1-2021

A Partition of Unity Method for Divergence-Free or Curl-Free Radial Basis Function Approximation

Kathryn P. Drake
Boise State University

Edward J. Fuselier
High Point University

Grady B. Wright
Boise State University

A PARTITION OF UNITY METHOD FOR DIVERGENCE-FREE OR CURL-FREE RADIAL BASIS FUNCTION APPROXIMATION

KATHRYN P. DRAKE*, EDWARD J. FUSELIER†, AND GRADY B. WRIGHT*

Abstract. Divergence-free (div-free) and curl-free vector fields are pervasive in many areas of science and engineering, from fluid dynamics to electromagnetism. A common problem that arises in applications is that of constructing smooth approximants to these vector fields and/or their potentials based only on discrete samples. Additionally, it is often necessary that the vector approximants preserve the div-free or curl-free properties of the field to maintain certain physical constraints. Div/curl-free radial basis functions (RBFs) are a particularly good choice for this application as they are meshfree and analytically satisfy the div-free or curl-free property. However, this method can be computationally expensive due to its global nature. In this paper, we develop a technique for bypassing this issue that combines div/curl-free RBFs in a partition of unity framework, where one solves for local approximants over subsets of the global samples and then blends them together to form a div-free or curl-free global approximant. The method is applicable to div/curl-free vector fields in $\mathbb{R}^2$ and tangential fields on two-dimensional surfaces, such as the sphere, and the curl-free method can be generalized to vector fields in $\mathbb{R}^d$. The method also produces an approximant for the scalar potential of the underlying sampled field. We present error estimates and demonstrate the effectiveness of the method on several test problems.

Key words. divergence-free, solenoidal, curl-free, irrotational, partition of unity, potential, radial basis functions

AMS subject classifications. 65D12, 41A05, 41A30

1. Introduction.

Approximating vector fields from scattered samples is a pervasive problem in many scientific applications, including, for example, fluid dynamics, meteorology, magnetohydrodynamics, electromagnetics, gravitational lensing, imaging, and computer graphics. Often these vector fields have certain differential invariant properties related to an underlying physical principle. For example, in incompressible fluid dynamics the velocity of the fluid is divergence-free (div-free) as a consequence of the conservation of mass. Similarly, in electromagnetics the electric field is curl-free in the absence of a time varying magnetic field as a consequence of the conservation of energy. Additionally, the fields may have properties of being tangential to a surface (e.g., the sphere $\mathbb{S}^2$) and have a corresponding surface div-free or curl-free property, as occurs in many areas of geophysical sciences [16]. In several of these applications it is necessary for the approximants to preserve these differential invariants to maintain certain physical constraints. For example, in incompressible flow simulations using the immersed boundary method, excessive volume loss can occur if the approximated velocity field of the fluid is not div-free [4].

To enforce these differential invariants on the approximant, one cannot approximate the individual components of the field separately, but must combine them in a particular way. One idea uses the property that div-free fields (in two dimensions) and curl-free fields can be defined in terms of a scalar potential (e.g., a stream function or electric potential). These methods then compute an approximant for the potential of the field by solving a Poisson equation involving the divergence or curl of the sampled field [5]. A separate idea is to use a vector basis for the approximant that satisfies the underlying differential invariant. This paper develops a radial basis function (RBF) method that uses latter approach, but has similarities to the former.

*Department of Mathematics, Boise State University, Boise, ID (KathrynDrake@u.boisestate.edu, gradywright@boisestate.edu)

†Department of Mathematics, High Point University, High Point, NC (efuselie@highpoint.edu)

RBFs are a main tool for scattered data approximation [18, 20, 47]. In the early 1990s, researchers began to focus on the problem of developing vector RBF interpolants for div-free fields that analytically satisfy the div-free constraint [2, 28, 36]. The idea, as presented in [36], is to use linear combinations of shifts of a matrix-valued kernel, whose columns satisfy the div-free property, to interpolate the samples of given field. Since these kernels are constructed from scalar-valued RBFs, they are referred to as div-free RBFs. These ideas were later extended to curl-free fields in [14, 23]. Further extensions of the idea to vector fields tangential to a two-dimensional surface (e.g., $\mathbb{S}^2$) that are surface div-free or curl-free were given in [37]. Some applications of these div/curl-free RBFs can, for example, be found in [11, 25, 31, 34, 35, 42, 48].

There are, however, issues with scaling div/curl-free RBF interpolation to large data sets. For a data set with N scattered nodes $X = \{\mathbf{x}_j\}_{j=1}^N$, the method requires solving a dN-by-dN linear system, where $d = 2, 3$ is the dimension of the underlying domain. Additionally, each evaluation of the resulting interpolant involves dN terms. If the div/curl-free RBFs are constructed from scalar-valued RBFs with global support, then the linear system is dense and not well suited to iterative methods. To ameliorate these issues, a multilevel framework has been developed for compactly supported div/curl-free RBFs in [17]. However, we take a different approach to reducing the computational cost using the partition of unity method (PUM) [6, 18, 32, 33, 46].

In RBF-PUM, one only needs to solve for local approximants over small subsets of the global data set and then blend them together to form a smooth global approximant. A particular challenge with extending this idea to div/curl-free RBFs is in enforcing that the global approximant is analytically div/curl-free. To overcome this challenge, we use the local div/curl-free RBFs to obtain local approximants to scalar potentials for the field and then blend these together to form a global scalar potential for the entire field. A div/curl-free vector approximant is then obtained by applying the appropriate differential operator to the global scalar potential. The method as presented here will only work for fields that can be defined by scalar potentials, which includes div/curl-free vector fields in $\mathbb{R}^2$, surface div/curl-free tangential fields on two-dimensional surfaces, and curl-free fields in $\mathbb{R}^d$, but not div-free fields in $\mathbb{R}^3$. However, there are several benefits of the method. First, for node sets X that are quasiuniform, the algorithm parameters can be chosen to produce global approximants to the field in $\mathcal{O}(N \log N)$ operations. Second, we have error estimates showing the method can give high rates of convergence, and numerical evidence that rates faster than algebraic with increasing N are possible. Unlike the method from [17], these convergence rates are possible with the fixed complexity of $\mathcal{O}(N \log N)$. Finally, a global approximant for the scalar potential is given directly from the samples without having to compute derivatives of the sampled field or solving a Poisson problem.

As far as we are aware, the only other computationally scalable div-free approximation technique for scattered data is the div-free moving least squares (MLS) method from [45]. The method is used for generating finite difference type discretizations for Stokes' equations. While it worked quite successfully for this application, it can be computationally expensive for more general approximation problems, since it requires solving a new (small) linear system for each evaluation point. For the method we develop, the (small) linear systems are independent of the evaluation points. Additionally, the div-free MLS method does not directly allow the potential for the field to also be approximated.

The rest of the paper is organized as follows. In the next section we introduce some background material necessary for the presentation of the method. Section 3 contains a review of PUM and then presents the div/curl-free RBF-PUM. Error estimates for

the new method are presented in Section 4. Section 5 contains numerical experiments demonstrating the convergence rates of the method on three model problems. The final section contains some concluding remarks.

2. Div/Curl-free RBFs. We review the generalized vector RBF techniques for reconstructing vector fields below, first for div-free fields and then for curl-free fields. In both cases, we focus on approximations of tangential vector fields on smooth, orientable, surfaces embedded in $\mathbb{R}^3$ (which includes $\mathbb{R}^2$ and $\mathbb{S}^2$). In the curl-free case the method extends trivially to $\mathbb{R}^d$. Before discussing these two techniques, we introduce some notation and review some relevant background material.

2.1. Notation and preliminaries. Let $\mathcal{P}$ denote a smooth, orientable surface embedded in $\mathbb{R}^3$, possibly with a boundary, and let $\mathbf{n} \in \mathbb{R}^3$ denote the unit normal vector to $\mathcal{P}$ expressed in the Cartesian basis. When discussing tangential vector fields on $\mathcal{P}$, we use the terms divergence and curl to be tacitly understood to refer to surface divergence and surface curl for $\mathcal{P}$. The surface curl (or *rot*) operator $\mathbf{L}$ and the surface gradient operator $\mathbf{G}$ play a central role in defining div-free and curl-free tangential fields on $\mathcal{P}$. We can express these operators in extrinsic (Cartesian) coordinates as follows:

$$\mathbf{L} = \mathbf{n} \times \nabla, \qquad \mathbf{G} = (I - \mathbf{n}\mathbf{n}^T)\nabla,$$

where ∇ is the standard $\mathbb{R}^3$ gradient, and I is the 3-by-3 identity matrix. It is a well known consequence of Poincaré's Lemma that div-free and curl-free fields are locally images of these operators [13][1]

PROPOSITION 2.1. *Let* $\mathbf{u}$ *be a tangential vector field defined on* $\mathcal{P}$ *then*

1. $\mathbf{u}$ *is div-free iff for each point* $\mathbf{x} \in \mathcal{P}$ *there exists a neighborhood* $U \subset \mathcal{P}$ *and a scalar potential* $\psi : U \longrightarrow \mathbb{R}$ *such that* $\mathbf{u} = \mathbf{L}(\psi)$
2. $\mathbf{u}$ *is curl-free for each point* $\mathbf{x} \in \mathcal{P}$ *there exists a neighborhood* $U \subset \mathcal{P}$ *and a scalar potential* $\varphi : U \longrightarrow \mathbb{R}$ *such that* $\mathbf{u} = \mathbf{G}(\varphi)$

Note that since $\mathbf{L}$ and $\mathbf{G}$ only annihilate constant functions along $\mathcal{P}$, the scalar potentials are unique up to the addition of a constant.

The present method relies on this property as it solves for scalar potentials on overlapping patches covering the domain of interest. Since each of these potentials is unique up to a constant, a straightforward procedure can be derived to determine these values so that the potentials can be shifted to agree over the domain. In three dimensions, div-free vector fields have vector potentials unique up to the addition of the gradient of a harmonic scalar function, and it not clear to us how to adapt the current method to this situation. However, the method will be applicable to curl-free fields in higher dimensions since a vector field $\mathbf{u}$ on $\mathbb{R}^d$ is curl-free if and only if $\mathbf{u} = \nabla\varphi$ for some scalar potential.

In what proceeds, we use the following notation for the $\mathbf{L}$ operator:

$$\mathbf{L} = \underbrace{\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}}_{Q_{\mathbf{x}}} \nabla, \tag{2.1}$$

[1] Poincaré's Lemma is typically given in terms of the exterior derivatve operator d. In this case applying the *Hodge star* operator * to $\mathbf{u}$ before applying Poincaré's Lemma gives the div-free result. For the curl-free result, one starts with $*d\mathbf{u} = 0$ and applying the Hodge star operator to this allows one to apply the lemma.

where $\mathbf{n} = (a_1, a_2, a_3)$ is the unit normal to $\mathcal{P}$ at $\mathbf{x}$. Note that applying $Q_{\mathbf{x}}$ to a vector in $\mathbb{R}^3$ gives the cross product of $\mathbf{n}$ with that vector. Similarly, we express $\mathbf{G}$ as

$$\mathbf{G} = P_{\mathbf{x}}\nabla, \tag{2.2}$$

where $P_{\mathbf{x}} = \mathbf{I} - \mathbf{n}\mathbf{n}^T$ projects any vector at $\mathbf{x}$ on $\mathcal{P}$ into a plane tangent to $\mathcal{P}$ at $\mathbf{x}$.

Two important cases of $\mathcal{P}$ are $\mathcal{P} = \mathbb{R}^2$ and $\mathcal{P} = \mathbb{S}^2$. For the former case, the unit normal is independent of its position and is typically chosen as $\mathbf{n} = (0, 0, 1)$. Using this with (2.1) and (2.2), leads to the standard definition for these operators for vector fields on $\mathbb{R}^2$:

$$\mathbf{L} = \begin{bmatrix} -\partial_y \\ \partial_x \\ 0 \end{bmatrix} \text{ and } \mathbf{G} = \begin{bmatrix} \partial_x \\ \partial_y \\ 0 \end{bmatrix}, \tag{2.3}$$

which can be truncated to remove the unnecessary third component. For $\mathcal{P} = \mathbb{S}^2$, the unit normal at $\mathbf{x}$ is $\mathbf{n} = \mathbf{x}$, but $\mathbf{L}$ and $\mathbf{G}$ do not simplify beyond this.

2.2. Div-free RBF interpolation. Div-free vector RBF interpolants are similar to scalar RBF interpolants in the sense that one constructs them from linear combinations of shifts of a kernel at each of the given data sites. The difference between the approaches is that in the vector case one uses a *matrix-valued kernel* whose columns are div-free. For the sake of brevity, we give the final construction of these kernels and refer the reader to [37] for a rigorous derivation. For more information on scalar-valued RBFs, which we do not discuss here, see any of the books [18, 20, 47].

Let $\phi : \mathbb{R}^3 \times \mathbb{R}^3 \longrightarrow \mathbb{R}$ be a radial kernel in the sense that $\phi(\mathbf{x}, \mathbf{y}) = \eta(\|\mathbf{x} - \mathbf{y}\|)$, for some $\eta : [0, \infty) \longrightarrow \mathbb{R}$, where $\|\cdot\|$ is the vector 2-norm. It is common in this case to simply write $\phi(\mathbf{x}, \mathbf{y}) = \phi(\|\mathbf{x} - \mathbf{y}\|)$. Supposing ϕ has two continuous derivatives, then the matrix kernel Φ_{div} is constructed using the operator $\mathbf{L}$ in (2.1) as

$$\begin{aligned} \Phi_{\text{div}}(\mathbf{x}, \mathbf{y}) = \mathbf{L}_{\mathbf{x}}\mathbf{L}_{\mathbf{y}}^T\phi\left(\|\mathbf{x} - \mathbf{y}\|\right) &= Q_{\mathbf{x}}\left(\nabla_{\mathbf{x}}\nabla_{\mathbf{y}}^T\phi\left(\|\mathbf{x} - \mathbf{y}\|\right)\right)Q_{\mathbf{y}}^T \\ &= Q_{\mathbf{x}}\left(\nabla\nabla^T\phi\left(\|\mathbf{x} - \mathbf{y}\|\right)\right)Q_{\mathbf{y}}, \end{aligned} \tag{2.4}$$

where the subscripts in the differential operators indicate which variables they operate on and, for simplicity, no subscript means they operate on the $\mathbf{x}$ component. Here we have used the fact that the matrix $Q_{\mathbf{y}}$ in (2.1) is skew-symmetric and $\nabla_{\mathbf{y}}^T\phi\left(\|\mathbf{x} - \mathbf{y}\|\right) = -\nabla^T\phi\left(\|\mathbf{x} - \mathbf{y}\|\right)$. For any $\mathbf{c} \in \mathbb{R}^3$ and fixed $\mathbf{y} \in \mathcal{P}$, the vector field $\Phi_{\text{div}}(\mathbf{x}, \mathbf{y})\mathbf{c}$ is tangent to $\mathcal{P}$ and div-free in $\mathbf{x}$, which follows from Proposition 2.1 since

$$\Phi_{\text{div}}(\mathbf{x}, \mathbf{y})\mathbf{c} = Q_{\mathbf{x}}\nabla\left(\nabla^T\phi\left(\|\mathbf{x} - \mathbf{y}\|\right)Q_{\mathbf{y}}\mathbf{c}\right) = \mathbf{L}(\psi(\mathbf{x})), \tag{2.5}$$

where ψ is the potential for $\Phi_{\text{div}}(\mathbf{x}, \mathbf{y})\mathbf{c}$. The second argument of Φ_{div} acts as a shift of the kernel and indicates where the field $\Phi_{\text{div}}\mathbf{c}$ is "centered."

An interpolant to a div-free tangential vector field $\mathbf{u} : \mathcal{P} \longrightarrow \mathbb{R}^3$ sampled at distinct points $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathcal{P}$ can be constructed using Φ_{div} as follows:

$$\mathbf{s}(\mathbf{x}) = \sum_{j=1}^{N} \Phi_{\text{div}}(\mathbf{x}, \mathbf{x}_j)\mathbf{c}_j, \tag{2.6}$$

where the coefficients $\mathbf{c}_j \in \mathbb{R}^3$ are tangent to $\mathcal{P}$ at $\mathbf{x}_j$ (this is necessary to make the interpolation problem well-posed as discussed below) and are chosen so that $\mathbf{s}\big|_X = \mathbf{u}\big|_X$. We refer to (2.6) as a div-free RBF interpolant.

Instinctively, one may try to solve for the expansion coefficients in (2.6) by imposing $\mathbf{s}(\mathbf{x}_j) = \mathbf{u}_j$, $j = 1, \ldots, N$, where $\mathbf{u}_j = \mathbf{u}(\mathbf{x}_j)$. However, this will lead to a singular system of equations since each $\mathbf{u}_j$ can be expressed using only two degrees of freedom rather than three. To remedy this, let $\{\mathbf{d}_j, \mathbf{e}_j, \mathbf{n}_j\}$ be orthonormal vectors at the node $\mathbf{x}_j$, where $\mathbf{n}_j$ is the outward normal to $\mathcal{P}$, $\mathbf{e}_j$ is a unit tangent vector to $\mathcal{P}$, and $\mathbf{d}_j = \mathbf{n}_j \times \mathbf{e}_j$. Since $\mathbf{u}_j$ is tangent to $\mathcal{P}$ we can write it in this basis as $\mathbf{u}_j = \gamma_j \mathbf{d}_j + \delta_j \mathbf{e}_j$, where $\gamma_j = \mathbf{d}_j^T \mathbf{u}_j$ and $\delta_j = \mathbf{e}_j^T \mathbf{u}_j$. We may also express each tangent $\mathbf{c}_j$ as $\mathbf{c}_j = \alpha_j \mathbf{d}_j + \beta_j \mathbf{e}_j$, which leads us to express (2.6) as

$$\mathbf{s}(\mathbf{x}) = \sum_{j=1}^{N} \Phi_{\text{div}}(\mathbf{x}, \mathbf{x}_j) \left[\alpha_j \mathbf{d}_j + \beta_j \mathbf{e}_j\right], \tag{2.7}$$

and to write the interpolation conditions as $\mathbf{d}_i^T \mathbf{s}(\mathbf{x}_i) = \gamma_i$ and $\mathbf{e}_i^T \mathbf{s}(\mathbf{x}_i) = \delta_i$. This leads to the $2N$-by-$2N$ system of equations

$$\sum_{j=1}^{N} \underbrace{\left(\begin{bmatrix} \mathbf{d}_i^T \\ \mathbf{e}_i^T \end{bmatrix} \Phi_{\text{div}}(\mathbf{x}_i, \mathbf{x}_j) \begin{bmatrix} \mathbf{d}_j & \mathbf{e}_j \end{bmatrix} \right)}_{A^{(i,j)}} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} = \begin{bmatrix} \gamma_i \\ \delta_i \end{bmatrix}, \quad 1 \le i \le N. \tag{2.8}$$

The interpolation matrix that arises from this system (with its $(i,j)^{\text{th}}$ 2-by-2 block given by $A^{(i,j)}$) is positive definite if Φ_{div} is constructed from an appropriately chosen scalar-valued RBF (e.g., a positive definite ϕ) [37].

When $\mathcal{P} = \mathbb{R}^2$, the div-free RBF interpolant can be simplified considerably since in this case we can choose $\mathbf{d}_j = (1, 0, 0)$ and $\mathbf{e}_j = (0, 1, 0)$ and use (2.3) for defining Φ_{div}. Using this in (2.7) and truncating the unnecessary third component of the vector interpolant (since it is always zero) gives the expansion

$$\tilde{\mathbf{s}}(\mathbf{x}) = \sum_{j=1}^{N} \widetilde{\Phi}_{\text{div}}(\mathbf{x}, \mathbf{x}_j) \tilde{\mathbf{c}}_j, \tag{2.9}$$

where $\tilde{\mathbf{s}}, \tilde{\mathbf{c}}_j \in \mathbb{R}^2$, and

$$\widetilde{\Phi}_{\text{div}}(\mathbf{x}, \mathbf{x}_j) = \begin{bmatrix} -\partial_{yy} & \partial_{xy} \\ \partial_{xy} & -\partial_{xx} \end{bmatrix} \phi(\|\mathbf{x} - \mathbf{x}_j\|).$$

This expression for $\widetilde{\Phi}_{\text{div}}$ can be written as $\widetilde{\Phi}_{\text{div}} = -I\Delta\phi + \nabla\nabla^T\phi$, which is the standard way to express div-free kernels for general $\mathbb{R}^d$ [23].

An important consequence from the construction of the div-free RBF interpolant (2.6) is that we can extract a scalar potential ψ for the interpolated field. Using (2.5) for Φ_{div} in (2.6) we have

$$\mathbf{s}(\mathbf{x}) = \sum_{j=1}^{N} \Phi_{\text{div}}(\mathbf{x}, \mathbf{x}_j) \mathbf{c}_j = \underbrace{Q_{\mathbf{x}} \nabla}_{\mathbf{L}} \underbrace{\left(\sum_{j=1}^{N} \nabla^T \phi\left(\|\mathbf{x} - \mathbf{x}_j\|\right) Q_{\mathbf{x}_j} \mathbf{c}_j \right)}_{\psi(\mathbf{x})} = \mathbf{L}(\psi(\mathbf{x})). \tag{2.10}$$

This potential will play a crucial role in developing the PUM in Section 3.

2.3. Curl-free RBF interpolation. Curl-free vector RBF interpolants are constructed in a similar fashion to the div-free ones, the only difference being that $\mathbf{G}$ is applied instead of $\mathbf{L}$ in the construction of the matrix kernel. Given a scalar RBF ϕ and using a derivation similar to (2.4), Φ_{curl} is given as

$$\Phi_{\text{curl}}(\mathbf{x},\mathbf{y}) = \mathbf{G}_{\mathbf{x}}\mathbf{G}_{\mathbf{y}}^T\phi\left(\|\mathbf{x}-\mathbf{y}\|\right) = -P_{\mathbf{x}}\left(\nabla\nabla^T\phi\left(\|\mathbf{x}-\mathbf{y}\|\right)\right)P_{\mathbf{y}}, \tag{2.11}$$

where we have used the fact that the $P_{\mathbf{x}}$ matrix in (2.2) is symmetric. For any $\mathbf{c}\in\mathbb{R}^3$ and fixed $\mathbf{y}\in\mathcal{P}$, the vector field $\Phi_{\text{curl}}(\mathbf{x},\mathbf{y})\mathbf{c}$ is tangential to $\mathcal{P}$ and curl-free in $\mathbf{x}$. This follows from Proposition 2.1 since

$$\Phi_{\text{curl}}(\mathbf{x},\mathbf{y})\mathbf{c} = P_{\mathbf{x}}\nabla\left(-\nabla^T\phi\left(\|\mathbf{x}-\mathbf{y}\|\right)P_{\mathbf{y}}\mathbf{c}\right) = \mathbf{G}(\varphi(\mathbf{x})), \tag{2.12}$$

where φ is the potential for $\Phi_{\text{curl}}(\mathbf{x},\mathbf{y})\mathbf{c}$. As with the div-free kernel (2.5), the second argument of Φ_{curl} acts as a shift of the kernel and indicates where the field $\Phi_{\text{curl}}\mathbf{c}$ is "centered".

Interpolants to a curl-free tangential vector field $\mathbf{u}:\mathcal{P}\longrightarrow\mathbb{R}^3$ sampled at distinct points $X=\{\mathbf{x}_j\}_{j=1}^N\subset\mathcal{P}$ are constructed from Φ_{curl} as

$$\mathbf{s}(\mathbf{x}) = \sum_{j=1}^{N}\Phi_{\text{curl}}(\mathbf{x},\mathbf{x}_j)\mathbf{c}_j, \tag{2.13}$$

where the coefficients $\mathbf{c}_j\in\mathbb{R}^3$ are tangent to $\mathcal{P}$ at $\mathbf{x}_j$ and are chosen so that $\mathbf{s}\big|_X = \mathbf{u}\big|_X$. The procedure for determining these coefficients is identical to the div-free case, one just needs to replace Φ_{div} with Φ_{curl} in (2.7) & (2.8). The matrix from the linear system (2.8) with Φ_{curl} is similarly positive definite for the same ϕ. Further, a scalar potential φ can also be extracted from the curl-free field (2.13) using (2.12):

$$\mathbf{s}(\mathbf{x}) = \underbrace{P_{\mathbf{x}}\nabla}_{\mathbf{G}}\underbrace{\left(-\sum_{j=1}^{N}\nabla^T\phi\left(\|\mathbf{x}-\mathbf{x}_j\|\right)P_{\mathbf{x}_j}\mathbf{c}_j\right)}_{\varphi(\mathbf{x})} = \mathbf{G}(\varphi(\mathbf{x})). \tag{2.14}$$

In the Euclidean case $\mathbb{R}^d$, the curl-free kernel is simply given as $\Phi_{\text{curl}}(\mathbf{x},\mathbf{y}) = -\nabla\nabla^T\phi(\|\mathbf{x}-\mathbf{y}\|)$ [23], where ∇ is the d-dimensional gradient. The interpolation conditions $\mathbf{s}\big|_X = \mathbf{u}\big|_X$ also lead to the simplified linear system for the expansion coefficients $\mathbf{c}_j\in\mathbb{R}^d$:

$$\sum_{j=1}^{N}\Phi_{\text{curl}}(\mathbf{x}_i,\mathbf{x}_j)\mathbf{c}_j = \mathbf{u}_i,\ i=1,2,\ldots,N, \tag{2.15}$$

which is dN-by-dN. A scalar potential φ for the vector interpolant can be extracted as

$$\mathbf{s}(\mathbf{x}) = \nabla\underbrace{\left(-\sum_{j=1}^{N}\nabla^T\phi\left(\|\mathbf{x}-\mathbf{x}_j\|\right)\mathbf{c}_j\right)}_{\varphi(\mathbf{x})}. \tag{2.16}$$

3. A div-free/curl-free partition of unity method. The cost associated with solving the linear systems (2.8) and (2.15) is $\mathcal{O}(N^3)$, which is prohibitively high when the number of nodes N in X is large. In this section, we develop a partition of unity method (PUM) that requires solving several linear systems associated with subsets X_ℓ of X with $n_\ell << N$ nodes, which reduces the computational cost significantly regardless of the nature of the RBF used.

3.1. Partition of unity methods. Let $\Omega \subset \mathbb{R}^d$ be an open, bounded domain of interest for approximating some function $f : \Omega \longrightarrow \mathbb{R}$. Let $\Omega_1, \ldots, \Omega_M$ be a collection of distinct overlapping patches that form an open cover of Ω, i.e., $\cup_{\ell=1}^{M} \Omega_\ell \supseteq \Omega$, and let the overlap between patches be limited such that at most $K << M$ patches overlap at any given point $\mathbf{x} \in \Omega$. For each $\ell = 1, \ldots, M$, let $w_\ell : \Omega_\ell \longrightarrow [0, 1]$ be a weight function such that w_ℓ is compactly supported on Ω_ℓ and the set of weight functions $\{w_\ell\}$ have the property that $\sum_{\ell=1}^{M} w_\ell \equiv 1$. Suppose s_ℓ is some approximation to f on each patch Ω_ℓ. The partition of unity approach of Babuška and Melenk [3] is to form an approximant s to f over the whole domain Ω by "blending" the local approximants s_ℓ with w_ℓ via $s = \sum_{\ell=1}^{M} w_\ell s_\ell$.

When samples of f are given at N "scattered" nodes $X = \{\mathbf{x}_j\}_{j=1}^{N} \subset \Omega$, RBF interpolants are a natural choice for the local approximants s_ℓ, as pointed out in [3]. RBF-PUM was first explored for interpolation in 2002 by Wendland [46] and Lazzaro and Montefusco [33], and then later in 2007 by Fasshauer [18, Ch. 29]. More recent work has explored various aspects of the method in terms of applications, methods, and implementations, especially by Cavoretto, De Rossi, and colleagues (e.g., [7–9]), and also extensions to problems on the sphere [6, 42]. Additionally, the method has been adapted for approximating the solution of partial differential equations (e.g., [1, 32, 40, 44]).

Common choices for the patches in RBF-PUM are disks for problems in $\mathbb{R}^2$, spherical caps for problems on $\mathbb{S}^2$, and balls for problems in $\mathbb{R}^3$, and these are the choices we use throughout this paper. Figure 1 gives an example of a set of patches for a problem in $\mathbb{R}^2$. Techniques for choosing the patches are discussed in, e.g., [9, 32, 42] (see Section 3.3 for more discussion). Other choices for patches commonly used in PUM methods are rectangles and procedures for generating these can be found, for example, in [27].

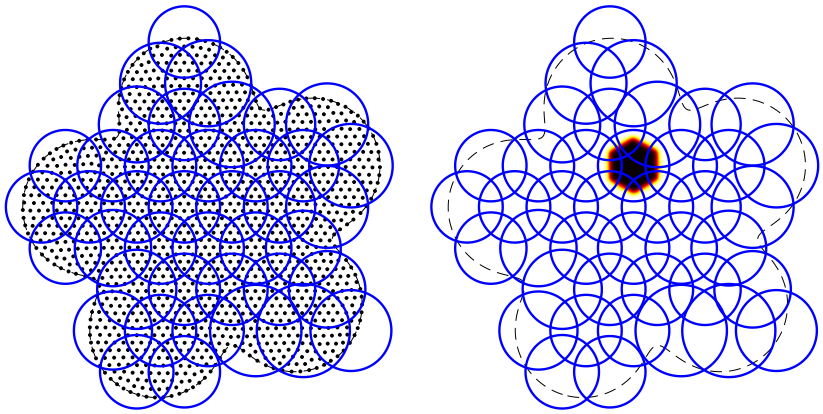
Based on the choices of patches, the weight functions w_ℓ can be constructed using Shepard's method as follows. Let $\kappa : \mathbb{R}^+ \to \mathbb{R}$ have compact support over the interval $[0, 1)$. For each patch Ω_ℓ, let $\boldsymbol{\xi}_\ell$ denote its center and ρ_ℓ denote its radius, and define $\kappa_\ell(\mathbf{x}) := \kappa\left(\|\mathbf{x} - \boldsymbol{\xi}_\ell\|/\rho_\ell\right)$. The weight functions are then given by

$$w_\ell(\mathbf{x}) = \kappa_\ell(\mathbf{x}) / \sum_{j=1}^{M} \kappa_j(\mathbf{x}), \ \ell = 1, \ldots, M.$$

Note that each w_ℓ is only supported over Ω_ℓ and that the summation on the bottom only involves terms that are non-zero over patch Ω_ℓ, which is bounded by K. Figure 1 (b) illustrates one of these weights functions for the example domain in part (a), where κ is chosen as the C^1 quadratic B-spline

$$\kappa(r) = \begin{cases} 1 - 3r^2, & 0 \le r \le \frac{1}{3}, \\ \frac{3}{2}(1 - r)^2, & \frac{1}{3} \le r \le 1. \end{cases} \tag{3.1}$$

This is the weight function we use throughout the paper.



(a) (b)

FIGURE 1. *(a) Illustration of partition of unity patches (outlined in blue lines) for a node set X (marked with black disks) contained in a domain Ω (marked with a dashed line). (b) Illustration of one of the PU weight functions for the patches from part (a), where the color transition from white to yellow to red to black correspond to weight function values from* 0 *to* 1.

3.2. Description of the method. A first approach at a vector RBF-PUM may be to construct local vector approximants $\mathbf{s}_\ell$ for the patches Ω_ℓ that make up the PU using either (2.6) for div-free fields or (2.13) for curl-free fields. These approximants can then be "blended" into a global approximant for the underlying field:

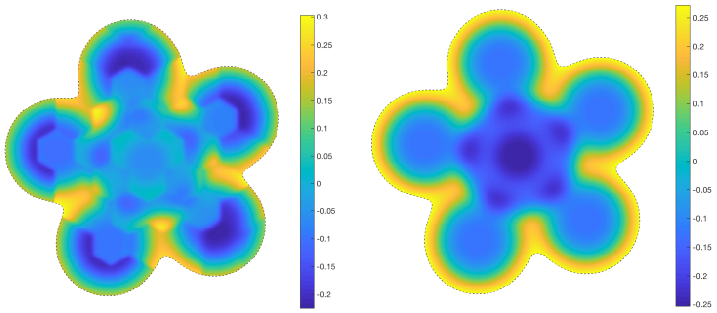
$$\mathbf{s} = \sum_{\ell=1}^{M} w_\ell \mathbf{s}_\ell. \tag{3.2}$$

The issue with this approach is that $\mathbf{s}$ will not necessarily inherit the div-free or curl-free properties of $\mathbf{s}_\ell$ because of the multiplication by the weight functions w_ℓ. We instead use the local scalar potentials that are recovered from each $\mathbf{s}_\ell$ and then blend those together. A div-free or curl-free approximant can then be recovered by applying the appropriate differential operator to the blended potentials. Since the essential ingredients are very similar for all the kernels treated from Section 2, for brevity we describe the method only for the div-free case in $\mathbb{R}^2$ and mention any relevant differences as needed.

Let X_ℓ denote the nodes from $X \subset \mathbb{R}^2$ that belong to patch Ω_ℓ, and let $\mathbf{s}_\ell$ denote the div-free RBF interpolant (2.6) to the target div-free field $\mathbf{u}$ over X_ℓ. Our interest is also in the scalar potential for each interpolant given in (2.10), which we denote as ψ_ℓ. While we could try to construct a global PU approximant for the scalar potential of the field ψ and then apply the operator $\mathbf{L}$ to the result, we would immediately run into problems since the scalar potentials are only unique up to a constant. This means that for two patches Ω_ℓ and Ω_k that overlap, ψ_ℓ and ψ_k could be off up to the addition of a constant in the overlap region and thus lead to an inaccurate PU approximant. To rectify this situation, we need to "shift" each ψ_ℓ by a constant b_ℓ such that $\psi_\ell + b_\ell \approx \psi_k + b_k$ if Ω_ℓ and Ω_k overlap.

To summarize, the main steps of the div-free PUM are as follows:

1. On each patch Ω_ℓ, compute a divergence free interpolant $\mathbf{x}_\ell$ and extract its scalar potential ψ_ℓ using (2.10).
2. Determine constants $\{b_\ell\}_{\ell=1}^{M}$ such that $\widetilde{\psi}_\ell := \psi_\ell + b_\ell \approx \psi_k + b_k =: \widetilde{\psi}_k$ whenever



(a) (b)

FIGURE 2. *Div-free RBF partition of unity approximant of the potential from Section 5.1 (a) without the patch potentials shifted (ψ_k) (b) with the patch potentials shifted ($\widetilde{\psi}_k$).*

$\Omega_\ell \cap \Omega_k \neq \emptyset$.

3. Blend the shifted potentials with the PU weight functions to obtain a global approximant for the underlying potential:

$$\widetilde{\psi}(\mathbf{x}) := \sum_{\ell=1}^{M} w_\ell(\mathbf{x})\widetilde{\psi}_\ell(\mathbf{x}). \tag{3.3}$$

4. Apply $\mathbf{L}$ to $\widetilde{\psi}$ to obtain a global div-free approximant to the underlying field:

$$\widetilde{\mathbf{s}}(\mathbf{x}) := \sum_{\ell=1}^{M} \mathbf{L}\left(w_\ell(\mathbf{x})\widetilde{\psi}_\ell(\mathbf{x})\right) = \sum_{\ell=1}^{M} w_\ell(\mathbf{x})\mathbf{s}_\ell(\mathbf{x}) + \sum_{\ell=1}^{M} \widetilde{\psi}_\ell(\mathbf{x})\mathbf{L}(w_\ell(\mathbf{x})). \tag{3.4}$$

Note that the second term in the last equality acts as a correction to the PU approximant formed by blending just the div-free RBF interpolants. Figure 2 illustrates the necessity of shifting the patch potentials by way of an example from Section 5.1. The figure shows a div-free RBF-PU approximant of a potential when the local patch potentials are not shifted (i.e., using ψ_ℓ in (3.3) rather than $\widetilde{\psi}_\ell$) and when they are shifted.

We now turn our attention to a technique for determining the constants $\{b_\ell\}_{\ell=1}^{M}$ for shifting the potential. The idea is to pick a point in the overlap region of each pair of overlapping patches and enforce that the potentials for the each of these patches are equal at this point. We refer to these points as the "glue points" since they are where the potentials between neighboring patches are "glued" to one another. We have found the following procedure for choosing these points to be effective. If Ω_ℓ and Ω_k overlap, then let $\bar{\mathbf{x}}_\ell^k$ denote the center of the overlap region: $\bar{\mathbf{x}}_\ell^k := (\rho_k\boldsymbol{\xi}_\ell + \rho_\ell\boldsymbol{\xi}_k)/(\rho_k+\rho_\ell)$, where $\ell < k$ to avoid redundancy; see Figure 3 for an illustration. We denote the collection of all such points by $\bar{X} := \{\bar{\mathbf{x}}_\ell^k \,|\, \Omega_\ell \cap \Omega_k \neq \emptyset,\ \ell < k\} = \{\bar{\mathbf{x}}_i\}_{i=1}^{L}$, where $L = |\bar{X}|$ and we have reindexed the set so that each $\bar{\mathbf{x}}_i = \bar{\mathbf{x}}_\ell^k$ for some unique overlapping pair of patches Ω_ℓ and Ω_k.

On this set we want to impose the conditions

$$\psi_\ell(\bar{\mathbf{x}}_\ell^k) + b_\ell = \psi_k(\bar{\mathbf{x}}_\ell^k) + b_k$$

for some constants b_ℓ, $\ell = 1, \ldots, M$, which we refer to as the "potential shifts". This can be arranged into a sparse L-by-M over-determined linear system

$$Pb = c \tag{3.5}$$

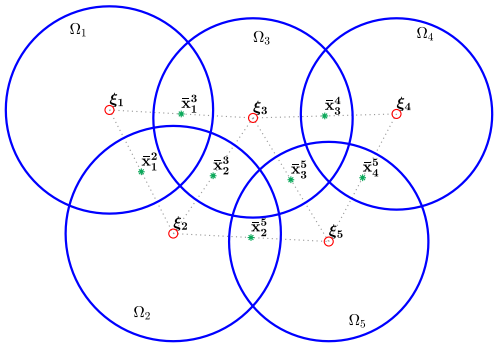


FIGURE 3. *Illustration of the glue points for shifting the potentials. The asterisks denote the glue points and the small circles denote the patch centers.*

with the following properties. The L-by-M matrix P is sparse with two non-zeros per row: the i^{th} row, where $\bar{\mathbf{x}}_i$ corresponds to $\bar{\mathbf{x}}_\ell^k$, has a 1 in the ℓ^{th} column and a -1 in the k^{th} column. The vector b contains the potential shifts, and the vector c is given by $c_i = \psi_k(\bar{\mathbf{x}}_i) - \psi_\ell(\bar{\mathbf{x}}_i) = \psi_k(\bar{\mathbf{x}}_\ell^k) - \psi_\ell(\bar{\mathbf{x}}_\ell^k)$. The matrix P also has rank $M-1$. This follows since P is the (oriented) incidence matrix for the graph with vertices being the patch centers Ω_ℓ and edges corresponding to non-empty intersections of patches. Based on the assumption that $\{\Omega_\ell\}_{\ell=1}^M$ is an overlapping open covering, this graph is connected, so $\operatorname{rank}(P) = M-1$ [12, Thm. 10.5]. In the next section we discuss the procedure we use to determine the potential shifts from (3.5).

Remark 3.1. The procedure described above works exactly the same for curl-free fields in $\mathbb{R}^2$ and $\mathbb{R}^3$ using (2.16) for the interpolants and potential fields on each patch. The procedure also extends to more general surfaces $\mathcal{P}$ for div-free fields (using (2.10)) and curl-free fields (using (2.14)). However, in this case determining the glue points using the above technique can be more difficult, but for $\mathcal{P} = \mathbb{S}^2$, this is easy since the center of the overlap region is trivial to determine.

3.3. Implementation details. We now discuss how the patches $\{\Omega_\ell\}_{\ell=1}^M$ are chosen as well as how one might compute the potential shifts from the system (3.5). In what follows, we assume that the nodes X are quasiuniformly distributed (i.e., have low discrepancy) in the underlying domain Ω, so that the mesh-norm for X,

$$h := \sup_{\mathbf{y}\in\Omega} \min_{\mathbf{x}\in X} \operatorname{dist}(\mathbf{x},\mathbf{y}), \tag{3.6}$$

satisfies $h = \mathcal{O}(1/\sqrt[d]{N})$, where d is the dimension of Ω. We also assume that there is a signed distance function for the domain to distinguish the interior from the exterior.

3.3.1. Patch centers. To determine the patches $\{\Omega_\ell\}$ for domains in $\mathbb{R}^2$ and $\mathbb{R}^3$, we use an approach similar to the one described in [32]. The idea is to start with a regular grid structure of spacing H that covers the domain Ω of interest and then remove the grid points that are not contained in the domain. The remaining grid points are chosen as the patch centers $\{\boldsymbol{\xi}_\ell\}_{\ell=1}^M$. Next, an initial radius ρ is chosen proportional to H so the patches $\{\Omega_\ell\}_{\ell=1}^M$ form an open cover and there is sufficient overlap between patches (specifics on this are given below). Finally, for any node in X that is not contained in one of the patches, the nearest patch center $\boldsymbol{\xi}_j$ is determined and the radius ρ_j for that patch is enlarged to enclose the node. We perform all range queries on patch centers using a k-d tree.

For domains in $\mathbb{R}^2$, we choose the initial grid structure for the patch centers as regular hexagonal lattice of spacing H. Neighboring patches will not overlap if the initial radius is less than or equal to $H/2$. Therefore, to guarantee overlap, we set the initial radii for the patches to $\rho = (1+\delta)H/2$, where $\delta > 0$. See Figure 1 for an illustration of the patches chosen using this algorithm for $\delta = 1/2$. For domains in $\mathbb{R}^3$, we choose the initial grid structure for the patch centers as a regular Cartesian lattice of spacing H. In this case, neighboring patches along the longest diagonal directions will not overlap if the initial radius is less than or equal to $\sqrt{3}H/2$. To guarantee overlap, we thus set the initial radii for the patches to $\rho = (1+\delta)\sqrt{3}H/2$.

To determine the patches for $\mathbb{S}^2$, we use an approach similar to the one described in [42]. The idea is to use M quasi-uniformly spaced points on $\mathbb{S}^2$ for the set of patch centers. We choose these as near minimum energy (ME) point sets [30], and use the pre-computed near ones from [49]. For a set with M points, the average spacing H between the points can be estimated as $H \approx \sqrt{4\pi/M}$. We select a value of H and then determine M as $M = \lceil 4\pi/H^2 \rceil$. Since the ME points are typically arranged in hexagonal patterns (with a few exceptions [30]), we choose the radius for each patch as $\rho_\ell = (1+\delta)H/2$, where the parameter δ again determines the overlap.

To keep the overall cost under control, the initial radii of the patches H should decrease as N increases. The rate at which H should decrease can be determined as follows. Assuming that the patches that intersect the boundary have similar radii to the interior patches, and using the assumption that X is quasiuniform, a simple volume argument gives that number of nodes in each patch satisfies $n = \mathcal{O}(\rho^d N) = \mathcal{O}(H^d N)$, where d is the dimension of Ω. So, to keep the work roughly constant per patch, we need $H = \mathcal{O}(1/N^{1/d})$. In our implementation of the vector PUM, we choose

$$H = q\,(A/N)^{1/d}, \tag{3.7}$$

where A is related to the area/volume of Ω, and q is a parameter that controls the average number of nodes per patch. Note that from the above analysis, the computational cost increases as the overlap parameter increases and as q increases. Based on the assumptions on X and the patches, choosing H according to (3.7) results in a computational cost of $\mathcal{O}(N)$ for constructing the vector PUM approximants, and $\mathcal{O}(N \log N)$ for the range queries involved for determining the patch structure. However, in practice, the cost is dominated by the former part of the method.

3.3.2. Potential shifts. Since $\operatorname{rank}(P) = M - 1$ and its nullspace consists of constant vectors, we first set one of the shifts b_j to zero, for some $1 \le j \le M$, and then compute the remaining shifts using the least squares solution of (3.5). For this problem we can form the normal equations directly since the matrix $P^T P$ is just a graph Laplacian (recall P is an oriented incidence matrix). We have found that the accuracy of the reconstructed field (3.4) can often be improved if a weighted least squares approach is used. In this case, we use a diagonal weight matrix W with entries that depend on the distance between the glue points and the patch centers. Specifically, we set r_i as the closer of the two distances between the i^{th} glue point $\bar{\mathbf{x}}_i$ and the centers of the two patches it was formed from, and then set

$$W_{ii} = \exp\left(-\gamma\left(1 - \frac{r_i}{r_{\min}}\right)^2\right), \tag{3.8}$$

where $r_{\min} = \min_j r_j$ and $\gamma > 0$. The normal equations in this case now look like a weighted graph Laplacian.

4. Error Estimates. The error bounds will be expressed in terms of local mesh norms h_ℓ, which are given by (3.6), with $\Omega = \Omega_\ell$ and $X = X_\ell$. Error rates for RBF interpolation, including divergence-free (curl-free) RBF approximation, both in flat space and on the sphere, have been known for some time. Many of these estimates are valid for target functions within the *native space*, which we denote by $\mathcal{N}(\Omega)$, of the RBF used - which for infinitely smooth RBFs are subspaces of analytic functions and for kernels of finite smoothness are essentially Sobolev spaces (with norms equivalent to Sobolev norms on bounded subsets)[2]. For the RBF kernels considered here, there is a continuous embedding from the native space of the matrix kernel into a Sobolev space of order $\tau > d/2$. In this situation we get the estimate below. In what follows, we let $\mathbf{H}^\tau(\Omega_\ell)$ denote the space of vector fields with each coordinate function in the Sobolev space $H^\tau(\Omega)$ with smoothness τ.

PROPOSITION 4.1. *Suppose that $\mathbf{u} \in \mathcal{N}(\Omega)$ and that each $\Omega_\ell \subset \Omega$ satisfies an interior cone condition with radius R_ℓ and angle θ independent of ℓ. Suppose also that there is a continous embedding of the native space into $\mathbf{H}^\tau(\Omega)$. Then there are constants $Q := Q(\theta, \tau)$ and $C := C(\theta, \tau, d)$ such that if $h_\ell < QR_\ell$, then*

$$\|\mathbf{u} - \mathbf{s}_\ell\|_{L_\infty(\Omega_\ell)} \leq \mathcal{E}(h_\ell)\|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)},$$

where $\mathcal{E}(h) = Ch^{\tau - d/2}$.

Proof. Estimates like these have been worked out for div/curl-free RBFs on subsets of $\mathbb{R}^d$ and on $\mathbb{S}^2$ [23, 24, 26]. However, in the papers referenced the domain was fixed and the dependence of the constants on the cone condition radius was not emphasized, so we should briefly review the arguments here.

First, note that the function $\mathbf{u} - \mathbf{s}_\ell$ will be zero on X_ℓ. On domains satifying an interior cone condition, in the Euclidean case and on surfaces, we may therefore employ a "zeros lemma" in each coordinate function. These give constants $Q := Q(\theta, \tau)$ and $C := C(\theta, \tau, d)$ such that if $h_\ell < QR_\ell$, then

$$\|\mathbf{u} - \mathbf{s}_\ell\|_{L_\infty(\Omega_\ell)} \leq Ch_\ell^{\tau - d/2}\|\mathbf{u} - \mathbf{s}_\ell\|_{\mathbf{H}^\tau(\Omega_\ell)}.$$

See for example [47, Theorem 11.32] and [29, Theorems A.4 and A.11]).

Next, since $\mathbf{u} \in \mathcal{N}(\Omega)$, then $\mathbf{u} \in \mathcal{N}(\Omega_\ell)$ and there is an isometric extension $E : \mathcal{N}(\Omega_\ell) \to \mathcal{N}(\Omega)$ such that $\|E\mathbf{u}\|_{\mathcal{N}(\Omega)} = \|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)}$ (see [47, Theorem 10.46,10.47][3]). With this, since $\mathcal{N}(\Omega)$ is continuously embedded in $\mathbf{H}^\tau(\Omega)$ for some $\tau > d/2$, we get

$$\|\mathbf{u} - \mathbf{s}_\ell\|_{\mathbf{H}^\tau(\Omega_\ell)} = \|E\mathbf{u} - \mathbf{s}_{E\mathbf{u},\ell}\|_{\mathbf{H}^\tau(\Omega_\ell)} \leq \|E\mathbf{u} - \mathbf{s}_{E\mathbf{u},\ell}\|_{\mathbf{H}^\tau(\Omega)} \leq C\|E\mathbf{u} - \mathbf{s}_{E\mathbf{u},\ell}\|_{\mathcal{N}(\Omega)},$$

where we write $\mathbf{s}_{E\mathbf{u},\ell} = \mathbf{s}_\ell$ to emphasize that the interpolant on X_ℓ of the extension is also $\mathbf{s}_\ell$. Note that the constant here may depend on Ω, but not on Ω_ℓ. Finally, it is well-known that the interpolation error is always orthogonal to the kernel interpolant in the native space, which implies the bound

$$\|E\mathbf{u} - \mathbf{s}_{E\mathbf{u},\ell}\|_{\mathcal{N}(\Omega)} \leq \|E\mathbf{u}\|_{\mathcal{N}(\Omega)} = \|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)},$$

where the last equality follows because E is an isometry. This completes the proof. ∎

[2] See [47, Ch. 10] for native spaces of scalar valued functions, and see [22, 24] for the vector cases on $\mathbb{R}^d$ and the sphere.

[3] The theorems referenced are given in the Euclidean scalar-valued context, but the arguments are general enouch to apply to matrix valued positive definite kernels on any set.

Thus it is possible to acheive high order convergence with patch sizes that are proportional to the mesh norm. In what follows we assume that the patch radii and local mesh norms are such that Proposition 4.1 is satisfied.

In addition to the estimate above, our arguments that follow will also rely on the Mean Value Theorem, which for a scalar function $\psi\colon \mathbb{R}^d \to \mathbb{R}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ we express as

$$|\psi(\mathbf{x}) - \psi(\mathbf{y})| \leq |\nabla(\psi)|_{\mathbf{x}^*}| \ \mathrm{dist}(\mathbf{x}, \mathbf{y}),$$

where $\mathbf{x}^*$ is on the line segment between $\mathbf{x}$ and $\mathbf{y}$. Here we use the notation $|\cdot|$ to denote the Euclidean length when the argument is a vector. To derive a similar estimate on surfaces, let $\mathbf{x}, \mathbf{y} \in \mathcal{P}$ and let $\gamma : [0, \mathrm{dist}_{\mathcal{P}}(\mathbf{x}, \mathbf{y})] \to \mathcal{P}$ denote a shortest path in $\mathcal{P}$ connecting $\mathbf{x}$ and $\mathbf{y}$ with $\gamma(0) = \mathbf{x}$, $\gamma(\mathrm{dist}_{\mathcal{P}}(\mathbf{x}, \mathbf{y})) = \mathbf{y}$, parameterized by arclength. This implies that γ' is tangent to $\mathcal{P}$ and $|\gamma'| = 1$. Applying the single variable Mean Value Theorem to the real-valued function $\psi \circ \gamma$ implies that

$$|\psi(\mathbf{x}) - \psi(\mathbf{y})| \leq |\nabla\psi \cdot \gamma'|_{t^*}|\mathrm{dist}_{\mathcal{P}}(\mathbf{x}, \mathbf{y}),$$

where $t^* \in [0, \mathrm{dist}_{\mathcal{P}}(\mathbf{x}, \mathbf{y})]$. Since γ' is tangent to $\mathcal{P}$ and has length 1, we get $|\nabla\psi \cdot \gamma'| = |\mathbf{G}\psi \cdot \gamma'| \leq |\mathbf{G}\psi|$. Combining the above with the fact that $|\mathbf{G}(\psi)| = |\mathbf{L}(\psi)|$ gives us the following

$$|\psi(\mathbf{x}) - \psi(\mathbf{y})| \leq |\mathbf{G}(\psi)|_{\mathbf{x}^*}| \ \mathrm{dist}_{\mathcal{P}}(x, y) = |\mathbf{L}(\psi)|_{\mathbf{x}^*}| \ \mathrm{dist}_{\mathcal{P}}(\mathbf{x}, \mathbf{y}), \tag{4.1}$$

where $\mathbf{x}^* \in \mathcal{P}$.

Before proceeding we summarize some of the important assumptions on the partition of unity. Recall that each $\mathbf{x} \in \Omega$ is covered by only a small number of patches (say at most K patches). We also assume that the number of patches that intersect a given patch is uniformly bounded by some constant m. Additionally, we suppose that there are roughly the same number of nodes in each patch, and that the node distribution in each patch is quasi-uniform. This leads to an estimate of the form $ch_\ell \leq \mathrm{diam}(\Omega_\ell) \leq Ch_\ell$ for some constants c, C independent of ℓ. Lastly, we assume that the partition is "1-stable" (see [47][Def. 15.16]), meaning that first order derivatives of the weight functions satisfy a bound of the form $|\nabla w_\ell| \leq C(\mathrm{diam}(\Omega_\ell))^{-1}$, where C is some constant independent of ℓ. This with the quasi-uniformity supposition gives the bound $|\nabla w_\ell| = |\mathbf{L}w_\ell| \leq Ch_\ell^{-1}$ for some C independent of ℓ.

Now we give an estimate for the pointwise error of the divergence-free approximant in a two dimensional domain. Note that the bound is local in the sense that it comprised of a local interpolation error plus an expression involving the residuals $r_\ell^k := \widetilde{\psi}_\ell(\bar{\mathbf{x}}_\ell^k) - \widetilde{\psi}_k(\bar{\mathbf{x}}_\ell^k)$ from adjusting neighboring potential functions.

THEOREM 4.2. *Suppose that the conditions in Proposition 4.1 are satisfied. Given a div-free vector field $\mathbf{u} = \mathbf{L}(\psi) \in \mathcal{N}(\Omega)$, let $\widetilde{\psi}$ and $\widetilde{\mathbf{s}} = \mathbf{L}(\widetilde{\psi})$ denote the PUM approximants from (3.3) and (3.4). Then the error at $\mathbf{x} \in \Omega$ satisfies*

$$\begin{aligned} \left|\mathbf{G}(\widetilde{\psi} - \psi)(\mathbf{x})\right| = \left|\mathbf{L}(\widetilde{\psi} - \psi)(\mathbf{x})\right| &= |\mathbf{u}(\mathbf{x}) - \widetilde{\mathbf{s}}(\mathbf{x})| \\ &\leq mC \max_{\ell \,|\, \mathbf{x} \in \Omega_\ell} \left(\mathcal{E}(h_\ell)\|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)}\right) + C \sum_{\ell | \mathbf{x} \in \Omega_\ell,\, \ell \neq k} h_\ell^{-1} |r_\ell^k|, \end{aligned} \tag{4.2}$$

where k is any index such that $\mathbf{x} \in \Omega_k$.

Proof. The first equality follows from the fact that $\mathbf{G}f$ and $\mathbf{L}f$ have the same magnitude. Next, note that

$$\widetilde{\mathbf{s}} = \sum_\ell w_\ell \mathbf{s}_\ell + \sum_\ell \mathbf{L}(w_\ell)\widetilde{\psi}_\ell. \tag{4.3}$$

The first term is a weighted average of RBF interpolants to $\mathbf{u}$ and the weight functions sum to 1, so we have

$$\begin{aligned}\left|\mathbf{u}(\mathbf{x}) - \sum_\ell w_\ell(\mathbf{x})\mathbf{s}_\ell(\mathbf{x})\right| &= \left|\sum_\ell w_\ell(\mathbf{x})\mathbf{u}(\mathbf{x}) - \sum_\ell w_\ell(\mathbf{x})\mathbf{s}_\ell(\mathbf{x})\right| \leq \sum_\ell w_\ell(\mathbf{x})|\mathbf{u}(\mathbf{x}) - \mathbf{s}_\ell(\mathbf{x})| \\ &\leq \sum_\ell w_\ell(\mathbf{x}) C\mathcal{E}(h_\ell)\|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)} = C \max_{\ell \,|\, \mathbf{x}\in\Omega_\ell} \mathcal{E}(h_\ell)\|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)}.\end{aligned}$$

To complete the proof we need to bound the second term in (4.3). Given $\mathbf{x} \in \Omega$, fix a k such that $\mathbf{x} \in \Omega_k$. Since $\sum \mathbf{L}(w_\ell) = 0$ and $w_\ell(\mathbf{x}) = 0$ for $\mathbf{x} \notin \Omega_\ell$ we get

$$\sum_\ell \mathbf{L}(w_\ell)\widetilde{\psi}_\ell(\mathbf{x}) = \sum_{\ell \,|\, \mathbf{x}\in\Omega_\ell} \mathbf{L}(w_\ell)\left(\widetilde{\psi}_\ell(\mathbf{x}) - \widetilde{\psi}_k(\mathbf{x})\right).$$

This and our assumptions on the weight functions give us the estimate

$$\left|\sum_\ell \mathbf{L}(w_\ell)\widetilde{\psi}_\ell(\mathbf{x})\right| \leq \sum_{\ell \,|\, \mathbf{x}\in\Omega_\ell} Ch_\ell^{-1}\left|\widetilde{\psi}_\ell(\mathbf{x}) - \widetilde{\psi}_k(\mathbf{x})\right|. \tag{4.4}$$

If $\ell = k$, the corresponding term in the sum is zero. If $\ell \neq k$, we let $g := \widetilde{\psi}_\ell - \widetilde{\psi}_k$ and $\bar{\mathbf{x}}_\ell^k$ be the adjustment point for Ω_ℓ and Ω_k, we can rewrite

$$\widetilde{\psi}_\ell(\mathbf{x}) - \widetilde{\psi}_k(\mathbf{x}) = g(\mathbf{x}) - g(\bar{\mathbf{x}}_\ell^k) + g(\bar{\mathbf{x}}_\ell^k) = g(\mathbf{x}) - g(\bar{\mathbf{x}}_\ell^k) + r_\ell^k.$$

To bound $g(\mathbf{x}) - g(\bar{\mathbf{x}}_\ell^k)$, we use (4.1) and the fact that $\mathbf{L}(g) = \mathbf{s}_\ell - \mathbf{s}_k$ to get

$$\begin{aligned}|g(\mathbf{x}) - g(\bar{\mathbf{x}}_\ell^k)| &\leq \|\mathbf{L}(g)\|_{L_\infty(\Omega_k\cap\Omega_\ell)}\mathrm{dist}(\mathbf{x}, \bar{\mathbf{x}}_\ell^k) \leq \|\mathbf{L}(g)\|_{L_\infty(\Omega_k\cap\Omega_\ell)}h_\ell \\ &\leq h_\ell\left(\|\mathbf{s}_\ell - \mathbf{u}\|_{L_\infty(\Omega_k\cap\Omega_\ell)} + \|\mathbf{u} - \mathbf{s}_k\|_{L_\infty(\Omega_k\cap\Omega_\ell)})\right) \\ &\leq Ch_\ell\left(\mathcal{E}(h_\ell)\|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)} + \mathcal{E}(h_k)\|\mathbf{u}\|_{\mathcal{N}(\Omega_k)}\right),\end{aligned}$$

which when applied to (4.4) gives

$$\begin{aligned}\left|\sum_\ell \mathbf{L}(w_\ell)\widetilde{\psi}_\ell(\mathbf{x})\right| &\leq \sum_{\ell|\mathbf{x}\in\Omega_\ell,\,\ell\neq k} C\left(\mathcal{E}(h_\ell)\|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)} + \mathcal{E}(h_k)\|\mathbf{u}\|_{\mathcal{N}(\Omega_k)}\right) + Ch_\ell^{-1}|r_\ell^k| \\ &\leq mC \max_{\ell \,|\, \mathbf{x}\in\Omega_\ell} \mathcal{E}(h_\ell)\|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)} + C\sum_{\ell|x\in\Omega_\ell,\,\ell\neq k} h_\ell^{-1}|r_\ell^k|.\end{aligned}$$

The result follows. ∎

Note that very similar arguments follow through also for curl-free vector fields on surfaces, i.e. an estimate identical to (4.2) holds for the curl-free case. The proof also carries directly over to $\mathbb{R}^d$ - namely if $\mathbf{u} = \nabla\varphi$, and $\widetilde{\mathbf{s}} = \nabla\widetilde{\varphi}$ denotes the curl-free RBF-PUM approximant, one has an estimate of the form

$$|\nabla(\widetilde{\varphi} - \varphi)(\mathbf{x})| = |\mathbf{u}(\mathbf{x}) - \widetilde{\mathbf{s}}(\mathbf{x})| \leq mC \max_{\ell \,|\, \mathbf{x}\in\Omega_\ell}\left(\mathcal{E}(h_\ell)\|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)}\right) + C\sum_{\ell|\mathbf{x}\in\Omega_\ell,\,\ell\neq k} h_\ell^{-1}|r_\ell^k|.$$

Now we discuss the residual in shifting the local potentials. We begin by showing that good constants for the shifts exist.

PROPOSITION 4.3. *Let $\mathbf{s}_\ell = \mathbf{L}\psi_\ell$ be the local RBF interpolant on $X_\ell \subset \Omega_\ell$ and let $\bar{X}_\ell = \bar{X} \cap \Omega_\ell$ be the collection of glue points on Ω_ℓ. Given any v such that $\mathbf{u} = \mathbf{L}(v)$, the constant*

$$b_\ell^* := \frac{1}{|\bar{X}_\ell|} \sum_{\mathbf{y} \in \bar{X}_\ell} v(\mathbf{y}) - \psi_\ell(\mathbf{y})$$

gives

$$\|\psi_\ell + b_\ell^* - v\|_{L_\infty(\Omega_\ell)} \leq C h_\ell \mathcal{E}(h_\ell) \|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)}.$$

Proof. Let $\mathbf{x} \in \Omega_\ell$. First we apply the triangle inequality and the Mean Value Theorem to obtain

$$\begin{aligned} |\psi_\ell(\mathbf{x}) + b_\ell^* - v(\mathbf{x})| &\leq \frac{1}{|\bar{X}_\ell|} \sum_{\mathbf{y} \in \bar{X}_\ell} |\psi_\ell(\mathbf{x}) - v(\mathbf{x}) - (\psi_\ell(\mathbf{y}) - v(\mathbf{y}))| \\ &\leq \frac{1}{|\bar{X}_\ell|} \sum_{\mathbf{y} \in \bar{X}_\ell} \|\mathbf{s}_j - \mathbf{u}\|_{L_\infty(\Omega_\ell)} \operatorname{dist}(\mathbf{x}, \mathbf{y}). \end{aligned}$$

Next, an application of Proposition 4.1 and the fact that $\operatorname{diam}(\Omega_\ell) \leq C h_\ell$ finishes the proof. ∎

Letting $r^* := Pb^* - c$, i.e., the residual in the system (3.5) using the shifts given in the above proposition, with a triangle inequality and using the fact that $h_k \sim h_\ell$ for neighboring patches, we get

$$(r^*)_\ell^k \leq C h_\ell \mathcal{E}(h_\ell) \|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)} + C h_\ell \mathcal{E}(h_k) \|\mathbf{u}\|_{\mathcal{N}(\Omega_k)}. \tag{4.5}$$

Applying this to the residual term from (4.2) becomes:

$$\sum_{\ell | \mathbf{x} \in \Omega_\ell,\, \ell \neq k} h_\ell^{-1} (r^*)_\ell^k \leq mC \max_{\ell \,|\, \mathbf{x} \in \Omega_\ell} \mathcal{E}(h_\ell) \|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)} \tag{4.6}$$

Thus if the shifts are chosen appropriately the method can achieve the same approximation order as that of local interpolation. However, we compute the shifts according to the overdetermined (3.5). The residual from that system satisfies the following.

PROPOSITION 4.4. *Let b be the least squares solution to (3.5). The residual $r := Pb - c$ satisfies the bound*

$$|r|^2 \leq m\, C \sum_\ell h_\ell^2 \mathcal{E}(h_\ell)^2 \|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)}^2.$$

Proof. Choose any scalar potential v such that $\mathbf{u} = \mathbf{L}(v)$, and let b^* be the vector whose ℓ^{th} element is b_ℓ^* as defined in Proposition 4.3. Then we have $|r| \leq |r^*|$. Next, we square the left-most inequality in (4.5) and estimate further to get

$$((r^*)_\ell^k)^2 \leq C \left(\mathcal{E}(h_\ell)^2 h_\ell^2 \|\mathbf{u}\|_{\mathcal{N}(\Omega_\ell)}^2 + \mathcal{E}(h_k)^2 h_k^2 \|\mathbf{u}\|_{\mathcal{N}(\Omega_k)}^2 \right). \tag{4.7}$$

Now sum the estimate over all glue points, and note that each Ω_ℓ (and Ω_k) will appear in the sum at most m times (the maximum number of patches that intersect any given patch). This gives the result. ∎

In an attempt to bound the error solely in terms of the point distribution and target function, let us look at an application of this estimate to the residual term from (4.2). For simplicity, assume that all $h_\ell \sim h$ for all h_ℓ. Since there are at most m terms in the sum, a Cauchy-Schwarz inequality gives

$$\sum_{\ell | \mathbf{x} \in \Omega_\ell,\, \ell \neq k} h_\ell^{-1} |r_\ell^k| \leq h^{-1}\sqrt{m}|r| \leq Cm\mathcal{E}(h)\sqrt{\sum_\ell \|\mathbf{u}\|^2_{\mathcal{N}(\Omega_\ell)}}.$$

Due to the sum over all patches, this bound may or may not match the expected error rates. It is reasonable to guess that this sum is equivalent to $\|\mathbf{u}\|^2_{\mathcal{N}(\Omega)}$. Numerical experiments for scalar RBF interpolants, not presented here, suggest that such a sum may be uniformly bounded in the case of a thin plate spline, and may grow very slowly for Matérn kernels. We leave exploring a tight bound for this term as an open question. A very rough estimate of the sum would introduce a factor of $\sqrt{M}$, where M is the number of patches. In the quasi-uniform case, a volume argument gives $\sqrt{M} \sim h^{-d/2}$. Thus a worst-case scenario is that the method converges according to $\mathcal{E}(h)h^{-d/2}$. However, numerical experiments suggest that the errors decay according to $\mathcal{E}(h)$ (see for example Section 5.2) and do not seem to depend on the number of patches - which suggests that the estimate $\mathcal{E}(h)h^{-d/2}$ is pessimistic.

5. Numerical experiments. In this section, we numerically study the vector RBF-PUM for three different test problems: a div-free field in a star-shaped domain in $\mathbb{R}^2$, a div-free field on $\mathbb{S}^2$, and a curl-free field in the unit ball in $\mathbb{R}^3$. For each of these cases, we numerically test the convergence rates of the method and compare them to the estimates from Section 4. The point sets we use in the experiments are all quasiuniform, so rather than compute the mesh-norm h and use this to measure convergence rates, we simply use $h \sim N^{-1/d}$.

To illustrate the different convergence rates that are possible, we use the inverse multiquadric (IMQ) kernel $\phi(r) = 1/\sqrt{1+(\varepsilon r)^2}$ and the Matérn kernel $\phi(r) = e^{-\varepsilon r}\left(1 + (\varepsilon r) + \frac{3}{7}(\varepsilon r)^2 + \frac{2}{21}(\varepsilon r)^3 + \frac{1}{105}(\varepsilon r)^4\right)$. The latter kernel is piecewise smooth and the local error from Proposition 4.1, in terms of N, is given by $\mathcal{E}(N) = (\sqrt{N})^{-3.5}$ for $d = 2$ (see [26] for more details). The IMQ kernel is analytic and therefore the local error decreases faster than any algebraic rate. For scalar interpolation with the IMQ, the local error estimate is $\mathcal{E}(N) = e^{-C\log(N)N^{1/2d}}$ [39], where $C > 0$ is a constant. We demonstrate that this also appears to be the correct rate for the vector case. While the error estimates are in terms the ∞-norm, we also include results on the 2-norm for comparison purposes. Since we are interested in demonstrating the convergence rates from the theory, we fix the shape parameter ε in all the tests, as using different ε on a per patch level will lead to different constants in the estimates. The values were selected so that conditioning of the linear systems (2.8) (or (2.15)) is not an issue. Choosing variable shape parameters in scalar RBF-PUM is explored in [10] and may be adapted to the current method, but we leave that to a separate study. For brevity we report results for one kernel per example, with the IMQ kernel used for the first and third test and the Matérn used for the second. However, we note that the estimated convergence rates for each kernel were consistent with the theory across all tests. Finally, we set the weighted least squares parameter in (3.8) to $\eta = 4$. This value produced good results over all the numerical experiments performed.

All results were obtained from a MATLAB implementation of the vector RBF-PUM method executed on a MacBook Pro with 2.4 GHz 8-Core Intel Core i9 processor and 32 GB RAM. No explicit parallelization was implemented.

5.1. Div-free field on $\mathbb{R}^2$. The target field and domain for this numerical test are defined as follows. Let the potential for the field be

$$\psi^{(1)}(\mathbf{x}) = -2g(\tfrac{27}{2}\|\mathbf{x}\|^4) - \frac{1}{2}g(27\|\mathbf{x}\|^2) - 2\sum_{j=0}^{4} g(9\|\mathbf{x}-\boldsymbol{\xi}_j\|^2), \tag{5.1}$$

where $\boldsymbol{\xi}_j = (\cos(2\pi j/5 + 0.1), \sin(2\pi j/5 + \frac{1}{2}))$ and

$$g(r) = \exp(r)/(1+\exp(r))^2. \tag{5.2}$$

The target domain is set from the potential as $\Omega^{(1)} = \{\mathbf{x} \in \mathbb{R}^2 | \psi^{(1)}(\mathbf{x}) \leq -\frac{1}{10}\}$, and target div-free vector field is $\mathbf{u}^{(1)}_{\text{div}} = \mathbf{L}\psi^{(1)}$. This gives a star-like domain with a non-trivial field that is tangential to $\partial\Omega$; see Figure 4 for a visualization of the potential and field.

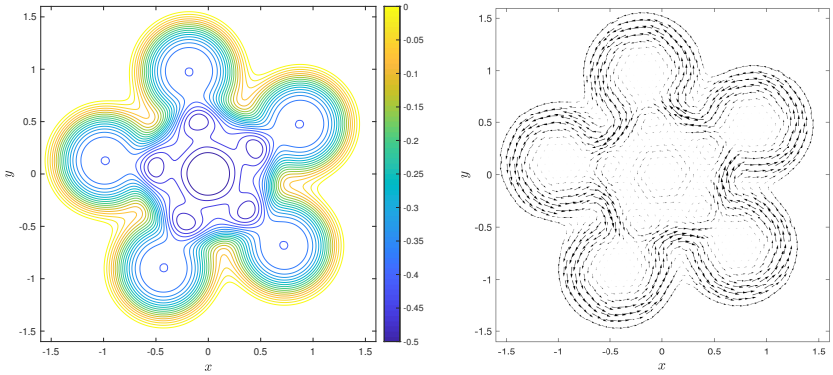
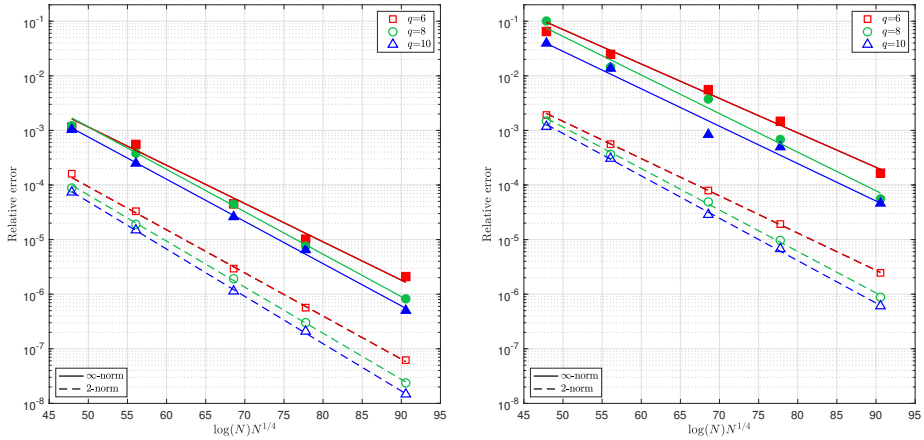


FIGURE 4. *Contours of the potential $\psi^{(1)}$ (left) and corresponding div-free velocity field $\mathbf{u}^{(1)}_{\text{div}}$ (right) for the numerical experiment on $\mathbb{R}^2$.*

The node sets X for this test were initially generated from DistMesh [38], but then perturbed by a small amount to remove any regular structures. The sizes of the node sets for the tests are $N = 11149$, 17405, 30943, 44570, and 69635[4]. We estimate A in (3.7) to be 6, and use an overlap parameter for the patches of $\delta = 1/2$. We test three different values of q to see how the errors are effected by increasing the nodes per patch. For $q = 6, 8, 10$, there are an average of $63, 112, 173$ nodes per patch, respectively. The boundaries create some variability in the nodes per patch and lead to minimum values of $32, 57, 85$ and the maximums of $109, 191, 300$, respectively. As mentioned above, we only report results for the IMQ kernel, for which the shape parameter is set to $\varepsilon = 13$ for all tests. Errors in the approximations of the target potential and field are computed at a dense set of 94252 points over the domain. Errors in the approximation of the target potential are computed after first normalizing the approximant and the potential to have a mean of zero over the evaluation points. For each N and q, the error reported is the average of the ∞-norm (2-norm) errors using 20 different random perturbations of the initial node set X. This reduces fluctuations in the errors caused by particularly good samples of the target field. We observed

[4]These node sets were produced from DistMesh when setting the "spacing" parameter to $h0 = 0.025, 0.02, 0.015, 0.0125, 0.01$



(a) Errors for the potential $\psi^{(1)}$ (b) Errors for the field $\mathbf{u}_{\text{div}}^{(1)}$

FIGURE 5. *Convergence results for the numerical experiment on the star domain in* $\mathbb{R}^2$ *for the IMQ kernel and different values of* q. *Filled (open) markers correspond to the relative* ∞*-norm (2-norm) errors and solid (dashed) lines indicate the fit to the estimate* $\mathcal{E}(N) = e^{-C\log(N)N^{1/4}}$, *without the first values included.*

that the relative standard deviation in the norms of the errors using this sampling technique varied from 5% to 10% for the 2-norm and 20% to 40% for the ∞-norm across the N we used.

Figure 5 displays the relative ∞-norm and 2-norm errors in the approximation of the target potential and field as a function of $\log(N)N^{1/4}$. Included in the figures are the lines of best fit to the errors using the error estimate $\mathcal{E}(N) = e^{-C\log(N)N^{1/4}}$ from scalar RBF theory. We see from the figure that this error estimate provides a good fit to both the ∞-norm and 2-norm errors for the potential and the field. The ∞-norm errors for the potential have more variability especially for $q = 6$, but the 2-norm errors are quite consistent. As expected, the errors in reconstructing the potential are lower than those for reconstructing the field, and the 2-norm errors are lower than the ∞-norm errors. Increasing q leads to a consistent decrease in the 2-norm errors, but the decrease is more variable for the ∞-norm errors.

5.2. Div-free field on $\mathbb{S}^2$. Let $\mathbf{x} = (x, y, z) \in \mathbb{S}^2$, and the potential for the target field be defined as

(5.3)
$$\psi^{(2)}(\mathbf{x}) = -\frac{1}{1+e^{-20(z+1/\sqrt{2})}} - \frac{1}{1+e^{-20(z-1/\sqrt{2})}} - 3\sum_{j=0}^{5}(-1)^j g(\|\mathbf{x}-\mathbf{y}_j\|^2, a_j),$$

where g is given in (5.2), $\mathbf{y}_j = (\cos(\lambda_j)\cos(\theta_j), \sin(\lambda_j)\cos(\theta_j), \sin(\theta_j))$ for $\{\lambda_j\}_{j=0}^5 = \{0.05, 1.1, 2.12, 3.18, 4.22, 5.26\}$ and $\{\theta_j\}_{j=0}^5 = \{0.79, -0.82, 0.76, -0.81, 0.8, -0.77\}$, and $a_j = 4 + j/2$. The div-free field is then given as $\mathbf{u}_{\text{div}}^{(2)} = \mathbf{L}\psi^{(2)}$. The values used in (5.3) were chosen to produce a zonal jet in the mid-latitudes with three superimposed vortices in each of the northern and southern hemispheres; see Figure 6 for a visualization of the potential and field.

The node sets X for this test are chosen as Hammersley nodes, which give quasiuniform, but random sampling points for $\mathbb{S}^2$ [49]. The sizes of the node sets for the

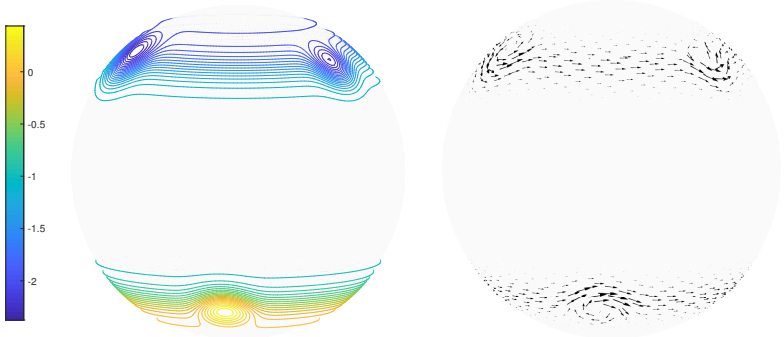
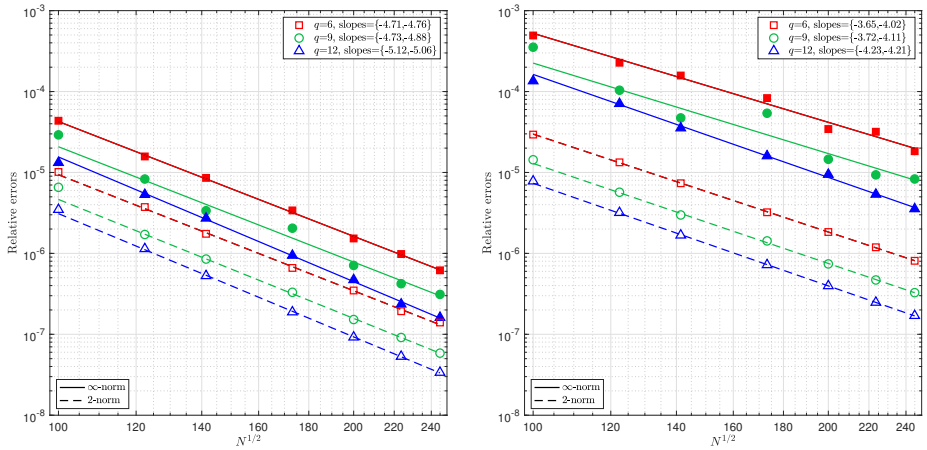


FIGURE 6. *Contours of the potential* $\psi^{(2)}$ *(left) and corresponding div-free velocity field* $\mathbf{u}_{\text{div}}^{(2)}$ *(right) for the numerical experiment on* $\mathbb{S}^2$.



(c) Errors for the potential $\psi^{(2)}$ (d) Errors for the field $\mathbf{u}_{\text{div}}^{(2)}$

FIGURE 7. *Convergence rates for the numerical experiment on* $\mathbb{S}^2$ *for the Matérn kernel and different values of* q. *Filled (open) markers correspond to the relative* ∞*-norm (2-norm) errors and solid (dashed) lines indicate the lines of best fit to the* ∞*-norm (2-norm) errors as a function of* $\sqrt{N}$ *on a loglog scale. The legend indicates the slopes of these lines with the first number corresponding to the* ∞*-norm and the second the 2-norm, which give estimates for the algebraic convergence rates.*

tests are $N = 10000$, 15000, 20000, 30000, 40000, 50000 and 60000. We use $A = 4\pi$ in (3.7) and set the overlap parameter to $\delta = 9/16$. We again use three different values of q to see how the errors are effected by increasing the nodes per patch. For $q = 6, 9, 12$, there are an average of $63, 143, 252$ nodes per patch, respectively. Since there are no boundaries for this domain, the number of nodes per patch is much more consistent across all patches. The minimum nodes per patch are $58, 137, 245$ and the maximums are $69, 150, 261$, respective to the q values. For this example, we only report results for the Matérn kernel, for which the shape parameter is set to $\varepsilon = 7.5$ for all tests. Errors in the approximations of the target potential and field are computed at a quasiuniform set of 92163 points over $\mathbb{S}^2$. Errors in the approximation of the target potential are again computed after first normalizing the approximant and the potential to have a mean of zero over the evaluation points. Similar to the previous experiment, for each N and q, the error reported is the average of the ∞-norm (2-

norm) errors from 20 different random rotations of the initial Hammersley node set X. We observed similar results on the relative standard deviations of the norms of the errors as the previous experiment using this sampling technique.

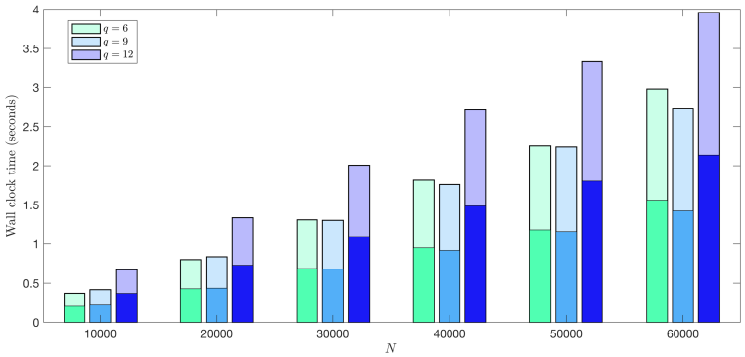


FIGURE 8. *Timing results for the numerical experiment on $\mathbb{S}^2$ with different values of q. The darker region of each bar marks the time it takes to compute the interpolation coefficients on each patch and solve for the potential shifts, while the full bar includes this time and the time it takes to evaluate the approximant of the field and the potential at N points.*

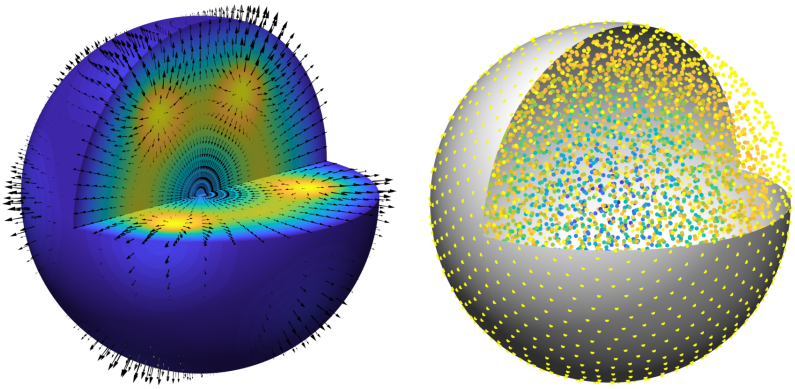
Figure 7 displays the relative ∞-norm and 2-norm errors in the approximation of the target potential and field as a function of $N^{1/2}$. Included in the figure are the lines of best fit to the log of the errors vs. the log of $N^{1/2}$ for each q, and the slopes of these lines are reported in the legend of the figure (where the first number is for ∞-norm and second for the 2-norm). We see from this figure that the computed rates of convergence for the ∞-norm are slightly higher than the theoretical rate of -3.5. Thus the residual estimate from Proposition 4.4 is not leading to a reduction in the convergence rates as discussed at the end of Section 4. We also see from the figure that the estimated rates for the 2-norm errors are higher than the ∞-norm errors as one would expect. Finally, similar to the previous experiment, we see that the errors in reconstructing the potential are lower than those for reconstructing the field.

We also display timing results for this experiment in Figure 8. For these results, we scaled the evaluation points with N and measured the time for the fitting phase of the method (determining the interpolation coefficients on each patch and the potential shifts) and the evaluation phase (evaluating the approximants of the field and potential on each patch and combining these using the PU weight functions). The results for $q = 9$ and $q = 12$ show a clear linear scaling with N, but the rate appears to be a bit higher for $q = 6$, which we anticipate is due to not being in the asymptotic range of N for this case. Also, the predicted $\mathcal{O}(N \log N)$ complexity is most likely not visible over the range of N considered. In all the results, we see that the evaluation phase takes less time than the fitting phase, which is expected since the cost for this phase is $\mathcal{O}(n^2)$ per patch vs. $\mathcal{O}(n^3)$ for fitting. Interestingly, with this serial version of the code, $q = 9$ is overall the fastest. Since the number of patches is inversely proportional to q^2, these results indicate that there is an optimal value that balances solving fewer larger systems to more smaller systems.

5.3. Curl-free field on the unit ball. The target curl-free field for this test is generated as follows. Let $g(r,a) = (a+r^2)^{-1/2}$ and define the following potential:

$$\psi^{(3)}(\mathbf{x}) = -\frac{1}{4}g(\|\mathbf{x}\|, 0.1) + \frac{1}{8}\sum_{j=1}^{1} 2g(\|\mathbf{x} - \boldsymbol{\xi}_j\|, 0.04), \tag{5.4}$$

where $\{\mathbf{x}_j\}_{j=1}^{12}$ are the vertices of a regular icosahedron with each vertex a distance of $2/3$ from the origin. The target curl-free is then generated by $\mathbf{u}^{(3)}_{\text{curl}} = -\nabla\psi^{(3)}$. This field can be interpreted as the (idealized) electric field that is generated from a negative (smoothed) point charge at the origin, surrounded by 12 positive (smoothed) point charges, equidistance from one another; see Figure 9(a) for a visualization of the potential and field.

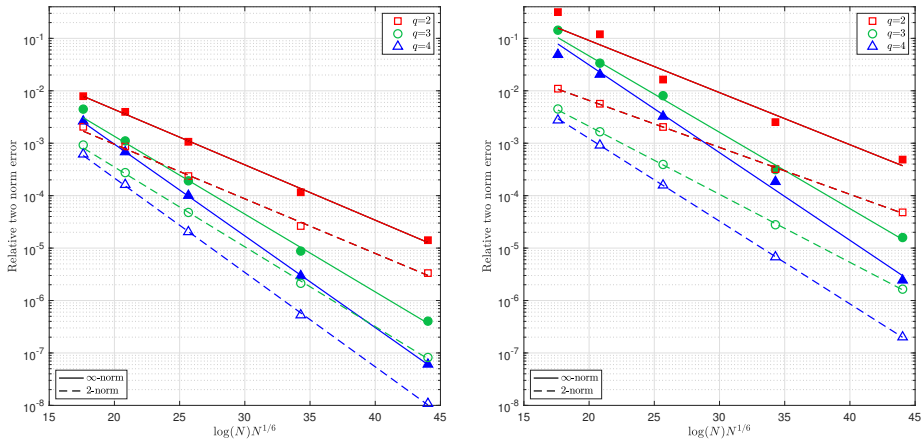


(a) Potential and field (b) Nodes

FIGURE 9. *(a) Visualization of the potential $\varphi^{(3)}$ and corresponding curl-free velocity field $\mathbf{u}^{(3)}_{\text{curl}} = -\nabla\varphi^{(3)}$ for the numerical experiment on the unit ball. (b) Example of $N = 4999$ node set (small solid disks) used in the numerical experiment on the unit ball, where colors of the nodes are proportional to their distance from the origin (yellow=1, green = 0.5, blue=0). The plots in both figures show the unit ball with a wedge removed to aid in the visualization.*

The node sets X for this test are obtained from the meshfree node generator described in [41], which produces quasiuniform but unstructured nodes in general domains; see Figure 9 (b) for an example of the nodes used for the unit ball. The sizes of the node sets for the tests are $N = 4999$, 9103, 19636, 59116, and 158474[5]. We use $A = 4/3\pi$ in (3.7) and an overlap parameter of $\delta = 1/4$. We again test three different values of q: $q = 2, 3, 4$. For $q = 2$, the minimum, average, and maximum nodes per patch are 18, 37, 83, for $q = 3$ these values are 72, 120, 238, and for $q = 4$ these values are 186, 271, 512. As with the first experiment, we only present results for the IMQ kernel, for which the shape parameter is set to $\varepsilon = 4$ for all tests. Errors in the approximations of the target potential and field are computed at a set of 208707 points over the unit ball. Errors in the approximation of the target potential are again computed after first normalizing the approximant and the potential to have a mean of zero over the evaluation points. Similar to the previous experiments, for each N and q, the error reported is the average of the ∞-norm (2-norm) errors from 20 different random rotations of the initial node set X.

[5]These node sets were produced from the node generator [41] when setting the "spacing" parameter to $h0 = 0.1, 0.08, 0.06, 0.04, 0.028$



(a) Errors for the potential $\varphi^{(3)}$ (b) Errors for the field $\mathbf{u}^{(3)}_{\text{curl}}$

FIGURE 10. *Convergence results for the numerical experiment on the unit ball in* $\mathbb{R}^3$ *for the IMQ kernel and different values of* q*. Filled (open) markers correspond to the relative* ∞*-norm (2-norm) errors and solid (dashed) lines indicate the fit to the expected error estimate* $\mathcal{E}(N) = e^{-C\log(N)N^{1/6}}$*, without the first values included.*

Figure 10 displays the relative ∞-norm and 2-norm errors in the approximation of the target potential and field as a function of $\log(N)N^{1/6}$. As in the first experiment, we have included the lines of best fit to the errors, but now using $\mathcal{E}(N) = e^{-C\log(N)N^{1/6}}$. We see from the Figure that the error estimate again generally provides a good fit to both the ∞-norm and 2-norm errors for the potential and the field. The ∞-norm errors deviate more from the estimates than the 2-norm errors, especially for field in the $q = 2$ case. However, for this case the minimum number of points per patch can be quite small.

Remark 5.1. In practice, there are several parameters a user needs to choose in the algorithm that effect the computational cost and accuracy. In the experiments reported here, and several others not reported, we have explored these parameters and come up with the following suggestions. For the q parameter, which controls the average nodes per patch, we recommend a value in the range of $8 \leq q \leq 9$ for 2D problems and $3 \leq q \leq 4$ for 3D problems. For the overlap parameter, δ, we recommend a value in the range $1/2 \leq \delta \leq 3/4$. For the shape parameter ε, we recommend choosing it as small as possible on each patch before ill-conditioning sets in when solving the local linear systems (2.8). This is similar to the method [43] used for generating RBF finite difference formulas. For smooth vector fields, this typically gives the best accuracy for a given N.

6. Concluding remarks. We have presented a new method based on div/curl-free RBFs and PUM for approximating div/curl-free vector fields in $\mathbb{R}^2$ and $\mathbb{S}^2$, and for curl-free fields in $\mathbb{R}^3$. The method produces approximants that are analytically div/curl-free and also produces an approximant potential for the field at no additional cost. For quasi-uniform samples, we have shown how the parameters can be selected so that the computational complexity of the method is $\mathcal{O}(N \log N)$. We have proved error estimates for the approximants based on local estimates for the div/curl-free interpolants on the PU patches. We have also demonstrated the high-order convergence rates of the method on three different test problems with samples ranging from

thousands to hundreds of thousands of nodes—all done on a standard laptop.

While we have only focused on div/curl-free interpolation over local patches, a future area to explore is to instead use a least squares approach similar to the one used for scalar RBFs in [32]. Here one can choose fewer centers in the local patches for the div/curl-free RBFs than data samples, a technique referred to as regression splines in the statistics literature [18, ch. 19]. This has the benefit of further reducing the cost of the local patch solves for the approximation coefficients and could provide some regularization. Another future area to explore is the adaption of stable algorithms for "flat" RBFs [19, 21] to the div/curl-free RBFs. These algorithms are especially important in scalar RBF-PUM methods based on smooth RBFs for reaching high accuracies [32]. Some work has been done along these lines for $\mathbb{S}^2$ in [15], but not for the local setting on patches. A final promising area for future research is in developing adaptive algorithms for the method along the lines of [10].

Acknowledgments. We thank Elisabeth Larsson for helpful discussions regarding the PU patch distribution algorithm and Varun Shankar for generating the node sets used for the unit ball example. KPD's work was partially supported by the SMART Scholarship funded by The Under Secretary of Defense-Research and Engineering, National Defense Education Program/BA-1, Basic Research. GBW's work was partially supported by National Science Foundation grant 1717556.

REFERENCES

[1] K. A. Aiton, *A Radial Basis Function Partition of Unity Method for Transport on the Sphere*, master's thesis, Boise State University, USA, 2014.

[2] L. Amodei and M. N. Benbourhim, *A vector spline approximation*, J. Approx. Theory, 67 (1991), pp. 51–79.

[3] I. Babuška and J. M. Melenk, *The partition of unity method*, Int. J. Numer. Meths. Eng., 40 (1997), pp. 727–758.

[4] Y. Bao, A. Donev, B. E. Griffith, D. M. McQueen, and C. S. Peskin, *An immersed boundary method with divergence-free velocity interpolation and force spreading*, J. Comput. Phys., 347 (2017), pp. 183–206.

[5] H. Bhatia, G. Norgard, V. Pascucci, and P.-T. Bremer, *The Helmholtz-Hodge decomposition—a survey*, IEEE Transactions on Visualization and Computer Graphics, 19 (2013), pp. 1386–1404.

[6] R. Cavoretto and A. De Rossi, *Fast and accurate interpolation of large scattered data sets on the sphere*, Comput. Appl. Math., (2010), pp. 1505–1521.

[7] R. Cavoretto and A. De Rossi, *A trivariate interpolation algorithm using a cube-partition searching procedure*, SIAM J. Sci. Comput., 37 (2015), pp. A1891–A1908.

[8] R. Cavoretto, A. De Rossi, G. E. Fasshauer, M. J. McCourt, and E. Perracchione, *Anisotropic weights for RBF-PU interpolation with subdomains of variable shapes*, in Radu F., Kumar K., Berre I., Nordbotten J., Pop I. (eds) Numerical Mathematics and Advanced Applications ENUMATH 2017. Lecture Notes in Computational Science and Engineering, Springer, Cham, 2019.

[9] R. Cavoretto, A. De Rossi, and E. Perracchione, *Partition of unity interpolation on multivariate convex domains*, Int. J. Model. Simul. Sci. Comp., 06 (2015), p. 1550034.

[10] R. Cavoretto, A. De Rossi, and E. Perracchione, *RBF-PU interpolation with variable subdomain sizes and shape parameters*, in AIP Conference Proceedings, vol. 1776, AIP Publishing, 2016, p. 070003.

[11] D. Coe, E. Fuselier, N. Benítez, T. Broadhurst, B. Frye, and H. Ford, *LensPerfect: Gravitational lens mass map reconstructions yielding exact reproduction of all multiple images*, Astrophys. J, 681 (2008), pp. 814–830.

[12] A. Dharwadker and S. Pirzad, *Graph Theory*, CreateSpace Independent Publishing Platform, North Charleston, SC, USA, 2011.

[13] M. P. do Carmo, *Differential forms and applications*, Universitext, Springer-Verlag, Berlin, 1994, https://doi.org/10.1007/978-3-642-57951-6, https://doi.org/10.1007/978-3-642-57951-6. Translated from the 1971 Portuguese original.

[14] F. Dodu and C. Rabut, *Irrotational or divergence-free interpolation*, Numer. Math., 98 (2004), pp. 477–498.
[15] K. P. Drake and G. B. Wright, *A stable algorithm for divergence-free radial basis functions in the flat limit*, J. Comput. Phys., 417 (2020), p. 109595.
[16] M. Fan, D. Paul, T. C. M. Lee, and T. Matsuo, *Modeling tangential vector fields on a sphere*, Journal of the American Statistical Association, 113 (2018), pp. 1625–1636.
[17] P. Farrell, K. Gillow, and H. Wendland, *Multilevel interpolation of divergence-free vector fields*, IMA J. Numer. Anal., 37 (2016), pp. 332–353.
[18] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB, Interdisciplinary Mathematical Sciences*, World Scientific Publishers, Singapore, 2007.
[19] G. E. Fasshauer and M. J. McCourt, *Stable evaluation of Gaussian radial basis function interpolants*, SIAM J. Sci. Comput., 34 (2012), pp. A737–A762.
[20] B. Fornberg and N. Flyer, *A Primer on Radial Basis Functions with Applications to the Geosciences*, SIAM, Philadelphia, 2014.
[21] B. Fornberg, E. Larsson, and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput., 33 (2011), pp. 869–892.
[22] E. J. Fuselier, *Improved stability estimates and a characterization of the native space for matrix-valued RBFs*, Adv. Comput. Math., 29 (2008), pp. 269–290.
[23] E. J. Fuselier, *Sobolev-type approximation rates for divergence-free and curl-free RBF interpolants*, Math. Comp., 77 (2008), pp. 1407–1423.
[24] E. J. Fuselier, F. J. Narcowich, J. D. Ward, and G. B. Wright, *Error and stability estimates for surface-divergence free RBF interpolants on the sphere*, Math. Comp., 78 (2009), pp. 2157–2186.
[25] E. J. Fuselier, V. Shankar, and G. B. Wright, *A high-order radial basis function (RBF) Leray projection method for the solution of the incompressible unsteady Stokes equations*, Comput. Fluids, 128 (2016), pp. 41–52.
[26] E. J. Fuselier and G. B. Wright, *Stability and error estimates for vector field interpolation and decomposition on the sphere with RBFs*, SIAM J. Numer. Anal., 47 (2009), pp. 3213–3239.
[27] M. Griebel and M. A. Schweitzer, *A particle-partition of unity method–part II: Efficient cover construction and reliable integration*, SIAM J. Sci. Comput., 23 (2002), pp. 1655–1682.
[28] D. Handscomb, *Local recovery of a solenoidal vector field by an extension of the thin-plate spline technique*, Numer. Algorithms, 5 (1993), pp. 121–129. Algorithms for approximation, III (Oxford, 1992).
[29] T. Hangelbroek, F. J. Narcowich, and J. D. Ward, *Polyharmonic and related kernels on manifolds: Interpolation and approximation*, Foundations of Computational Mathematics, 12 (2012), pp. 625–670.
[30] D. P. Hardin and E. B. Saff, *Discretizing manifolds via minimum energy points*, Notices Amer. Math. Soc., 51 (2004), pp. 1186–1194.
[31] U. Harlander, T. von Larcher, G. B. Wright, M. Hoff, K. Alexandrov, and C. Egbers, *Orthogonal decomposition methods to analyze PIV, LDA and thermography data of a thermally driven rotating annulus laboratory experiment*, in Modelling Atmospheric and Oceanic flows: insights from laboratory experiments and numerical simulations, T. von Larcher and P. D. Williams, eds., American Geophysical Union, Washington D.C., 2014.
[32] E. Larsson, V. Shcherbakov, and A. Heryudono, *A least squares radial basis function partition of unity method for solving PDEs*, SIAM J. Sci. Comput., 39 (2017), pp. A2538–A2563.
[33] D. Lazzaro and L. B. Montefusco, *Radial basis functions for the multivariate interpolation of large scattered data sets*, J. Comp. Appl. Math., 140 (2002), pp. 521–536.
[34] S. Lowitzsch, *Error estimates for matrix-valued radial basis function interpolation*, J. Approx. Theory, 137 (2005), pp. 238–249.
[35] A. A. Mitrano and R. B. Platte, *A numerical study of divergence-free kernel approximations*, Appl. Numer. Math., 96 (2015), pp. 94 – 107.
[36] F. J. Narcowich and J. D. Ward, *Generalized Hermite interpolation via matrix-valued conditionally positive definite functions*, Math. Comp., 63 (1994), pp. 661–687.
[37] F. J. Narcowich, J. D. Ward, and G. B. Wright, *Divergence-free RBFs on surfaces*, J. Fourier Anal. Appl., 13 (2007), pp. 643–663.
[38] P.-O. Persson and G. Strang, *A simple mesh generator in Matlab*, SIAM Rev., 46 (2004), pp. 329–345.
[39] C. Rieger and B. Zwicknagl, *Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning*, Adv. Comput. Math., 32 (2010), pp. 103–

129.
[40] A. Safdari-Vaighani, A. Heryudono, and E. Larsson, *A radial basis function partition of unity collocation method for convection–diffusion equations arising in financial applications*, J. Sci. Comput., 64 (2015), pp. 341–367.
[41] V. Shankar, R. Kirby, and A. Fogelson, *Robust node generation for mesh-free discretizations on irregular domains and surfaces*, SIAM J. Sci. Comput., 40 (2018), pp. A2584–A2608.
[42] V. Shankar and G. B. Wright, *Mesh-free semi-Lagrangian methods for transport on a sphere using radial basis functions*, J. Comput. Phys., 366 (2018), pp. 170–190.
[43] V. Shankar, G. B. Wright, R. M. Kirby, and A. L. Fogelson, *A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction-diffusion equations on surfaces*, J. Sci. Comput., 63 (2014), pp. 745–768.
[44] V. Shcherbakov, *Radial basis function partition of unity operator splitting method for pricing multi-asset American options*, BIT, 56 (2016), pp. 1401–1423.
[45] N. Trask, M. Maxey, and X. Hu, *A compatible high-order meshless method for the Stokes equations with applications to suspension flows*, J. Comput. Phys., 355 (2018), pp. 310–326.
[46] H. Wendland, *Fast evaluation of radial basis functions : Methods based on partition of unity*, in Approximation Theory X: Wavelets, Splines, and Applications, Vanderbilt University Press, 2002, pp. 473–483.
[47] H. Wendland, *Scattered data approximation*, vol. 17 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005.
[48] H. Wendland, *Divergence-free kernel methods for approximating the Stokes problem*, SIAM J. Numer. Anal., 47 (2009), pp. 3158–3179.
[49] G. B. Wright, *SpherePts*. https://github.com/gradywright/spherepts/, 2017.