

Generalized moving least squares vs. radial basis function finite difference methods for approximating surface derivatives

Andrew M. Jones^a, Peter A. Bosler^b, Paul A. Kuberry^b, Grady B. Wright^{a,*}

^a*Boise State University, 1910 University Drive, 83725, Boise, Idaho, USA*

^b*Sandia National Laboratories, P.O. Box 5800, 87185, Albuquerque, New Mexico, USA*

Abstract

Approximating differential operators defined on two-dimensional surfaces is an important problem that arises in many areas of science and engineering. Over the past ten years, localized mesh-free methods based on generalized moving least squares (GMLS) and radial basis function finite differences (RBF-FD) have been shown to be effective for this task as they can give high orders of accuracy at low computational cost, and they can be applied to surfaces defined only by point clouds. However, there have yet to be any studies that perform a direct comparison of these methods for approximating surface differential operators (SDOs). The first purpose of this work is to fill that gap. For this comparison, we focus on an RBF-FD method based on polyharmonic spline kernels and polynomials (PHS+Poly) since they are most closely related to the GMLS method. Additionally, we use a relatively new technique for approximating SDOs with RBF-FD called the tangent plane method since it is simpler than previous techniques and natural to use with PHS+Poly RBF-FD. The second purpose of this work is to relate the tangent plane formulation of SDOs to the local coordinate formulation used in GMLS and to show that they are equivalent when the tangent space to the surface is known exactly. The final purpose is to use ideas from the GMLS SDO formulation to derive a new RBF-FD method for approximating the tangent space for a point cloud surface when it is unknown. For the numerical comparisons of the methods, we examine their convergence rates for approximating the surface gradient, divergence, and Laplacian as the point clouds are refined for various parameter choices. We also compare their efficiency in terms of accuracy per computational cost, both when including and excluding setup costs.

Keywords: PDEs on surfaces, Meshfree, Meshless, RBF-FD, GMLS, Polyharmonic spline

2008 MSC: 65D05, 65D25, 65M06, 65M75, 65N06, 65N75, 41A05, 41A10, 41A15

1. Introduction

The problem of approximating differential operators defined on two dimensional surfaces embedded in \mathbb{R}^3 arises in many multiphysics models. For example, simulating atmospheric flows with Eulerian or Lagrangian numerical methods requires approximating the surface gradient, divergence, and Laplacian on the two-sphere [1–4]. Similar surface differential operators (SDOs) on more geometrically complex surfaces appear in models of ice sheet dynamics [5], biochemical signaling on cell membranes [6], morphogenesis [7], texture synthesis [8], and sea-air hydrodynamics [9].

Localized meshfree methods based on generalized moving least squares (GMLS) and radial basis function finite differences (RBF-FD) have become increasingly popular over the last ten years for approximating SDOs and solving surface partial differential equations (PDEs); see, for example, [10–14] for GMLS and [15–26] for RBF-FD. These methods can be applied to surfaces defined by point clouds, without having to form a triangulation of the surface like surface finite element methods [27] or a level-set representation of the surface like embedded finite element methods [28]. Additionally,

*Corresponding author

Email addresses: andrewjones237@u.boisestate.edu (Andrew M. Jones), pabosle@sandia.gov (Peter A. Bosler), pakuber@sandia.gov (Paul A. Kuberry), gradywright@boisestate.edu (Grady B. Wright)

for the special case of the sphere, RBF-FD has been shown to be highly competitive with element based methods in terms of accuracy per degree of freedom [16, 17, 23]. While there is one study dedicated to comparing GMLS and RBF-FD for approximating functions and derivatives in \mathbb{R}^2 and \mathbb{R}^3 [29], there are no studies that compare them for approximating SDOs. The first purpose of the present work is to fill this gap.

The RBF-FD methods referenced above use different approaches for approximating SDOs, while the GMLS methods essentially use the same approach based on (weighted) polynomial least squares. To keep the comparison to GMLS manageable, we will limit our focus to an RBF-FD method based on polyharmonic spline (PHS) kernels augmented with polynomials (or PHS+Poly) since they are most closely related to GMLS [29]. Additionally, these RBF-FD methods are becoming more and more prevalent as they can give high orders of accuracy that are controlled by the augmented polynomial degree [30] and they do not require choosing a shape parameter, which can be computationally intensive to do in an automated way.

The techniques for formulating SDOs also vary significantly in the RBF-FD methods referenced above, while the formulations used in GMLS are similar, being based on local coordinates to the surface. In this work, we limit our focus to the so-called tangent plane formulation with RBF-FD, as it provides a more straightforward technique for incorporating polynomials in RBF-FD methods than [15, 16, 18–21, 24, 25] and is related to the local coordinate formulation used in GMLS (see below). Additionally, the comparison in [22] of several RBF-FD methods for approximating the surface Laplacian (Laplace-Beltrami operator) revealed the tangent plane approach to be the most computationally efficient in terms of accuracy per computational cost. The tangent plane method was first introduced by Demanet [31] for approximating the surface Laplacian using polynomial based approximations. Suchde & Kuhnert [13] generalized this method to other SDOs using polynomial weighted least squares. Shaw [22] (see also [26]) was the first to use this method for approximating the surface Laplacian with RBF-FD and Gundersen et. al. [23] independently developed the method for RBF-FD specialized to the surface gradient and divergence on the unit two-sphere. The second purpose of the present work is to analytically compare the local coordinate formulation of SDOs used in GMLS to the tangent plane formulation and to show that these formulations are in fact identical when the tangent space for the surface is known exactly for the given point cloud.

When the tangent space is unknown, which is generally the case for surface represented by point clouds, it must be approximated. There has been little attention given in the literature on RBF-FD methods for how to do these approximations; commonly it is assumed that they are computed by some separate techniques (e.g., [15, 18, 24]). However, for GMLS, these approximations are incorporated directly in the methods (e.g., [11, 12, 14]). The third purpose of this work is use the ideas from GMLS to develop a new RBF-FD technique for approximating the tangent space directly using PHS+Poly. By combining this with the tangent plane method, we arrive at the first comprehensive PHS+Poly RBF-FD framework for approximating SDOs on point cloud surfaces.

The GMLS and RBF-FD methods both use weighted combinations of function values over a local stencil of points to approximate SDOs. They also feature a parameter ℓ for controlling the degree of polynomial precision of the formulas. For the numerical comparisons of the methods, we investigate how the size of the stencils and the polynomial degree effect the convergence rates of the methods for approximating SDOs under refinement. We focus on approximations of the surface gradient, divergence, and Laplacian operators on two topologically distinct surfaces, the unit two-sphere and the torus, which are representative of a broad range of application domains. In the case of the sphere, we also study the convergence rates of the methods for different point sets, including icosahedral points that are popular in applications. Finally, we investigate the efficiency of the methods in terms of their accuracy versus computational cost, both when including and excluding setup costs.

Our numerical results demonstrate that RBF-FD and GMLS give similar convergence rates for the same choice of polynomial degree ℓ , but overall RBF-FD results in lower errors. We also show that the often-reported super convergence of GMLS for the surface Laplacian only happens for highly structured, quasi-uniform point sets, and when the point sets are more general (but still possibly quasi-uniform), this convergence rate drops to the theoretical rate. Additionally, we find that the errors for RBF-FD can be further reduced with increasing stencil sizes, but that this does not generally hold for GMLS, and the errors can actually deteriorate. Finally, we find that when

setup costs are included, GMLS has an advantage in terms of efficiency, but if these are neglected then RBF-FD is more efficient.

The remainder of the paper is organized as follows. In Section 2, we provide some background and notation on stencil-based approximations and on surface differential operators. We follow this with a detailed overview of the GMLS and RBF methods in Section 3 and 4, respectively. In particular, Section 4.1 shows the equivalence of the local coordinate and tangent plane formulations of some SDOs, and Section 4.4 introduces an RBF-FD method for approximating the tangent space. A comparison of some theoretical properties of the two methods is given in Section 5, while extensive numerical comparisons are given in Section 6. We end with some concluding remarks in Section 7.

2. Background and notation

2.1. Stencils

The RBF-FD and GMLS methods both discretize SDOs by weighted combinations of function values over a local *stencil* of points. This makes them similar to traditional finite-difference methods, but the lack of a grid, a tuple indexing scheme, and inherent awareness of neighboring points requires that some different notation and concepts be introduced. In this section we review the stencil notation that will be used in the subsequent sections.

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ be a global set of points (point cloud) contained in some domain Ω . A *stencil* of \mathbf{X} is a subset of $n \leq N$ nodes of \mathbf{X} that are close (see discussion below for what this means) to some point $\mathbf{x}_c \in \Omega$, which is called the *stencil center*. In this work, the stencil center is some point from \mathbf{X} , so that $\mathbf{x}_c = \mathbf{x}_i$, for some $1 \leq i \leq N$, and this point is always included in the stencil. We denote the subset of points making up the stencil with stencil center \mathbf{x}_i as \mathbf{X}^i and allow the number of points in the stencil to vary with \mathbf{x}_i . To keep track of which points in \mathbf{X}^i belong to \mathbf{X} , we use *index set* notation and let σ^i denote the set of indices of the $1 < n_i \leq N$ points from \mathbf{X} that belong to \mathbf{X}^i . Using this notation, we write the elements of the stencil as $\mathbf{X}^i = \{\mathbf{x}_j\}_{j \in \sigma^i}$. We also use the convention that the indices are sorted by the distance the stencil points are from the stencil center \mathbf{x}_i , so that the first element of σ^i is i .

With the above notation, we can define a general stencil-based approximation method to a given (scalar) linear differential operator \mathcal{L} . Let u be a scalar-valued function defined on Ω that is smooth enough so that $\mathcal{L}u$ is defined for all $\mathbf{x} \in \Omega$. The approximation to $\mathcal{L}u$ at any $\mathbf{x}_i \in \mathbf{X}$ is given as

$$\mathcal{L}u|_{\mathbf{x}=\mathbf{x}_i} \approx \sum_{j \in \sigma^i} c_{ij} u(\mathbf{x}_j). \quad (1)$$

The weights c_{ij} are determined by the method of approximation, which in this study will be either GMLS or RBF-FD. These weights can be assembled into a sparse $N \times N$ “stiffness” matrix, similar to mesh-based methods. Vector linear differential operators (e.g., the gradient) can be similarly defined where (1) is used for each component and \mathcal{L} is the scalar operator for that component.

There are two main approaches used in the meshfree methods literature for determining the stencil points, one based on k -nearest neighbors (KNN) and one based on ball searches. These are illustrated in Figure 1 for a scattered point set \mathbf{X} in the plane. The approach that uses KNN is straightforward since it amounts to simply choosing the stencil \mathbf{X}^i as the subset of n_i points from \mathbf{X} that are closest to \mathbf{x}_i . The approach that uses ball searches is a bit more involved, so we summarize it in Algorithm 1. Both methods attempt to select points such that the stencil satisfies polynomial unisolvency conditions (see the discussion in Section 3.1). In this work, we use the method in Algorithm 1 since

- it is better for producing stencils with symmetries when \mathbf{X} is regular, which can be beneficial for improving the accuracy of the approximations;
- it is more natural to use with the weighting kernel inherent to GMLS; and
- it produces stencils that are not biased in one direction when the spacing of the points in \mathbf{X} are anisotropic.

Algorithm 1 Procedure for determining the stencil points based on ball searches.

- 1: **Input:** Point cloud \mathbf{X} ; stencil center \mathbf{x}_c ; number initial stencil points n ; radius factor $\tau \geq 1$
- 2: **Output:** Indices σ^c in \mathbf{X} for the stencil center \mathbf{x}_c
- 3: Find the n nearest neighbors in \mathbf{X} to \mathbf{x}_c , using the Euclidean distance
- 4: Compute the max distance h_{\max} between \mathbf{x}_c and its n nearest neighbors
- 5: Find the indices σ^c of the points in \mathbf{X} contained in the ball of radius τh_{\max} centered at \mathbf{x}_c

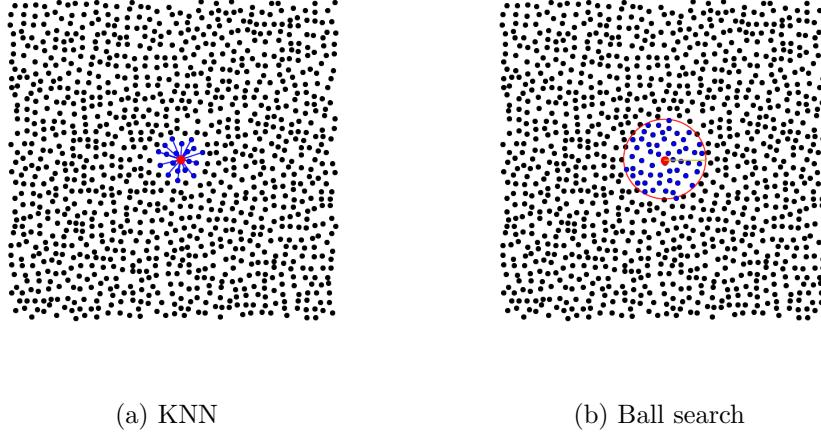


Figure 1: Comparison of the two search algorithms used in this paper for determining a stencil. The nodes \mathbf{X} are marked with solid black disks and all the stencil points are marked with solid blue disks, except for the stencil center, which is marked in red.

To measure distance in the ball search, we use the standard Euclidean distance measured in \mathbb{R}^3 rather than distance on the surface since this is simple to compute for any surface. We also use a k -d tree to efficiently implement the method. Finally, the choice of parameters we use in Algorithm 1 are discussed in Section 3.3.

2.2. Surface differential operators in local coordinates

Here we review some differential geometry concepts that will be used in the subsequent sections. We refer the reader to the books [32–34] for a thorough discussion of these concepts and the derivations of what follows.

We assume that $\mathcal{M} \subset \mathbb{R}^3$ is a regular surface and let $T_{\mathbf{x}}\mathcal{M}$ denote the set of all vectors in \mathbb{R}^3 that are tangent to \mathcal{M} at $\mathbf{x} \in \mathcal{M}$ (i.e., the tangent space to \mathcal{M} at \mathbf{x}). This assumption implies that for each point $\mathbf{x} \in \mathcal{M}$ there exists a local parameterization in $T_{\mathbf{x}}\mathcal{M}$ of a neighborhood (or patch) of \mathcal{M} containing \mathbf{x} of the form

$$\mathbf{f}(\hat{x}, \hat{y}) = (\hat{x}, \hat{y}, f(\hat{x}, \hat{y})), \quad (2)$$

where \hat{x}, \hat{y} are local coordinates for $T_{\mathbf{x}}\mathcal{M}$, and f is a smooth function for the “height” of the surface patch over $T_{\mathbf{x}}\mathcal{M}$ [34]. This local parametric representation of a surface is called a Monge patch or Monge form [33] and is illustrated for a bumpy sphere surface in Figure 2. As we see below, it is particularly well suited for computing SDOs.

Using the parameterization (2), the local metric tensor G about \mathbf{x} for the surface is given as

$$G = \begin{bmatrix} \partial_{\hat{x}}\mathbf{f} \cdot \partial_{\hat{x}}\mathbf{f} & \partial_{\hat{x}}\mathbf{f} \cdot \partial_{\hat{y}}\mathbf{f} \\ \partial_{\hat{y}}\mathbf{f} \cdot \partial_{\hat{x}}\mathbf{f} & \partial_{\hat{y}}\mathbf{f} \cdot \partial_{\hat{y}}\mathbf{f} \end{bmatrix} = \begin{bmatrix} 1 + (\partial_{\hat{x}}f)^2 & (\partial_{\hat{x}}f)(\partial_{\hat{y}}f) \\ (\partial_{\hat{x}}f)(\partial_{\hat{y}}f) & 1 + (\partial_{\hat{y}}f)^2 \end{bmatrix}. \quad (3)$$

Letting g^{ij} denote the (i, j) entry of G^{-1} , the surface gradient operator locally about \mathbf{x} is given as

$$\hat{\nabla}_{\mathcal{M}} = (\partial_{\hat{x}}\mathbf{f}) (g^{11}\partial_{\hat{x}} + g^{12}\partial_{\hat{y}}) + (\partial_{\hat{y}}\mathbf{f}) (g^{21}\partial_{\hat{x}} + g^{22}\partial_{\hat{y}}). \quad (4)$$

However, this is the surface gradient with respect to the horizontal $\hat{x}\hat{y}$ -plane (see Figure 2 (b)), and subsequently needs to be rotated so that it is with respect to $T_{\mathbf{x}}\mathcal{M}$ in its original configuration. If ξ^1 and ξ^2 are orthonormal vectors that span $T_{\mathbf{x}}\mathcal{M}$ and $\boldsymbol{\eta}$ is the unit outward normal to \mathcal{M} at \mathbf{x} , then the surface gradient in the correct orientation is given as

$$\nabla_{\mathcal{M}} = \underbrace{[\xi^1 \quad \xi^2 \quad \boldsymbol{\eta}]}_R \hat{\nabla}_{\mathcal{M}}. \quad (5)$$

Using this result, the surface divergence of a smooth vector $\mathbf{u} \in T_{\mathbf{x}}\mathcal{M}$ can be written as

$$\nabla_{\mathcal{M}} \cdot \mathbf{u} = (g^{11}\partial_{\hat{x}} + g^{12}\partial_{\hat{y}})(\partial_{\hat{x}}\mathbf{f})^T R^T \mathbf{u} + (g^{21}\partial_{\hat{x}} + g^{22}\partial_{\hat{y}})(\partial_{\hat{y}}\mathbf{f})^T R^T \mathbf{u} \quad (6)$$

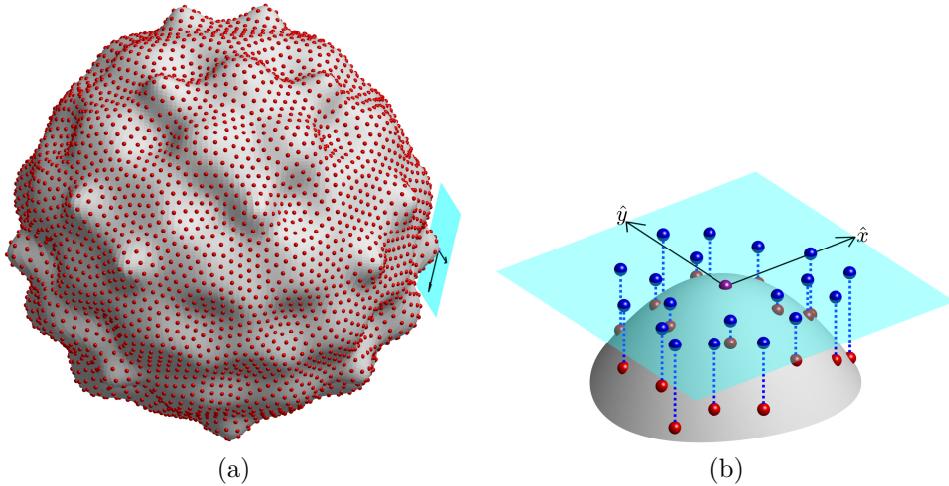


Figure 2: Illustration of a Monge patch parameterization for a local neighborhood of a regular surface \mathcal{M} in 3D. (a) Entire surface (in gray) together with the tangent plane (in cyan) for a point \mathbf{x}_c where the Monge patch is constructed (i.e., $T_{\mathbf{x}_c}\mathcal{M}$); red spheres mark a global point cloud \mathbf{X} on the surface. (b) Close-up view of the Monge patch parameterization, together with the points from a stencil \mathbf{X}_c (red spheres) formed from \mathbf{X} and the projection of the stencil to the tangent plane (blue spheres); the stencil center \mathbf{x}_c is at the origin of the axes for the $\hat{x}\hat{y}$ -plane and is marked with a violet sphere.

The surface Laplacian operator locally about \mathbf{x} is given as

$$\Delta_{\mathcal{M}} = \frac{1}{\sqrt{|g|}} \left(\partial_{\hat{x}} \left(\sqrt{|g|} g^{11} \partial_{\hat{x}} \right) + \partial_{\hat{x}} \left(\sqrt{|g|} g^{12} \partial_{\hat{y}} \right) + \right. \\ \left. \partial_{\hat{y}} \left(\sqrt{|g|} g^{21} \partial_{\hat{x}} \right) + \partial_{\hat{y}} \left(\sqrt{|g|} g^{22} \partial_{\hat{y}} \right) \right), \quad (7)$$

where $|g| = \det(G)$. This operator is invariant to rotations of the surface in \mathbb{R}^3 , so no subsequent modifications of (7) are necessary.

3. GMLS using local coordinates

The formulation of GMLS on a manifold was introduced by Liang & Zhao [11] and further refined by Trask, Kuberry, and collaborators [12, 14]. It uses local coordinates to approximate SDOs as defined in (5)–(7) and requires a method to also approximate the metric terms. Both approximations are computed for each $\mathbf{x}_i \in \mathbf{X} \subset \mathcal{M}$ using GMLS over a local stencil of points $\mathbf{X}^i \subset \mathbf{X}$. Below we give a brief overview of the method assuming that the tangent/normal vectors for the surface are known for each $\mathbf{x}_i \in \mathbf{X}$. We then discuss a method for approximating these that is used in the Compadre Toolkit [35], which we use in the numerical experiments.

We present the GMLS method through the lens of derivatives of MLS approximants as we feel this makes the analog to RBF-FD clearer, it is also closer to the description from [11]. Other derivations of GMLS are based on weighted least squares approximants of general linear functionals given at some set of points, e.g. [36–38]. However, both techniques produce the same result in the end [37]. For a more thorough discussion of MLS approximants, see for example [39, ch. 22] and the references therein.

3.1. Approximating the metric terms

The metric terms are approximated from an MLS reconstruction of the Monge patch of \mathcal{M} centered at each target point \mathbf{x}_i using a local stencil of n_i points $\mathbf{X}^i \subset \mathbf{X}$. This procedure is illustrated in Figure 2 and can be described as follows. First, the stencil \mathbf{X}^i is expressed in the form of (2) (i.e., $(\hat{x}_j, \hat{y}_j, f_j)$, $j \in \sigma^i$), where (\hat{x}_j, \hat{y}_j) are the coordinates for the stencil points in $T_{\mathbf{x}_i}\mathcal{M}$, and $f_j = f(\hat{x}_j, \hat{y}_j)$ are samples of the surface as viewed from the $\hat{x}\hat{y}$ -plane. These can be computed explicitly as

$$\begin{bmatrix} \hat{x}_j \\ \hat{y}_j \\ f_j \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{\xi}_i^1 & \boldsymbol{\xi}_i^2 & \boldsymbol{\eta}_i \end{bmatrix}^T}_{R_i^T} (\mathbf{x}_j - \mathbf{x}_i), \quad (8)$$

where $\boldsymbol{\xi}_i^1$ and $\boldsymbol{\xi}_i^2$ are orthonormal vectors that span $T_{\mathbf{x}_i}\mathcal{M}$ and $\boldsymbol{\eta}_i$ is the unit normal to \mathcal{M} at \mathbf{x}_i . To simplify the notation that follows, we let $\hat{\mathbf{x}}_j = (\hat{x}_j, \hat{y}_j)$ and $\hat{\mathbf{X}}^i = \{\hat{\mathbf{x}}_j\}_{j \in \sigma^i}$ denote the projection of the stencil \mathbf{X}^i to $T_{\mathbf{x}_i}\mathcal{M}$. Note that for convenience in what comes later we have shifted the coordinates so that the center of the projected stencil is $\hat{\mathbf{x}}_i = (0, 0)$.

In the second step, the approximate Monge patch at \mathbf{x}_i is constructed from a MLS approximant of the data $(\hat{\mathbf{x}}_j, f_j)$, $j \in \sigma^i$, which can be written as

$$q(\hat{\mathbf{x}}) = \sum_{k=1}^L b_k(\hat{\mathbf{x}}) p_k(\hat{\mathbf{x}}), \quad (9)$$

where $\{p_1, \dots, p_L\}$ is a basis for \mathbb{P}_ℓ^2 (the space of bivariate polynomials of degree ℓ) and $L = \dim(\mathbb{P}_\ell^2) = (\ell+1)(\ell+2)/2$ is the dimension of this space. The coefficients $b_k(\hat{\mathbf{x}})$ of the approximant are determined from the data according to the weighted least squares problem

$$\underline{b}^*(\hat{\mathbf{x}}) = \underset{\underline{b} \in \mathbb{R}^L}{\operatorname{argmin}} \sum_{j \in \sigma^i} w_\rho(\hat{\mathbf{x}}_j, \hat{\mathbf{x}}) (q(\hat{\mathbf{x}}_j) - f_j)^2 = \underset{\underline{b} \in \mathbb{R}^L}{\operatorname{argmin}} \|W_\rho(\hat{\mathbf{x}})^{1/2} (P\underline{b} - \underline{f})\|_2^2, \quad (10)$$

where $w_\rho : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^{\geq 0}$ is a weight kernel that depends on a support parameter ρ , $W_\rho(\hat{\mathbf{x}})$ is the $n_i \times n_i$ diagonal matrix $W_\rho(\hat{\mathbf{x}}) = \operatorname{diag}(w_\rho(\hat{\mathbf{x}}_j, \hat{\mathbf{x}}))$, and P is the $n_i \times L$ Vandermonde-type matrix

$$P = [p_1(\hat{\mathbf{x}}_j) \quad p_2(\hat{\mathbf{x}}_j) \quad \cdots \quad p_L(\hat{\mathbf{x}}_j)], \quad j \in \sigma^i \quad (11)$$

Here we use underlines to denote vectors (i.e., \underline{b} and \underline{f} denote vectors containing coefficients and data from (10), respectively). Note that the coefficients \underline{b}_k depend on $\hat{\mathbf{x}}$ because the kernel w_ρ depends on $\hat{\mathbf{x}}$ (this gives origin to the term “moving” in MLS). We discuss the selection of the stencils and weighting kernel below, but for now it is assumed that $n_i > L$ and \mathbf{X}^i is unisolvant on the space \mathbb{P}_ℓ^2 (i.e., P is full rank), so that (10) has a unique solution.

The MLS approximant q is used in place of f in the Monge patch (2) and it is used to approximate the metric terms in (5)–(7). To compute these terms, various derivatives need to be approximated at the projected stencil center $\hat{\mathbf{x}}_i$. Considering, for example, $\partial_{\hat{x}} q$, the approximation is computed as follows:

$$\partial_{\hat{x}} q|_{\hat{\mathbf{x}}_i} \approx \sum_{k=1}^L b_k^*(\hat{\mathbf{x}}_i) \partial_{\hat{x}} (p_k(\hat{\mathbf{x}}))|_{\hat{\mathbf{x}}_i}, \quad (12)$$

where $b_k^*(\hat{\mathbf{x}}_i)$ come from (10) with $\hat{\mathbf{x}} = \hat{\mathbf{x}}_i$. Other derivatives of metric terms in (5)–(7) are approximated in a similar way to (12). We note if the standard monomial basis is used for $\{p_1, \dots, p_L\}$, then by centering the projected stencil in (8) about the origin, only one of the derivatives of p_k in (12) is non-zero when evaluated at $\hat{\mathbf{x}}_i$.

Note that (12) is only an approximation of $\partial_{\hat{x}} q$ because it does not include the contribution of $\partial_{\hat{x}}(b_k^*(\hat{\mathbf{x}}))|_{\hat{\mathbf{x}}_i}$. This approximation is referred to as a “diffuse derivative” in the literature and is equivalent to the GMLS formulation of approximating derivatives [37]. The term “GMLS derivatives” is preferred over “diffuse derivatives” to describe (12), since the approximation is not diffuse or uncertain and has the same order of accuracy as the approximations that include the derivatives of the weight kernels [40].

3.2. Approximating SDOs

The procedure for approximating any of the SDOs in (5)–(7) is similar to the one for approximating the metric terms, but for this task we are interested in computing stencil weights as in (1) instead of the value of a derivative at a point. Since these SDOs involve computing various partial derivatives with respect to \hat{x} and \hat{y} , we can use (12) as a starting point for generating these stencil weights. If $\{u_j\}_{j \in \sigma^i}$ are samples of a function u over the projected stencil $\hat{\mathbf{X}}^i$, then we can again approximate $\partial_{\hat{x}} u|_{\hat{\mathbf{x}}=\hat{\mathbf{x}}_i}$ using (12), with $b_k^*(\hat{\mathbf{x}}_i)$ defined in terms of the samples of u . To write this in stencil form we note that (12) can be written using vector inner products as

$$\partial_{\hat{x}} u|_{\hat{\mathbf{x}}_i} \approx \partial_{\hat{x}} q|_{\hat{\mathbf{x}}_i} \approx \underbrace{\begin{bmatrix} \partial_{\hat{x}} p_1|_{\hat{\mathbf{x}}_i} & \cdots & \partial_{\hat{x}} p_L|_{\hat{\mathbf{x}}_i} \end{bmatrix}}_{(\partial_{\hat{x}} \underline{p}(\hat{\mathbf{x}}_i))^T} \underline{b}^*(\hat{\mathbf{x}}_i) = \underbrace{\begin{bmatrix} c_1^i & \cdots & c_{n^i}^i \end{bmatrix}}_{(\underline{c}_{\hat{x}}^i)^T} \underline{u}, \quad (13)$$

where we have substituted the solution of $\underline{b}^*(\hat{\mathbf{x}}_i)$ in (10) to obtain the term in the last equality. Using the normal equation solution for $\underline{b}^*(\hat{\mathbf{x}}_i)$, the stencil weights $\underline{c}_{\hat{x}}^i$ can be expressed as

$$\underline{c}_{\hat{x}}^i = W_{\rho}(\hat{\mathbf{x}}_i) P (P^T W_{\rho}(\hat{\mathbf{x}}_i) P)^{-1} (\partial_{\hat{x}} \underline{p}(\hat{\mathbf{x}}_i)). \quad (14)$$

This is typically computed using a QR factorization of $W_{\rho}(\hat{\mathbf{x}}_i)^{1/2} P$ to promote numerical stability.

Stencil weights $\underline{c}_{\hat{y}}^i$, $\underline{c}_{\hat{x}\hat{x}}^i$, $\underline{c}_{\hat{x}\hat{y}}^i$, and $\underline{c}_{\hat{y}\hat{y}}^i$ for the other derivative operators appearing in (5)–(7) can be computed in a similar manner for each stencil $\hat{\mathbf{X}}^i$, $i = 1, \dots, N$. These can then be combined together with the approximate metric terms to define the weights $\{c_{ij}\}$ in (1) for any of the SDOs in (5)–(7).

3.3. Choosing the stencils and weight kernel

As discussed in Section 2.1, we use Algorithm 1 to choose the stencil weights. For the initial stencil size, we use $L = \dim(\mathbb{P}_{\ell}^2)$. The radius factor τ controls the size of the stencil, with larger τ resulting in larger stencils, and we experiment with this parameter in the numerical results section.

There are many choices for the weight kernel w_{ρ} in (10). Typically, a single radial kernel is used to define w_{ρ} as $w_{\rho}(\mathbf{x}, \mathbf{y}) = w(\|\mathbf{x} - \mathbf{y}\|/\rho)$, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\|\cdot\|$ is the standard Euclidean norm for \mathbb{R}^d . In this work, we use the same family of compactly supported radial kernels as [12, 14] and implemented in [35]:

$$w_{\rho}(\mathbf{x}, \mathbf{y}) = \left(1 - \frac{\|\mathbf{x} - \mathbf{y}\|}{\rho}\right)_+^{2m}, \quad (15)$$

where m is a positive integer and $(\cdot)_+$ is the positive floor function. These C^0 kernels have support over the ball of radius ρ centered at \mathbf{y} . While smoother kernels can be used such as Gaussians, splines, or Wendland kernels [39], we have not observed any significant improvement in the accuracy of GMLS derivative approximations with smoother kernels. In general, proofs on how the choice of kernels effects the accuracy of GMLS approximations have yet to be found.

Finally, we note that the support parameter ρ is chosen on a per stencil basis and is set equal to τh_{\max} from Algorithm 1. Picking an optimal value for τ to minimize the approximation error

is a difficult problem. In general, the optimal value depends locally on the point set and the function (or its derivative) being approximated [41]. While there are some algorithms that attempt to approximate this value to minimize the local pointwise error (e.g., [41, 42]), they are computationally expensive. Typically, one chooses a single $\tau > 1$ such that the minimization problem (10) is well-posed (i.e., P is full rank). This can be easily monitored for each stencil to adjust τ appropriately.

3.4. Approximating the tangent space

When the tangent space $T_{\mathbf{x}_i} \mathcal{M}$ is unknown, a coarse approximation to it can be computed for each stencil \mathbf{X}^i using principal component analysis [11]. In this method, one computes the eigenvectors of the covariance matrix $\bar{\mathbf{X}}_i \bar{\mathbf{X}}_i^T$, where $\bar{\mathbf{X}}_i$ is the 3-by- n_i matrix formed from the stencil points \mathbf{X}^i centered about their mean. The two dominant eigenvectors of this matrix are taken as a coarse approximation to $T_{\mathbf{x}_i} \mathcal{M}$ and the third is taken as a coarse approximation to the normal to \mathcal{M} at \mathbf{x}_i ; we denote these by $\tilde{\xi}_i^1$, $\tilde{\xi}_i^2$, and $\tilde{\eta}_i$, respectively. Next, an approximate Monge patch parameterization is formed with respect to this approximate tangent space using MLS following the same procedure outlined at the beginning of Section 3.1. This procedure is illustrated in Figure 3 (a), where the coarse approximate tangent plane is given in yellow. A refined approximation to the true tangent plane and normal at the stencil center \mathbf{x}_c can be obtained by computing the tangent plane and normal to the MLS approximant of the Monge patch at \mathbf{x}_c ; this plane is given in cyan in Figure 3 (a). Once this plane is computed, a new Monge patch parameterization with respect to this refined tangent plane approximation is formed, as illustrated in Figure 3 (b). This procedure is repeated for each stencil \mathbf{X}^i and the refined tangent space computed for each stencil is used in the procedure described in Section 3.1 for approximating the metric terms.

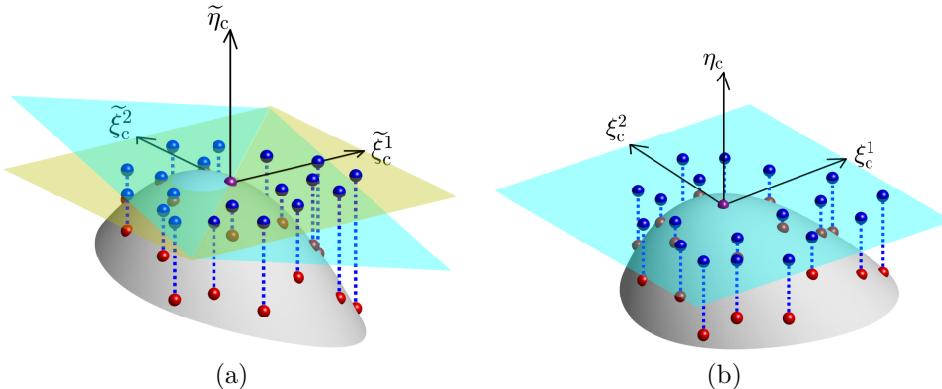


Figure 3: Illustration of the tangent plane correction method. (a) Monge patch parameterization for a local neighborhood of a regular surface \mathcal{M} (in gray) in 3D using a coarse approximation to the tangent plane (in yellow) at the center of the stencil \mathbf{x}_c and the refined approximation to the tangent plane (in cyan). (b) Same as (a), but for the Monge patch with respect to the refined tangent plane. The red spheres denote the points from the stencil and the blue spheres mark the projection of the stencil to the (a) coarse and (b) refined tangent planes. The coarse and refined approximations to the tangent and normal vectors are given as ξ_c^1 , ξ_c^2 , and η_c , respectively, with tildes on these variables denoting the coarse approximation.

4. RBF-FD using the tangent plane

As discussed in the introduction, there are several RBF-FD methods that have been developed over the past ten years for approximating SDOs. We use the one based on the tangent plane method for formulating SDOs and PHS+Poly interpolants for approximating the derivatives that appear in this formulation. The subsections below provide a detailed overview of these respective techniques.

4.1. Tangent plane method

The tangent plane method similarly uses local coordinates for the surface in the tangent plane formed at each $\mathbf{x}_i \in \mathbf{X}$, but unlike the method from Section 3.1, it does not use approximations to

the metric terms. It instead approximates the SDOs at each \mathbf{x}_i using the standard definitions for the derivatives in the tangent plane. So, using local coordinates (2) about \mathbf{x}_i , the surface gradient for the tangent plane method is taken as

$$\nabla_{\mathcal{M}} = R_i \begin{bmatrix} \partial_{\hat{x}} \\ \partial_{\hat{y}} \\ 0 \end{bmatrix}, \quad (16)$$

and the surface divergence of a smooth vector $\mathbf{u} \in T_{\mathbf{x}_i} \mathcal{M}$ is taken as

$$\nabla_{\mathcal{M}} \cdot \mathbf{u} = [\partial_{\hat{x}} \quad \partial_{\hat{y}} \quad 0] R_i^T \mathbf{u}, \quad (17)$$

where R_i is the rotation matrix given in (8). Similarly, the surface Laplacian in the tangent plane method is

$$\Delta_{\mathcal{M}} = \partial_{\hat{x}\hat{x}} + \partial_{\hat{y}\hat{y}}. \quad (18)$$

We next show that if $T_{\mathbf{x}_i} \mathcal{M}$ is known exactly for each $\mathbf{x}_i \in \mathbf{X}$ and the point at which the SDOs are evaluated is \mathbf{x}_i , then the SDOs (16)–(18) are equivalent to the corresponding ones involving metric terms (5)–(7). This was shown indirectly in [31] for the surface Laplacian using the distributional definition of the surface Laplacian. Here we show the result follows explicitly for each surface differential operator (5)–(7) from the local coordinate formulation in Section 2.2.

The first step is to note that the vectors $\partial_{\hat{x}} \mathbf{f}|_{\hat{\mathbf{x}}_i}$ and $\partial_{\hat{y}} \mathbf{f}|_{\hat{\mathbf{x}}_i}$ from the Monge parameterization (2) are tangential to the $\hat{x}\hat{y}$ -plane and must therefore be orthogonal to the vector $[0 \ 0 \ 1]$. This implies $\partial_{\hat{x}} f = \partial_{\hat{y}} f = 0$ at $\hat{\mathbf{x}}_i$, which means the metric tensor (3) reduces to the identity matrix when evaluated at $\hat{\mathbf{x}}_i$. Using this result in (5) for $\hat{\nabla}_{\mathcal{M}}$ means that the surface gradient formula (5) is exactly (16) when evaluated at $\hat{\mathbf{x}}_i$. The equivalence of the surface divergence formulas (6) and (17) also follow immediately from this result.

The steps for showing the equivalence of the surface Laplacian operator are more involved. To simplify the notation in showing this result, we denote partial derivatives of f with subscripts. For the first step of this process, we substitute the explicit metric terms, $|g| = (1 + f_{\hat{x}}^2)(1 + f_{\hat{y}}^2) - (f_{\hat{x}} f_{\hat{y}})^2$, $g^{11} = (1 + f_{\hat{y}})/|g|$, $g^{12} = g^{21} = -(f_{\hat{x}})(f_{\hat{y}})/|g|$, and $g^{22} = (1 + f_{\hat{x}})/|g|$, into (7) and expand the derivatives. Next, we simplify to obtain the following formula:

$$\begin{aligned} \Delta_{\mathcal{M}} = & \frac{1}{(f_{\hat{x}}^2 + f_{\hat{y}}^2 + 1)^2} \left((f_{\hat{y}} f_{\hat{x}\hat{y}} (1 + 2f_{\hat{x}}^2 + f_{\hat{y}}^2) - (f_{\hat{x}} f_{\hat{x}\hat{x}} + f_{\hat{y}} f_{\hat{x}\hat{y}})(1 + f_{\hat{y}}^2) - f_{\hat{x}} f_{\hat{y}\hat{y}} (1 + f_{\hat{x}}^2)) \partial_{\hat{x}} + \right. \\ & \left. (f_{\hat{x}} f_{\hat{x}\hat{y}} (1 + 2f_{\hat{y}}^2 + f_{\hat{x}}^2) - (f_{\hat{y}} f_{\hat{y}\hat{y}} + f_{\hat{x}} f_{\hat{x}\hat{y}})(1 + f_{\hat{x}}^2) - f_{\hat{y}} f_{\hat{x}\hat{x}} (1 + f_{\hat{y}}^2)) \partial_{\hat{y}} \right) + \\ & g^{11} \partial_{\hat{x}\hat{x}} + 2g^{12} \partial_{\hat{x}\hat{y}} + g^{22} \partial_{\hat{y}\hat{y}} \end{aligned}$$

Using $f_{\hat{x}} = f_{\hat{y}} = g^{12} = 0$ and $g^{11} = g^{22} = 1$ at $\hat{\mathbf{x}}_i$, this formula reduces to (18).

4.2. Approximating the SDOs

Since the tangent plane method does not require computing approximations to any metric terms, we only need to describe the RBF-FD method for approximating the derivatives that appear in (16)–(18). We derive this method from derivatives of interpolants over the projected stencils for each point $\mathbf{x}_i \in \mathbf{X}$ using the same notation as Section 3 and we assume that the tangent space is known. A method for approximating the tangent space also using RBF-FD is discussed in Section 4.4.

Let $\{u_j\}_{j \in \sigma^i}$ be samples of some function u over the projected stencil $\hat{\mathbf{X}}^i = \{\hat{\mathbf{x}}_j\}_{j \in \sigma^i}$. The PHS+Poly interpolant to this data can be written

$$s(\hat{\mathbf{x}}) = \sum_{j=1}^{n_i} a_j \phi(\|\hat{\mathbf{x}} - \hat{\mathbf{x}}_{\sigma_j^i}\|) + \sum_{k=1}^L b_k p_k(\hat{\mathbf{x}}), \quad (19)$$

where $\phi(r) = r^{2\kappa+1}$ is the PHS kernel of order $2\kappa+1$, $\kappa \in \mathbb{Z}^{\geq 0}$, σ_j^i is the j th index in σ^i , $\|\cdot\|$ denotes the Euclidean norm, and $\{p_1, \dots, p_L\}$ are a basis for \mathbb{P}_ℓ^2 . The expansion coefficients are determined by the n_i interpolation conditions and L additional moment conditions:

$$s(\hat{\mathbf{x}}_j) = u_j, \quad j \in \sigma^i \quad \text{and} \quad \sum_{j=1}^{n_i} a_j p_k(\hat{\mathbf{x}}_{\sigma_j^i}) = 0, \quad k = 1, \dots, L. \quad (20)$$

These conditions can be written as the following $(n_i + L) \times (n_i + L)$ linear system

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} = \begin{bmatrix} \underline{u} \\ \mathbf{0} \end{bmatrix}, \quad (21)$$

where $A_{jk} = \|\hat{\mathbf{x}}_{\sigma_j^i} - \hat{\mathbf{x}}_{\sigma_k^i}\|^{2\kappa+1}$ ($j, k = 1, \dots, n_i$) and P is the same Vandermonde-type matrix given in (11). The PHS parameter κ controls the smoothness of the kernel and should be chosen such that $0 \leq \kappa \leq \ell$. With this restriction on κ , it can be shown that A is positive definite on the subspace of vectors in \mathbb{R}^n satisfying the L moment conditions in (20) [36]. Hence, if the stencil points \mathbf{X}^i are such that $\text{rank}(P) = L$ (i.e., they are unisolvant on the space \mathbb{P}_ℓ^2), then the system (21) is non-singular and the PHS+Poly interpolant is well-posed. Note that this is the same restriction on \mathbf{X}^i for the MLS problem (10) to have a unique solution.

The stencil weights for approximating any of the derivatives appearing in the SDOs (16)–(18) can be obtained from differentiating the PHS+Poly interpolant (19). Without loss of generality, consider approximating the operator $\partial_{\hat{x}}$ over the stencil $\hat{\mathbf{X}}^i$. Using vector inner products as in (13), the stencil weights for this operator are determined from the approximation

$$\partial_{\hat{x}} u|_{\hat{\mathbf{x}}_i} \approx \partial_{\hat{x}} s|_{\hat{\mathbf{x}}_i} = [\partial_{\hat{x}} \underline{\phi}(\hat{\mathbf{x}}_i) \quad \partial_{\hat{x}} \underline{p}(\hat{\mathbf{x}}_i)]^T \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix}.$$

where $\partial_{\hat{x}} \underline{\phi}(\hat{\mathbf{x}}_i)$ and $\partial_{\hat{x}} \underline{p}(\hat{\mathbf{x}}_i)$ are vectors containing the entries $\partial_{\hat{x}} \|\hat{\mathbf{x}} - \hat{\mathbf{x}}_{\sigma_j^i}\|^{2\kappa+1}|_{\hat{\mathbf{x}}_i}$, $j = 1, \dots, n_i$, and $\partial_{\hat{x}} p_k(\hat{\mathbf{x}})|_{\hat{\mathbf{x}}_i}$, $k = 1, \dots, L$, respectively. Using (21) in the preceding expression gives the stencil weights as the solution to the following linear system

$$\begin{bmatrix} A & P \\ P^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{c}_{\hat{x}}^i \\ \underline{\lambda} \end{bmatrix} = \begin{bmatrix} \partial_{\hat{x}} \underline{\phi}(\hat{\mathbf{x}}_i) \\ \partial_{\hat{x}} \underline{p}(\hat{\mathbf{x}}_i) \end{bmatrix}, \quad (22)$$

where the entries in $\underline{\lambda}$ are not used as part of the weights. Note that this description is equivalent to applying $\partial_{\hat{x}}$ to the PHS+Poly cardinal basis functions defined over the stencil and evaluating them at $\hat{\mathbf{x}}_i$ [3].

Stencil weights \underline{c}_y^i , $\underline{c}_{\hat{x}\hat{y}}^i$, $\underline{c}_{\hat{x}\hat{y}}^i$, and $\underline{c}_{\hat{y}\hat{y}}^i$ for the other partial derivatives can be computed in an analogous way for each stencil $\hat{\mathbf{X}}^i$, $i = 1, \dots, N$. These can then be combined together to define the weights $\{c_{ij}\}$ in (1) for any of the SDOs in (16)–(18).

4.3. Choosing the stencils and PHS order

Similar to GMLS, we use Algorithm 1 to choose the stencils and also use the same initial stencil size of $n = L$ for this algorithm. The parameter κ used to determine the PHS order should be chosen with an upper bound of $\kappa \leq \ell$ (so that (22) is well posed) and a lower bound such that the derivatives of the PHS kernels make sense for whatever operator the RBF-FD stencils are being used to approximate. In this work we use $\kappa = \ell$ as we have found that this choice works well for approximating various SDOs across a wide range of surfaces. Choosing $\kappa < \ell$ can be useful for improving the conditioning of the system (22) and for reducing Runge Phenomenon-type edge effects in RBF-FD approximations near boundaries [43].

4.4. Approximating the tangent space

If $T_{\mathbf{x}_i} \mathcal{M}$ is unknown for any $\mathbf{x}_i \in \mathbf{X}$, then we use a similar procedure to the one discussed for GMLS in Section 3.4 (and illustrated in Figure 3) to approximate it. The difference for RBF-FD

is that instead of using an MLS reconstruction of the Monge patch parameterization formed from the coarse tangent plane approximation at each \mathbf{x}_i , we use the PHS+Poly interpolant (19) for the reconstruction. The refined approximation to the tangent plane at each \mathbf{x}_i is then obtained from derivatives of the PHS+Poly interpolant of the Monge patch for stencil \mathbf{X}^i . We note that this approach is new amongst the different tangent plane methods, as previous approaches assumed the tangent space was computed by some other, possibly unrelated techniques, and not directly from the stencils (e.g., [13, 22, 26]). By combining this technique with the tangent plane method, we arrive at the first comprehensive PHS+Poly RBF-FD framework for approximating SDOs on point cloud surfaces.

5. Theoretical comparison of GMLS and RBF-FD

In this section, we make comparisons of the GMLS and RBF-FD methods in terms of some of their theoretical properties, including the different approaches in formulating SDOs, the parameters of the approximations, and the computational cost.

One of the main differences between the GMLS and RBF-FD approaches is that the former uses the local coordinate method to formulate SDOs, while the latter uses the tangent plane method. As shown in Section 4.1 these methods are equivalent if the tangent space for \mathcal{M} is known for each $\mathbf{x}_i \in X$ and the SDOs are evaluated at the stencil center \mathbf{x}_i . However, the GMLS method does not take advantage of this and instead includes metric terms in the formulation. These metric terms are approximated with the same order of accuracy as the GMLS approximation of the derivatives (see below), so that these errors are asymptotically equivalent as the spacing of the points in the stencil goes to zero. When the tangent space is unknown, both methods again approximate it to the same order of accuracy as their respective approximations of the derivatives.

The GMLS and RBF-FD methods each feature the parameter ℓ , which controls the degree of the polynomials used in the approximation. For a given ℓ , the formulas for either method are exact for all bivariate polynomials of degree ℓ in the tangent plane formed by the stencil center \mathbf{x}_i . Unsurprisingly, ℓ also effects the local accuracy of the formulas in the tangent plane with increasing ℓ giving higher orders of accuracy for smooth problems; see [11, 40] for a study of the accuracy of GMLS and [44, 45] for RBF-FD. The order of accuracy of both methods depends on the highest order derivative appearing in the SDOs, and is generally ℓ if the derivative order is 1 and $\ell - 1$ if the derivative order is two. However, for certain quasi-uniform point clouds with symmetries, the order has been shown to be ℓ for GMLS applied to second order operators like the surface Laplacian [11].

The computational cost of the methods can be split between the setup cost and the evaluation cost. The setup cost depends on ℓ and n_i (which depends on τ). For each stencil \mathbf{X}^i , the dominant setup cost of GMLS comes from solving the $n_i \times L$ system (14), while the dominant cost for RBF-FD comes from solving the $(n_i + L) \times (n_i + L)$ system (22). We use QR factorization to solve the GMLS system and LU factorization to solve the RBF-FD system, which gives the following (to leading order):

$$\text{Setup cost GMLS} \sim 2 \sum_{i=1}^N n_i L^2 \quad \text{and} \quad \text{Setup cost RBF-FD} \sim \frac{2}{3} \sum_{i=1}^N (n_i + L)^3. \quad (23)$$

The stencil sizes depend on ℓ and τ , and for quasi-uniform point clouds \mathbf{X} , n_i is typically some multiple γ of L . In this case, the setup cost of RBF-FD is higher by approximately $(1 + \gamma)^3 / (3\gamma)$. We note that the setup procedures for both methods are an embarrassingly parallel process, as each set of stencil weights can be computed independently of every other set. The evaluation costs of both methods are the same and can be reduced to doing sparse matrix-vector products. So, for a scalar SDO like the surface Laplacian

$$\text{evaluation cost GMLS \& RBF-FD:} \sim 2 \sum_{i=1}^N n_i. \quad (24)$$

If the ℓ and τ parameters remain fixed so that size of the stencils remain fixed as N increases, then both the setup and evaluation cost are linear in N .

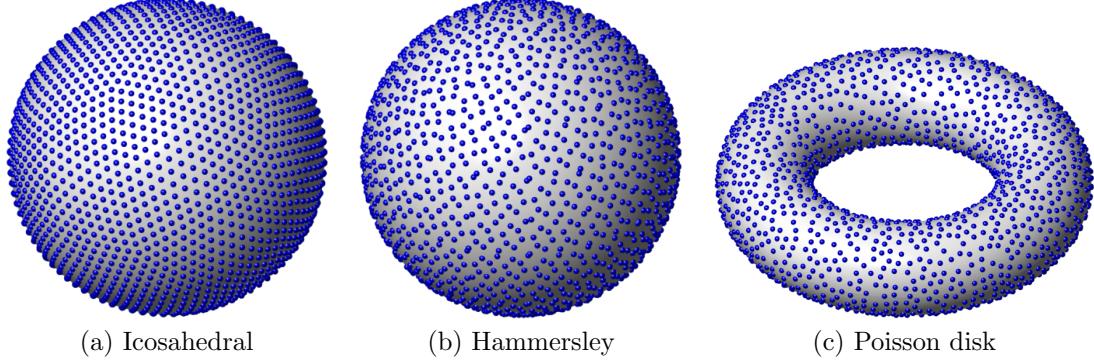


Figure 4: Examples from the three node sets considered in the numerical experiments: (a) $N = 2562$, (b) $N = 2048$, (c) $N = 2038$.

6. Numerical comparison of GMLS and RBF-FD

We perform a number of numerical experiments comparing GMLS and RBF-FD for approximating the gradient, divergence, and Laplacian on two topologically distinct surfaces: the unit two sphere \mathbb{S}^2 and the torus defined implicitly as

$$\mathbb{T}^2 = \left\{ (x, y, z) \in \mathbb{R}^3 \mid (1 - \sqrt{x^2 + y^2})^2 + z^2 - 1/9 = 0 \right\}. \quad (25)$$

For the experiments with the sphere, we consider two different node sets \mathbf{X} , icosahedral and Hammersley; see Figure 4 (a) & (b) for examples. The first are highly structured, quasi-uniform points that are commonly used in numerical weather prediction [2, 4]. They have also been used in other studies on GMLS [12] and RBF-FD [16] methods on the sphere. Hammersley are low discrepancy point sequences commonly used in Monte-Carlo integration on the sphere [46]. They are highly unstructured with some points that nearly overlap. For the experiments on the torus, we use Poisson disk points generated using the weighted sample elimination (WSE) algorithm [47]. These points are also unstructured, but are quasi-uniform; see Figure 4 (c) for an example. They have also previously been used in studies on GMLS and RBF-FD methods [26]. Convergence results with other point sets can be found in the PhD thesis of the first author [48].

Error estimates for GMLS and RBF-FD typically require the nodes to be quasi-uniform in the sense that the average spacing between the points h (or more generally the mesh-norm) decreases like $h \sim N^{-1/2}$ [36, 39]. As mentioned above, the icosahedral and Poisson disk node sets have this property and are thus well-suited for numerically testing convergence rates of GMLS and RBF-FD methods with increasing N (i.e., convergence as the density of the sampling of the surfaces increases). Specifically, we experimentally examine the algebraic convergence rates β versus \sqrt{N} , assuming the error behaves like $\mathcal{O}(N^{-\beta/2})$, and include results for polynomial degrees $\ell = 2, 4$, and 6 . The Hammersley node sets are well suited to testing how stable the methods are to stencils with badly placed points. Since these nodes have low discrepancy over the sphere, it also makes sense to test convergence in a similar manner to the other point sets. The exact values of N used in the experiments for each of the node sets are as follows. For Icosahedral, $N = 10242, 40962, 163842, 655362$, and for Hammersley and Poisson disk: $N = 8153, 32615, 130463, 521855$.

All RBF-FD results that follow were obtained from a Python implementation of the method that only utilizes the scientific computing libraries SciPy and NumPy. For the GMLS results, we use the software package Compadre [35], which is implemented in C++ and uses the portable performance library Kokkos.

6.1. Convergence comparison: Sphere

We base all the convergence comparisons for the sphere on the following function consisting of a random linear combination of translates of 50 Gaussians of different widths on the sphere:

$$u(\mathbf{x}) = \sum_{j=1}^{50} d_j \exp(-\gamma_j \|\mathbf{x} - \mathbf{y}_j\|^2), \quad \mathbf{x}, \mathbf{y}_j \in \mathbb{S}^2, \quad (26)$$

where \mathbf{y}_j are the centers and are randomly placed on the sphere, and d_j & γ_j are sampled from the normal distributions $\mathcal{N}(0, 1)$ & $\mathcal{N}(15, 4)$, respectively. This function has also been used in other studies on RBF-FD methods [15]. We use samples of u in the surface gradient tests and measure the error against the exact surface gradient, which can be computed using the Cartesian gradient ∇ in \mathbb{R}^3 as $\nabla_{\mathcal{M}} u = \nabla u - \boldsymbol{\eta}(\boldsymbol{\eta} \cdot \nabla u)$, where $\boldsymbol{\eta}$ is the unit outward normal to \mathbb{S}^2 [49] (which is just \mathbf{x}). Applying this to (26) gives

$$\nabla_{\mathcal{M}} u = 2 \sum_{j=1}^{50} d_j \gamma_j (\mathbf{y}_j - \mathbf{x}(\mathbf{x} \cdot \mathbf{y}_j)) \exp(-\gamma_j \|\mathbf{x} - \mathbf{y}_j\|^2). \quad (27)$$

We use samples of this field in the surface divergence tests. Since $\nabla_{\mathcal{M}} \cdot \nabla_{\mathcal{M}} u = \Delta_{\mathcal{M}} u$, we compare the errors in this test against the exact surface Laplacian of u , which can be computed using the results of [3] as

$$\Delta_{\mathcal{M}} u = - \sum_{j=1}^{50} d_j \gamma_j (4 - \|\mathbf{x} - \mathbf{y}_j\|^2 (2 + \gamma_j (4 - \|\mathbf{x} - \mathbf{y}_j\|^2))) \exp(-\gamma_j \|\mathbf{x} - \mathbf{y}_j\|^2).$$

We also use this in the tests of the surface Laplacian using samples of u .

For all these tests, we set radius factor τ in the stencil selection Algorithm 1 to 1.5, which gave good results for both RBF-FD and GMLS (see the next section for some results on the effects of increasing τ). While the exact tangent space for the sphere is trivially determined, we approximate it in all the results using the methods discussed in the Section 3.4 for GMLS and Section 4.4 for RBF-FD. These approximations are done with the same parameters for approximating the different SDOs to keep the asymptotic orders of accuracy comparable. Although not included here, we did experiments with the exact tangent space and obtained similar results to those presented here.

Figures 5 and 6 display the convergence results for GMLS and RBF-FD as a function of N . Each figure is for a different point set type and contains the results for approximating the surface gradient, divergence, and Laplacian in both the relative two- and max-norms and for different polynomial degrees ℓ . We see from all the results that the measured convergence rates for GMLS and RBF-FD are similar, but that RBF-FD gives lower errors for the same N and ℓ for approximating the surface gradient and divergence. This is also true for the surface Laplacian when $\ell = 4$ and $\ell = 6$, but not for $\ell = 2$. For this case, GMLS gives lower errors for the same N on the icosahedral nodes and about the same error for the Hammersley nodes. We also see from Figure 6 that both methods do not appear to be effected by stability issues associated with badly placed points in the stencils for the Hammersley nodes.

The measured convergence rates in the two-norm for the surface gradient and divergence approximations are close to the expected rates of ℓ for both point sets. However, when looking at the convergence rates of the surface Laplacian, we see from Figure 5 that the icosahedral nodes have higher rates than for the Hammersley nodes in Figure 6. These improved convergence rates have been referred to as superconvergence in the GMLS literature and rely on the point set being structured so that the stencils have certain symmetries [11]. When these symmetries do not exist, as is the case for the Hammersley nodes, the convergence rates for the surface Laplacian more closely follow the expected rates of $\ell - 1$.

6.2. Convergence comparison: Torus

The convergence comparisons on the torus are based on the target function

$$u(\mathbf{x}) = \frac{x}{8} (x^4 - 10x^2y^2 + 5y^4)(r^2 - 60z^2), \quad \mathbf{x} \in \mathbb{T}^2, \quad (28)$$

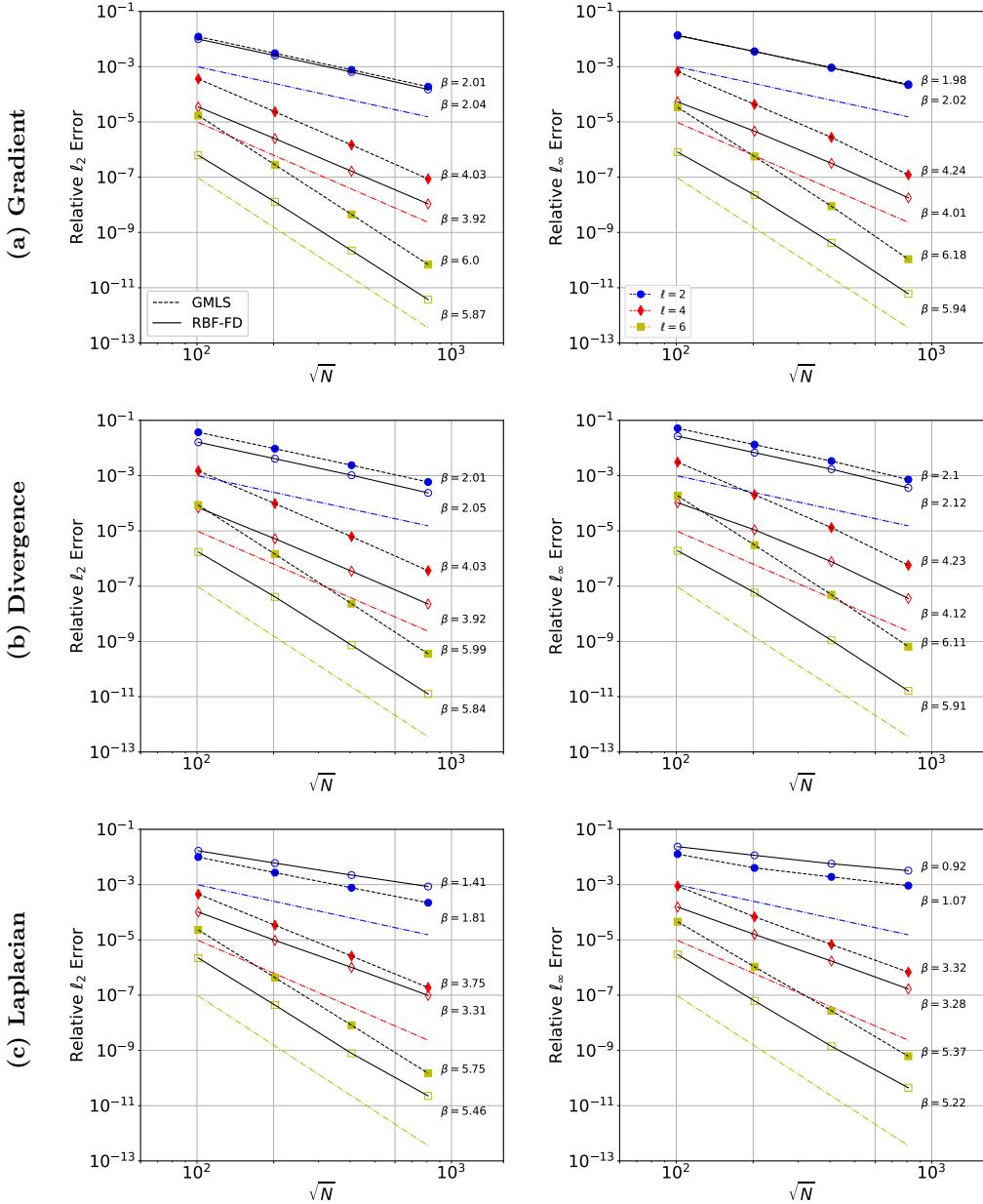


Figure 5: Convergence results for (a) surface gradient, (b) divergence, and (c) Laplacian on the sphere using icosahedral node sets. Errors are given in relative two-norms (first column) and max-norms (second column). Markers correspond to different ℓ : filled markers are GMLS and open markers are RBF-FD. Dash-dotted lines without markers correspond to 2nd, 4th, and 6th order convergence with $1/\sqrt{N}$. β are the measured order of accuracy computed using the lines of best fit to the last three reported errors.

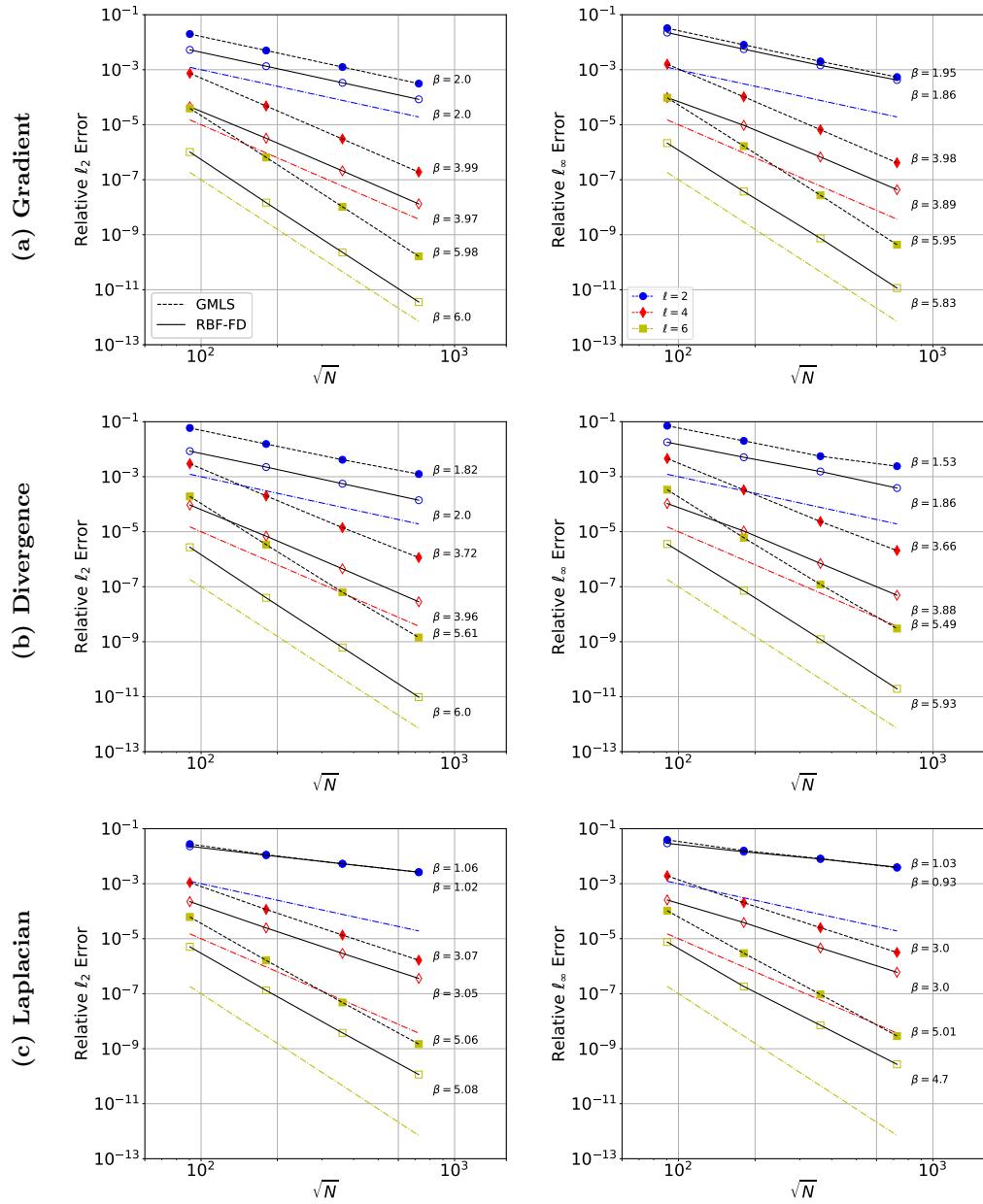


Figure 6: Same as Figure 5, but for the Hammersley nodes on the sphere.

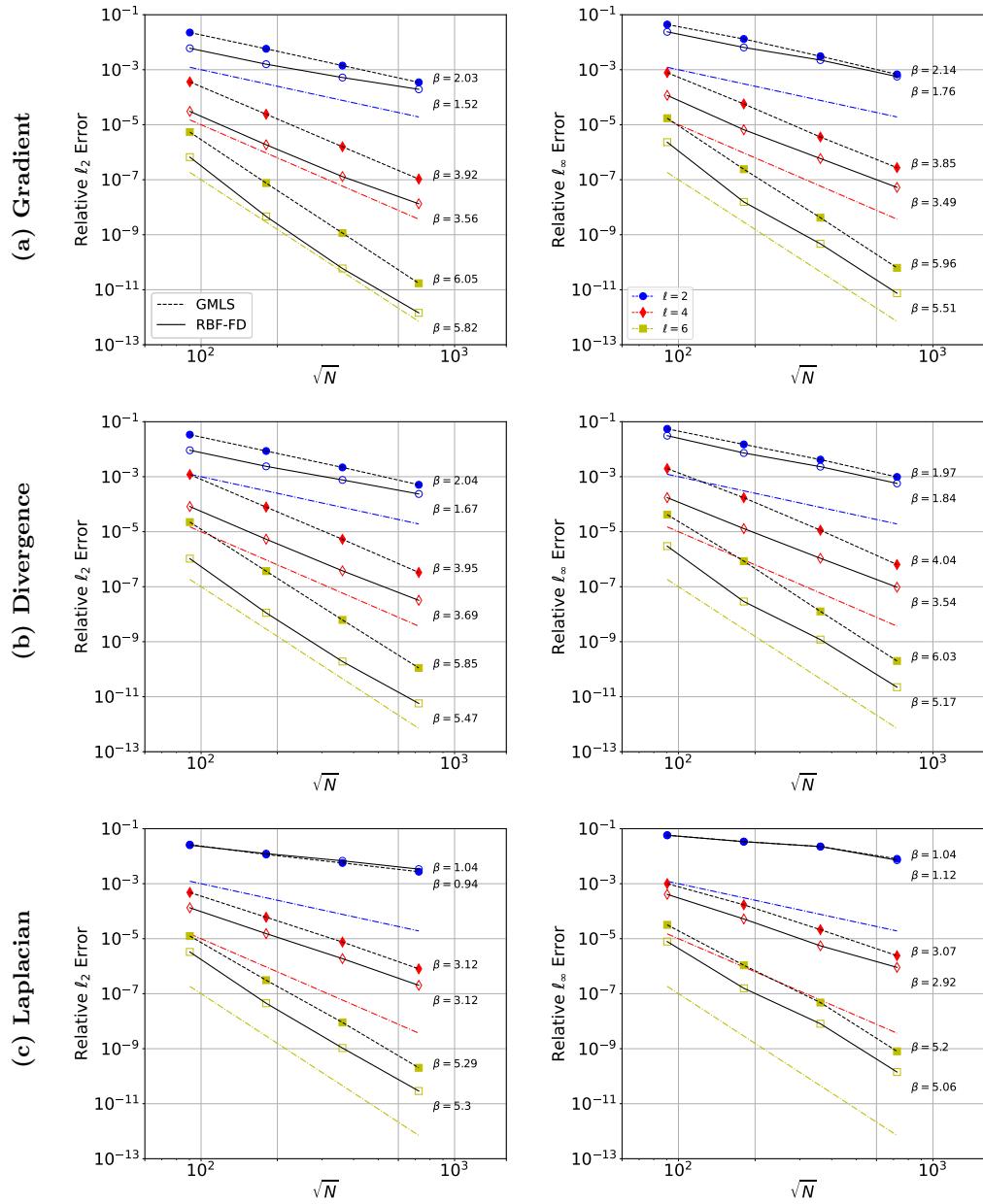


Figure 7: Same as Figure 5, but for torus using Poisson disk points.

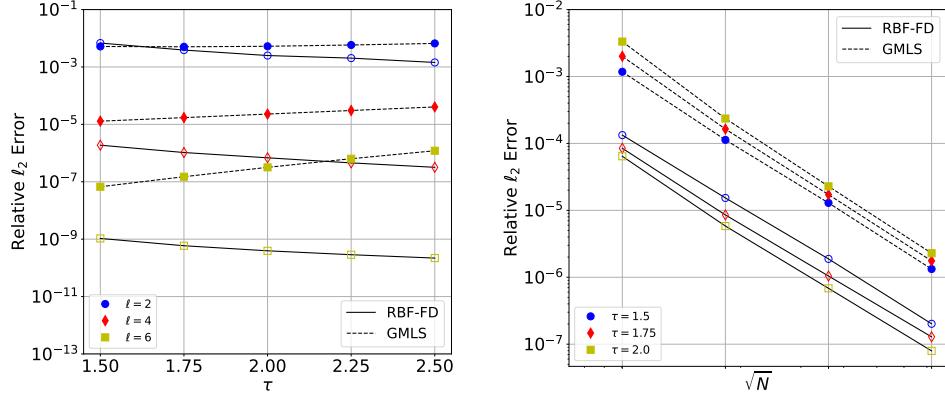


Figure 8: Relative two-norm errors of the surface Laplacian approximations as the stencil radius parameter τ varies. Left figure shows errors for several different values of τ and a fixed $N = 130463$. Right figure shows the convergence rates of the methods for different τ and a fixed $\ell = 4$.

where $r = \sqrt{x^2 + y^2}$. This function has also been used in other studies of RBF methods for surfaces [49]. As with the sphere example, the surface gradient of u can be computed as $\nabla_{\mathcal{M}}u = \nabla u - \boldsymbol{\eta}(\boldsymbol{\eta} \cdot \nabla u)$, where $\boldsymbol{\eta}$ is the unit outward normal to \mathbb{T}^2 , which can be computed from the implicit equation (25). The surface Laplacian of (28) is given in [49] as

$$\Delta_{\mathcal{M}}u(\mathbf{x}) = -\frac{3x}{8r^2}(x^4 - 10x^2y^2 + 5y^4)(10248r^4 - 34335r^3 + 41359r^2 - 21320r + 4000), \quad \mathbf{x} \in \mathbb{T}^2.$$

Similar to the sphere, we use samples of $\nabla_{\mathcal{M}}u$ in the tests of the divergence and compare the results with $\Delta_{\mathcal{M}}u$ above.

We first study the convergence rates with the stencil radius scaling $\tau = 1.5$ and approximate the tangent space, as we did with the sphere tests. Figure 7 displays the results for the surface gradient, divergence, and Laplacian. We see that errors for RBF-FD are again smaller than the errors for GMLS in almost all cases over the range of N tested. However, GMLS has a slightly higher convergence rates in the case of the surface gradient and divergence, but not for the Laplacian. Both methods have convergence rates that are close to the expected rates of ℓ for these surface gradient and divergence and $\ell - 1$ for the Laplacian.

Next we investigate how the approximation properties of the two methods change when τ is increased, which results in larger stencil sizes. We focus on approximating the surface Laplacian as similar results were found for the other SDOs. In the left plot of Figure 8, we show the relative two-norm errors of the approximations for a fixed N as τ varies from 1.5 to 2.5. We see that increasing τ has opposite effects on the two methods: the errors decrease for RBF-FD and increase with GMLS. We see similar results in the right plot of Figure 8, where we show the convergence of the methods with increasing N for different fixed values of τ (and ℓ fixed at 4). While the convergence rates do not appear to change with τ , the overall errors decrease for RBF-FD and increase for GMLS. It should be noted that the errors eventually increase for GMLS as τ decreases to 1 (which has been observed in other studies) and picking an optimal τ in an automated way is challenging (e.g. [41, 42]).

These results make sense when one considers the different types of approximations the methods are based on: RBF-FD is based on interpolation, while GMLS is based on least squares approximation. As the stencil sizes increase, RBF-FD has a larger approximation space consisting of more shifts of PHS kernels, which can reduce the errors [44]. However, GMLS has the same fixed approximation space of polynomials of degree ℓ regardless of the stencil size.

Finally, we compare the errors when the exact and approximate tangent spaces are used in the two methods. We focus only on the surface Laplacian and for $\ell = 4$ since similar results were obtained for the other operators and other ℓ . Table 1 shows the results for both methods. The approximate tangent spaces were computed using the methods from Sections 3.4 (GMLS) and 4.4 (RBF-FD) also using the polynomial degree $\ell = 4$. As discussed in Section 5, this choice is made

so that the tangent spaces are approximated with the same asymptotic order of accuracy as the approximation of the metric terms with GMLS. We see from the table that the differences between using the exact or the approximate tangent spaces for approximating the surface Laplacian is minor.

N	GMLS		RBF-FD	
	Exact	Approx.	Exact	Approx.
8153	4.7984e-04	4.8004e-04	1.3311e-04	1.3312e-04
32615	6.0457e-05	6.04654e-05	1.5321e-05	1.5322e-05
130463	7.5486e-06	7.5488e-06	1.8811e-06	1.8811e-06
521855	8.0158e-07	8.0159e-07	2.0177e-07	2.0176e-07

Table 1: Comparison of the relative ℓ_2 errors for the surface Laplacian on the torus using the exact tangent space for the torus and approximations to it based on the methods from Sections 3.4 (GMLS) and 4.4 (RBF-FD). In all cases, $\ell = 4$ and the points are based on Poisson disk sampling.

6.3. Efficiency comparison

The results in Section 6 demonstrate that RBF-FD and GMLS have similar asymptotic convergence rates for the same ℓ , but that RBF-FD can achieve lower errors for the same N and stencil sizes. In this section, we consider which of the methods are more computationally efficient in terms of error per computational cost. We examine both the efficiency when the setup costs are included and when just the evaluation costs are included, as measured by (23) and (24), respectively. We limit this comparison to $\tau = 1.5$, but note that it may be possible to tune this parameter to (marginally) optimize the efficiency of either method over this case. Figure 9 displays the results of this examination for the case of computing the surface Laplacian on the torus discretized with Poisson disk sampling. Similar results were obtained for other SDOs and for the sphere, so we omit them. We see from the figure that GMLS is more efficient when the setup costs are included, but that RBF-FD is more efficient when only evaluation costs are included. For problems where the point sets are fixed and approximations to a SDO are required to be performed multiple times—as occurs when solving a time-dependent surface PDEs—the setup costs are not as important as the evaluation costs since they are amortized across all time-steps. In this scenario RBF-FD is the more efficient method.

7. Concluding remarks

We presented a thorough comparison of the GMLS and RBF-FD methods for approximating the three most common SDOs: the gradient, divergence, and Laplacian (Laplace-Beltrami). Our analysis of the two different formulations of SDOs used in the methods revealed that if the exact tangent space for the surface is used, these formulations are identical. We further derived a new RBF-FD method for approximating the tangent space of surfaces represented only by point clouds. Our numerical investigation of the methods showed that they appear to converge at similar rates when the same polynomial degree ℓ is used, but that RBF-FD generally gives lower errors for the same N and ℓ . We additionally examined the dependency of the stencil size on the methods (as measured by the τ parameter) and found that the errors produced by GMLS deteriorate as the stencil size increases. The errors for RBF-FD, contrastingly, appear to keep improving as the stencil size increases. However, we don’t expect this trend to continue indefinitely, as eventually the tangent plane formulation breaks down when the stencil size becomes too large. Finally, we investigated the computational efficiency of the methods in terms of error versus computational cost and found GMLS to be more efficient when setup costs are included and RBF-FD to be more efficient when only considering evaluation costs.

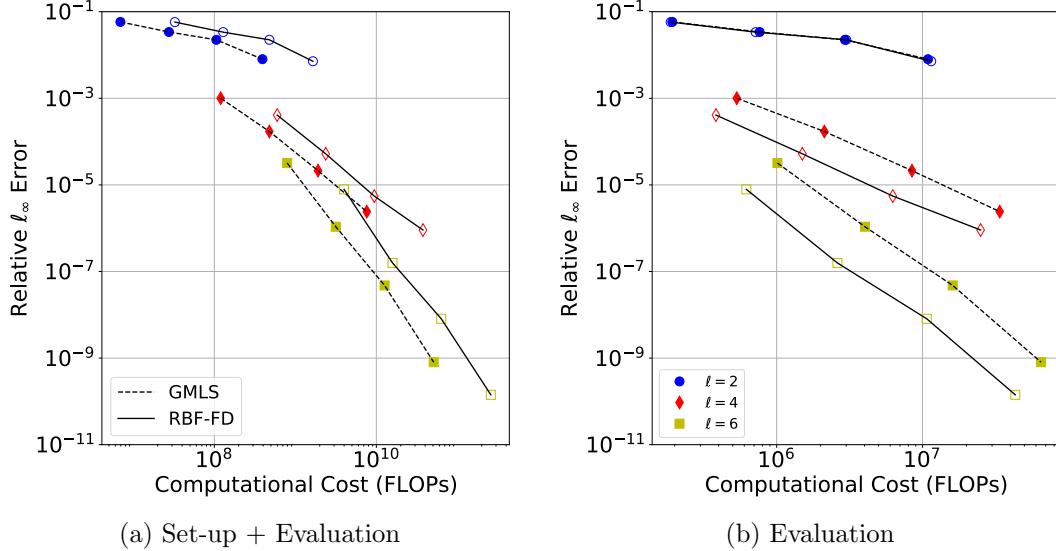


Figure 9: Comparison of the computational efficiency of GMLS and RBF-FD for approximating the surface Laplacian in terms of accuracy per computational cost. (a) Shows the efficiency when considering the setup and evaluation costs as defined in (23) and (24), respectively, while (b) show the efficiency when only considering the evaluation cost.

Acknowledgements

Funding. AMJ was partially supported by US NSF grant CCF-1717556. PAB & PAK were supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR) Program and Biological and Environmental Research (BER) Program under a Scientific Discovery through Advanced Computing (SciDAC 4) BER partnership pilot project. PAK was additionally supported by the Laboratory Directed Research & Development (LDRD) program at Sandia National Laboratories and ASCR under Award Number DE-SC-0000230927. AMJ was also partially supported by the Climate Model Development and Validation (CMDV) program, funded by BER. Part of this work was conducted while AMJ was employed at the Computer Science Research Institute at Sandia National Laboratories. GBW was partially supported by U.S. NSF grants CCF-1717556 and DMS-1952674.

References

- [1] G. K. Vallis, Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-scale Circulation, Cambridge University Press, 2006. doi:10.1017/CBO9780511790447.
- [2] D. L. Williamson, The evolution of dynamical cores for global atmospheric models, *J. Meteorol. Soc. Jpn.* 85B (2007) 241–269.
- [3] B. Fornberg, N. Flyer, A Primer on Radial Basis Functions with Applications to the Geosciences, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2015.
- [4] P. Bosler, L. Wang, C. Jablonowski, R. Krasny, A Lagrangian particle/panel method for the barotropic vorticity equations on a rotating sphere, *Fluid Dyn. Res.* 46 (2014). doi:doi:10.1088/0169-5983/46/3/031406.
- [5] E. J. Gowan, X. Zhang, S. Khosravi, A. Rovere, P. Stocchi, A. L. C. Hughes, R. Gyllencreutz, J. Mangerud, J.-I. Svendsen, G. Lohmann, A new global ice sheet reconstruction for the past 80 000 years, *Nature Communications* 12 (1) (2021) 1199. doi:10.1038/s41467-021-21469-w.

[6] J. Liu, J. F. Totz, P. W. Miller, A. D. Hastewell, Y.-C. Chao, J. Dunkel, N. Fakhri, Topological braiding and virtual particles on the cell membrane, *Proc. Natl. Acad. Sci. U.S.A.* 118 (34) (2021). doi:10.1073/pnas.2104191118.

[7] N. Stoop, R. Lagrange, D. Terwagne, P. M. Reis, J. Dunkel, Curvature-induced symmetry breaking determines elastic surface patterns, *Nature Materials* 14 (3) (2015) 337–342.

[8] M. S. Mikkelsen, Surface gradient-based bump mapping framework, *Journal of Computer Graphics Techniques (JCGT)* 9 (4) (2020) 60–91.

[9] S. Banerjee, D. Lakehal, M. Fulgosi, Surface divergence models for scalar exchange between turbulent streams, *International Journal of Multiphase Flow* 30 (7) (2004) 963–977.

[10] V. S. Mahadevan, J. E. Guerra, X. Jiao, P. Kuberry, Y. Li, P. Ullrich, D. Marsico, R. Jacob, P. Bochev, P. Jones, Metrics for Intercomparison of Remapping Algorithms (MIRA) protocol applied to Earth system models, *Geosci. Model Dev.* 15 (17) (2022) 6601–6635.

[11] J. Liang, H. Zhao, Solving partial differential equations on point clouds, *SIAM J. Sci. Comput.* 35 (3) (2013) A1461–A1486. doi:10.1137/120869730.

[12] N. Trask, P. Kuberry, Compatible meshfree discretization of surface PDEs, *Computational Particle Mechanics* 7 (2020) 271–277. doi:10.1007/s40571-019-00251-2.

[13] P. Suchde, J. Kuhnert, A meshfree generalized finite difference method for surface PDEs, *Comp. Math. Appl.* 78 (8) (2019) 2789–2805. doi:<https://doi.org/10.1016/j.camwa.2019.04.030>.

[14] B. J. Gross, N. Trask, P. Kuberry, P. J. Atzberger, Meshfree methods on manifolds for hydrodynamic flows on curved surfaces: A generalized moving least-squares (GMLS) approach, *J. Comput. Phys.* 409 (2020) 109340. doi:<https://doi.org/10.1016/j.jcp.2020.109340>.

[15] E. Lehto, V. Shankar, G. B. Wright, A radial basis function (RBF) compact finite difference (FD) scheme for reaction-diffusion equations on surfaces, *SIAM J. Sci. Comput.* 39 (2017) A219–A2151. doi:10.1137/16M1095457.

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

[17] N. Flyer, G. B. Wright, B. Fornberg, Radial Basis Function-generated Finite Differences: A Mesh-free Method for Computational Geosciences, in: W. Freeden, M. Z. Nashed, T. Sonar (Eds.), *Handbook of Geomathematics*, 2nd Edition, Springer-Verlag, Berlin, 2014, pp. 1–30.

[18] V. Shankar, G. B. Wright, R. M. Kirby, A. L. Fogelson, A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction-diffusion equations on surfaces, *J. Sci. Comput.* 63 (3) (2014) 745–768. doi:10.1007/s10915-014-9914-1.

[19] C. Piret, J. Dunn, Fast RBF OGr for solving PDEs on arbitrary surfaces, *AIP Conference Proceedings* 1776 (1) (2016) 070005. doi:10.1063/1.4965351.

[20] V. Shankar, A. Narayan, R. M. Kirby, RBF-LOI: Augmenting radial basis functions (RBFs) with least orthogonal interpolation (LOI) for solving PDEs on surfaces, *J. Comput. Phys.* 373 (2018) 722–735. doi:<https://doi.org/10.1016/j.jcp.2018.07.015>.

[21] A. Petras, L. Ling, S. J. Ruuth, An RBF-FD closest point method for solving PDEs on surfaces, *J. Comput. Phys.* 370 (2018) 43–57.

[22] S. Shaw, Radial basis function finite difference approximations of the Laplace-Beltrami operator, Master’s thesis, Boise State University, USA, 1587 (2019). doi:10.18122/td/1587/boisestate.

[23] D. Gunderman, N. Flyer, B. Fornberg, Transport schemes in spherical geometries using spline-based RBF-FD with polynomials, *J. Comput. Phys.* 408 (2020) 109256. doi:<https://doi.org/10.1016/j.jcp.2020.109256>.

[24] H. Wendland, J. Künemund, Solving partial differential equations on (evolving) surfaces with radial basis functions, *Advances in Computational Mathematics* 46 (2020) 64. doi:10.1007/s10444-020-09803-0.

[25] D. Álvarez, P. González-Rodríguez, M. Kindelan, A local radial basis function method for the Laplace–Beltrami operator, *J. Sci. Comput.* 86 (3) (2021) 28. doi:10.1007/s10915-020-01399-3.

[26] G. B. Wright, A. M. Jones, V. Shankar, MGM: A meshfree geometric multilevel method for systems arising from elliptic equations on point cloud surfaces, *SIAM J. Sci. Comput.* 45 (2023) A312–A337.

[27] G. Dziuk, C. M. Elliott, Finite element methods for surface PDEs, *Acta Numerica* 22 (2013) 289–396. doi:10.1017/S0962492913000056.

[28] M. Bertalmío, L. Cheng, S. Osher, G. Sapiro, Variational problems and partial differential equations on implicit surfaces, *J. Comput. Phys.* 174 (2001) 759–780.

[29] V. Bayona, Comparison of moving least squares and RBF+Poly for interpolation and derivative approximation, *J. Sci. Comput.* 81 (1) (2019) 486–512. doi:10.1007/s10915-019-01028-8.

[30] V. Bayona, N. Flyer, B. Fornberg, G. A. Barnett, On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs, *J. Comput. Phys.* 332 (2017) 257–273.

[31] L. Demanet, Painless, highly accurate discretizations of the Laplacian on a smooth manifold, Technical report, Stanford University (2006).

[32] S. W. Walker, *The Shapes of Things*, SIAM, Philadelphia, 2015. doi:10.1137/1.9781611973969.

[33] B. O’Neill, *Elementary Differential Geometry*, 2nd Edition, Elsevier, 2006.

[34] J. J. Koenderink, *Solid shape*, MIT press, 1990.

[35] P. Kuberry, P. Bosler, N. Trask, Compadre toolkit (Jan. 2019). doi:10.11578/dc.20190411.1. URL <https://github.com/sandialabs/compadre>

[36] H. Wendland, *Scattered Data Approximation*, Vol. 17, Cambridge University Press, Cambridge, 2005.

[37] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives, *IMA J. Numer. Anal.* 32 (2011) 983–1000. doi:10.1093/imanum/drr030.

[38] N. Trask, M. Perego, P. Bochev, A high-order staggered meshless method for elliptic problems, *SIAM J. Sci. Comput.* 39 (2) (2017) A479–A502.

[39] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences, World Scientific Publishers, Singapore, 2007.

[40] D. Mirzaei, Error bounds for GMLS derivatives approximations of Sobolev functions, *J. Comput. Appl. Math.* 294 (2016) 93–101.

[41] Y. Lipman, D. Cohen-Or, D. Levin, Error bounds and optimal neighborhoods for MLS approximation, in: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 2006, pp. 71–80.

[42] H. Wang, C. E. Scheidegger, C. T. Silva, Optimal bandwidth selection for MLS surfaces, in: *2008 IEEE International Conference on Shape Modeling and Applications*, IEEE, 2008, pp. 111–120.

[43] V. Bayona, N. Flyer, B. Fornberg, On the role of polynomials in RBF-FD approximations: III. Behavior near domain boundaries, *J. Comput. Phys.* 380 (2019) 378–399.

[44] O. Davydov, R. Schaback, Optimal stencils in Sobolev spaces, *IMA J. Num. Analy.* 39 (1) (2019) 398–422.

[45] V. Bayona, An insight into RBF-FD approximations augmented with polynomials, *Comput. Math. Appl.* 77 (9) (2019) 2337–2353.

[46] J. Cui, W. Freeden, Equidistribution on the sphere, *SIAM J. Sci. Stat. Comput.* 18 (1997) 595–609.

[47] C. Yuksel, Sample elimination for generating Poisson disk sample sets, *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2015)* 34 (2) (2015) 25–32. doi:10.1111/cgf.12538.

[48] A. M. Jones, Meshfree methods for PDEs on surfaces, Ph.D. thesis, Boise State University, USA (2022). doi:10.18122/td.2023.boisestate.

[49] E. J. Fuselier, G. B. Wright, A high-order kernel method for diffusion and reaction-diffusion equations on surfaces, *J. Sci. Comput.* 56 (3) (2013) 535–565.

Statements and Declarations

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan.

Competing Interests. The authors have no relevant financial or non-financial interests to disclose.

Author Contributions.

- AMJ: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing: Original draft preparation, Writing: Review & Editing.
- PAB: Methodology, Resources, Writing: Review & Editing, Funding acquisition, Supervision
- PAK: Resources, Software, Writing: Review & Editing
- GBW: Methodology, Writing: Original draft preparation, Writing: Review & Editing, Funding acquisition, Supervision