The Journal of Fourier Analysis and Applications

Volume 13, Issue 6, 2007

Divergence-Free RBFs on Surfaces

Francis J. Narcowich, Joseph D. Ward, and Grady B. Wright

Communicated by S. Dahlke

ABSTRACT. This article presents a new tool for fitting a divergence-free vector field tangent to a two-dimensional orientable surface $\mathcal{P} \in \mathbb{R}^3$ to samples of such a field taken at scattered sites on $\mathcal{P}$. This method, which involves a kernel constructed from radial basis functions, has applications to problems in geophysics, and has the advantage of avoiding problems with poles. Numerical examples testing the method on the sphere are included.

1. Introduction

In this article, we develop a new tool, based on radial basis functions (RBFs), for fitting a divergence-free vector field tangent to a two-dimensional orientable surface $\mathcal{P} \in \mathbb{R}^3$ to samples of such a field taken at scattered sites on $\mathcal{P}$. In the case where $\mathcal{P}$ is a sphere, there are important physical applications.

The shallow water wave equations on the surface of a rotating sphere describe the nonlinear flow of an incompressible fluid in a single hydrostatic atmospheric layer [31]. The incompressibility assumption gives rise to the constraint that the velocity field has vanishing surface divergence. Similar constraints on the velocity also arise in the barotropic vorticity equation on the surface of the sphere, which provides a good model for 500-mb short-term weather forecasts in mid-latitudes [14, pp. 108–110]. Fitting divergence-free tangent vector fields to data taken in these cases would help in modeling the incompressible velocity fields involved.

There are similar problems that arise in $\mathbb{R}^3$. For example, both the velocity field of an incompressible fluid and the magnetic field from a system of currents and charges are divergence free. One would like to fit such data with a divergence-free vector field. Also,

Math Subject Classifications. 41A05, 41A63, 76M25, 86-08.

Keywords and Phrases. Sphere, vector fields, incompressible fluid, radial basis function, numerical modeling.

Acknowledgements and Notes. First and second author were supported by grant DMS-0504353 from the National Science Foundation; third author was supported by grant ATM-0620090 from the National Science Foundation.

ISSN 1069-5869 DOI: 10.1007/s00041-006-6903-2

if a set of velocity data is generated by an irrotational fluid, then fitting it with a curl-free vector field is appropriate. Several years ago, divergence-free RBFs and curl-free RBFs were introduced [23] to help in handling such problems. Unfortunately, when restricted to a sphere or to a surface $\mathcal{P}$, these RBFs, which are constructed to be divergence-free or curl free in $\mathbb{R}^3$, lose those properties.

Our goal is to use RBFs to construct a positive definite kernel that can be used to obtain a divergence-free field of tangent vectors on a sphere or a surface $\mathcal{P}$. As we mentioned above, these are *not* merely matrix-valued RBFs restricted to a surface. Moreover, one may use *extrinsic* coordinates in conjunction with these kernels. For the case of the sphere, this means that polar regions can be treated in the same way as any other region of the sphere. We will call these kernels *divergence-free RBFs on* $\mathcal{P}$, where it will be tacitly understood that the term *divergence* refers to that for the surface $\mathcal{P}$ as imbedded in $\mathbb{R}^3$. In Section 2, we will carry out the construction of the kernels, and, in Section 3, we will show how to implement them to create divergence-free vector-field interpolants. In particular, we show that the interpolation matrices involved are positive definite and therefore invertible.

Section 4 contains the results of numerical experiments conducted when the surface is a sphere. In Section 4.1, we list three vector fields that we will sample, including a field similar to the one used in Test Case 4 of [31]. This field in particular is meant to exhibit some of the properties of flow in the middle level troposphere. The other fields are used to test various aspects of the method. We then employ our method to reproduce these test fields under various conditions, with various RBFs, and we then compare the results for our method with those from more traditional methods. The numerical results we get are listed in the tables in Section 4.5, and demonstrate the superiority of our method over more traditional RBF methods.

The numerical evidence suggests that these divergence-free RBF interpolants on $\mathbb{S}^2$ converge nicely. This is not a surprise. Very recently Fuselier [7, 8, 9] and Lowitzsch [19, 20] have studied both stability and convergence properties for interpolants generated by the matrix-valued RBFs introduced in [23]. In particular, Fuselier [8] obtained good, Sobolev-type error estimates for the matrix-valued interpolants to smooth divergence-free or curl-free vector fields in $\mathbb{R}^2$ and $\mathbb{R}^3$. Currently, we are working on error estimates and stability properties for these new interpolants.

2. Divergence-Free RBFs on $\mathcal{P}$

We now turn to constructing divergence-free RBFs on a surface $\mathcal{P}$, assuming that the surface, which is imbedded in $\mathbb{R}^3$, is smooth and orientable. The easiest way to get things straight is to use differential forms. The books by Bishop and Goldberg [1] and by Flanders [4] are good references here. One can also use old-fashioned vector calculus, but the arguments need to be "tweaked" to be made rigorous. In the next few paragraphs, we will review some standard results from differential geometry. This also serves to establish notation.

The general setup is this. A vector field $\mathbf{v}$ on $\mathcal{P}$ (i.e., one that is tangent to $\mathcal{P}$ at every point) can be expressed in a local patch as $\mathbf{v} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2$, where $\{\mathbf{e}_1, \mathbf{e}_2\}$ form a smoothly varying orthonormal frame of tangent vectors at each $\mathbf{x}$ in the local patch. The normal to $\mathcal{P}$ at $\mathbf{x}$ is $\mathbf{n} = \mathbf{e}_1 \times \mathbf{e}_2$, which makes $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}\}$ a right-handed orthonormal frame. (All vectors vary with $\mathbf{x}$, but for the moment we will suppress this.)

We can identify $\mathbf{v}$ with a differential form ν via duality,

$$\nu = \mathbf{v} \cdot d\mathbf{x} = v_1 \underbrace{\mathbf{e}_1 \cdot d\mathbf{x}}_{\sigma_1} + v_2 \underbrace{\mathbf{e}_2 \cdot d\mathbf{x}}_{\sigma_2} = v_1\sigma_1 + v_2\sigma_2 \,.$$

The divergence of $\mathbf{v}$ is just $d * \nu$, where $*$ is the (two-dimensional) Hodge star operator. For a right-handed orthonormal basis for the tangent space, the $*$ operator has the following effect on σ_1 and σ_2: $*\sigma_1 = \sigma_2$ and $*\sigma_2 = -\sigma_1$.

It would be easy to compute the divergence of $\mathbf{v}$, now. However, this isn't the object here. Instead, we want to produce a divergence-free vector field. On $\mathcal{P}$, every 1-form looks like $\omega = a_1\sigma_1 + a_2\sigma_2$. Since $*\omega = a_1\sigma_2 - a_2\sigma_1 = b_1\sigma_1 + b_2\sigma_2$, it is still an arbitrary 1-form. Take $\nu = *\omega$, and note that $d * \nu = d * *\omega = -d\omega$. By the converse of the Poincaré lemma, if $d * \nu = d\omega = 0$, then one has $f : \mathcal{P} \to \mathbb{R}$ such that $\omega = df$, locally.

We can put this in terms of vector fields. First, $df = \nabla f \cdot d\mathbf{x}$, so $\omega = df \leftrightarrow \nabla f$. Second, $\nu = *\omega \leftrightarrow \mathbf{n} \times \nabla f$. Finally, $d * \nu = 0 \leftrightarrow \operatorname{div}_{\mathcal{P}}(\mathbf{n} \times \nabla f) = 0$. We have thus shown that *a vector field* $\mathbf{v}$ *on* $\mathcal{P}$ *is divergence free if and only if locally there exists* $f : \mathcal{P} \to \mathbb{R}$ *such that* $\mathbf{v} = \mathbf{n} \times \nabla f$.

The function f can be defined on $\mathbb{R}^3$ and then restricted to $\mathcal{P}$, because the cross product $\mathbf{n} \times \nabla f$ eliminates any component along $\mathbf{n}$. This means that we may use *extrinsic* coordinates to deal with $\mathcal{P}$.

At this point, we will use Cartesian coordinates in $\mathbb{R}^3$ and regard vectors there as columns. Let

$$\Phi(\mathbf{x}) = \int_{\mathbb{R}^3} \left[e^{i\xi\cdot\mathbf{x}} - 1 - i\xi \cdot \mathbf{x} \right] \widehat{\Phi}(\xi) \frac{d^3\xi}{(2\pi)^3} \,, \tag{2.1}$$

where $\widehat{\Phi}(\xi) > 0$ and $\int_{\mathbb{R}^3} |\xi|^2 \widehat{\Phi}(\xi) d\xi < \infty$. The function Φ is conditionally positive definite of order 1 [10, 21, 28] on $\mathbb{R}^3$. Moreover, Φ is in $C^2(\mathbb{R}^3)$ [28, Theorem 4.1], so we can form its Hessian,

$$\nabla\nabla^T \Phi = -\int_{\mathbb{R}^3} \widehat{\Phi}(\xi) \xi\xi^T e^{i\mathbf{x}^T\xi} \frac{d^3\xi}{(2\pi)^3} \,.$$

Note that

$$\mathbf{n}_\mathbf{x} \times \nabla_\mathbf{x} \left(e^{i\mathbf{x}^T\xi} \right) = i e^{i\mathbf{x}^T\xi} \mathbf{n}_\mathbf{x} \times \xi = i\, \mathrm{X}_{\mathbf{n}_\mathbf{x}} \xi e^{i\mathbf{x}^T\xi} \,,$$

where $\mathbf{n}_\mathbf{a}$ is the normal to $\mathcal{P}$ at $\mathbf{a}$, $\nabla_\mathbf{a}$ is the gradient with respect to $\mathbf{a}$, and

$$\mathrm{X}_\mathbf{a} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}, \text{ and } \mathrm{X}_\mathbf{a}\mathbf{b} = \mathbf{a} \times \mathbf{b}, \ \mathbf{a}, \mathbf{b} \in \mathbb{R}^3 \,.$$

In the same way, we get

$$\left\{ \mathbf{n}_\mathbf{y} \times \nabla_\mathbf{y} \left(e^{-i\mathbf{y}^T\xi} \right) \right\}^T = -i\xi^T \mathrm{X}_{\mathbf{n}_\mathbf{y}}^T e^{-i\mathbf{y}^T\xi} \,.$$

From these two equations we easily see that when $\mathbf{x}$ and $\mathbf{y}$ belong to $\mathcal{P}$ the kernel

$$\begin{aligned} \Psi(\mathbf{x}, \mathbf{y}) \ &:= \ \mathrm{X}_{\mathbf{n}_\mathbf{x}} \left\{ \int_{\mathbb{R}^3} \widehat{\Phi}(\xi) \xi\xi^T e^{i(\mathbf{x}-\mathbf{y})^T\xi} \frac{d^3\xi}{(2\pi)^3} \right\} \mathrm{X}_{\mathbf{n}_\mathbf{y}}^T \\ &= \ -\mathrm{X}_{\mathbf{n}_\mathbf{x}} \left(\nabla\nabla^T \Phi(\mathbf{x} - \mathbf{y}) \right) \mathrm{X}_{\mathbf{n}_\mathbf{y}}^T \end{aligned} \tag{2.2}$$

is divergence free and nonnegative — i.e., the corresponding quadratic form is positive semi-definite. We summarize these observations below.

Theorem 1. *Let $\mathcal{P}$ be a smooth, orientable manifold in $\mathbb{R}^3$, possibly with boundary, and let $\mathbf{n}_\mathbf{x}$ denote the unit normal to $\mathcal{P}$ at $\mathbf{x} \in \mathcal{P}$. Then, the kernel given in* (2.2) *is nonnegative and divergence free when* $\mathbf{x}$ *and* $\mathbf{y}$ *belong to* $\mathcal{P}$.

We now turn to calculating Ψ when the function Φ is radial; that is, when $\Phi(\mathbf{x}) = \Phi(|\mathbf{x}|)$. We will start by finding the Hessian matrix $\nabla\nabla^T\Phi(\mathbf{x})$ in Cartesian coordinates. Let $r = |\mathbf{x}|$. The j, k component of the matrix is then easily shown to be

$$\frac{\partial^2\Phi}{\partial x_j \partial x_k} = \delta_{j,k} \underbrace{\frac{1}{r}\Phi'(r)}_{F(r)} + x_j x_k \underbrace{\frac{1}{r}\left(\frac{1}{r}\Phi'(r)\right)'}_{G(r)} .$$

From this we have that

$$\nabla\nabla^T\Phi(\mathbf{x}-\mathbf{y}) = F(r)I + G(r)(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T ,$$

where $r = |\mathbf{x}-\mathbf{y}|$ and I is the identity matrix. Carrying out the multiplications implicit in (2.2), we arrive at

$$\begin{aligned}&\Psi(\mathbf{x},\mathbf{y})\\ &\quad = F(r)\big(\mathbf{n}_\mathbf{y}\mathbf{n}_\mathbf{x}^T - \mathbf{n}_\mathbf{y}^T\mathbf{n}_\mathbf{x} I\big) - G(r)\left(\mathbf{n}_\mathbf{x} \times (\mathbf{x}-\mathbf{y})\right)\left(\mathbf{n}_\mathbf{y} \times (\mathbf{x}-\mathbf{y})\right)^T .\end{aligned} \tag{2.3}$$

When $\mathcal{P} = \mathbb{S}^2$, the unit sphere in $\mathbb{R}^3$, the normals are just $\mathbf{n}_\mathbf{x} = \mathbf{x}$ and $\mathbf{n}_\mathbf{y} = \mathbf{y}$. Using this in the previous formula gives us

$$\Psi(\mathbf{x},\mathbf{y}) = F(r)\big(\mathbf{y}\mathbf{x}^T - \mathbf{y}^T\mathbf{x}I\big) - G(r)(\mathbf{x}\times\mathbf{y})(\mathbf{x}\times\mathbf{y})^T . \tag{2.4}$$

Of course, here we have $r = |\mathbf{x}-\mathbf{y}| = 2\sin(\theta/2)$, where θ is the angle between $\mathbf{x}$ and $\mathbf{y}$.

We return to the general case of a surface patch $\mathcal{P}$. When $\mathbf{x}$ and $\mathbf{y}$ are fixed points on $\mathcal{P}$, the kernel $\Psi(\mathbf{x},\mathbf{y})$ is a linear map that takes a tangent vector $\mathbf{s}$ at $\mathbf{y}$ to a tangent vector at $\mathbf{x}$. Set $\tilde{\mathbf{s}} = \mathbf{n}_\mathbf{y} \times \mathbf{s}$. From (2.3), we see that

$$\Psi(\mathbf{x},\mathbf{y})\mathbf{s} = \mathbf{n}_\mathbf{x} \times \left\{F(r)\tilde{\mathbf{s}} + G(r)(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^T\tilde{\mathbf{s}}\right\} . \tag{2.5}$$

If $\mathbf{t}$ is tangent to $\mathcal{P}$ at $\mathbf{x}$, and if $\tilde{\mathbf{t}} = \mathbf{n}_\mathbf{x} \times \mathbf{t}$, then we also have

$$\mathbf{t}^T\Psi(\mathbf{x},\mathbf{y})\mathbf{s} = -F(r)\tilde{\mathbf{t}}^T\tilde{\mathbf{s}} - G(r)(\mathbf{x}-\mathbf{y})^T\tilde{\mathbf{t}}(\mathbf{x}-\mathbf{y})^T\tilde{\mathbf{s}} . \tag{2.6}$$

3. Divergence-Free RBF Interpolants on $\mathcal{P}$

Let the vectors $\mathbf{t}_1, \dots, \mathbf{t}_N$ be tangent to $\mathcal{P}$ at points $\mathbf{x}_1, \dots, \mathbf{x}_N$, respectively. We want to interpolate the $\mathbf{t}_j$'s with a divergence-free vector field of the form

$$\mathbf{t}(\mathbf{x}) = \sum_{k=1}^{N} \Psi(\mathbf{x}, \mathbf{x}_k)\mathbf{s}_k \ , \tag{3.1}$$

where $\mathbf{s}_k$ and $\mathbf{t}(\mathbf{x})$ are tangent to $\mathcal{P}$ at $\mathbf{x}_k$ and $\mathbf{x}$, respectively. Doing this requires that we solve the linear system below for the $\mathbf{s}_k$'s:

$$\mathbf{t}_j = \sum_{k=1}^{N} \Psi(\mathbf{x}_j, \mathbf{x}_k)\mathbf{s}_k, \ j = 1, \dots, N \ .$$

At each $\mathbf{x}_k$, choose $\mathbf{e}_k$ to be a unit vector tangent to $\mathcal{P}$ at $\mathbf{x}_k$ and let $\mathbf{f}_k = \mathbf{n}_k \times \mathbf{e}_k$. The triple $\{\mathbf{e}_k, \mathbf{f}_k, \mathbf{n}_k\}$ is a right-handed, orthonormal 3-frame at $\mathbf{x}_k$, and $\{\mathbf{e}_k, \mathbf{f}_k\}$ is an orthonormal basis for the tangent space there. Express the $\mathbf{t}_j$'s and $\mathbf{s}_k$'s in terms of the appropriate basis:

$$\mathbf{s}_k = \alpha_k\mathbf{e}_k + \beta_k\mathbf{f}_k \quad \text{and} \quad \mathbf{t}_j = \gamma_j\mathbf{e}_j + \delta_j\mathbf{f}_j \ .$$

Using these in conjunction with the linear system above, we obtain the following system of equations,

$$\begin{pmatrix} \gamma_j \\ \delta_j \end{pmatrix} = \sum_{k=1}^{N} \underbrace{\begin{pmatrix} \mathbf{e}_j^T \\ \mathbf{f}_j^T \end{pmatrix} \Psi(\mathbf{x}_j, \mathbf{x}_k) \begin{pmatrix} \mathbf{e}_k & \mathbf{f}_k \end{pmatrix}}_{A^{(2)}_{jk}} \begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix}, \ j = 1, \dots, N \ . \tag{3.2}$$

Next, we will use (2.6) to find the 2×2 matrix $A^{(2)}_{jk}$ in the sum above. In doing so, keep in mind that $\tilde{\mathbf{e}} = \mathbf{f}$ and $\tilde{\mathbf{f}} = -\mathbf{e}$.

$$\begin{aligned} A^{(2)}_{jk} &= \begin{pmatrix} \mathbf{e}_j^T \\ \mathbf{f}_j^T \end{pmatrix} \Psi(\mathbf{x}_j, \mathbf{x}_k) \begin{pmatrix} \mathbf{e}_k & \mathbf{f}_k \end{pmatrix} \\ &= -\begin{pmatrix} \mathbf{f}_j^T \\ -\mathbf{e}_j^T \end{pmatrix} \left\{ F(r_{jk})I + G(r_{jk})(\mathbf{x}_j - \mathbf{x}_k)(\mathbf{x}_j - \mathbf{x}_k)^T \right\} \begin{pmatrix} \mathbf{f}_k & -\mathbf{e}_k \end{pmatrix} , \end{aligned} \tag{3.3}$$

where $r_{jk} = |\mathbf{x}_j - \mathbf{x}_k|$. Note that we have $(A^{(2)}_{jk})^T = A^{(2)}_{kj}$.

In the case of the sphere, this formula simplifies somewhat. The normal $\mathbf{n}_j$ is just $\mathbf{x}_j$. Consequently, the terms $\mathbf{x}_j^T\mathbf{e}_j$ and $\mathbf{x}_j^T\mathbf{f}_j$ vanish. Switching to "dot" product notation, this gives us

$$\begin{aligned} &A^{(2)}_{jk} \\ &= F(r_{jk}) \begin{pmatrix} -\mathbf{f}_j \cdot \mathbf{f}_k & \mathbf{f}_j \cdot \mathbf{e}_k \\ \mathbf{e}_j \cdot \mathbf{f}_k & -\mathbf{e}_j \cdot \mathbf{e}_k \end{pmatrix} + G(r_{jk}) \begin{pmatrix} \mathbf{f}_j \cdot \mathbf{x}_k \\ -\mathbf{e}_j \cdot \mathbf{x}_k \end{pmatrix} \begin{pmatrix} \mathbf{x}_j \cdot \mathbf{f}_k & -\mathbf{x}_j \cdot \mathbf{e}_k \end{pmatrix} . \end{aligned} \tag{3.4}$$

The object now is to solve the system (3.2). Let A be the $2N \times 2N$ matrix composed of the blocks $A^{(2)}_{jk}$ in (3.3), let $c = (\alpha_1 \, \beta_1 \, \cdots \, \alpha_N \, \beta_N)^T$, and finally let $d = (\gamma_1 \, \delta_1 \, \cdots \, \gamma_N \, \delta_N)^T$.

The $2N \times 1$ column vector c contains the coefficients we want to find, and d contains the data we know. The system (3.2) then becomes $Ac = d$, and it has a unique solution if and only if we can invert A. The lemma and the theorem that follow establish that A is not only invertible, but also that it is positive definite.

Lemma 1. *Let A be as above, and let Ψ be given by* (2.2)*, then*

$$c^T Ac = \int_{\mathbb{R}^3} \widehat{\Phi}(\xi) \left| \xi^T \mathbf{v}(\xi) \right|^2 \frac{d^3\xi}{(2\pi)^3} \tag{3.5}$$

where $\mathbf{v}(\xi) = \sum_{k=1}^{N} \tilde{\mathbf{s}}_k e^{-i\mathbf{x}_k^T \xi}$ and $\tilde{\mathbf{s}}_k = \mathbf{n}_k \times \mathbf{s}_k$.

Proof. From (3.2), we have that

$$\begin{aligned} c^T Ac &= \sum_{j=1}^{N} \sum_{k=1}^{N} \begin{pmatrix} \alpha_j & \beta_j \end{pmatrix} \begin{pmatrix} \mathbf{e}_j^T \\ \mathbf{f}_j^T \end{pmatrix} \Psi(\mathbf{x}_j, \mathbf{x}_k) \begin{pmatrix} \mathbf{e}_k & \mathbf{f}_k \end{pmatrix} \begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} \\ &= \sum_{j=1}^{N} \sum_{k=1}^{N} \mathbf{s}_j^T \Psi(\mathbf{x}_j, \mathbf{x}_k) \mathbf{s}_k \, . \end{aligned} \tag{3.6}$$

Next, from (2.2) we have that

$$\mathbf{s}_j^T \Psi(\mathbf{x}_j, \mathbf{x}_k) \mathbf{s}_k = \mathbf{s}_j^T \, \mathrm{X}_{\mathbf{n}_j} \left\{ \int_{\mathbb{R}^3} \widehat{\Phi}(\xi) \xi \xi^T e^{i(\mathbf{x}_j - \mathbf{x}_k)^T \xi} \, \frac{d^3\xi}{(2\pi)^3} \right\} \mathrm{X}_{\mathbf{n}_k}^T \, \mathbf{s}_k \, .$$

Since we have $\mathrm{X}_{\mathbf{n}_k}^T \, \mathbf{s}_k = -\mathrm{X}_{\mathbf{n}_k} \mathbf{s}_k = -\mathbf{n}_k \times \mathbf{s}_k = -\tilde{\mathbf{s}}_k$ and, similarly, $\mathbf{s}_j^T \, \mathrm{X}_{\mathbf{n}_j} = -\tilde{\mathbf{s}}_j^T$, we see that

$$\mathbf{s}_j^T \Psi(\mathbf{x}_j, \mathbf{x}_k) \mathbf{s}_k = \int_{\mathbb{R}^3} \widehat{\Phi}(\xi) \left(\tilde{\mathbf{s}}_j^T \xi e^{i\mathbf{x}_j^T \xi} \right) \left(\xi^T \tilde{\mathbf{s}}_k e^{-i\mathbf{x}_k^T \xi} \right) \frac{d^3\xi}{(2\pi)^3} \, .$$

Using the expression above in (3.6) and simplifying, we obtain (3.5). □

Theorem 2. *A is positive definite and therefore invertible.*

Proof. By the lemma above, A is positive semi-definite. To show that it is positive definite, we need only show that $c^T Ac = 0$ forces $c = 0$. From (3.5), $c^T Ac = 0$ implies that the integrand, $\widehat{\Phi}(\xi)|\xi^T \mathbf{v}(\xi)|^2$, which is continuous and nonnegative, is 0. Since $\widehat{\Phi}(\xi) > 0$, we may cancel it to get $\xi^T \mathbf{v}(\xi) \equiv 0$, or, equivalently,

$$\sum_{k=1}^{N} \xi^T \tilde{\mathbf{s}}_k e^{-i\mathbf{x}_k^T \xi} \equiv 0 \, .$$

The expression on the left is a tempered distribution in ξ. Take the inverse Fourier transform of both sides and cancel constant factors to obtain

$$\sum_{k=1}^{N} \tilde{\mathbf{s}}_k \cdot \nabla \delta(\mathbf{x} - \mathbf{x}_k) \equiv 0 \, .$$

Integrate this distribution against a smooth "bump" function supported in a small ball, which is centered at a given $\mathbf{x}_k$ and excludes all other $\mathbf{x}_k$'s. In addition, take the function's gradient at $\mathbf{x}_k$ equal to $\tilde{\mathbf{s}}_k$. The result will be $\tilde{\mathbf{s}}_k \cdot \tilde{\mathbf{s}}_k = \mathbf{s}_k \cdot \mathbf{s}_k = \alpha_k^2 + \beta_k^2 = 0$, and so $\alpha_k = \beta_k = 0$ for all k. Hence, $c = 0$. □

3.1 Implementation

We summarize the steps in finding the interpolant $\mathbf{t}(\mathbf{x})$, given points $\mathbf{x}_1, \dots, \mathbf{x}_N$ on $\mathcal{P}$ and data vectors $\mathbf{t}_1, \dots, \mathbf{t}_N$ tangent at each of these points.

(1) At each $\mathbf{x}_k$, find the (unit) normal $\mathbf{n}_k$, then choose a unit tangent vector $\mathbf{e}_k$, and finally find $\mathbf{f}_k = \mathbf{n}_k \times \mathbf{e}_k$. On the unit sphere, these can be chosen so that $\mathbf{n}_k = \mathbf{x}_k$, $\mathbf{e}_k$ is the north-pointing unit vector along the longitude, and $\mathbf{f}_k$ is the east-pointing unit vector along the latitude. If the north or south poles are among the data points, simply choose vectors there.

(2) Find the coordinates γ_k and δ_k for $\mathbf{t}_k$ relative to $\{\mathbf{e}_k, \mathbf{f}_k\}$. Form the data vector $d = (\gamma_1\ \delta_1\ \cdots\ \gamma_N\ \delta_N)^T$.

(3) Find the 2×2 matrices $A_{jk}^{(2)}$ from (3.3), or from (3.4) in the case of the sphere. Use these to form the matrix A.

(4) Solve $Ac = d$ for $c = (\alpha_1\ \beta_1\ \cdots\ \alpha_N\ \beta_N)^T$.

(5) Find $\tilde{\mathbf{s}}_k = -\beta_k \mathbf{e}_k + \alpha_k \mathbf{f}_k$. From (2.5), we have

$$\Psi(\mathbf{x}, \mathbf{x}_k)\mathbf{s}_k = \mathbf{n}_{\mathbf{x}} \times \left\{ F(|\mathbf{x} - \mathbf{x}_k|)\tilde{\mathbf{s}}_k + G(|\mathbf{x} - \mathbf{x}_k|)(\mathbf{x} - \mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)^T \tilde{\mathbf{s}}_k \right\} ,$$

which we then sum to obtain $\mathbf{t}(\mathbf{x})$ via (3.1).

In the last step, the output will be in Cartesian coordinates. If instead one would like the output to be given in terms of a basis of tangent vectors $\{\mathbf{e}(\mathbf{x}), \mathbf{f}(\mathbf{x})\}$ (e.g., the latitude and longitude vectors on the sphere), then we can use (2.6) to calculate the appropriate components. Specifically, in the case of the unit sphere, where $\mathbf{x} = \mathbf{n}$, we have

$$\begin{cases} \mathbf{e} \cdot \mathbf{t} &= \sum_{k=1}^{N} \left\{ -F(|\mathbf{x} - \mathbf{x}_k|)\mathbf{f} \cdot \tilde{\mathbf{s}}_k + G(|\mathbf{x} - \mathbf{x}_k|)(\mathbf{f} \cdot \mathbf{x}_k)\big(\mathbf{x} \cdot \tilde{\mathbf{s}}_k\big) \right\} \\ \mathbf{f} \cdot \mathbf{t} &= \sum_{k=1}^{N} \left\{ F(|\mathbf{x} - \mathbf{x}_k|)\mathbf{e} \cdot \tilde{\mathbf{s}}_k - G(|\mathbf{x} - \mathbf{x}_k|)(\mathbf{e} \cdot \mathbf{x}_k)\big(\mathbf{x} \cdot \tilde{\mathbf{s}}_k\big) \right\} . \end{cases} \tag{3.7}$$

We now turn to a discussion of numerical results for the case in which $\mathcal{P}$ is the sphere, which is one of the most important applications of the method.

4. Numerical Examples

In this section we apply the theory from the previous sections to specific examples. We focus on the case of divergence-free vector fields on the surface of the sphere. As we mentioned in Section 1, this case has many relevant physical applications. For several different divergence-free fields, we compare the accuracy of the divergence-free RBF interpolation method presented above to the standard RBF interpolation method where each component of the vector field is interpolated separately (see, Section 4.2).

Following the notation from the previous section applied to the sphere, we let $\mathbf{x}_k = (x_k, y_k, z_k)^T$, $k = 1, \dots, N$, be distinct data locations on the unit sphere and (θ_k, λ_k), $k = 1, \dots, N$, the corresponding spherical latitude-longitude coordinates such that

$$\begin{aligned} x_k &= \cos\lambda_k \cos\theta_k , \\ y_k &= \sin\lambda_k \cos\theta_k , \\ z_k &= \sin\theta_k . \end{aligned} \tag{4.1}$$

(Note that we measure latitude from the equator rather than the north pole). Let $\mathbf{u}_k = (\gamma_k, \delta_k, 0)^T = (\gamma(\theta_k, \lambda_k), \delta(\theta_k, \lambda_k), 0)^T$, $k = 1, \dots, N$, be samples of some divergence-free vector field in spherical coordinates at the data locations, where γ, δ, and 0 correspond to the θ, λ, and radial directions, respectively. Then $\mathbf{u}_k$ can be converted to Cartesian coordinates using the transformation

$$\mathbf{t}_k = \underbrace{\begin{bmatrix} -\sin\theta_k \cos\lambda_k & -\sin\lambda_k & \cos\theta_k \cos\lambda_k \\ -\sin\theta_k \sin\lambda_k & \cos\lambda_k & \cos\theta_k \sin\lambda_k \\ \cos\theta_k & 0 & \sin\theta_k \end{bmatrix}}_{Q_k} \mathbf{u}_k \, . \tag{4.2}$$

The first two columns of Q_k form an orthonormal basis for the tangent space of the unit sphere and we thus assign these vectors to $\mathbf{e}_k$ and $\mathbf{f}_k$, respectively:

$$\mathbf{t}_k = \gamma_k \underbrace{\begin{bmatrix} -\sin\theta_k \cos\lambda_k \\ -\sin\theta_k \sin\lambda_k \\ \cos\theta_k \end{bmatrix}}_{\mathbf{e}_k} + \delta_k \underbrace{\begin{bmatrix} -\sin\lambda_k \\ \cos\lambda_k \\ 0 \end{bmatrix}}_{\mathbf{f}_k} \, . \tag{4.3}$$

Note that since Q_k is orthogonal, the vector field in Cartesian coordinates can be translated back to spherical coordinates by $\mathbf{u}_k = Q_k^T \mathbf{t}_k$.

4.1 Test Vector Fields

The divergence of a vector field $\mathbf{u} = (\gamma, \delta, 0)^T$ on the surface of the unit sphere is given by

$$\nabla \cdot \mathbf{u} = \frac{1}{\cos\theta} \left(\frac{\partial(\gamma \cos\theta)}{\partial\theta} + \frac{\partial\delta}{\partial\lambda} \right) . \tag{4.4}$$

To generate vector fields such that $\nabla \cdot \mathbf{u} = 0$, we use a scalar "stream function" Ω. The latitude and longitude components of $\mathbf{u}$ are then given by

$$\gamma = \frac{1}{\cos\theta} \frac{\partial\Omega}{\partial\lambda} \, , \tag{4.5}$$

$$\delta = -\frac{\partial\Omega}{\partial\theta} \, . \tag{4.6}$$

For the numerical experiments, we use three vector fields of varying character and smoothness as described below.

(1) This field is similar to the one used in Test Case 4 of [31] and is meant to exhibit some of the properties of flow in the middle level troposphere. We will use it to test the accuracy of the divergence-free RBF technique for infinitely smooth fields. Let

$$\overline{\Omega}(\theta, \lambda; \sigma, \theta_c, \lambda_c) = e^{-(\sigma\rho)^2} \, , \tag{4.7}$$

where

$$\rho = \arccos(\sin\theta_c \sin\theta + \cos\theta_c \cos\theta \cos(\lambda - \lambda_c)) \tag{4.8}$$

is the geodesic (or great circle) distance from (θ_c, λ_c) to (θ, λ). Then the vector field is generated from the stream function

$$\Omega_1(\theta, \lambda) = \frac{2}{3} \cos^{15}\theta - \overline{\Omega}(\theta, \lambda; 8, \pi/4, 0) + \overline{\Omega}(\theta, \lambda; 8, -\pi/4, 0) \, . \tag{4.9}$$

In terms of a wind field, this flow models two low pressure systems in a jet stream that is symmetrical about the equator [31]. Figure 1(a) displays this field on the surface of the unit sphere.

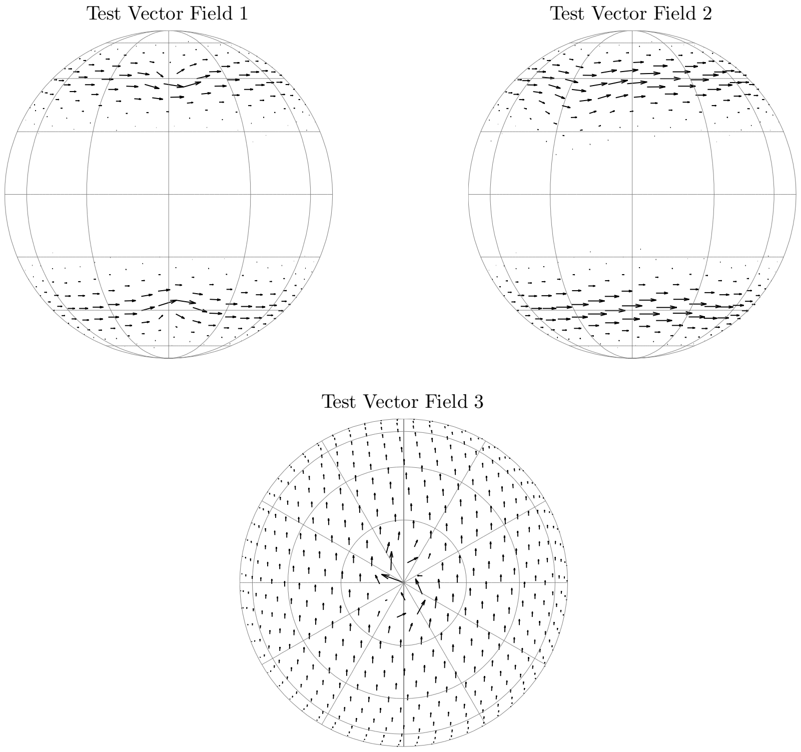

FIGURE 1 Vector fields used in the numerical examples generated from the stream functions (a) Ω_1 in (4.9), (b) Ω_2 in (4.10) and (c) Ω_3 in (4.11). (a) and (b) are orthographic projections of the field viewed from $(0, 0)$ degrees latitude-longitude, while (c) is viewed from the north pole.

(2) This field is somewhat similar to the first, however, the flow is compactly supported. It is designed to test the accuracy of the new RBF technique on divergence-free fields with finite smoothness. Let

$$\widetilde{\Omega}(\theta, \lambda; \sigma, \theta_c, \lambda_c) = \frac{\sigma^3}{12} \sum_{j=0}^{4} (-1)^j \binom{4}{j} \left| \rho - \frac{(j-2)}{\sigma} \right|^3 ,$$

where ρ is again the geodesic distance (4.8). $\widetilde{\Omega}$ is a cubic B-spline in ρ with two continuous derivatives and is positive for $|\rho| < 2/\sigma$ and zero elsewhere. We generate the vector field for this second test from the stream function

$$\Omega_2(\theta, \lambda) = \frac{2}{3} \cos^{15} \theta - \widetilde{\Omega}(\theta, \lambda; 4, -2\pi/9, -\pi/7) . \tag{4.10}$$

Since $\widetilde{\Omega}$ is only C^2, the resulting vector field generated from Ω_2 will only be C^1, which is a more realistic model of physical data. The field is displayed in

Figure 1(b).

(3) While the first two test fields have zero flow over both poles, the third test has flow directly over the poles with very strong flow over the north pole. A nonzero flow over either pole is known to cause difficulties with many interpolation/approximation methods on the sphere since, in spherical coordinates, the latitudinal and longitudinal components of the vector field will be discontinuous there [29]. This test is designed to show that the divergence-free RBF technique has no difficulties with nonzero flow over the poles since it operates on vector fields in Cartesian form, where the components are smooth over the whole sphere. The field is generated from the stream function

$$\begin{aligned}\Omega_3(\theta,\lambda) &= -5(\sin(\theta)\cos(\pi/2-0.05)-\cos\lambda\cos\theta\sin(\pi/2-0.05))\\ &\quad+\overline{\Omega}(\theta,\lambda;9,\pi/2-0.05,\pi/2-0.1)-\overline{\Omega}(\theta,\lambda;8,-\pi/2-0.1,-\pi/2+0.1)\,, \qquad (4.11)\end{aligned}$$

where $\overline{\Omega}$ is given in (4.7). The first term in this equation correspond to a stream function for zonal flow (or solid body rotation) at angle nearly coincident with the north pole. The second two terms generate a rotating flow near the north pole. See, Figure 1(c) for a plot of this field as viewed from the north pole.

4.2 Standard RBF Interpolation

We briefly discuss the standard RBF interpolation method used for comparison with the divergence-free method. We refer the reader to any of the books [2, 16, 30] for a more in depth discussion of the standard method.

Similar to the beginning of this section, let $\mathbf{x}_k = (x_k, y_k, z_k)^T$, $k = 1, \ldots, N$, be distinct data locations on the unit sphere and $\mathbf{u}_k = (\gamma_k, \delta_k, 0)^T$, $k = 1, \ldots, N$, be the corresponding samples of some divergence-free vector field in spherical coordinates at these locations. For the standard RBF interpolation method, we first translate $\mathbf{u}_k$ to Cartesian coordinates $\mathbf{t}_k$ using (4.3), and then interpolate the samples from each component in Cartesian coordinates separately.

Let (t_k^x, t_k^y, t_k^z) correspond to the components of $\mathbf{t}_k$ in the x, y, and z directions, respectively. We define the standard RBF interpolant $\mathbf{p}(\mathbf{x})$ to the vector field as

$$\mathbf{p}(\mathbf{x}) = \begin{bmatrix} p^x(\mathbf{x}) \\ p^y(\mathbf{x}) \\ p^z(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^N b_k^x \Phi(|\mathbf{x}-\mathbf{x}_k|) \\ \sum_{k=1}^N b_k^y \Phi(|\mathbf{x}-\mathbf{x}_k|) \\ \sum_{k=1}^N b_k^z \Phi(|\mathbf{x}-\mathbf{x}_k|) \end{bmatrix}, \qquad (4.12)$$

where the expansion coefficients b_k^x, b_k^y, and b_k^z, $k = 1, \ldots, N$, are determined from the interpolation conditions $\mathbf{p}(\mathbf{x}_k) = \mathbf{t}_k$, $k = 1, \ldots, N$. This leads to the following N-by-N symmetric matrix problem with three right-hand sides:

$$\begin{bmatrix} \Phi(|\mathbf{x}_1-\mathbf{x}_1|) & \Phi(|\mathbf{x}_1-\mathbf{x}_2|) & \cdots & \Phi(|\mathbf{x}_1-\mathbf{x}_N|) \\ \Phi(|\mathbf{x}_1-\mathbf{x}_1|) & \Phi(|\mathbf{x}_2-\mathbf{x}_2|) & \cdots & \Phi(|\mathbf{x}_2-\mathbf{x}_N|) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(|\mathbf{x}_N-\mathbf{x}_1|) & \Phi(|\mathbf{x}_N-\mathbf{x}_2|) & \cdots & \Phi(|\mathbf{x}_N-\mathbf{x}_N|) \end{bmatrix} \begin{bmatrix} b_1^x & b_1^y & b_1^z \\ b_2^x & b_2^y & b_2^z \\ \vdots & \vdots & \vdots \\ b_N^x & b_N^y & b_N^z \end{bmatrix} = \begin{bmatrix} t_1^x & t_1^y & t_1^z \\ t_2^x & t_2^y & t_2^z \\ \vdots & \vdots & \vdots \\ t_N^x & t_N^y & t_N^z \end{bmatrix}.$$

This matrix is positive definite for all positive definite radial functions $\Phi(r)$ and conditionally negative definite with $N-1$ negative eigenvalues and 1 positive eigenvalue for all order 1 positive definite radial functions (cf. [2, 16, 30]). Note that for the standard method a linear system of size N-by-N needs to be solved, whereas for the divergence-free method the solution of a $2N$-by-$2N$ system is required. However, the standard method completely decouples the components of the vector fields and does not exploit any information about its divergence-free nature.

If desired, the standard RBF interpolant $\mathbf{p}(\mathbf{x})$ can be translated back to spherical coordinates using $Q^T\mathbf{p}(\mathbf{x})$, where Q is given by in (4.2) with (θ, λ) the corresponding spherical coordinates of $\mathbf{x}$.

4.3 Node Distributions

Both standard and divergence-free RBF interpolation are well-posed for *any* distinct set of nodes on the surface of the sphere. For interpolation in $\mathbb{R}^d$ with the standard method, studies have shown that best results are achieved with roughly evenly distributed nodes [16, pp. 59–61]. Since only a maximum of 20 nodes can be evenly distributed on a sphere, there are a multitude of algorithms to define "even" distribution for larger numbers of nodes, such as equal partitioned area, convex hull approaches, electrostatic repulsion, etc. [12]. Although any of these will suffice, we have decided to use the electrostatic repulsion nodes (which are also known as the minimum energy (ME) nodes) for our tests since these nodes do not line up along any vertices or lines, emphasizing the arbitrary node layout of the RBF technique. Many different of these node sets are also readily available for download [32].

Let $\mathbf{x}_k = (x_k, y_k, z_k)^T$, $k = 1, \dots, N$ be a set of N nodes on the unit sphere $\mathbb{S}^2$ and consider the following measure of the density of the points:

$$h = \max_{\mathbf{x}\in\mathbb{S}^2} \min_{1\le k\le N} \operatorname{dist}(\mathbf{x}, \mathbf{x}_k) , \tag{4.13}$$

where dist is the geodesic distance from $\mathbf{x}$ to $\mathbf{x}_k$ [see, (4.8)]. This quantity is referred to as the mesh-norm [32, 17] and, geometrically, it represents the radius of the largest spherical cap that can be placed on the sphere without covering any nodes $\mathbf{x}_k$. 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. Figure 2 displays the $N = 1024$ ME node distribution, which is also the same node distribution used to display the vector fields in Figure 1.

The mesh norm is also of practical importance since it appears in many proofs of error bounds for standard RBF interpolation on the sphere (e.g., [17, 15]). Indeed, in the context of infinitely smooth radial kernels, it is shown in [17] that, provided the underlying function being interpolated is sufficiently smooth, the standard RBF interpolation method converges (in the max. norm) like $h^{-1/2}e^{-c/4h}$, i.e., at a geometric rate, for some constant $c > 0$ that depends on the radial kernel. For the ME node sets, convergence will thus proceed like $N^{1/4}e^{-c\sqrt{N}/4}$. In the experiments that follow, we test this convergence rate also for the divergence-free RBF method.

4.4 Radial Kernels

As proved in Theorem 2, any positive definite or order 1 positive definite radial kernel $\Phi(r)$ leads to a well-posed divergence-free RBF interpolation problem. As noted above, this

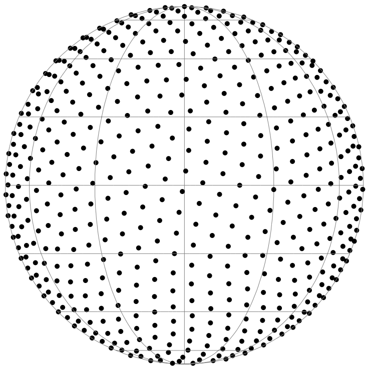

FIGURE 2 Orthographic projection of the $N = 1024$ minimum energy (ME) node distribution on the sphere used in the examples. Solid black circles mark the node locations.

result also holds for the standard RBF method. While this allows for a very large choice of available kernels, we have selected three for the numerical tests. These are listed in Table 1 together with the corresponding kernels $F(r)$ and $G(r)$ used in the divergence-free RBF method. We briefly describe them below:

- The Gaussian (GA) kernel is undoubtedly the most well-known of all positive definite radial kernels and it dates back to Schoenberg's pioneering work [27] in the field. As indicated in the table, it is an infinitely smooth kernel.
- The multiquadric (MQ) kernel is used extensively in applications and originated from Hardy's pioneering work [13] in RBFs. It is an order 1 positive definite kernel and is similarly infinitely smooth.
- The Matérn (MA_ν) class of kernels was introduced in [22] and is arguably the most important and most popular family of kernels for statistical applications of RBFs [11]. These kernels are positive definite, but have finite smoothness that depends on the parameter ν as indicated in Table 1. When ν is a positive half-integer, they reduce to the product of an exponential with a polynomial. For example, for $\nu = \frac{13}{2}$, we get the following Matérn kernel:

$$\mathrm{MA}_{\frac{13}{2}}: \quad \Phi(r) = e^{-\varepsilon r}\left(1 + (\varepsilon r) + \frac{5}{11}(\varepsilon r)^2 + \frac{4}{33}(\varepsilon r)^3 + \frac{2}{99}(\varepsilon r)^4 + \frac{1}{495}(\varepsilon r)^5 + \frac{1}{10395}(\varepsilon r)^6\right),$$

which, according to Table 1, is C^6 (recall $r = |x|$). For the numerical examples, we use Matérn kernels of differing smoothness that correspond to the smoothness of the test vector field. However, we note that, since the divergence-free RBF method depends on the second derivative of the radial kernels (the $G(r)$ function in the third column of Table 1), these interpolants will have two less orders of smoothness. In the case of the Matérn family, this means that we must use the kernel $\mathrm{MA}_{\nu+2}$ in the divergence-free RBF method to be consistent with the standard RBF method when using the kernel MA_ν. The kernels used for each test are summarized in Table 2.

TABLE 1

The radial kernels used in the numerical examples. In all cases, ε is the shape parameter and is always positive. For the Matérn class, K_ν corresponds to the K-Bessel function of order ν, ν is a positive real number, and m is a nonnegative integer.

Radial Kernel	$\Phi(r)$	$F(r) = \frac{1}{r}\Phi'(r)$	$G(r) = \frac{1}{r}F'(r)$
Gaussian (GA)	$e^{-(\varepsilon r)^2}$	$-2\varepsilon^2 e^{-(\varepsilon r)^2}$	$4\varepsilon^4 e^{-(\varepsilon r)^2}$
Smoothness	C^∞	C^∞	C^∞
Multiquadric (MQ)	$\sqrt{1+(\varepsilon r)^2}$	$\dfrac{\varepsilon^2}{\sqrt{1+(\varepsilon r)^2}}$	$-\dfrac{\varepsilon^4}{(1+(\varepsilon r)^2)^{3/2}}$
Smoothness	C^∞	C^∞	C^∞
Matérn (MA$_\nu$)	$\dfrac{2^{1-\nu}}{\Gamma(\nu)}(\varepsilon r)^\nu K_\nu(\varepsilon r),$	$-\dfrac{2^{1-\nu}}{\Gamma(\nu)}\varepsilon^2(\varepsilon r)^{\nu-1}K_{\nu-1}(\varepsilon r),$	$\dfrac{2^{1-\nu}}{\Gamma(\nu)}\varepsilon^4(\varepsilon r)^{\nu-2}K_{\nu-2}(\varepsilon r),$
Smoothness	$C^m,\ \nu > m \geq 0$	$C^{m-1},\ \nu > m \geq 1$	$C^{m-2},\ \nu > m \geq 2$

We can see from Table 1 that each of the kernels for the numerical examples features a free shape parameter $\varepsilon > 0$. In all cases, as ε decreases to zero the kernels become increasingly flat. It has generally been reported in the literature that, for the standard RBF method, there is typically an optimal value of ε that produces the best accuracy and this value tends to decrease with increasing smoothness of the underlying function being approximated (e.g., [24]). However, as ε decreases, the shifted radial kernels $\Phi(|\mathbf{x} - \mathbf{x}_k|)$ in the RBF interpolant become less and less distinguishable from one another, leading to ill-conditioning in the resulting interpolation matrices. This is the so-called "uncertainty principle" for RBF interpolation [26]. While it has been shown that RBF interpolants are overall well-conditioned even in the limit of $\varepsilon \to 0$ (cf. [3, 18]), special algorithms like Contour-Padé [6] and RBF-QR [5] are needed for these smaller values.

In the numerical results that follow, we do not employ any of these special algorithms or try to determine the "optimal ε." Instead, we select ε so that the condition numbers of the standard and divergence-free RBF interpolation matrices is roughly equal and remains roughly constant as the number of nodes increases. As shown next, we are able to get good results with this strategy. We leave the exploration of the dependence of the divergence-free RBF method on the shape parameter ε to a separate study.

TABLE 2

The Matérn (MA_ν) kernels used in the standard and divergence-free methods (and their smoothness) for each of the test vector fields.

Test Vector Field	Standard RBF Method	Divergence-free RBF Method	Interpolant Smoothness
Ω_1	$\mathrm{MA}_{\frac{13}{2}}$ 10pt	$\mathrm{MA}_{\frac{17}{2}}$	C^6
Ω_2	$\mathrm{MA}_{\frac{3}{2}}$	$\mathrm{MA}_{\frac{7}{2}}$	C^1
Ω_3	$\mathrm{MA}_{\frac{13}{2}}$	$\mathrm{MA}_{\frac{17}{2}}$	C^6

4.5 Numerical Results

For each of the three example vector fields (4.9)–(4.11), we compute the (vector-valued) divergence-free and standard RBF interpolants based on the three radial kernels mentioned above for ME node distributions of increasing size. We then evaluate these interpolants on a set of 21,952 nodes which densely cover the surface of the sphere and are generated by the "spiral points" algorithm in [25]. Finally, we compute the difference between these values and the true vector fields at the evaluation points, and compute the relative ℓ_∞ (i.e., max. norm) error in each component, (γ, δ), of the vector field. The results are presented numerically in Tables 3–5 and graphically in Figures 3–5 for the respective fields (4.9)–(4.11).

TABLE 3

Comparison of the relative ℓ_∞ error for the divergence-free RBF method (Div Free) vs. the standard RBF method (Std) for the vector field $\mathbf{u}$ generated from the stream function Ω_1 in (4.9). N is corresponds to number of ME nodes (see, Section 4.3) used for sampling the function. $\sqrt{N}$ is included since it is approximately inversely proportional to the spacing of the nodes [as measured by (4.13)]. The last column pair corresponds to the ℓ_∞ error in the divergence of interpolants of the two methods.

			Example Vector Field 1, $\mathbf{u} = (\gamma, \delta, 0)^T$					
			Relative ℓ_∞ Error, γ		Relative ℓ_∞ Error, δ		ℓ_∞ Error, $\nabla \cdot \mathbf{u}$	
	N	$\sqrt{N}$	Div. Free	Std.	Div. Free	Std.	Div. Free	Std.
GA Kernel	529	23	$5.40 \cdot 10^{-1}$	$4.58 \cdot 10^{-1}$	$1.64 \cdot 10^{-1}$	$1.94 \cdot 10^{-1}$	0	$2.04 \cdot 10^{1}$
	1024	32	$1.00 \cdot 10^{-2}$	$6.31 \cdot 10^{-2}$	$4.76 \cdot 10^{-3}$	$2.60 \cdot 10^{-2}$	0	$6.97 \cdot 10^{0}$
	1521	39	$1.26 \cdot 10^{-3}$	$1.86 \cdot 10^{-2}$	$6.02 \cdot 10^{-4}$	$7.41 \cdot 10^{-3}$	0	$1.36 \cdot 10^{0}$
	2025	45	$2.26 \cdot 10^{-4}$	$2.54 \cdot 10^{-3}$	$7.38 \cdot 10^{-5}$	$1.34 \cdot 10^{-3}$	0	$2.32 \cdot 10^{-1}$
	2601	51	$3.34 \cdot 10^{-5}$	$2.76 \cdot 10^{-4}$	$1.76 \cdot 10^{-5}$	$1.72 \cdot 10^{-4}$	0	$3.50 \cdot 10^{-2}$
MQ Kernel	529	23	$2.49 \cdot 10^{-1}$	$4.61 \cdot 10^{-1}$	$1.05 \cdot 10^{-1}$	$1.91 \cdot 10^{-1}$	0	$1.97 \cdot 10^{1}$
	1024	32	$1.60 \cdot 10^{-2}$	$6.65 \cdot 10^{-2}$	$8.58 \cdot 10^{-3}$	$2.78 \cdot 10^{-2}$	0	$6.21 \cdot 10^{0}$
	1521	39	$3.80 \cdot 10^{-3}$	$2.10 \cdot 10^{-2}$	$1.64 \cdot 10^{-3}$	$8.48 \cdot 10^{-3}$	0	$1.50 \cdot 10^{0}$
	2025	45	$5.88 \cdot 10^{-4}$	$2.99 \cdot 10^{-3}$	$2.18 \cdot 10^{-4}$	$1.46 \cdot 10^{-3}$	0	$3.18 \cdot 10^{-1}$
	2601	51	$8.63 \cdot 10^{-5}$	$3.49 \cdot 10^{-4}$	$4.68 \cdot 10^{-5}$	$2.49 \cdot 10^{-4}$	0	$5.36 \cdot 10^{-2}$
MA_ν Kernel	529	23	$2.39 \cdot 10^{-1}$	$4.63 \cdot 10^{-1}$	$1.03 \cdot 10^{-1}$	$1.92 \cdot 10^{-1}$	0	$2.00 \cdot 10^{1}$
	1024	32	$1.76 \cdot 10^{-2}$	$6.88 \cdot 10^{-2}$	$9.65 \cdot 10^{-3}$	$2.87 \cdot 10^{-2}$	0	$6.43 \cdot 10^{0}$
	1521	39	$4.61 \cdot 10^{-3}$	$2.23 \cdot 10^{-2}$	$1.95 \cdot 10^{-3}$	$8.96 \cdot 10^{-3}$	0	$1.58 \cdot 10^{0}$
	2025	45	$6.81 \cdot 10^{-4}$	$3.33 \cdot 10^{-3}$	$2.46 \cdot 10^{-4}$	$1.52 \cdot 10^{-3}$	0	$3.69 \cdot 10^{-1}$
	2601	51	$8.54 \cdot 10^{-5}$	$4.45 \cdot 10^{-4}$	$5.23 \cdot 10^{-5}$	$3.00 \cdot 10^{-4}$	0	$7.03 \cdot 10^{-2}$

TABLE 4

Same as Table 3, but for the vector field generated from the stream function Ω_2 in (4.10).

			Example Vector Field 2, $\mathbf{u} = (\gamma, \delta, 0)^T$					
			Relative ℓ_∞ Error, γ		Relative ℓ_∞ Error, δ		ℓ_∞ Error, $\nabla \cdot \mathbf{u}$	
	N	$\sqrt{N}$	Div. Free	Std.	Div. Free	Std.	Div. Free	Std.
GA Kernel	529	23	$4.36 \cdot 10^{-2}$	$3.39 \cdot 10^{-2}$	$1.11 \cdot 10^{-2}$	$1.04 \cdot 10^{-2}$	0	$9.31 \cdot 10^{-1}$
	1024	32	$5.88 \cdot 10^{-2}$	$4.80 \cdot 10^{-2}$	$1.12 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	0	$9.83 \cdot 10^{-1}$
	1521	39	$3.42 \cdot 10^{-2}$	$2.98 \cdot 10^{-2}$	$1.15 \cdot 10^{-2}$	$9.02 \cdot 10^{-3}$	0	$1.04 \cdot 10^{0}$
	2601	51	$2.71 \cdot 10^{-2}$	$1.15 \cdot 10^{-2}$	$5.31 \cdot 10^{-3}$	$2.38 \cdot 10^{-3}$	0	$5.86 \cdot 10^{-1}$
	4096	64	$1.42 \cdot 10^{-2}$	$7.85 \cdot 10^{-3}$	$3.53 \cdot 10^{-3}$	$2.12 \cdot 10^{-3}$	0	$4.51 \cdot 10^{-1}$
MQ Kernel	529	23	$3.83 \cdot 10^{-2}$	$3.38 \cdot 10^{-2}$	$1.16 \cdot 10^{-2}$	$1.19 \cdot 10^{-2}$	0	$1.07 \cdot 10^{0}$
	1024	32	$2.23 \cdot 10^{-2}$	$4.24 \cdot 10^{-2}$	$5.99 \cdot 10^{-3}$	$1.14 \cdot 10^{-2}$	0	$9.01 \cdot 10^{-1}$
	1521	39	$1.87 \cdot 10^{-2}$	$3.02 \cdot 10^{-2}$	$6.44 \cdot 10^{-3}$	$8.94 \cdot 10^{-3}$	0	$9.28 \cdot 10^{-1}$
	2601	51	$7.54 \cdot 10^{-3}$	$1.09 \cdot 10^{-2}$	$2.16 \cdot 10^{-3}$	$2.54 \cdot 10^{-3}$	0	$5.82 \cdot 10^{-1}$
	4096	64	$4.05 \cdot 10^{-3}$	$7.76 \cdot 10^{-3}$	$1.02 \cdot 10^{-3}$	$2.02 \cdot 10^{-3}$	0	$4.12 \cdot 10^{-1}$
MA$_\nu$ Kernel	529	23	$4.12 \cdot 10^{-2}$	$5.64 \cdot 10^{-2}$	$1.15 \cdot 10^{-2}$	$2.10 \cdot 10^{-2}$	0	$2.33 \cdot 10^{0}$
	1024	32	$3.38 \cdot 10^{-2}$	$4.11 \cdot 10^{-2}$	$9.36 \cdot 10^{-3}$	$9.06 \cdot 10^{-3}$	0	$1.08 \cdot 10^{0}$
	1521	39	$2.40 \cdot 10^{-2}$	$3.17 \cdot 10^{-2}$	$7.04 \cdot 10^{-3}$	$8.39 \cdot 10^{-3}$	0	$7.58 \cdot 10^{-1}$
	2601	51	$9.16 \cdot 10^{-3}$	$1.44 \cdot 10^{-2}$	$2.56 \cdot 10^{-3}$	$3.84 \cdot 10^{-3}$	0	$7.10 \cdot 10^{-1}$
	4096	64	$6.57 \cdot 10^{-3}$	$8.67 \cdot 10^{-3}$	$1.66 \cdot 10^{-3}$	$2.25 \cdot 10^{-3}$	0	$4.85 \cdot 10^{-1}$

TABLE 5

Same as Table 3, but for the vector field generated from the stream function Ω_3 in (4.11).

			Example Vector Field 3, $\mathbf{u} = (\gamma, \delta, 0)^T$					
			Relative ℓ_∞ Error, γ		Relative ℓ_∞ Error, δ		ℓ_∞ Error, $\nabla \cdot \mathbf{u}$	
	N	$\sqrt{N}$	Div. Free	Std.	Div. Free	Std.	Div. Free	Std.
GA Kernel	529	23	$2.95 \cdot 10^{-1}$	$1.71 \cdot 10^{-1}$	$2.34 \cdot 10^{-1}$	$5.39 \cdot 10^{-1}$	0	$4.63 \cdot 10^{1}$
	1024	32	$1.31 \cdot 10^{-2}$	$4.73 \cdot 10^{-2}$	$1.21 \cdot 10^{-2}$	$1.08 \cdot 10^{-1}$	0	$2.02 \cdot 10^{1}$
	1521	39	$2.04 \cdot 10^{-3}$	$1.82 \cdot 10^{-2}$	$1.93 \cdot 10^{-3}$	$1.89 \cdot 10^{-2}$	0	$8.23 \cdot 10^{0}$
	2025	45	$3.70 \cdot 10^{-4}$	$4.59 \cdot 10^{-3}$	$2.72 \cdot 10^{-4}$	$4.32 \cdot 10^{-3}$	0	$1.21 \cdot 10^{0}$
	2601	51	$5.03 \cdot 10^{-5}$	$1.22 \cdot 10^{-3}$	$3.57 \cdot 10^{-5}$	$1.41 \cdot 10^{-3}$	0	$5.45 \cdot 10^{-1}$
MQ Kernel	529	23	$2.33 \cdot 10^{-1}$	$1.69 \cdot 10^{-1}$	$2.64 \cdot 10^{-1}$	$5.25 \cdot 10^{-1}$	0	$4.55 \cdot 10^{1}$
	1024	32	$1.14 \cdot 10^{-2}$	$4.16 \cdot 10^{-2}$	$1.27 \cdot 10^{-2}$	$1.02 \cdot 10^{-1}$	0	$1.92 \cdot 10^{1}$
	1521	39	$4.00 \cdot 10^{-3}$	$1.77 \cdot 10^{-2}$	$3.31 \cdot 10^{-3}$	$2.08 \cdot 10^{-2}$	0	$7.74 \cdot 10^{0}$
	2025	45	$1.16 \cdot 10^{-3}$	$5.11 \cdot 10^{-3}$	$7.14 \cdot 10^{-4}$	$4.83 \cdot 10^{-3}$	0	$1.22 \cdot 10^{0}$
	2601	51	$7.59 \cdot 10^{-5}$	$1.98 \cdot 10^{-3}$	$8.83 \cdot 10^{-5}$	$2.15 \cdot 10^{-3}$	0	$7.90 \cdot 10^{-1}$
MA_ν Kernel	529	23	$2.13 \cdot 10^{-1}$	$1.70 \cdot 10^{-1}$	$2.66 \cdot 10^{-1}$	$5.22 \cdot 10^{-1}$	0	$4.56 \cdot 10^{1}$
	1024	32	$1.46 \cdot 10^{-2}$	$3.89 \cdot 10^{-2}$	$1.66 \cdot 10^{-2}$	$1.00 \cdot 10^{-1}$	0	$1.90 \cdot 10^{1}$
	1521	39	$5.14 \cdot 10^{-3}$	$1.82 \cdot 10^{-2}$	$4.19 \cdot 10^{-3}$	$2.19 \cdot 10^{-2}$	0	$7.89 \cdot 10^{0}$
	2025	45	$1.47 \cdot 10^{-3}$	$5.56 \cdot 10^{-3}$	$8.74 \cdot 10^{-4}$	$5.17 \cdot 10^{-3}$	0	$1.35 \cdot 10^{0}$
	2601	51	$1.30 \cdot 10^{-4}$	$2.30 \cdot 10^{-3}$	$1.33 \cdot 10^{-4}$	$2.53 \cdot 10^{-3}$	0	$9.17 \cdot 10^{-1}$

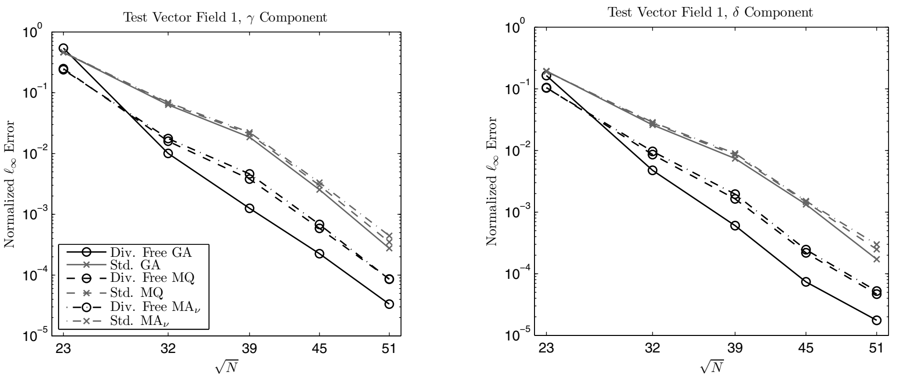

FIGURE 3 Plot of the results from Table 3 for the vector field $\mathbf{u} = (\gamma, \delta, 0)^T$ generated from the stream function Ω_1 in (4.9). Note the log vs. linear scale on the plots.

Comparing the results for the divergence-free and standard interpolants for the first vector field (4.9) (cf. Table 3 and Figure 3), we see that the error in the former is smaller than the latter for all three of the kernels and values of N (except the GA kernel and $N = 529$). This difference becomes more pronounced as N increases, with the error in the divergence-free method about an order of magnitude smaller. From the plot of the errors in the two components of the vector field in Figure 3 (displayed on a log vs. linear scale), we see that both the divergence-free and standard methods are exhibiting geometric convergence [recall that $\sqrt{N}$ is approximately inversely proportional to the spacing of the nodes as measured by (4.13)]. This is expected for the latter method, as discussed in Section 4.3, since both components of the vector field are C^∞ functions. As mentioned in the introduction, a theoretical understanding of the accuracy of the divergence-free method is not yet available, but will be pursued in a separate study. The GA seems to be the most accurate of the three kernels in both the divergence-free and standard methods, while the MQ is second. This is expected since these are both C^∞ kernels, while the MA$_\nu$ is only C^6. By increasing the smoothness of the MA$_\nu$ kernel, we expect to get results closer to those of the GA and MQ.

The second vector field (4.10) is only C^1 and, as expected, we see from Table 4 and Figure 4 that the convergence rate of both the divergence-free and standard interpolants has been diminished to low-order algebraic (note the log vs. log scale on both plots in the figure) and is much less steady than for the first field. For the MQ and MA$_\nu$ kernels, we see that the divergence-free method is again out performing the corresponding standard methods, with the MQ producing the best results. This is a bit surprising since the MQ kernel is C^∞ while the MA$_\nu$ kernel used here is C^1, matching that of the vector field. Interestingly, for the GA kernel, the error in both components of the vector field for the standard method is smaller than for the divergence-free method. However, as we can see in the last pair of columns of Table 4, the latter method results in an exactly divergent-free field while the divergence of the former is relatively large.

The results for the last vector field (4.11) in Table 5 and Figure 5 are similar to those of the first, with the divergence-free method far out performing the standard method for all the radial kernels. While the components of this field in spherical coordinates are discontinuous at the poles, we see that this has no effect on the high accuracy that is achieved with either the divergence-free or standard method. Indeed, Figure 4 (plotted on a log vs. linear scale)

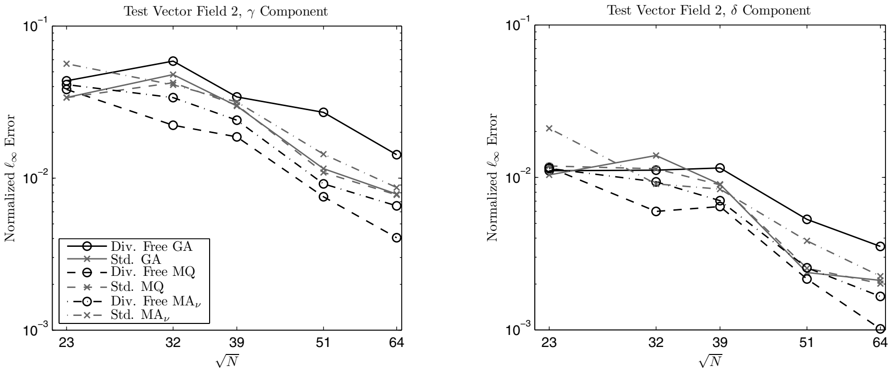

FIGURE 4 Plot of the results from Table 4 for the vector field $\mathbf{u} = (\gamma, \delta, 0)^T$ generated from the stream function Ω_2 in (4.10). Note the log vs. log scale on the plots.

again shows both of the these interpolation methods exhibiting geometric convergence.

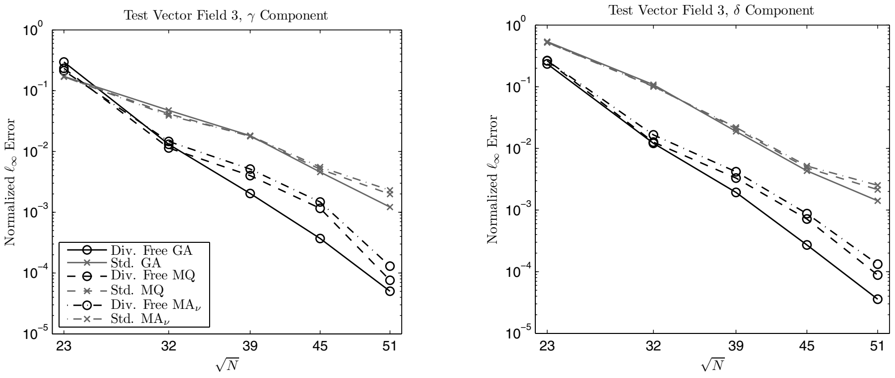

FIGURE 5 Plot of the results from Table 5 for the vector field $\mathbf{u} = (\gamma, \delta, 0)^T$ generated from the stream function Ω_3 in (4.11). Note the log vs. linear scale on the plots.

As mentioned in Section 4.2, the divergence-free method requires solving a $2N$-by-$2N$ linear system of equations whereas the standard method requires solving an N-by-N system. So, if comparing the two methods in terms of computational cost, one should compare the accuracy of the divergence-free method with N nodes and the standard method with $2N$ nodes. When this is done for the results from Tables 3–5, we see that the error in the standard method is generally smaller. However, regardless of the number of nodes, the standard method fails to preserve the divergence-free nature of the field. Furthermore, for experimental data, it is not always possible to increase the number of observations in order to increase the accuracy of the approximation. As the results indicate, in case of a fixed number of nodes N, the divergence-free method generally produces much better results than the standard method.

5. Concluding Remarks

We have presented a new RBF method for fitting divergence-free vector fields defined on some surface imbedded in $\mathbb{R}^3$. The method allows for scattered data locations (i.e., no grids or meshes), does not suffer from any coordinate singularities (e.g., pole singularities in the case of the sphere), results in a positive definite interpolation matrix (i.e., it is well-posed), and preserves the divergence-free nature of the field. For the case of the sphere, numerical results indicate that the method far out performs more traditional RBF methods.

We conclude by noting that, while we have focused primarily on interpolation, it is sometimes more desirable when fitting "real-world" data to use approximation, such as least-squares fitting. In this case, an approximant is constructed from fewer linear combinations of radial kernels than available data (cf. [16, pp. 61–65]). The divergence-free RBF method carries over to this more general situation and still results in an approximant that is divergence-free.

References

[1] Bishop, R. L. and Goldberg, S. I. *Tensor Analysis on Manifolds,* Macmillan, New York, (1968).

[2] Buhmann, M. D. *Radial Basis Functions: Theory and Implementations,* Vol. 12 of *Cambridge Monographs on Applied and Computational Mathematics,* Cambridge University Press, Cambridge, (2003).

[3] Driscoll, T. A. and Fornberg, B. (2002). Interpolation in the limit of increasingly flat radial basis functions, *Comput. Math. Appl.* **43**, 413–422.

[4] Flanders, H. *Differential Forms,* Academic Press, New York, (1963).

[5] Fornberg, B. and Piret, C. (2006). A stable algorithm for flat radial basis functions on a sphere, to be submitted.

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

[7] Fuselier, E. (2006). Improved stability estimates and a characterization of the native space for marix-valued RBFs, preprint.

[8] Fuselier, E. (2006). Sobolev-type approximation rates for divergence-free and curl-free RBF interpolants, preprint.

[9] Fuselier, E. Refined error estimates for matrix-valued radial basis functions, Ph.D. thesis, Texas A&M University, College Station, TX 77843-3368, (2006).

[10] Gel'fand, I. M. and Vilenkin, N. Ya. *Generalized Functions,* Vol. 4, Academic Press, New York and London, (1964).

[11] Gneiting, T. (1999). Correlation functions for atmospheric data analysis, *Q. J. R. Meteorol. Soc.* **125**, 2449–2464.

[12] Hardin, D. P. and Saff, E. B. (2004). Discretizing manifolds via minimum energy points, *Notices Amer. Math. Soc.* **51**, 1186–1194.

[13] Hardy, R. L. (1971). Multiquadric equations of topograpy and other irregular surfaces, *J. Geophy. Res.* **76**, 1905–1915.

[14] Holton, J. R. (1992). *An Introduction to Dynamic Meteorology,* Academic Press, San Francisco, 3rd ed., (1992).

[15] Hubbert, S. and Morton, T. (2004). L^p-error estimates for radial basis function interpolation on the sphere, *J. Approx. Theory* **129**, 58–77.

[16] Iske, A. *Multiresolution Methods in Scattered Data Modelling,* Vol. 37 of *Lecture Notes in Computational Science and Engineering,* Springer-Verlag, Heidelberg, (2004).

[17] Jetter, K., Stöckler, J., and Ward, J. D. (1999). Error estimates for scattered data interpolation on spheres, *Math. Comput.* **68**, 733–747.

[18] Larsson, E. and Fornberg, B. (2005). Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, *Comput. Math. Appl.* **49**, 103–130.

[19] Lowitzsch, S. (2005). Error estimates for matrix-valued radial basis function interpolation, *J. Approx. Theory* **137**, 238–249.

[20] Lowitzsch, S. (2005). Matrix-valued radial basis functions: Stability estimates and applications, *Adv. Comput. Math.* **23**, 299–315.

[21] Madych, W. R. and Nelson, S. A. (1990). Multivariate interpolation and conditionally positive definite functions II, *Math. Comp.* **54**, 211–230.

[22] Matérn, B. *Spatial Variation,* Vol. 36 of *Lecture Notes in Statistics,* Springer-Verlag, Heidelberg, 2nd ed., (1986).

[23] Narcowich, F. J. and Ward, J. D. (1994). Generalized Hermite interpolation via matrix-valued conditionally positive definite functions, *Math. Comp.* **63**, 661–687.

[24] Rippa, S. (1999). An algorithm for selecting a good value for the parameter c in radial basis function interpolation, *Adv. Comput. Math.* **11**, 193–210.

[25] Saff, E. B. and Kuijlaars, A. B. J. (1997). Distributing many points on a sphere, *Math. Intelligencer* **19**, 5–11.

[26] Schaback, R. (1995). Error estimates and condition numbers for radial basis function interpolants, *Adv. Comput. Math.* **3**, 251–264.

[27] Schoenberg, I. J. (1938). Metric spaces and completely monotone functions, *Ann. Math.* **39**, 811–841.

[28] Sun, X. (1993). Conditionally positive definite functions and their application to multivariate interpolations, *J. Approx. Theory* **74**, 159–180.

[29] Swarztrauber, P. N. (1981). The approximation of vector functions and their derivatives on the sphere, *SIAM J. Numer. Anal.* **18**, 191–210.

[30] Wendland, H. *Scattered Data Approximation,* Vol. 17 of *Cambridge Monographs on Applied and Computational Mathematics,* Cambridge University Press, Cambridge, (2005).

[31] Williamson, D. L., Drake, J. B., Hack, J. J., Jakob, R., and Swarztrauber, P. N. (1992). A standard test set for numerical approximations to the shallow water equations in spherical geometry, *J. Comput. Phys.* **102**, 211–224.

[32] Womersley, R. S. and Sloan, I. H. Interpolation and cubature on the sphere, Website. `http://web.maths.unsw.edu.au/~rsw/Sphere/`.

Received

Revision received September 22, 2006

Department of Mathematics, Texas A&M University, College Station, TX 77843-3368
e-mail: fnarc@math.tamu.edu

Department of Mathematics, Texas A&M University, College Station, TX 77843-3368

Department of Mathematics, University of Utah, Salt Lake City, UT 84112-0090