the movement of the grid points. Instead of becoming triple-valued with a high concentration of grid points in the frontal zone, the algorithm produces a grid that is almost as uniform as the initial grid and a single-valued solution that is contaminated by large spurious oscillations as illustrated in Fig. 4.

This illustrates longstanding *mokita*[1] in the vortex method community: Lagrangian methods are intrinsically adaptive only for *advection-dominated* flows.

10 Numerical Examples

10.1 Rossby-Haurwitz Waves

On a more positive Lagrangian note, Fig. 5 shows the trajectories of the grid points during the numerical integration of a Rossby-Haurwitz wave by the Lagrangian algorithm: the trajectories are smooth and do not degrade accuracy, which is very high even with just twelve basis functions. The exact solution is independent of time except for a steady westward propagation. Table 1 shows that accuracy for these smooth solutions increases exponentially fast with the number of basis functions.

Table 1: Rossby-Haurwitz: $\psi = a\sin(\theta)\cos(\lambda - t/2)$. The relative error is the L_∞ norm of the absolute error divided by the L_∞ norm of the relative vorticity.

N	model	t_{end}	α	ϵ	no. timesteps	a	rel. error
12	Lagrangian	4π	1/4	0.2378	200	0.5	1.55E-5
12	Lagrangian	4π	1/4	0.2378	400	0.5	1.53E-5
42	Lagrangian	4π	1/4	0.4575	200	0.5	5.63E-7
42	Lagrangian	4π	1/4	0.4575	400	0.5	2.23E-8
42	Lagrangian	4π	1/4	0.4575	800	0.5	1.26E-9
42	Lagrangian	4π	1/4	0.4575	1600	0.5	5.17E-10
92	Lagrangian	4π	1/4	0.4575	200	0.5	0.00145
92	Lagrangian	4π	1/4	0.4575	1600	0.5	2.99E-8

10.2 Vortex merger on a rotating sphere

When two corotating vortices of equal strength are far apart, their mutual interaction is a slow orbit around a point halfway between them. This is the Fujiwhara Effect, often observed in Pacific typhoons, but seen (less frequently) in both tropical and

[1] "Mokita" is a word in a language of Papua New Guinea which means "something we all know but refuse to talk about".

extra-tropical cyclones throughout the globe. When cyclones are sufficiently close, the straining fields of one vortex, differentially advecting the other and vice-versa, stretches filaments of vorticity until these wrap around one another. The cores merge; the outer edges are flung off as thin, spiral filaments ("braids"). Thus, hurricanes Francesca and Gretchen merged in the northeast Pacific in 1974, Typhoon Pat and Tropical Storm Ruth completed a full orbit around their centroid before merging in 1994 in the northwest Pacific and Tropical Cyclone Fame and Tropical Cyclone Guia orbited and then merged in the southwest Indian Ocean in 2008, three mergers among many such recorded events.

Levy, Nair and Tufo [52] and Shin, Han and Baik [65] numerically simulated two vortices of the same sign of vorticity on a rotating sphere. Fig. 6 shows as expected that when two vortices are close enough, they merge; when sufficiently far apart, they merely orbit.

Merger is a severe challenge because the vortices are stretched and wrapped around each other, but our vortex-RBF method had no difficulties even at modest resolution.

10.3 Vortex roll-up on a rotating sphere

The initial thin line of vorticity is an idealization of a latitudinally varying jet such as the East African Jet whose instabilities are father and mother to Atlantic hurricanes. Vortex methods are extremely efficient because the vortex blobs cover only a tiny fraction of the globe. Nevertheless, Kelvin-Helmholtz roll-up into a chain of large, filament-cored vortices is captured in exquisite, spiriform detail as shown in Fig. 7.

10.4 Accuracy, cost and vortex regridding

The number of floating point operations to factor a dense $N \times N$ matrix is approximately $(2/3)N^3$ operations. Since it is necessary to form and factor a new RBF interpolation on each time step as the grid points move with the flow, one expects that the running time of the Lagrangian coordinates model will scale with N proportional to N^3. The left panel of Fig. 8 shows two important things.

The first is that the Eulerian coordinates model is much cheaper than the Lagrangian model. This is hardly surprising because the grid points are fixed; the factorization of the dense RBF interpolation is done only once. The Eulerian model has a per-time step cost that is proportional to N^2.

The second is that for the range of N typical of our desktop experiments $O(10^3)$, the cost of the Lagrangian model grows only slightly faster than N^2. How is this possible?

The right panel of the same figure provides the answer. On contemporary computer architecture, small matrices are manipulated at a slower rate of floating point operations per second (flops) because small sizes cannot fully exploit the multiple caches and vector-processing tricks. The two guidelines proportional to N^2 and N^3 show that the factorization cost grows roughly as N^2 for $N \leq 1000$ even though the number of floating point operations is rising proportional to N^3 because of increasing efficiency and flops rate as data is processed in larger and larger chunks. The expected cubic time/per time step growth is observed for $N > 3000$ after the desktop has reached its peak speed, increasing no further for $N > 3000$.

11 Interpretation

The Lagrangian coordinate vortex-RBF model has two forms of merit. First, it automatically adapts the grid for advection-dominated flow. Second, advection is solved in Lagrangian fashion by implicitly solving a Riemann problem. However, it is possible to regrid after every time step so as to move the grid points through the predicted displacements over that time step. This is not as efficient as the vortex-RBF method, which is fully Lagrangian. However, regridding does produce a similar adaptivity.

Irrevocably lost is the merit of solving advection via a Riemann problem. It therefore behooves us to determine exactly what are the advantages of solving advection as a Riemann problem.

The Lagrangian vortex method does enforce strict conservation of absolute vorticity. However, the vorticity field is still erroneous. In the first place, the trajectories of the centers do not quite move as the true trajectories, so each interpolation has, as it were, the right absolute vorticity in slightly the wrong place. Furthermore, if we plot the vorticity field on a fine grid, we make interpolation errors, partly an N-dependent mathematical interpolation error and partly an error that arises because the trajectories are not exact, and the true vorticity at the interpolation points is slightly different from the conserved values. However, except for the trajectory errors,the Lagrangian errors are *non-secular* on each blob in the sense that the vorticity on each Lagrangian blob is exactly conserved.

In contrast, the errors in an Eulerian coordinate model can accumulate steadily over time. After a million timesteps, the total enstrophy (squared vorticity averaged over the sphere) may have drastically changed even with high spatial resolution and high order time-marching. The enstrophy will be almost perfectly conserved in a Lagrangian model.

The Lagrangian approach has few advantages for small amplitude wave propagation, however. A wave pulse moves fast on a path controlled by refraction, reflection and diffraction. The Lagrangian interpolation points execute tiny closed (or nearly closed) orbits. Adaptation may be helpful to cluster grid points in wave caustics. Advection does not move the interpolation points very much when the waves are small, and thus is useless for wave adaptation.

12 Topics for the Future

12.1 Abrupt versus continuous variations in resolution

Many contemporary finite volume and spectral elements require conformal subdomains or grid boxes, requiring that the resolution jump *discontinuously* by a factor of two. The conventional wisdom has been that this is a Bad Thing [39] Zukas and Scheffler [84]). The last have summarized the consensus: “Care must be taken not to change element sizes too rapidly, however. A change in element size represents a change in element stiffness even if the material properties remain the same. Traveling waves, when encountering this stiffness difference, will act as if an impedance mismatch had occurred. Part of the wave will be reflected, part transmitted. If element-to-element size variation is kept below 10 %, acceptable results can generally be achieved.”

A WKB-argument provides a more precise answer: when a wave of wavescale L propagates through a slowly-varying medium whose scale of variation is M— whether the variations are due to changes in numerics or the physical index of refraction — then the reflection is proportional to $\exp(-q/\epsilon)$ where $\epsilon = L/M$ and the constant q can be calculated only by some form of hyperasymptotic (“beyond-all-orders”) perturbation theory [10, 11, 13]. Spurious reflection can thus be controlled — made exponentially small — by varying the grid sufficiently slowly.

In contrast, RBFs, because of their meshlessness, allow smooth resolution variations on completely unstructured grids; grid-halving is unnecessary on meshless grids.

12.2 Scaling with degree and remedies: PUM, treecodes, iteration and GPUs

The main drawback of our present codes is that the cost grows as the *square* [Eulerian fixed grid version] or *cube* [vortex-RB] of the number of grid points. A laptop can quickly do simulations with several thousand points, and this is sufficient for process modeling [illustrated by vortex merger and the filamentation of a single vortex here] and for classroom laboratories, engineering consulting and so on. However, 10,000 grid points is pitifully small compared to current operational weather forecasting models where a million points on each of a hundred vertical levels is routine.

There are at least six remedies: (i) dimensional restriction (ii) RBF-FD (iii) Partition-of-Unity (PUM) subdivision (iv) massive parallelism (v) treecodes and (vi) preconditioned iterations.

Wright, Flyer and Yuen [78] employed dimensional restriction. This model of mantle convection in an earth with non-spherical boundaries employed RBFs on horizontal layers with a Chebyshev pseudospectral basis in the radial direction. Each latitude-longitude series involved only a moderate number of RBFs, interacting only with RBFs on the same layer.

This is somewhat special; because of the rapid growth in cost with the number of degrees of freedom N, Flyer and Fornberg have abandoned global bases in favor of "radial basis function finite differences" (RBF-FD) [9, 30, 32, 33]. These are identical to the "differential quadrature" scheme of Bellman, Kashef and Casti [6, 83] except that RBFs, instead of polynomials, are fitted through a few points to approximate the derivative near the center of an irregular point cluster. But this is no longer a spectral method though still high order.

12.3 Partition-of-Unity (PUM)

In the context of our model, a PUM strategy begins with a window function $\mathfrak{w}_j$ that is one over a small portion of the sphere and zero elsewhere with an annular smoothing layer between the two. A set of such overlapping windows can be made into a Partition-of-Unity by modifying the j-th window to

$$\mathfrak{W}_j = \frac{\mathfrak{w}_j}{\sum_k \mathfrak{w}_k} \tag{57}$$

The resulting set of windows has the PUM property in the sense that (i) $\sum_k \mathfrak{W}_j(x,y) = 1$ for all x and y, which implies that (ii) the spatially localized, windowed functions $f_j(x,y)$ *exactly* reproduce the global function $f(x,y)$::

$$f(x,y) = \sum_j f_j \tag{58}$$

$$f_j \equiv \mathfrak{W}_j f(x,y) \tag{59}$$

Thus, a Partition of Unity Method (PUM) allows a single large interpolation matrix (such as used in this article) to be replaced by smaller matrices [7,59,68,74]. The idea is to replace the global interpolation problem, or the discretization of a partial differential equation on a global domain, by a set of subproblems. Each subproblem is obtained by multiplying the global inhomogeneous term ζ by the j-th window function $\mathfrak{W}_j$ of the PUM. The window is negligibly small over most of the global domain by construction, so each subproblem can be solved in a *small subdomain* with a *small* number of grid points. With M windows, an $O(N^3)$ global problem is replaced by M subproblems of size $O(N/M)$ at a cost that is $O(N^3/M^2)$. Fuselier *et al.* [35–37] and Cavoretto and De Rossi [19] have made a good start at applying PUM , but there are many practicalities still in flux. What is the optimum subdomain size? Best window function? How does nonlinear coupling change the answers? And so on.

12.4 Regridding and RBF Technologies

An active research frontier is to accelerate RBF interpolation. Cost can also be greatly reduced by using fast multipole methods and treecodes such as [1,50,81] to sum the "far field" parts of RBF series. Unfortunately, treecodes are not accelerative for interactions between neighboring RBFs [15].

Preconditioned iterations are also strongly accelerative [1, 53, 54, 64, 82]. Barba and Yokota argue that these improvements are very well-suited to emerging computer architectures [2]. The only drawbacks of these improvements are that (i) both treecodes and preconditioners are still evolving and (ii) the software engineering tasks are considerable. The obvious preconditioner is a sparse matrix that retains only nearest-neighbor interactions, but this is ineffective. Practical conditioners are matrices that incorporate a hierarchy of interactions on different scales and therefore tedious to construct.

Graphic Processing Units (GPUs) have added multiple processors and parallelism to a single desktop machine, but also have been profitably combined with massively parallel computing clusters. Bollig, Flyer and Erlebacher [9] and Yokota and Barba [2, 81, 82] have shown that spectrally-accurate RBFs play nicely with GPUs, but this is "early days".

Thus, the restriction to a few thousand RBF vortex blobs is not intrinsic; rather, the number of RBFs N can be enormously increased by using technologies already developed. The downside is that the simplicity of RBFs, which can solve an elliptic PDE with half a page of code, is buried under a blizzard of subroutines.

Because other research groups have taken up the challenges of treecodes, GPUs and so on, we have chosen to orient our future plans in a different direction.

12.5 RBF paintbrush and topologically-complex adaptation

For many years, a common practice in operational weather forecasting has been to embed a high resolution "Limited Area Model" (LAM) in a global, low-resolution model. With proper use of blending functions, it is possible to use a Fourier LAM without loss of spectral accuracy [14,24,67]. To resolve the spiral rainbands and "hot towers" [active convective cumulus clouds], however, a highly nonuniform RBF model might be much better.

LAMs and Large-Eddy Simulation (LES) models both employ creative filtering to pass unresolved high spatial frequencies to high resolution local models. The meshlessness of RBFs means that it is possible to stroke an image with a moving cursor like an artist using a brush in Photoshop or Painter. Our "RBF paintbrush" does not turn on pixels with color, but rather brushes curving swaths of additional grid points and RBFs to resolve curving filaments of vorticity, frontal zones and so on. RBF paintbrushes are a near-future technology in computer graphics and fluid mechanics, but a more distant improvement to atmospheric models because the blending/filtering complexity escalate rapidly with physics added to hydrodynamics.

Wong, Hon and Gohberg built a time-dependent model of tidal flow in Tolo Harbor (Hong Kong) with a global spatial basis of compactly-supported RBFs which agreed well with measurements despite the geometric complexity of islands, winding estuary, docks and breakwaters [77]. "Meshless" is a synonym for "freely adjustable to curving, topologically complex boundaries". But complex mountains and coastlines and the "RBF paintbrush" each warrant separate publications.

13 Concluding Remarks

We have demonstrated that spectrally-accurate vortex models, are feasible, stable and efficient for solving the barotropic vorticity equation on the sphere. Lagrangian, semi-Lagrangian and Eulerian coordinates versions of the basic model have been tested successfully. Both hyperviscosity and Tikohnov regularization have been used to control high-frequency noise and suppress aliasing instabilities. Hyperviscosity is discussed here, but Tikhonov Regularization is discussed at length in our companion paper [79] and Chapter 3 of Xiao's thesis [80].

With N gridpoints, the cost of the Eulerian (fixed grid) code is $O(N^2)$ floating point operations per time step; the vortex-RBF (Lagrangian coordinates) code has a cost of $O(N^3)$. This implies these algorithms are unfeasible for large N. However, a good workstation never sleeps. These RBF methods provide spectral accuracy for small and medium-sized N on a laptop.

Another future direction is to deploy the additional numerical technologies discussed in the preceding section to massively parallelize at least the rate-determining linear algebra steps and to accelerate the convergence of RBF series. Our present codes are very simple to program, spectrally-accurate but scale poorly to large N. At the cost of additional programming, tricks like Partition-of–Unity-Methods (PUM) allow us to sacrifice program simplicity for much greater speed.

A smooth domain like the surface of a sphere is not traditional for RBFs which because of their meshlessness are well-suited to complex domains such as an ocean basin with complicated peninsulas, bays and estuaries and islands and archipelagoes. An obvious future direction is to apply these methods to small-N and medium-N problems in oceans with complicated coastlines. The study by Boyd and Sanjaya [16] is a beginning.

The Lagrangian/Eulerian controversy has the answer: Use the Eulerian coordinates code, which is *not* a vortex method, except when the flow is dominated by advection and maybe not even then. When waves are present, Lagrangian RBF codes will advectively move the grid points, but fronts move at a non-advective speed because they are moved by wave propagation, too. A good front-tracking subroutine can cluster the RBF interpolation points at the front. Once again, though, performance is greatly improved at the expense of piling new algorithms and complexity on top of the simple methods discussed here.

The interpolation/quasi-interpolation controversy is equally important, but the complexities of this contest require postpone a separate article.

Numerous additional details and further numerical examples can be found in the doctoral theses of Wang and Xiao [73,80]. The latter also gives the details of constructing a spherical icosahedral grid.

This work was supported by NSF Grants OCE998636, OCE0451951, OCE 1059703, , ATM-0620100 and ATM 0723440. We thank Robert Krasny for his constant support. We thank the reviewer for helpful comments. We also thank Marlon Allen for help with the icosahedral grid, and also the Undergraduate Research Opportunities Program (UROP) at Michigan which supported his participation. We are grateful to Piotr K. Smolarkiewicz for his many detailed suggestions.

References

[1] L. A. Barba and L. F. Rossi, *Global field interpolation for particle methods*, J. Comput. Phys., 229 (2010), pp. 1292–1310.

[2] L. A. Barba and R. yokota, *How will the Fast Multipole Method fare in the Exascale Era?*, SIAM News, 73 (2013), pp. 1–2.

[3] E. W. Barrett, *Eccentric circumpolar vortices in a barotropic atmosphere*, Tellus, 10 (1958), pp. 395–400.

[4] J. Beale and A. Majda, *Vortex methods. II. Higher order accuracy in two or three dimensions*, Math. Comput., 39 (1982), pp. 29–52.

[5] J. T. Beale, *On the accuracy of vortex methods at large times*, in Computational Fluid Dynamics and Reacting Gas Flows, B. E. *et al.*, ed., Springer-Verlag, New York, 1988, pp. 19–32. Also a (1987) technical report from the Institute for Mathematics and its Applications, University of Minnesota.

[6] R. E. Bellman, B. G. Kashef, and J. Casti, *Differential quadrature: A technique for the rapidl solution of nonlinear partial differential equations*, Journal of Computational Physics, 10 (1972), pp. 40–52.

[7] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, *Meshless methods: An overview and recent developments*, Comput. Meth. Appl. Mech. Engrg., 139 (1996), pp. 3–47.

[8] V. A. Bogomolov, *On the motion of a vortex on a rotating sphere*, Izvestiya, 21 (1985), pp. 298–302.

[9] E. F. Bollig, N. Flyer, and G. Erlebacher, *Solution to PDEs using radial basis function finte-differences (rbf-fd) on multiple GPUs*, J. Comput. Phys., 231 (2012), pp. 7133–7151.

[10] J. P. Boyd, *Weakly Nonlocal Solitary Waves and Beyond-All-Orders Asymptotics: Generalized Solitons and Hyperasymptotic Perturbation Theory*, vol. 442 of Mathematics and Its Applications, Kluwer, Amsterdam, 1998. 608 pp.

[11] ——, *The devil's invention: Asymptotic superasymptotics and hyperasymptotics*, Acta Applicandae, 56 (1999), pp. 1–98.

[12] ——, *Chebyshev and Fourier Spectral Methods*, Dover, Mineola, New York, 2d ed., 2001. 665 pp.

[13] ——, *Hyperasymptotics and the linear boundary layer problem: Why asymptotic series diverge*, SIAM Rev., 47 (2005), pp. 553–575.

[14] ——, *Limited-area Fourier spectral models and data analysis schemes: Windows, Fourier extension, Davies relaxation and all that*, Month. Weather Rev., 133 (2005), pp. 2030–2042.

[15] ——, *The uselessnesss of the Fast Gauss Transform for summing Gaussian radial basis functions series*, J. Comput. Phys., 229 (2009), pp. 1311–1326.

[16] J. P. Boyd and E. Sanjaya, *Geometrical effects on western intensification of wind- driven ocean currents: the rotated-channel Stommel model, coastal orientation, and curvature*, Dynamics of the Atmosphere and Ocean, 65 (2014), pp. 17–38.

[17] J. P. Boyd and C. Zhou, *Three ways to solve the Poisson equation on a sphere with gaussian forcing*, J. Comput. Phys., 228 (2009), pp. 4702–4713.

[18] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, vol. 12 of Cambridge monographs on applied and computational mathematics, Cambridge University Press, 2003.

[19] R. Cavoretto and A. D. Rossi, *Spherical interpolation using the partition of unity method: An efficient and flexible algorithm*, Appl. Math. Lett., 26 (2012), pp. 1251–1256.

[20] J. G. Charney, R. Fjörtoft, and J. von Neumann, *Numerical integration of the barotropic vorticity equation*, Tellus, 2 (1950), pp. 237–254.

[21] A. J. Chorin, *Numerical study of slightly viscous flow*, J. Fluid Mech., 57 (1973), pp. 785–796.

[22] A. J. Chorin and P. S. Bernard, *Discretization of a vortex sheet, with an example of roll-up*, J. Comput. Phys., (1973), pp. 423–428.

[23] G.-H. Cottet and P. D. Koumoutsakos, *Vortex Methods Theory and Practice*, Cambridge University Press, Cambridge, 2000.

[24] D. Degrauwe, S. Caluwaerts, F. Voitus, R. Hamdi, and P. Termonia, *Application of Boyd's periodization and relaxation method in a spectral atmospheric limited-area model. Part II. Accuracy analysis and detailed study of operational impact*, Mon. Wea. Rev., 140 (2012), pp. 3149–3162.

[25] M. DeMaria, *Tropical cyclone motion in a nondivergent barotropic model*, Monthly Weather Rev., 113 (1985), pp. 1199–1210.

[26] M. Dibattista and L. Polvani, *Barotropic vortex pairs on a rotating sphere*, J. Fluid Mech., 358 (1998), pp. 107–133.

[27] G. F. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences, World Scientific Publishing Company, Singapore, 2007.

[28] N. Flyer and B. Fornberg, *Radial basis functions: Developments and applications to planetary scale flows*, Comput. Fluid, 46 (2011), pp. 23–32.

[29] N. Flyer and E. Lehto, *Rotational transport on a sphere: Local node refinement with radial basis functions*, J. Comput. Phys., 229 (2010), pp. 1954–1969.

[30] N. Flyer, E. Lehto, S. Blaise, G. B. Wright, and A. St-Cyr, *A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere*, J. Comput. Phys., 231 (2012), pp. 4078–4095.

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

[32] B. Fornberg and E. Lehto, *Stabilization of RBF-generated finite difference methods for convective PDEs*, J. Comput. Phys., 230 (2011), pp. 2270–2285.

[33] B. Fornberg, E. Lehto, and C. Powell, *Stable calculation of Gaussian-based RBF-FD stencils*, Comput. Math. Applics, 65 (2013), pp. 627–637.

[34] S. Friedlander, *Interactions of vortices in a fluid on a surface of a rotating sphere*, Tellus, 27 (1975), pp. 1–.

[35] E. Fuselier, T. Hangelbroek, F. Narcowich, J. Ward, and G. Wright, *Localized bases for kernel spaces on the unit sphere*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 2538–2562.

[36] ———, *Kernel based quadrature on spheres and other homogeneous spaces*, Numerische Mathematik, 127 (2014), pp. 57–92.

[37] E. J. Fuselier and G. B. Wright, *Stability and error estimates for vector field interpolation and decomposition on the sphere with RBFs*, SINUM J. Numer. Anal., 47 (2009), pp. 3213–3239.

[38] W. L. Gates and C. A. Riegel, *Study of numerical errors in integration of barotropic flow on a spherical grid*, J. Geophys. Res., 67 (1962), pp. 773–.

[39] S. Gravel and A. Staniforth, *Variable resolution and robustness*, Mon. Weath. Rev., 120 (1992), pp. 2633–2640.

[40] V. M. Gryanik, *Localized vortices in the equatorial atmosphere: Structure and dynamics*, Izv. Atmos. Oceanic Phys., 24 (1988), pp. 13–19.

[41] N. G. Hogg and H. M. Stommel, *The heton, an elementary interaction between discrete baroclinic geostrophic vortices, and its implications concerning eddy heat-flow*, Proc. R. Soc. London, A397 (1985), pp. 1–20.

[42] A. Iske, *Multiresolution Methods in Scattered Data Modelling*, vol. 37 of Lecture Notes in Computational Science and Engineering, Springer, Heidelberg, 2004.

[43] M. I. Jamaloodeen and P. K. Newton, *The N-vortex problem on a rotating sphere. II. Heterogeneous Platonic solid equilibria*, Proc. R. Soc. London A, 462 (2007), pp. 3277–3299.

[44] E. J. Kansa, *A comparative study of finite difference and multiquadric schemes for the Euler equations*, Simulation, 51 (1988), pp. 180–183.

[45] R. Kidambi and P. K. Newton, *Streamline topologies for integrable vortex motion on a sphere*, Physica D, 140 (1998), pp. 95–125.

[46] Y. Kimura, *Vortex motion on surfaces with constant curvature*, Proc. R. Soc. London A, 455 (1987), pp. 245–259.

[47] Y. Kimura and H. Okamoto, *Vortex on a sphere*, J. Phys. Soc. Japan, 56 (1987), pp. 4203–4206.

[48] K. V. Klyatskin and G. M. Reznik, *Point vortices on a rotating sphere*, Oceanology, 29 (1989), pp. 12–16.

[49] R. Krasny, *Desingularization of periodic vortex sheet roll-up*, J. Comput. Phys., 65 (1986), pp. 292–313.

[50] R. Krasny and L. Wang, *Fast evaluation of multiquadric rbf sums by a Cartesian treecode*, SIAM J. Sci. Comput., 33 (2011), pp. 2341–2355.

[51] F. Laurent-Polz, *Point vortices on a rotating sphere*, Regular & Chaotic Dyn., 10 (2005), pp. 39–58.

[52] M. N. Levy, R. D. Nair, and H. A. Tufo, *A high-order element-based galerkin method for the barotropic vorticity equation*, Inter. J. Numer. Meths. Fluids, 59 (2009), pp. 1369–1387.

[53] L. V. Ling and E. J. Kansa, *A least-squares preconditioner for radial basis functions collocation methods*, Adv. Comput. Math., 23 (2005), pp. 31–54.

[54] F. Magoules, L. A. Diago, and I. Hagiwara, *A two-level iterative method for image reconstruction with radial basis functions*, JSME Inter. J. Ser. C-Mech. Systems Mach. Elements Manufacturing, 48 (2005), pp. 149–158.

[55] V. Maz'ya and G. Schmidt, *Approximate Approximations*, American Mathematical Society, Providence, 2007. 349 pp.

[56] J. C. McWilliams and N. J. Zabusky, *Interactions of isolated vortices. 1. modons colliding with modons*, Geophys. Astro. Fluid Dyn., 19 (1982), pp. 207–227.

[57] Natasha and G. B. Wright, *A radial basis function method for the shallow water equations on a sphere*, Proc. Roy. Soc. A, 465 (2009), pp. 1949–1976. submitted.

[58] P. K. Newton and H. Shokraneh, *Interacting dipole pairs on a rotating sphere*, Proc. R. Soc. London A, 464 (2008), pp. 1525–1541.

[59] Y. Ohtake, A. Belyaev, and H. Seidel, *Sparse surface reconstruction with adaptive partition of unity and radial basis functions*, Graphical Models, 68 (2006), pp. 15–24.

[60] G. M. Reznik and Z. Kizner, *Two-layer quasi-geostrophic singular vortices embedded in a regular flow. Part 1. Invariants of motion and stability of vortex pairs*, J. Fluid Mech., 584 (2007), pp. 185–202.

[61] L. Rosenhead, *The formation of vortices from a surface of discontinuity*, Proc. R. Soc. London A, 134 (1931), pp. 170–192.

[62] R. Sadourny, A. Arakawa, and Y. Mintz, *Integration of the nondivergent barotropic equation with an icosahedral hexagonal grid for the sphere*, Mon. Weather Rev., 96 (1968), pp. 351–356.

[63] T. Sakajo, *Non-self-similar, partial, and robust collapse of four point vortices on a sphere*, Phys. Rev. E, 78 (2008), p. Art. No. 016312.

[64] R. Schaback, *Multivariate interpolation by polynomials and radial basis functions*, Construct. Approx., 21 (2005), pp. 293–317.

[65] S.-E. Shin, J.-Y. Han, and J.-J. Baik, *On the critical separation distance of binary vortices in a nondivergent barotropic atmosphere*, J. Meteor. Soc. Japan, 84 (2006), pp. 853–869.

[66] G. R. Stuhne and W. R. Peltier, *New icosahedral grid-point discretization of the shallow water equations on the sphere*, J. Comput. Phys., 128 (1996), pp. 58–81.

[67] P. Termonia, F. Voitus, D. Degrauwe, S. Caluwaerts, and R. Hamdi, *Application of Boyd's periodization and relaxation method in a spectral atmospheric limited-area model. Part I. Implementation and reproducibility tests*, Mon. Wea. Rev., 140 (2012), pp. 3137–3148.

[68] I. Tobor, P. Reuter, and C. Schlick, *Reconstructing multi-scale variational partition of unity implicit surfaces with attributes*, Graphical Models, 68 (2006), pp. 25–41.

[69] C. E. Torres and L. A. Barba, *Fast radial basis function interpolation with Gaussians by localization and iteration*, J. Comput. Phys., 228 (2009), pp. 4976–4999.

[70] J. J. Tribbia, *Modons in spherical geometry*, Geophys. Astrophys. Fluid Dyn., 30 (1984), pp. 131–168.

[71] W. T. M. Verkley, *The construction of barotropic modons on a sphere*, J. Atmos. Sci., 41 (1984), pp. 2492–2504.

[72] E. H. Vestine, W. L. Sibley, and J. W. K. aand J. L. Carlstedt, *Integral and spherical-harmonic analyses of the geomagnetic field for 1955.0, part 2*, J. Geomagnetism and Geoelectricity, 15 (1963), pp. 73–89.

[73] L. Wang, *Radial Basis Functions and Vortex Methods and their Application to Vortex Dynamics on a Rotating Sphere*, PhD dissertation, University of Michigan, Applied and Interdisciplinary Mathematics, Department of Mathematics, 2010.

[74] H. Wendland, *Fast evaluation of radial basis functions: methods based on partition of unity*, in Approximation Theory X: Wavelets, Splines and Applications, C. K. Chui, L. L. Schumaker, and J. Stoeckler, eds., Nashville, 2002, Vanderbilt University Press, pp. 473–483.

[75] ——, *Scattered Data Approximation*, Cambridge University Press, 2005.

[76] D. L. Williamson, *Integration of the barotropic vorticity equation on a spherical geodesic grid*, Tellus, 20 (1968), pp. 642–653.

[77] S. M. Wong, Y. C. Hon, and M. A. Golberg, *Compactly supported radial basis functions for shallow water equations*, Appl. Math. Comput., 127 (2002), pp. 79–101.

[78] G. B. Wright, N. Flyer, and D. A. Yuen, *A hybrid radial basis function-pseudospectral method for thermal convection in a 3-d spherical shell*, Geochemistry, Geophysics and Geosystems, 11 (2010), p. Q07003.

[79] J. Xiao, *An analysis of Periodized RBF interpolation with Tikhonov regularization*, J. Comput. Phys., (2014). submitted.

[80] ——, *Periodized Radial Basis Functions and RBF-Vortex method for Barotropic Vorticity Equation*, PhD dissertation, University of Michigan, Department of Atmospheric, Oceanic and Space Science, April 2014.

[81] R. Yokota and L. A. Barba, *A tuned and scalable fast multipole method as a preeminent algorithm for exascale systems*, Internat. J. High Performance Comput., 26 (2012), pp. 337–346.

[82] R. Yokota, L. A. Barba, and M. J. Knepley, *Petrbfa parallel o(n) algorithm for radial basis function interpolation*, Comput. Meths. Appl. Mech. Engrg., 199 (2010), pp. 1793–1804.

[83] Z. Zong and Y. Zhang, *Advanced Differential Quadrature Methods*, CRC Press, Boca Raton, 2009.

[84] J. A. Zukas and D. R. Scheffler, *Practical aspects of numerical simulations of dynamics events: effects of meshing*, Inter. J. Impact Engrg., 24 (2000), pp. 925–945.

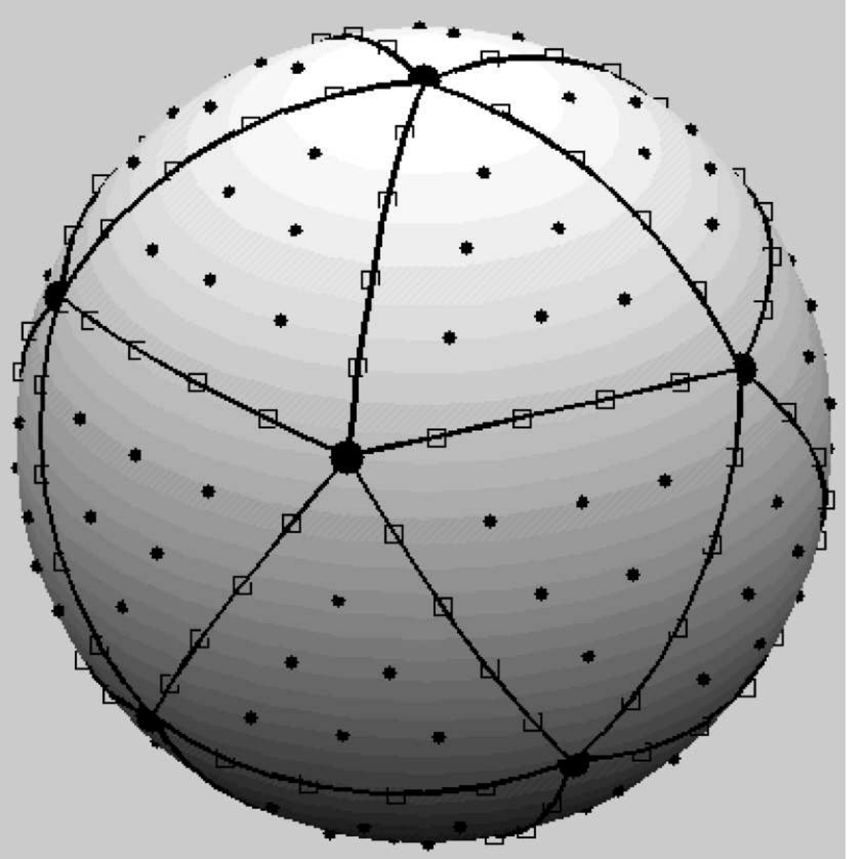

Figure 2: The icosahedral grid with 252 points. The twelve vertices of the icosahedron are large balls. They are connected by curves that are the edges of a spherical icosahedron. Hollow squares mark the four grid points on the interior of each edge. The points interior to each spherical triangle, 6 for each face of the icosahedron, are shown as red balls.

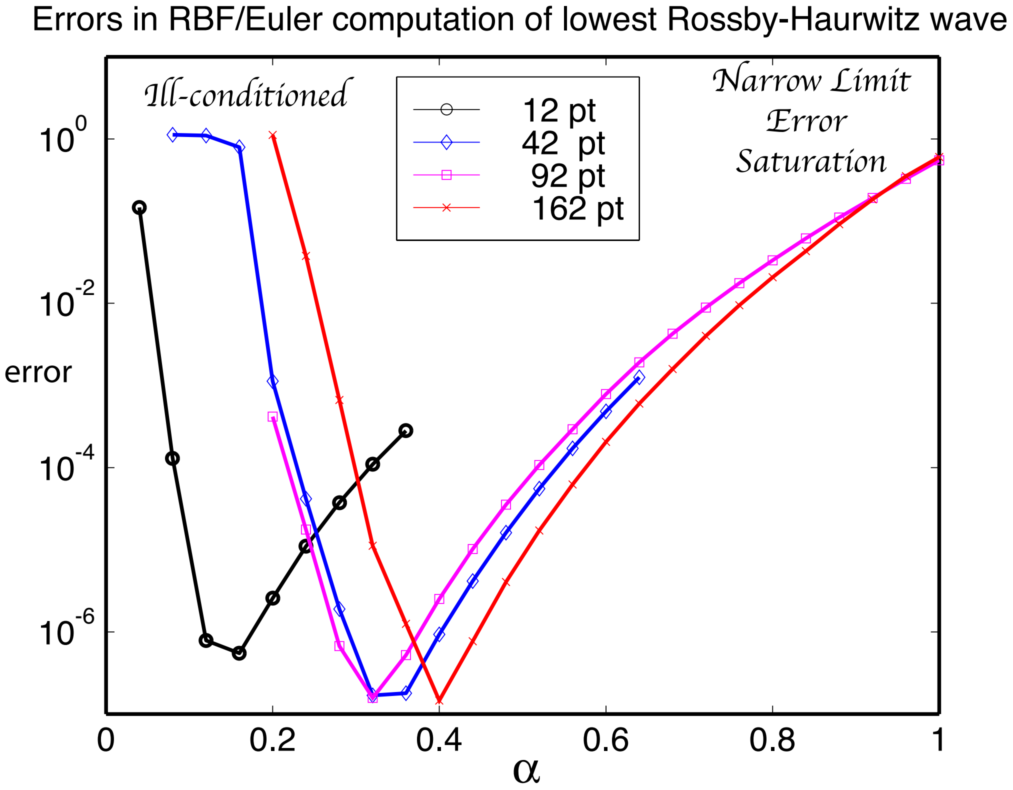

Figure 3: Errors in the solution of the lowest Rossby-Haurwitz wave on $t \in [0, 4\pi]$ as computed using the Eulerian RBF model for four different resolutions and various values of the relative width α.

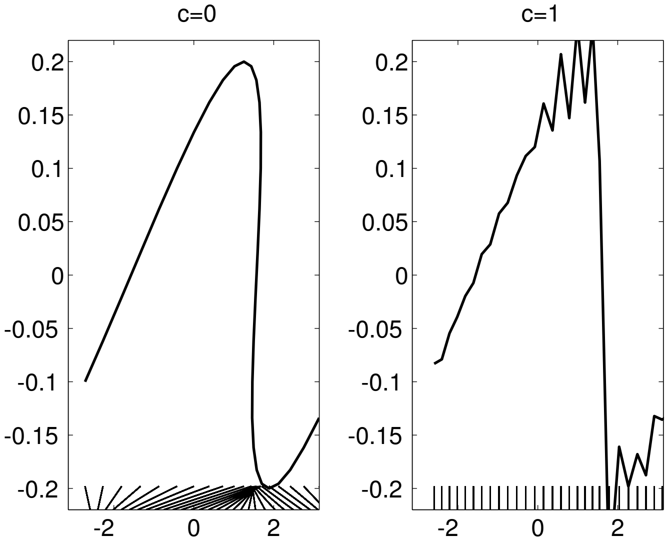

Figure 4: Solutions to $u_t + uu_x = -cu_x$ on a periodic domain, $x \in [-\pi, \pi]$, with thirty grid points as solved by a Lagrangian RBF algorithm. Left: $c = 0$. Right: $c = 1$. The thin lines at the bottom connect the initial and final locations of each grid points.

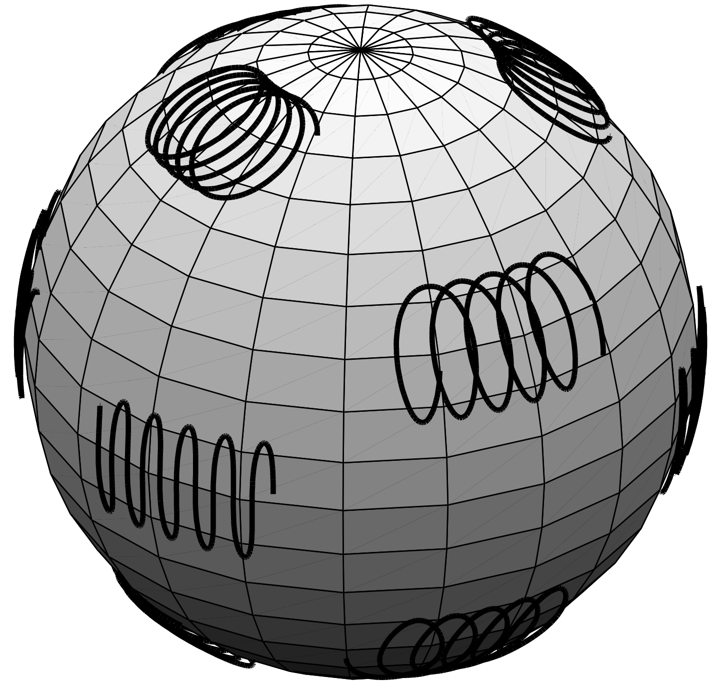

Figure 5: The trajectories of the twelve grid points, initialized at the vertices of an icosahedron, are cycloids moved by the fully-Lagrangian algorithm. Numerical integration of the Rossby-Haurwitz wave $\zeta = (1/5)\sin(\theta)\cos(\lambda - ct)$, $c = -1/2$, on the interval $t \in [0, 20\pi]$. Although the integration used only twelve vortex blobs, the relative error at the end of the integration was only 8.5×10^{-5}.

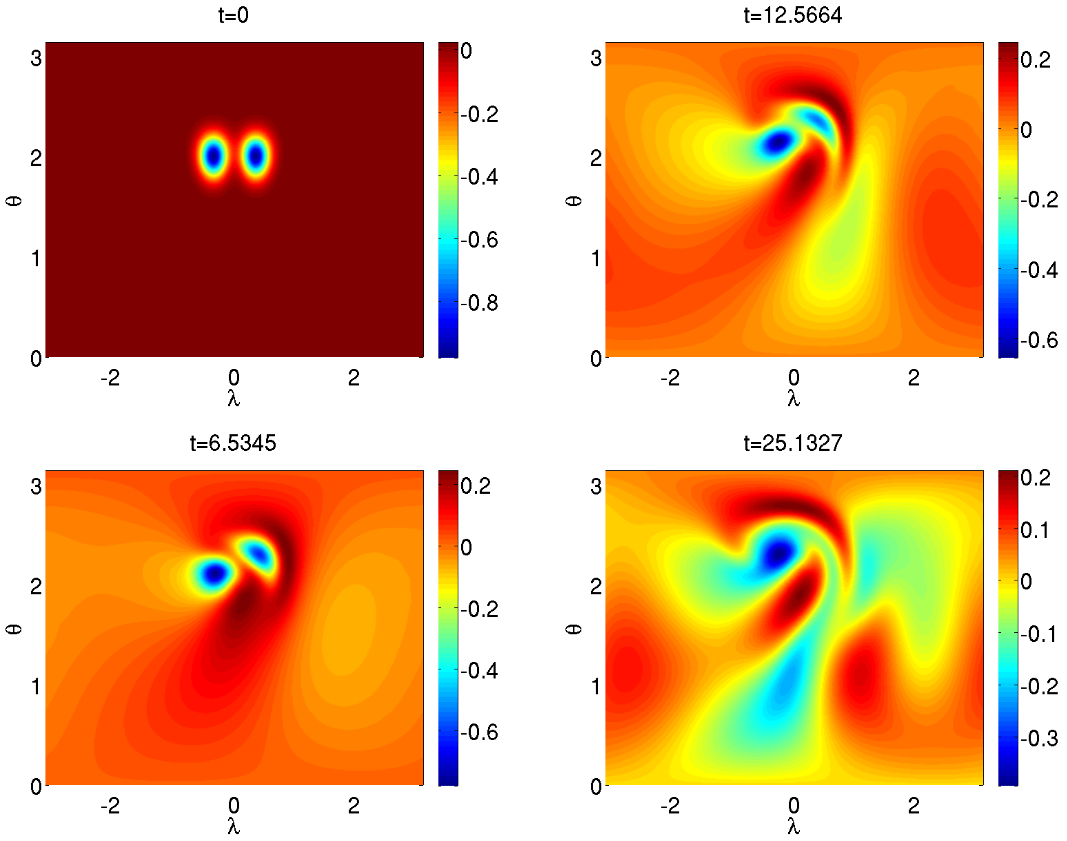

Figure 6: Vortex merger on a rotating sphere. The icosahedral grid had $N = 1442$ points, $\triangle t = 0.005, \epsilon = 3.6153$. The two vortices simulated are Gaussian vortices with $\zeta(\lambda,\theta) = \exp(-2\beta^2(1-\cos(\lambda,\theta,\lambda_c,\theta_c))) - \frac{1-\exp(-4\beta^2)}{4\beta^2}, \beta = 5$. The initial locations of the two Gaussian vortices are $(\lambda_{c1} = -0.35, \theta_{c1} = 2.000)$ and $(\lambda_{c2} = 0.35, \theta_{c2} = 2.000)$.

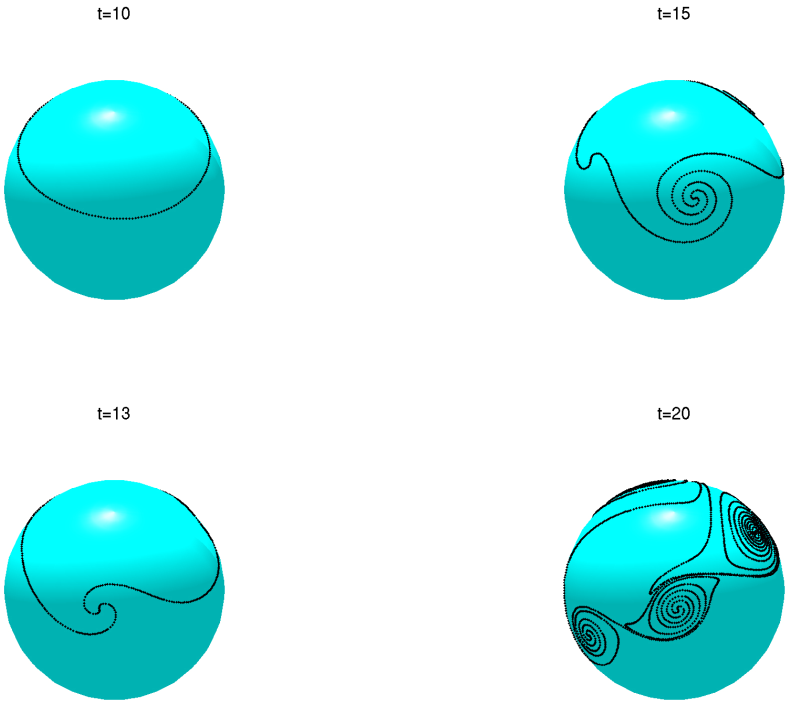

Figure 7: Vortex sheet evolution on a rotating sphere at $t = 10, 13, 15, 20$.

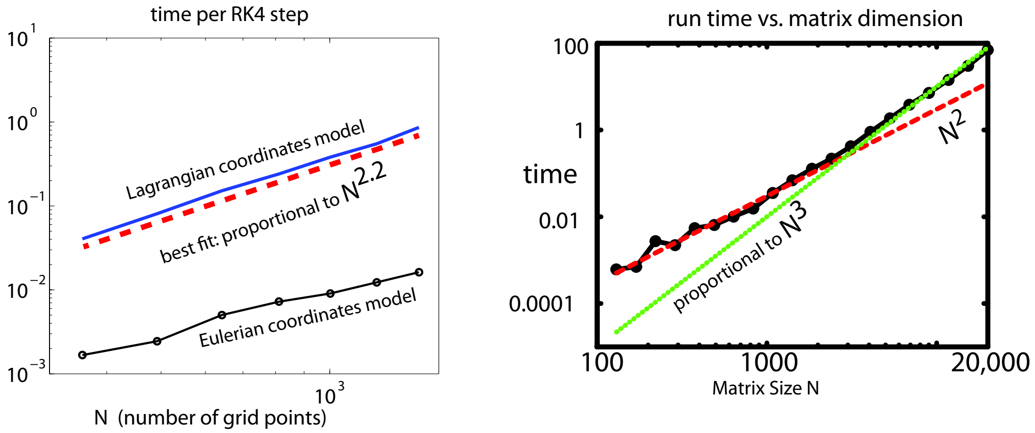

Figure 8: The left graph is the execution time per RK4 step versus the number of RBF basis functions and interpolations, plotted on a log-log scale so that power-law behavior is linear. The cost of the Lagrangian RBF-vortex (upper solid curve) is closely approximated by a best-fit line proportional to $N^{2.2}$ (red dashed line). The cost of the Eulerian coordinates RBF-Vortex model is the bottom curve (with disks). The simulation used MATLAB on an HP Z420 workstation. Right: cost of an LU factorization of a dense $N \times N$ matrix versus N (solid curve with disks). The dotted line is proportional to N^3 and is a good fit to the experimentally-measured execution times for $N > 3000$. The dashed line grows only as N^2 and approximately the actual Matlab-on-a-desktop execution times for $N < 3000$.