A localized meshless method for diffusion on folded surfaces

Ka Chun Cheung[a,*], Leevan Ling[a], Steve Ruuth[b]

[a] *Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.*
[b] *Department of Mathematics, Simon Fraser University, BC, Canada.*

Abstract

Partial differential equations (PDEs) on surfaces arise in a variety of application areas including biological systems, medical imaging, fluid dynamics, mathematical physics, image processing and computer graphics. In this paper, we propose a radial basis function (RBF) discretization of the closest point method. The corresponding localized meshless method may be used to approximate diffusion on smooth or folded surfaces. Our method has the benefit of having an a priori error bound in terms of percentage of the norm of the solution. A stable solver is used to avoid the ill-conditioning that arises when the radial basis functions (RBFs) become flat.

Keywords: Power function, radial basis function (RBF), closest point method, diffusion, surface.

1. Introduction

A variety of numerical methods have been proposed to approximate the solution of partial differential equations on surfaces. These include discretizations on parameterized surfaces, finite difference and finite element methods on triangulated surfaces, and embedding methods.

Each class of methods is characterized by certain advantages and disadvantages. Parametrization methods can be very efficient, but these methods require a parametrization of the surface. Typically, the construction of a parameterization leads to distortions in the surface as well as singularities [1, 2, 3, 4]. We may avoid this complication by discretizing on triangulated surfaces. Unfortunately, finite difference discretizations on triangulated surfaces are more difficult to construct on curved surfaces than on the plane and it can be difficult to compute geometric quantities such as the curvature and the normal to the surface

[*] Corresponding author
Email address: kachun.charles@gmail.com (Ka Chun Cheung)

[1] This project was supported by a CERG Grant of the Hong Kong Research Grant Council, an FRG Grant of HKBU, and a grant from NSERC Canada.

[5, 6, 7]. On the other hand, finite element methods on triangulated surfaces can be effective when applied to problems of elliptic or parabolic type [8, 9]. In the embedding class of methods, the surface and the corresponding surface PDE are represented in the underlying d-dimensional embedding space (e.g., a problem posed on a two-dimensional curved surface might be represented and solved in a three-dimensional neighbourhood of the surface) thereby enabling the use of standard Cartesian grid methods in $\mathbb{R}^d$. The most popular finite difference [5, 6] and RBF discretizations [10, 11] of this type project the surface Laplacian to the tangent space of the surface. Often, a level set representation of the surface is used, in which case there is no direct generalization to open surfaces or surfaces without orientation. Moreover, embedding methods typically introduce artificial boundary conditions at the boundary of the computational domain, leading to an addition error and first-order accuracy for diffusion problems. These complications can be overcome by using the closest point method [12], which is a high-order accurate method that embeds the surface within the Euclidean embedding space by means of a closest point mapping. The standard explicit closest point method is comprised of an evolution step and an interpolation step: the method must interpolate function values on the surface at each time step in order to replace derivatives intrinsic to the surface with standard Cartesian operators.

In [13], Piret proposed to discretize the closest point method using radial basis functions. As a consequence of using radial basis functions, the method is not restricted to Cartesian grids. This allows for two fundamental advantages over the standard closest point method. First, it is straightforward to use irregular or refined grids according to the needs of the problem. Second, the freedom of selecting nodal points naturally allows one to eliminate the interpolation step of the closest point method by making an appropriate node selection. By avoiding the interpolation step, the computational error and cost associated with interpolation is eliminated. The method proposed in this paper also combines the closest point method and the radial basis function projection method to solve PDEs on surfaces. As in Piret's approach, our approach does not involve an interpolation step, thereby eliminating a source of error and opening the possibility of significant savings in computational cost. Apart from this, we will show that our proposed method is error optimal, i.e., the spatial approximation has a minimal error bound among all meshless methods. Moreover, the error bound is of a-priori type: it depends only on the distribution of nodes and the type of kernel or basis function. On the other hand, the discretization of the spatial variables in our approach leads to a differentiation matrix W such that $\mathbf{u}_t = W\mathbf{u}$. While the differentiation matrix W is error optimal, it is also very ill-conditioned when the radial basis functions become flat or when the nodal points are very dense. Consequently, as we refine our discretization, the differentiation matrix becomes ill-conditioned [14, 15, 16]. This trade-off is inherent in meshless methods. See [17, 18] for a strategy to handle the ill-conditioning by changing the basis function to an orthogonal basis, thereby stabilizing the computation for flat radial basis functions. Throughout this paper, the differentiation matrix W arising in numerical experiments is solved by this RBF-QR strategy.

In addition to the development of an error-optimal RBF discretization, this paper introduces a generalization of the closest point method appropriate for diffusion on folded surfaces. The standard closest point method is built on the assumption that the closest point function is sufficiently smooth within the computational band. As a consequence, the standard method cannot be used to compute the solutions of PDEs on curves with corners or surfaces with folds. In [19], a transformation was introduced to give a smooth, transformed problem which is amenable to computation via the standard closest point method. Unfortunately, the transformed problem has increased dimensionality, and consequently is computationally expensive. In this paper, we develop an extension of the closest point method for solving diffusion on curves with corners and on folded surfaces. The method computes within the original embedding space and uses node placement along the folds. This makes a radial basis function discretization particularly appropriate. Numerical experiments demonstrate that the proposed method retains high-order accuracy for smooth curves and surfaces, while giving first order convergence for computations on geometries with folds.

2. Surface Embedding

The closest point method, originally proposed by Ruuth and Merriman [12], is an embedding method for solving PDEs on surfaces. The method embeds the surface operator to the Euclidean space $\mathbb{R}^n$ using a closest point mapping. Such an embedding is called a closest point extension. By alternating the closest point extension with a traditional PDE solver on $\mathbb{R}^n$, we can obtain an explicit method for solving a surface PDE in the embedding space.

Let $\mathcal{S} \subset \mathbb{R}^n$ be a smooth surface and ∇ be the standard Euclidean gradient in $\mathbb{R}^n$. Then, the surface gradient, denoted by $\nabla_{\mathcal{S}}$, is defined as:

$$\nabla_{\mathcal{S}} := \nabla - \hat{n}(\hat{n} \cdot \nabla), \tag{1}$$

where $\hat{n}$ is the unit normal to the surface $\mathcal{S}$. Consider a C^1 vector field $\phi : \mathbb{R}^n \to \mathbb{R}^n$ that is constant along the normal direction to $\mathcal{S}$. The directional derivative satisfies

$$0 = \frac{d}{d\hat{n}}(u \circ \phi) = \hat{n} \cdot \nabla(u \circ \phi),$$

which we combine with (1) to obtain

$$\nabla_{\mathcal{S}}(u \circ \phi) = \nabla(u \circ \phi) - \hat{n} \cdot \nabla(u \circ \phi)\,\hat{n} = \nabla(u \circ \phi), \text{ on } \mathcal{S}. \tag{2}$$

Thus, for any C^1 function that is constant along the normal direction to $\mathcal{S}$, we observe that, on the surface, standard Euclidean gradients (defined on the embedding space) coincide with the desired surface gradients. A particularly convenient vector field $\phi : \mathbb{R}^n \to \mathcal{S}$ is the *closest point mapping*

$$\phi_{\mathcal{S}}(x) = \arg\min_{s \in \mathcal{S}} |x - s|, \text{ for all } x \in \overline{\mathcal{S}}, \tag{3}$$

where $\overline{\mathcal{S}} \subseteq \mathbb{R}^n$ is a neighborhood of $\mathcal{S}$ over which the closest point mapping $\phi_{\mathcal{S}}$ is C^1. Generally, the maximum possible $\overline{\mathcal{S}}$ is determined by the curvature of $\mathcal{S}$. For any C^1 surface, we can properly embed $\mathcal{S} \subset \overline{\mathcal{S}}$. For example, if $\mathcal{S} \in \mathbb{R}^2$ is circle, then the maximum possible $\overline{\mathcal{S}}$ will be the whole plane $\mathbb{R}^2$ except the center of $\mathcal{S}$.

Similar conditions can be derived for the surface divergence for any differentiable vector field $\varphi : \mathbb{R}^n \to \mathbb{R}^n$. Similarly, if φ is tangent to the surface and all the surfaces displaced by a fixed distance from $\mathcal{S}$, then the surface divergence of φ coincides with the usual divergence at the surface

$$\nabla_{\mathcal{S}} \cdot \varphi = \nabla \cdot \varphi, \text{ on } \mathcal{S}. \tag{4}$$

A combination of these two principles demonstrates that we may evaluate the traditional Laplacian $\Delta(u \circ \phi_{\mathcal{S}}) = \nabla \cdot \nabla(u \circ \phi_{\mathcal{S}})$ in $\overline{\mathcal{S}}$ instead of the surface Laplacian $\Delta_{\mathcal{S}}(u \circ \phi_{\mathcal{S}}) = \nabla_{\mathcal{S}} \cdot \nabla_{\mathcal{S}}(u \circ \phi_{\mathcal{S}})$ on $\mathcal{S}$ for $u \in C^2$. In fact, any nonlinear diffusion operator can be embedded to a traditional one in $\overline{\mathcal{S}}$; see [12].

3. $\mathcal{S}$–Orthogonal Data Distribution

For the sake of clarity, let $u_{\mathcal{S}}$ be the restriction to the surface $\mathcal{S}$ for any function u defined on $\mathbb{R}^n$. Consider the heat equation intrinsic to the surface $\mathcal{S}$, i.e.,

$$\partial_t u_{\mathcal{S}} = \Delta_{\mathcal{S}} u_{\mathcal{S}} \text{ on } \mathcal{S}. \tag{5}$$

After embedding the surface PDE to $\overline{\mathcal{S}}$, we get an embedded PDE:

$$\partial_t u = \Delta(u \circ \phi_{\mathcal{S}}) \text{ in } \overline{\mathcal{S}}. \tag{6}$$

We can use any solver developed for the embedded PDE posed on $\mathbb{R}^n$ to solve the surface PDE (5).

Meshless methods are methods that do not require any mesh structure. When applied to the closest point method with an appropriate mesh, meshless methods give a trivial discretization of the closest point extension step. In this work, we adopt the localized meshless approach in [20, 21, 13, 22, 23, 24]. First, let $X = \{\mathbf{x}_i\}_{i=1}^N \subset \mathcal{S}$ be the set of data points used to discretize $\mathcal{S}$; see Figure 1 for a schematic demonstration. Using the set of normal vectors $\mathbf{n}(X) = \{\hat{n}(\mathbf{x}_i) : \mathbf{x}_i \in X\}$, the set X can be extended to a discretization for $\overline{\mathcal{S}}$. We define $G_h(X) : \mathcal{S} \to \overline{\mathcal{S}}$ such that

$$G_h(X) = X + h\mathbf{n}(X) = \{\mathbf{x}_i + h\,\hat{n}(\mathbf{x}_i) : \mathbf{x}_i \in X, h \in \mathbb{R}\}.$$

It is easy to see that $G_h(X)$ is a set of points extended from X by a distance h along its normal direction $\mathbf{n}(X)$. For simplicity, assume $h > 0$ is sufficiently small (w.r.t. k) such that, $\mathcal{G} : \mathcal{S} \to \overline{\mathcal{S}}$ and

$$\overline{X} = \mathcal{G}(X) := X \cup G_{\pm h}(X) \cup \cdots \cup G_{\pm kh}(X) \subset \overline{\mathcal{S}}. \tag{7}$$

With $N = |X|$ points on $\mathcal{S}$, we now have a set of $M = (2k+1)N$ data points in $\overline{\mathcal{S}}$. By construction, trivially, we also have a discretized closest point mapping (3), that maps $u_{|X} \to u_{|\overline{X}}$. The tradeoff is that the data points in $\overline{X}$ are not regular.

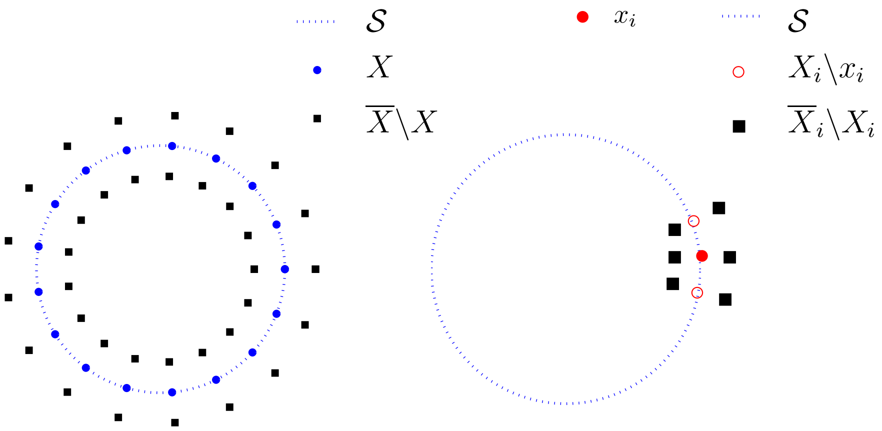


Figure 1: A schematic demonstration for a data point x_i on surfaces $\mathcal{S} \subset \mathbb{R}^2$ along with its symmetric 3-point local neighbourhood X_i and $\mathcal{S}$-orthogonal extension $\overline{X}_i$.

3.1. Point distribution for corners and folded surfaces

Suppose for now that our surface $\mathcal{S}$ is nonsmooth and $\mathcal{S} = \bigcup \mathcal{S}_j$, where each $\mathcal{S}_j$ is smooth. Define the singular curve Γ to be the intersection of these subsurfaces, i.e., the set on which $\mathcal{S}$ is nonsmooth. It is desirable to have data on Γ to enhance the resolution of the numerical approximation. We now describe mesh selection on singular surfaces. For each data point $x_i \in X \subset \mathcal{S}$, we either have x_i inside a smooth subsurface or on the singular curve. For $x_i \in \mathcal{S}_j \setminus \partial\mathcal{S}_j$ (for some j), its local neighbourhood should be restricted to the same subsurface; i.e., $X_i \subset \mathcal{S}_j$. This local neighbourhood can then be extended in the $\mathcal{S}$-orthogonal direction to yield $\overline{X}_i$.

Now, for $x_i \in \Gamma$, x_i must belong to two or more smooth subsurfaces. That is, $x_i \in \mathcal{S}_{i_1} \bigcap \cdots \bigcap \mathcal{S}_{i_k} \subseteq \Gamma$ where $k \geq 2$. Notice that our discretization requires smoothness of the closest point function in a band around each subsurface, a requirement that can be satisfied by taking h sufficiently small. This contrasts with the original closest point method that places a more strict, global restriction over $\mathcal{S}$. Figure 2 gives a simple case of two smooth curves meeting at a corner point x_i. At the corner point x_i, asymmetric local neighbourhoods are used to satisfy $X_{i_1} \subset \mathcal{S}_1$ and $X_{i_2} \subset \mathcal{S}_2$. In our method, these local neighbourhoods will lead to multiple sets of weighings for the corner x_i. An arithmetic average will be taken as the final weighting.

Note that the $\mathcal{S}$-orthogonal extensions $\overline{X}_{i_1}$ can extend past subsurface S_2, and vice versa for an acute angled corner with our method. These overlapping grids will not cause any trouble for our method.

Figure 3(a) shows an example of a dented sphere $\mathcal{S} \subset \mathbb{R}^3$ composed of $\mathcal{S}_1$, a part of the unit sphere, and $\mathcal{S}_2$, the dented top. Figure 3(b) is a bird's-eye

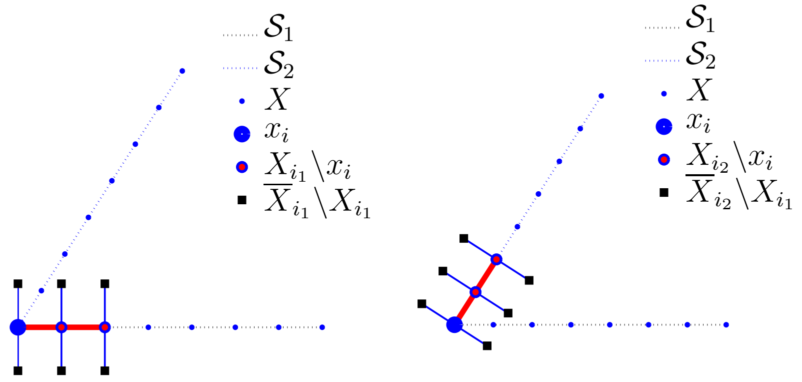
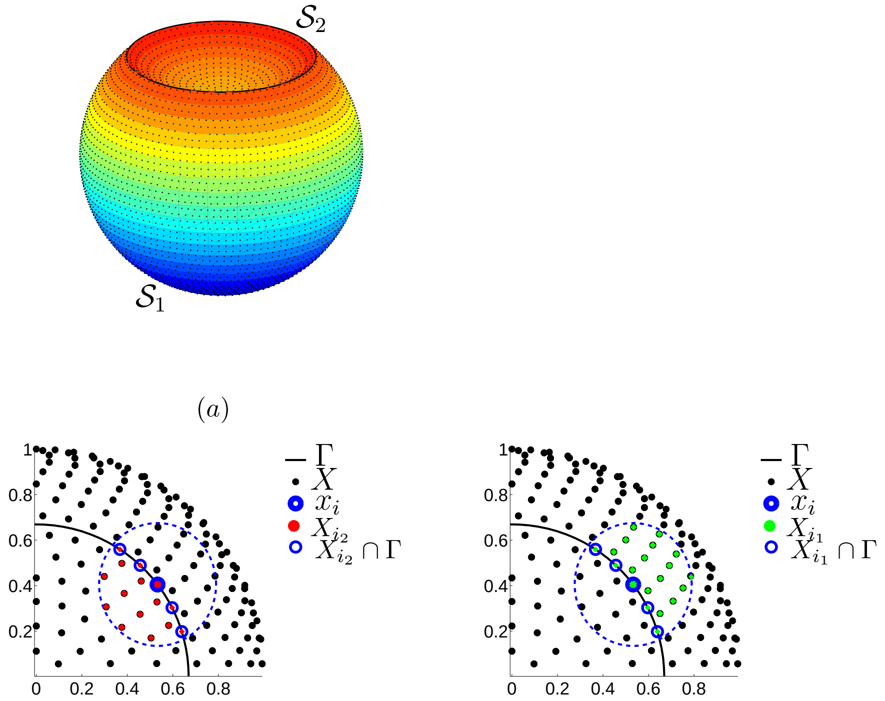


Figure 2: A schematic demonstration of the local neighbourhoods (X_{i_1} and X_{i_2}) and extensions ($\overline{X}_{i_1}$ and $\overline{X}_{i_2}$) for a corner point $x_i \in \mathcal{S}_1 \bigcap \mathcal{S}_2$.



(a)

(b)

(c)

Figure 3: A nonsmooth manifold in $\mathbb{R}^3$ and a schematic data point distribution.

view of some data points X near a part of Γ. In this projected view, the outer annular region belongs to $\mathcal{S}_1$ and the inner region belongs to $\mathcal{S}_2$. $\mathcal{S}_1$ and $\mathcal{S}_2$ meet at Γ (represented by the solid line). For a data point $x_i \in \mathcal{S}_1 \bigcap \mathcal{S}_2$, $\overline{X}_{i_1}$ and $\overline{X}_{i_2}$ are obtained via neighborhood searches. See Figures 3(b) and 3(c) for illustrations of these sets.

3.2. Reduced semi-discrete form

For each $\mathbf{x}_i \in X \subset \mathcal{S}$ and some $\varepsilon > 0$, let the i^{th} local neighbourhood be denoted by $X_i = \{\mathbf{x} \in X \subset \mathcal{S} : |\mathbf{x} - \mathbf{x}_i| < \varepsilon\}$. Then, this neighbourhood on $\mathcal{S}$ can be extended to $\overline{\mathcal{S}}$ by $\overline{X}_i = \mathcal{G}(X_i) \subset \overline{X}$. Assuming we have a set of *generalized finite-difference weightings* $w_{ij} \in \mathbb{R}$, such that for each $\mathbf{x}_i \in \overline{X}$,

$$\mathcal{L}u(\mathbf{x}_i) \approx \sum_j w_{ij}\, u(\boldsymbol{\xi}_j), \quad \boldsymbol{\xi}_j \in \overline{X}_i, \tag{8}$$

where $\mathcal{L}$ is a linear functional. For data $x_i \in \Gamma$ on the surface singularity, the linear combination in (8) involves all points $\boldsymbol{\xi}_j \in \overline{X}_{i_1} \bigcup \cdots \bigcup \overline{X}_{i_k}$. A detailed discussion on the generalized finite-difference method is presented in the next section. For now, suppose (8) is available for all data points in X. Collecting all these weights in an $M \times M$ sparse matrix W, as in the standard finite difference method, we obtain a semi-discrete formulation,

$$\partial_t \mathbf{u}(\overline{X}) = W(\mathbf{u} \circ \phi_{\mathcal{S}})(\overline{X}). \tag{9}$$

Advancing (9) via any time discretization scheme will yield a set of new nodal values at all points in $\overline{X}$. For simplicity, we consider the forward Euler scheme which yields

$$\mathbf{u}(\overline{X}, t + \Delta t) = (\mathbf{u} \circ \phi_{\mathcal{S}})(\overline{X}, t) + \Delta t W(\mathbf{u} \circ \phi_{\mathcal{S}})(\overline{X}, t). \tag{10}$$

It remains to discretize the closest point mapping $\mathbf{u} \circ \phi_{\mathcal{S}}$. This will yield a standard difference equation to update the approximate solution of the embedded PDE (6).

Recall that our target is the surface PDE (5) rather than the embedded (6). In other words, we are only interested in the function values of $\mathbf{u} \circ \phi_{\mathcal{S}}$ or, equivalently, $\mathbf{u}$ values at points X belonging to $\mathcal{S}$. Those at points $\overline{X} \setminus X$ are specified by the closest point mapping, i.e., $\mathbf{u} \circ \phi_{\mathcal{S}}(\overline{X}) = \mathbf{u}(\mathcal{G}^{-1}(\overline{X}))$. Notice, however, that our data points give the function values outside $\mathcal{S}$ as copies of those on the surface. Thus, in (10), the values of $\mathbf{u}(\overline{X} \backslash X)$ and the corresponding weightings in W are unnecessary. Applying this observation, the closest point mapping can be written in matrix form as

$$(\mathbf{u} \circ \phi_{\mathcal{S}})(\overline{X}, t) = E\mathbf{u}(X, t), \text{ for } E = \mathbf{1} \otimes I_N \in \{0,1\}^{M \times N},$$

where $\otimes$ is the kronecker product and E is an upsampling with the $(2k+1)$ vector $\mathbf{1}$ of all ones, assuming points in X are listed first in $\overline{X}$ as in (7). We also define the corresponding downsampling $E^{\dagger} = e_1^T \otimes I_N \in \{0,1\}^{N \times M}$ that extracts values at X. Applying operators E and $E^\dagger$, the $M \times M$ system (10) can be reduced to an $N \times N$ system:

$$\begin{aligned}
\mathbf{u}(X, t+\Delta t) &= E^\dagger \mathbf{u}(\overline{X}, t+\Delta t), \\
&= E^\dagger \big((\mathbf{u} \circ \phi_{\mathcal{S}})(\overline{X}, t) + \Delta t W (\mathbf{u} \circ \phi_{\mathcal{S}})(\overline{X}, t)\big), \\
&= E^\dagger \big(E\mathbf{u}(X,t) + \Delta t W E \mathbf{u}(X,t)\big), \\
&= \big(I_N + \Delta t\, E^\dagger W E\big)\mathbf{u}(X,t), \qquad (11)
\end{aligned}$$

where the $N \times N$ matrix $E^\dagger W E$ only requires N sets of finite difference weights w_{ij}, given by (8), for the points X instead of all M sets for $\overline{X}$.

Equation (11) gives the forward Euler approximation of the time discretization of the surface PDE $\partial_t u_{\mathcal{S}} = \mathcal{L}_{\mathcal{S}} u_{\mathcal{S}}$, (5), on data points $X \subset \mathcal{S}$. Other time-stepping methods may also be considered. Similar to the forward Euler case, such methods should be applied to the reduced semi-discrete form for the surface PDE,

$$\partial_t \mathbf{u}_{\mathcal{S}}(X) = (E^\dagger W E)\, \mathbf{u}_{\mathcal{S}}(X). \qquad (12)$$

4. Norm-Optimal Approximation

We now address how the generalized finite difference weightings (8) can be obtained for all $\mathbf{x}_i \in X \subset \mathcal{S}$ to evaluate the matrix $E^\dagger W E$ in the reduced semi-discrete surface PDE (12). Recall that $\mathbf{x}_i \in X$ is any given data point on the surface $\mathcal{S}$. The subset of neighboring points $X_i \in \mathcal{S}$ of x_i is then obtained by some neighborhood search technique over the set X, such as a kd-tree search. Then, the subset $\overline{X}_i \subset \overline{X}$ is obtained by extending points in the normal direction, as in (7), and we denote $n_i = |\overline{X}_i|$. Throughout this section, the index i will be used to denote the current point under consideration and will be omitted when no ambiguity arises. Generally speaking, we need a method to look for the weighting $\{w_{ij}\}$ in (8) such that the linear functional $\delta_{\mathbf{x}_i}\mathcal{L}$ is well-approximated by a linear combination of the function values of u at $\overline{X}_i$,

$$\delta_{\mathbf{x}_i}\mathcal{L}[u] = \mathcal{L}u(\mathbf{x}_i) \approx \sum_j w_{ij}\, u(\boldsymbol{\xi}_j), \quad \boldsymbol{\xi}_j \in \overline{X}_i. \qquad (13)$$

Equivalently, in matrix form, it suffices to find the matrix W in the following linear system

$$\mathcal{L}\mathbf{u}(X) \approx E^\dagger W \mathbf{u}(\overline{X}),$$

where $E^\dagger W$ is of size $N \times M$ with entries $[E^\dagger W]_{ij} = w_{ij}$ in (13). For the singular points $x_i \in \Gamma$, the linear functional $\delta_{\mathbf{x}_i}\mathcal{L}$ has multiple approximations, one for each subsurface to which x_i belongs.

To select the weighting so that the discretization on a set of given nodes is error optimal, we introduce a reproducing kernel Hilbert space. As will be shown, *a priori* error estimation is possible with our approach.

Let $K : \Omega \times \Omega \to \mathbb{R}$ be a symmetric, strictly positive definite kernel with $\Omega \supset \overline{X}_i$ and let $H_K(\Omega) = \text{span}\{K(\cdot, y) : y \in \Omega\}$. We define an associated bilinear form $\langle \cdot, \cdot \rangle_K$ for H_K by

$$\left\langle \sum_j c_j K(\cdot, x_j), \sum_k d_k K(\cdot, y_k) \right\rangle_K = \sum_j \sum_k c_j d_k K(x_j, y_k).$$

Thus, $H_K(\Omega)$ is a pre-Hilbert space with $\langle \cdot, \cdot \rangle_K$ as an inner product. Completion of $H_K(\Omega)$ yields a Hilbert space which is called the native space, denoted by $\mathcal{H}(\Omega)$ for simplicity. The space $\mathcal{H}(\Omega)$ satisfies

$$\begin{aligned} K(x,y) &= \langle K(x,\cdot), K(y,\cdot) \rangle_{\mathcal{H}}, \\ f(x) &= \langle f, K(x,\cdot) \rangle_{\mathcal{H}}, \\ \langle \delta_{\mathbf{x}_i}\mathcal{L}, \delta_{\mathbf{x}_j}\mathcal{L} \rangle_{\mathcal{H}^*} &= \delta_{\mathbf{x}_i}\mathcal{L}^x \delta_{\mathbf{x}_j}\mathcal{L}^y K(x,y) = \mathcal{L}^x \mathcal{L}^y K(\mathbf{x}_i, \mathbf{x}_j), \end{aligned}$$

for all $x, y \in \Omega$, $f \in \mathcal{H}$ and $\delta_{\mathbf{x}}\mathcal{L} \in \mathcal{H}^*$, where $\mathcal{H}^*$ is the dual space of $\mathcal{H}$ and the superscript in $\delta_{\mathbf{x}_i}\mathcal{L}^x$ is used to emphasize its action on the first variable of $K(x, y)$.

Suppose we want to find the weights in (8) to minimize the absolute error [25]. The approximation error is given by

$$\begin{aligned} \left| \mathcal{L}u(\mathbf{x}_i) - \sum_{\boldsymbol{\xi}_j \in \overline{X}_i} w_j u(\boldsymbol{\xi}_j) \right| &= \left| \Big(\delta_{\mathbf{x}_i}\mathcal{L} - \sum w_j \delta_{\boldsymbol{\xi}_j} \Big)[u] \right|, \\ &\le \left\| \delta_{\mathbf{x}_i}\mathcal{L} - \sum w_j \delta_{\boldsymbol{\xi}_j} \right\|_{\mathcal{H}^*} \|u\|_{\mathcal{H}}, \end{aligned} \tag{14}$$

where $w_j := w_{ij}$ is the weight in (8). We define the *power function* $\varepsilon_{\delta_{\mathbf{x}_i}\mathcal{L}, \overline{X}_i, \mathbf{w}} := \delta_{\mathbf{x}_i}\mathcal{L} - \sum w_j \delta_{\boldsymbol{\xi}_j}$ for a linear functional $\delta_{\mathbf{x}_i}\mathcal{L}$, node point $\overline{X}_i$, and weighting $\mathbf{w}$. Note that the square of the norm of the power function is a bilinear form such that:

$$\begin{aligned} \mathcal{P}^2(\mathbf{w}) := \left\| \varepsilon_{\delta_{\mathbf{x}_i}\mathcal{L}, \overline{X}_i, \mathbf{w}} \right\|^2_{\mathcal{H}^*} &= \left\| \delta_{\mathbf{x}_i}\mathcal{L} - \sum_{\boldsymbol{\xi}_j \in \overline{X}_i} w_j \delta_{\boldsymbol{\xi}_j} \right\|^2_{\mathcal{H}^*}, \\ &= \langle \delta_{\mathbf{x}_i}\mathcal{L}, \delta_{\mathbf{x}_i}\mathcal{L} \rangle_{\mathcal{H}^*} - 2 \sum w_j \langle \delta_{\mathbf{x}_i}\mathcal{L}, \delta_{\mathbf{x_j}} \rangle_{\mathcal{H}^*} + \sum \sum w_j w_k \langle \delta_{\mathbf{x}_k}, \delta_{\mathbf{x}_j} \rangle_{\mathcal{H}^*}, \\ &= \mathcal{L}^x \mathcal{L}^y K(\mathbf{x}_i, \mathbf{x}_i) - 2 \sum w_j \mathcal{L}^x K(\mathbf{x}_i, \mathbf{x}_j) + \sum \sum w_j w_k K(\mathbf{x}_k, \mathbf{x}_j) \end{aligned} \tag{15}$$

Finding an error optimal formula is equivalent to minimizing $\mathcal{P}^2$. It is easily shown that the solution $\mathbf{w}^*$ of the following system minimizes $\mathcal{P}^2$:

$$\sum_{\mathbf{x}_j \in \overline{X}_i} w_j^* K(\mathbf{x}_k, \mathbf{x}_j) = \mathcal{L}^x K(\mathbf{x}_i, \mathbf{x}_k), \quad \mathbf{x}_k \in \overline{X}_i. \tag{16}$$

Equivalently, in matrix form a minimizer is obtained if $\mathbf{w}^*$ satisfies

$$K(\overline{X}_i, \overline{X}_i)\mathbf{w}^* = \mathcal{L}^x K(\mathbf{x}_i, \overline{X}_i). \tag{17}$$

Applying this result to (15), we observe that the minimum of $\mathcal{P}^2$ (i.e., the square of absolute error of (8) at the point $\mathbf{x}_i$) is given by

$$\min_{\mathbf{w}} \mathcal{P}^2(\mathbf{w}) = \mathcal{P}^2(\mathbf{w}^*) = \mathcal{L}^x \mathcal{L}^y K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{\mathbf{x}_j \in \overline{X}_i} w_j^* \mathcal{L}^x K(\mathbf{x}_i, \mathbf{x}_j).$$

Consequently, the solution of (16) is the weighting such that the approximation (13) has the smallest error in the sense of the native space norm $\| \cdot \|_{\mathcal{H}}$. Note that the native space $\mathcal{H}$ is a reproducing kernel Hilbert space. For each type of kernel K, there is a corresponding native space $\mathcal{N}_K$. In our problem, we take $K(\mathbf{x}, \mathbf{y}) = e^{-(\frac{1}{c}\|\mathbf{x}-\mathbf{y}\|_2)^2}$. For the Matérn kernel $K(\mathbf{x}, \mathbf{y}) := \|\mathbf{x}-\mathbf{y}\|_2^{m-d/2} \mathcal{K}_{m-d/2}\left(\frac{1}{c}\|\mathbf{x}-\mathbf{y}\|_2\right)$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, where $\mathcal{K}_v(r)$ is the modified Bessel function of the second kind, the corresponding native space is the Sobolev space $W_2^m(\mathbb{R}^d)$ for $m > d/2$. For the native space of other kernels, see [26, 27]. In both cases, the shape parameter c controls the shape of the kernel. The tradeoff property for the radial basis function method [28, 29] is that the accuracy improves as the shape parameter c grows large or as the nodal point distribution becomes dense. Computationally, either case comes at the cost of more ill-conditioned matrices.

Once the kernel K is fixed, one can directly compute the power function and error bound without any knowledge of the exact solution. Thus, we can obtain an error bound for a given node before computing any numerical solution. Note that the error bound (14) can also be interpreted as the relative error of the discretization in terms of the $\mathcal{H}$-norm of the exact solution u.

4.1. The meshless finite difference method

The meshless finite difference method, a.k.a. the localized meshless method [20, 22, 23, 24], has been developed for the same purpose to find the weighting $\mathbf{w}$ in (13). Here, we shall show that the weighting found by the meshless finite difference method coincides in exact arithmetic with ours in (16).

Suppose we are given a set of data sites $\overline{X}_i$, but with unknown function values $\mathbf{u}(\overline{X}_i)$. Meshless methods, with a suitable kernel function K, give an interpolant of $\tilde{u}$ in the form

$$\tilde{u}(\mathbf{x}) = \sum_{\mathbf{x}_j \in \overline{X}_i} w_j K(\mathbf{x}, \mathbf{x}_j).$$

Imposing interpolation conditions $\tilde{u} = u$ at all $\mathbf{x} \in \overline{X}_i$, we express the weighting as

$$\mathbf{w} = K(\overline{X}_i, \overline{X}_i)^{-1} \mathbf{u}(\overline{X}_i),$$

where $\mathbf{u}(\overline{X}_i)$ is a vector of (unknown) function values and the matrix entries are defined by $[K(\overline{X}_i, \overline{X}_i)]_{jk} = K(\mathbf{x}_j, \mathbf{x}_k)$ for $\mathbf{x}_j, \mathbf{x}_k \in \overline{X}_i$. Note that the matrix $K(\overline{X}_i, \overline{X}_i)$ is nonsingular provided the kernel K is positive definite.

The meshless finite difference method uses information from the interpolant to approximate the unknown function. For our discussion we have

$$\begin{aligned}
\mathcal{L}u(\mathbf{x}_i) &\approx \mathcal{L}\tilde{u}(\mathbf{x}_i),\\
&= \sum_{\mathbf{x}_j\in\overline{X}_i} w_j \mathcal{L}^x K(\mathbf{x}_i,\mathbf{x}_j),\\
&= (\mathcal{L}^x K(\mathbf{x}_i,\overline{X}_i))^T\cdot\mathbf{w},\\
&= (\mathcal{L}^x, K(\mathbf{x}_i,\overline{X}_i))^T K(\overline{X}_i,\overline{X}_i)^{-1}\mathbf{u}(\overline{X}_i),\\
&= \mathbf{w}\cdot\mathbf{u}(\overline{X}_i) = \sum_{x_j\in\overline{X}_i} w_j u(\mathbf{x}_j).
\end{aligned}$$

Therefore, the meshless finite difference method returns the solution of

$$K(\overline{X}_i,\overline{X}_i)\mathbf{w} = \mathcal{L}^x K(\mathbf{x}_i,\overline{X}_i),$$

as the weighting $\mathbf{w}$, which is identical to our previous results (16) and (17). Thus, the meshless finite difference method is error optimal. Note, however, that the reproducing kernel Hilbert space approach [26, 25, 30] has the advantage of allowing us to measure error by using the corresponding power function.

4.2. Finding optimal-weights numerically

To numerically obtain the generalized finite difference weighting for point $\mathbf{x}_i\in X\subset\mathcal{S}$, we solve the $n_i\times n_i$ matrix system (17). This gives an $n_i=|\overline{X}_i|$ point interpolant over a neighbourhood of $\mathbf{x}_i$. For any commonly used global kernel, such as the Gaussian and Matérn kernels, the kernel matrix $\mathbf{K}:=K(\overline{X}_i,\overline{X}_i)$ is full and, generally speaking, ill-conditioned. To solve (17) directly, it is necessary to invert the kernel matrix, which gives rise to numerical instability. In our numerical computation, we adopt the Gaussian kernel $K(\mathbf{x},\mathbf{y})=e^{-(\frac{1}{c}\|\mathbf{x}-\mathbf{y}\|_2)^2}$ and apply the stable RBF-QR method by Fornberg *et al.* [17]. Note that the method can also be applied to the Matérn kernel [31, 17] in which case one will be working with error bounds in some standard Sobolev spaces, which are the native spaces of the Matérn kernel [26].

It is not difficult to see why the matrix $K(\overline{X}_i,\overline{X}_i)$ becomes ill-conditioned for large c-values. In the limit $c\to\infty$, the Gaussian kernels become flat and form a numerically unstable basis. To avoid this problem, Gaussian radial basis functions may be changed to another set of stable basis functions sharing the same span. It is shown in [17] that the Gaussian kernel $K(\mathbf{x},\mathbf{x}_k)=e^{-(\frac{1}{c}\|\mathbf{x}-\mathbf{x}_k\|_2)^2}$ has an expansion in 2D,

$$K(\mathbf{x},\mathbf{x}_k)=\sum_{j=0}^{\infty}\sum_{m=0}^{\frac{j-p}{2}} d_{j,m}c_{j,m}(\mathbf{x}_k)T^c_{j,m}(\mathbf{x})+\sum_{j=0}^{\infty}\sum_{m=1-p}^{\frac{j-p}{2}} d_{j,m}s_{j,m}(\mathbf{x}_k)T^s_{j,m}(\mathbf{x}), \tag{18}$$

for $0\le m\le\lfloor\frac{j}{2}\rfloor=\frac{j-p}{2}$ with $p=\mathrm{mod}_2 j$. In this expression, the functions

$$T^c_{j,m}(\mathbf{x})=e^{-r^2/c^2}r^{2m}T_{j-2m}(r)\cos\big((2m+p)\theta\big),\quad \text{for } 2m+p=0 \tag{19}$$

and

$$T^s_{j,m}(\mathbf{x}) = e^{-r^2/c^2} r^{2m} T_{j-2m}(r) \sin\big((2m+p)\theta\big), \quad \text{for } 2m+p \neq 0, \tag{20}$$

form a linearly independent set and are defined using the Chebyshev polynomials $T_m(r)$ where $\mathbf{x} = (r, \theta)$ is the polar coordinate of $\mathbf{x}$. The coefficients in (18) depend on the point distribution of $\overline{X}_i$ only and are given by

$$\begin{aligned}
d_{j,m} &= c^{-2j}\Big(2^{j-2m-1}\big(\frac{j+2m+p}{2}\big)!\big(\frac{j-2m-p}{2}\big)!\Big)^{-1}, \\
c_{j,m}(\mathbf{x}_k) &= b_{2m+p} t_{j-2m} e^{-r_k^2/c^2} r_k^j \cos\big((2m+p)\theta_k\big)\ {}_1F_2(\alpha_{j,m}, \beta_{j,m}, c^{-4} r_k^2), \\
s_{j,m}(\mathbf{x}_k) &= b_{2m+p} t_{j-2m} e^{-r_k^2/c^2} r_k^j \sin\big((2m+p)\theta_k\big)\ {}_1F_2(\alpha_{j,m}, \beta_{j,m}, c^{-4} r_k^2),
\end{aligned}$$

where $b_0 = 1$ and $b_m = 2, m > 0$, $t_0 = \frac{1}{2}$ and $t_j = 1, j > 0$. The function ${}_1F_2$ is the hypergeometric function with $\alpha_{j,m} = \frac{j-2m+p+1}{2}$ and $\beta_{j,m} = \left[\ j-2m+1 \quad \frac{j+2m+p+2}{2}\ \right]$.

Despite the complexity, if we truncate the infinite sum in (18) after $J > n_i$ terms, the kernel matrix can be analytically decomposed as

$$K(\overline{X}_i, \overline{X}_i)^T = CDV^T = (QR)DV^T, \tag{21}$$

where $C = QR$ is an $n_i \times J$ matrix consisting of coefficients $c_{j,m}(\overline{X}_i)$ and $s_{j,m}(\overline{X}_i)$ and its full QR-factorization, D, is a $J \times J$ diagonal matrix consisting of entries $d_{j,m}$. The $n \times J$ matrix $V = V(\mathbf{x})$ consists of the functions $T^c_{j,m}(\mathbf{x})$ and $T^s_{j,m}(\mathbf{x})$.

The QR-factorization of the matrix C gives $C = QR = Q\,[R_1 \quad R_2]$ where R_1 contains the left $n_i \times n_i$ block of R and R_2 contains the remaining columns. Let D_1 be the $n_i \times n_i$ principal submatrix of the diagonal matrix D and denote the remaining nonzero $(J - n_i) \times (J - n_i)$ submatrix by D_2. We observe that

$$\begin{aligned}
Q^T K(\overline{X}_i, \overline{X}_i)^T &= RDV^T, \\
&= [R_1 \quad R_2] \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} V^T, \\
&= [R_1 D_1 \quad R_2 D_2] V^T, \\
D_1^{-1} R_1^{-1} Q^T K(\overline{X}_i, \overline{X}_i)^T &= [I_{n_i} \quad D_1^{-1} R_1^{-1} R_2 D_2] V^T.
\end{aligned}$$

Once the above expansion is known, we define the new basis functions as

$$H(\overline{X}_i, \overline{X}_i) = K\,\underbrace{(Q R_1^{-T} D_1^{-T})}_{=:G} = \underbrace{V[I_{n_i} \quad D_1^{-1} R_1^{-1} R_2 D_2]^T}. \tag{22}$$

Note that the diagonal elements of D are computed analytically, whereas $R_1^{-1} R_2$ is computed by backward substitution. The relation between the new kernel matrix and the old kernel matrix is given by

$$H(\overline{X}_i, \overline{X}_i) = K(\overline{X}_i, \overline{X}_i)\, G.$$

Instead of solving the original system, $K(\overline{X}_i, \overline{X}_i)\mathbf{w} = \mathcal{L}^x K(\mathbf{x}_i, \overline{X}_i)$, we find $\mathbf{w}$ in the new (discrete) basis via

$$H(\overline{X}_i, \overline{X}_i)\mathbf{w} = \mathcal{L}^x H(\mathbf{x}_i, \overline{X}_i). \tag{23}$$

Note that the right-hand vector can be obtained analytically from (19) and (20), and that the interpolation matrix with respect to the new kernel, $H(\overline{X}_i, \overline{X}_i)$, is no longer symmetric.

In practice, some polynomial terms should be added to the expansion (13) in order to guarantee polynomial accuracy, or when the kernel K is conditionally positive definite. See [26, 17] for further details.

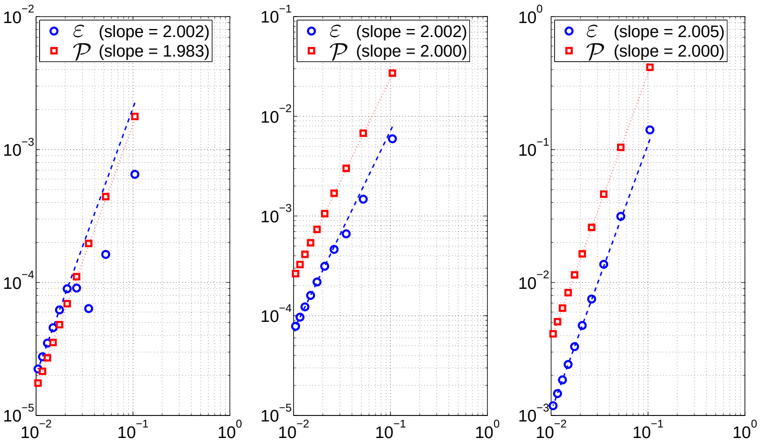
5. Numerical Demonstrations

To begin, we consider diffusion on various surfaces $\mathcal{S} \subset \mathbb{R}^2$. Three point local neighbourhoods X_i will be used to extend one layer outwards and one inwards, as shown in Figures 1 and 2. Throughout this section, we will use the (scaled) Gaussian kernel $\overline{K}(\mathbf{x}, \mathbf{y}) = e^{-(\frac{1}{c}\|\mathbf{x}-\mathbf{y}\|_2)^2}$, with various shape parameters c, to evaluate the generalized finite difference weights in (13) and the power function $\mathcal{P}$ in (15). The reduced semi-discrete system (12) is evolved in time by the forward Euler scheme. As our interest is the spatial discretization of the surface embedding, small time steps are used: $\Delta t = 0.001\hbar^2$ where $\hbar$ is the fill distance [26].

Our first example evolves diffusion on the unit circle. The convergence of the relative errors ε and the power function $\mathcal{P}$ for three different shape parameters, $c = 0.5$, 1, and 2 are given in Figure 4. Convergence rates are approximately second order for all three tested shape parameters (rates are estimated by the last 5 data points). Notably, the proposed scheme is convergent for all tested shape parameters. Moreover, the power function $\mathcal{P}$ also exhibits second order convergence.

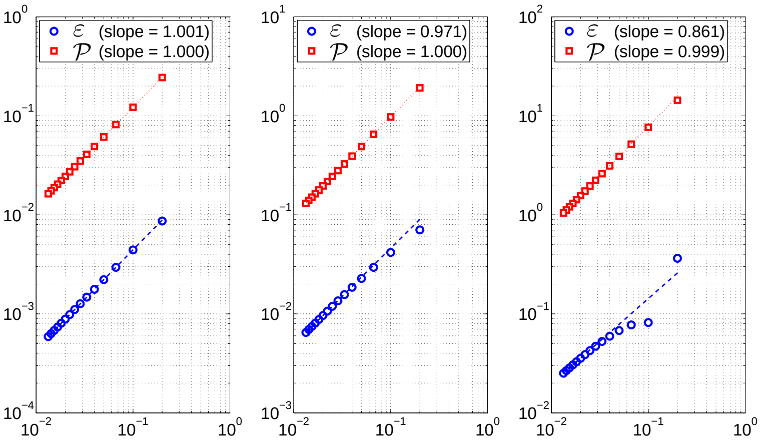
To test the performance on nonsmooth curves we apply the same method to a diffusion problem on the square $[-1, 1]^2$. As shown in Figure 5, the proposed method is still convergent but the rate drops from second order to first. As in the case of the unit circle, we observe a close agreement between the $\varepsilon-$ and $\mathcal{P}-$convergence rates. Provided the temporal discretization error is negligible, the numerically accessible convergence rate of $\mathcal{P}$ provides a good estimate for the convergence rate of the error. The power function can be viewed as the local truncation error in the finite difference method.

We emphasize that the evaluation of $\mathcal{P}$ is pointwise; that is, at the generalized finite difference stencils (8) and (13), the power function can be evaluated by the stencils and assigned weights for any SPD kernel, independent of the solution. The value of $\mathcal{P}$ yields an upper bound for the spatial discretization error: when the solution is unknown, which is usually the case, the discretization error is bounded by $\mathcal{P}$ times an unknown factor $\|u\|_{\mathcal{H}}$. In our convergence plots, we observe that the ratio $\varepsilon/\mathcal{P}$ changes with the shape parameter c. Since the native



$c = 2.0$ $c = 1.0$ $c = 0.5$

Figure 4: Convergence profiles for diffusion on the unit circle $\mathcal{S}$. The relative error ε and power function $\mathcal{P}$ are plotted for various grid spacings and shape parameters c.



$c = 2.0$ $c = 1.0$ $c = 0.5$

Figure 5: Convergence profiles for diffusion on the square $\mathcal{S} = [-1,\,1]^2$. The relative error ε and power function $\mathcal{P}$ are plotted for various grid spacings and shape parameters c.

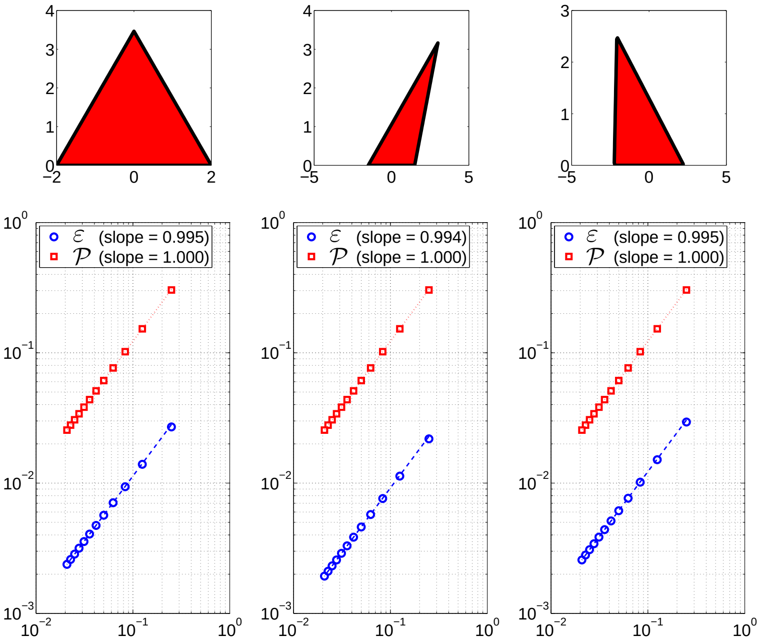


Figure 6: Convergence profiles for diffusion on various triangles. The relative error ε and power function $\mathcal{P}$ are plotted for various grid spacings with the shape parameter $c = 2.0$.

space $\mathcal{H}$ is generated by the c-dependent Gaussian kernel in our demonstrations, different shape parameters yield different spaces $\mathcal{H}$ and hence different $\|u\|_{\mathcal{H}}$.

Determining the optimal shape parameter is challenging and is not the purpose of this work. Nonetheless, the introduction of the power function allows an easy way to search for the optimal c by brute force. The values of $\mathcal{P}$ shown in Figures 4 and 5 are the maximum over all stencils used in a particular run. Intuitively, the first order spatial convergence observed in Figure 5 is due to the error at the corners (this conjecture can be confirmed numerically). We may determine the optimal shape parameters for a problem with corners by minimizing the power function at the corners. Experimentally, we find $c = 2$ gives good results, so we take it for all computations going forward. Figure 6 gives results for diffusion over 3 different folded, triangulated curves. All three cases give similar errors and first order convergence.

The observed quadratic convergence on smooth surfaces and linear convergence on folded surfaces is also found in 3D. In Figure 7, the convergence rates (as measured by the max-norm error) for ε and $\mathcal{P}$ are shown for the unit sphere and a punch-in sphere, which is constructed by reflecting all points above $z = 0.5$ in the place $z = 0.5$ (thereby allowing an analytic solution to be derived). For the unit sphere, data points X are quasi-uniformly placed on the manifold [32]. For the punch-in sphere we place points along the fold. The local neighborhood

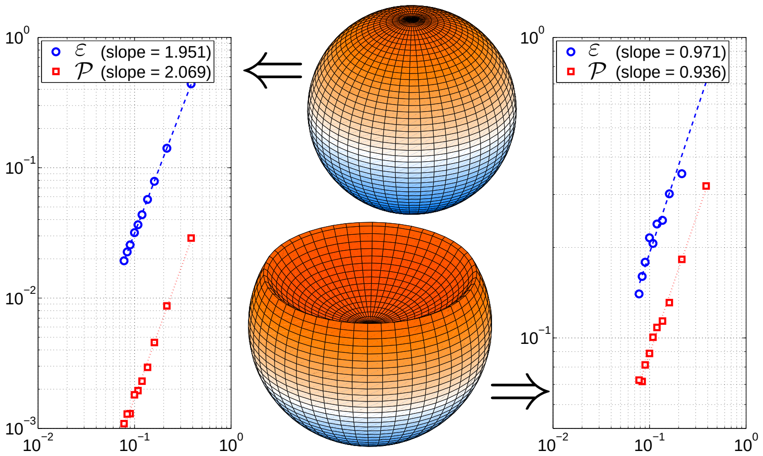
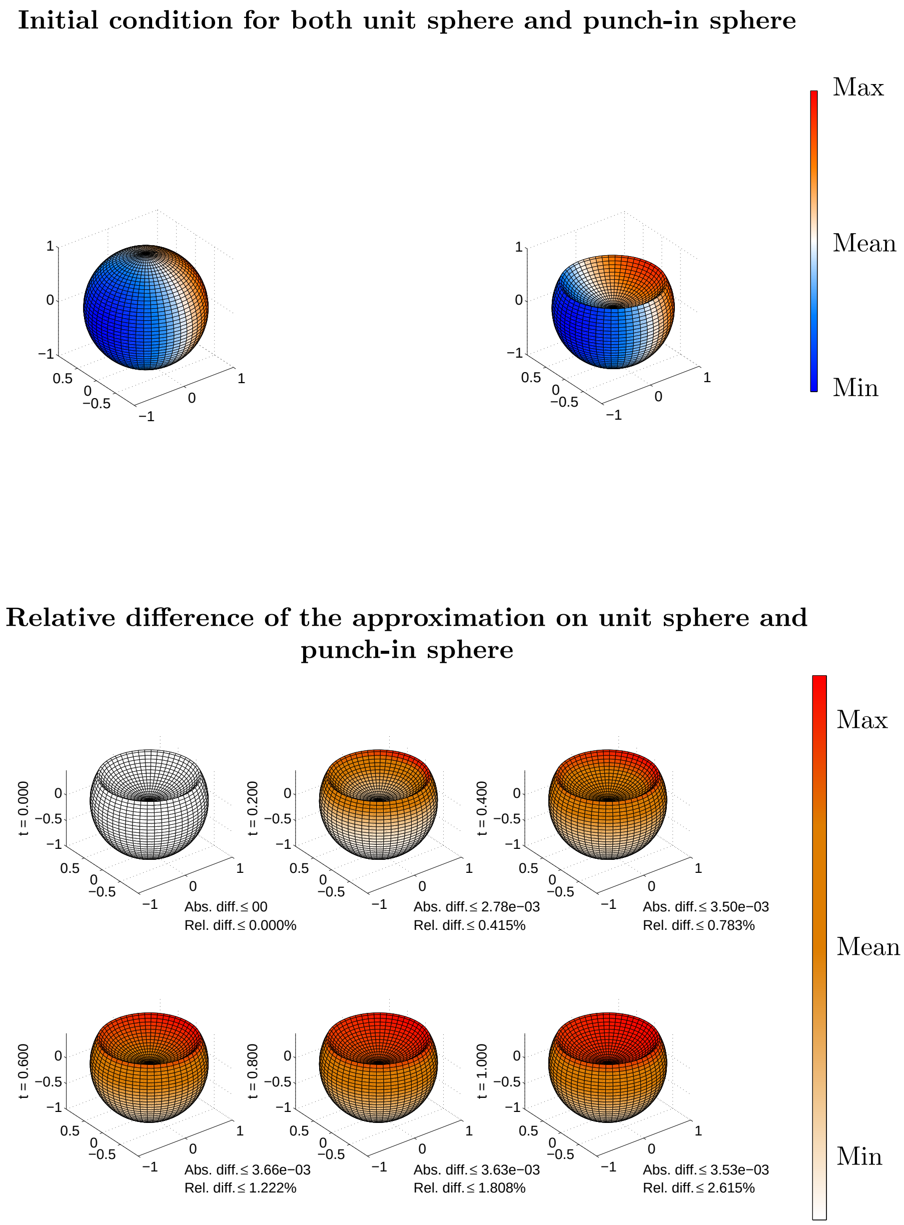


Figure 7: Convergence profiles for diffusion on the unit sphere $\mathcal{S}$ and a punch-in sphere. The relative error ε and power function $\mathcal{P}$ are plotted for various grid spacings and the shape parameter $c = 2.0$. Middle: snapshot at $t = 1$ for the same initial condition.

of x_i on each smooth (sub-) surface is chosen to contain 13 points: 5 points are chosen on the fold and 8 on each smooth subsurface. Similar to the treatment in 2D, we average the finite difference weighings over the two subsurfaces to yield the final spatial discretization for points on the fold.

The unit sphere and the punch-in sphere share identical solutions for any given initial condition despite the existence of a fold (we can map between the unit sphere and the punch-in sphere via a reflection in $z = 0.5$). The numerical schemes used for these two manifolds are clearly different and we observe quadratic spatial convergence for the unit sphere and linear convergence for the punch-in sphere. Figure 8 shows how the difference between the two numerical schemes evolves over time for one fixed spatial discretization. The top two subfigures are the respective initial conditions: they are identical if we unfold the punch-in sphere. From the snapshots in Figure 8, we can see that a disagreement between the two numerical solutions emerges and diffuses away from the fold, which is not surprising since the two numerical recipes are different at the fold. We can see that the relative difference grows gradually with time due to the decay of the solution, whereas the absolute difference is rather steady in time. Clearly, the difference is solely due to the treatment of the fold. Note that our initial conditions vary along the meridian lines. This choice was made to avoid over-simplifying the example via symmetric data at the fold.

Not all manifolds are easily unfolded. As an example, consider a flat-top sphere constructed by replacing the z-coordinates of the unit sphere by $\min(z, 0.5)$. Figures 9 and 10 present two simulated results with different initial conditions. The first simulation begins with two hot spots on the large smooth surface symmetric about the x-axis. As time evolves, we observe the


Initial condition for both unit sphere and punch-in sphere

Relative difference of the approximation on unit sphere and punch-in sphere

Figure 8: Top: initial conditions. Middle, Bottom: relative difference of the numerical approximation on the unit sphere and the punch-in sphere at various times t with "identical" initial conditions.

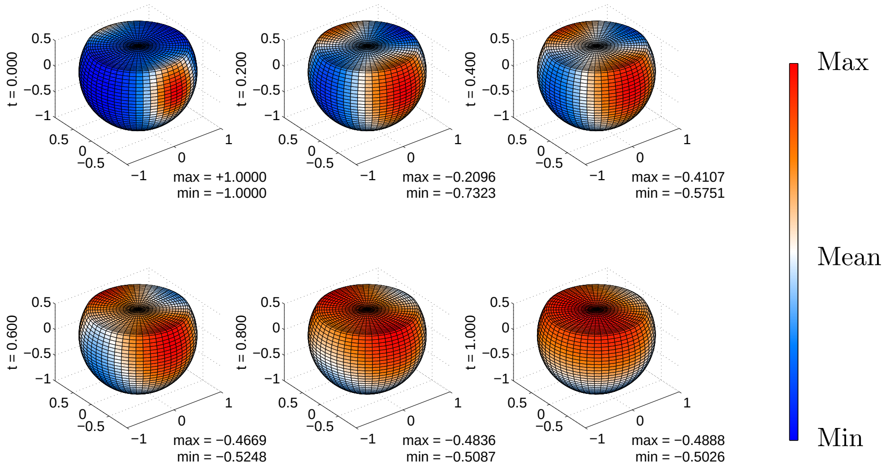


Figure 9: Simulation of diffusion on a flat-top sphere with two hot spots in the x-symmetric initial profile.

diffusion progress across the fold to the flat-top and across the large spherical surface. For each snapshot, the maximum and minimum values of the numerical solution are also presented to confirm that the maximum principle is satisfied at consecutive snapshots. The second simulation is initialized with a hot spot located near the edge of the flat-top in order to highlight the method's treatment of diffusion across a fold. Snapshots are presented at nonuniform times to better visualize the rapid initial flow. We observe that heat flows across the fold at a rate similar to the heat flow within the flat-top. At the final time, the location of hottest temperature coincides with the hot spot for the initial condition. Finally, we note that it will take longer for this simulation to reach steady state due to its more isolated initial profile.

6. Conclusions

We propose an embedding method to solve diffusion on manifolds with folds. Our discretization is based on meshless methods. This approach has the advantage of eliminating the interpolation step in the underlying embedding method, thereby eliminating a source of error and computational cost. Moreover, the flexibility of data point placement is central to our approach for representing folds and the associated differential operators. In future work we will develop extensions of the method to more general operators and more general surfaces. Higher order accurate methods will also be developed.

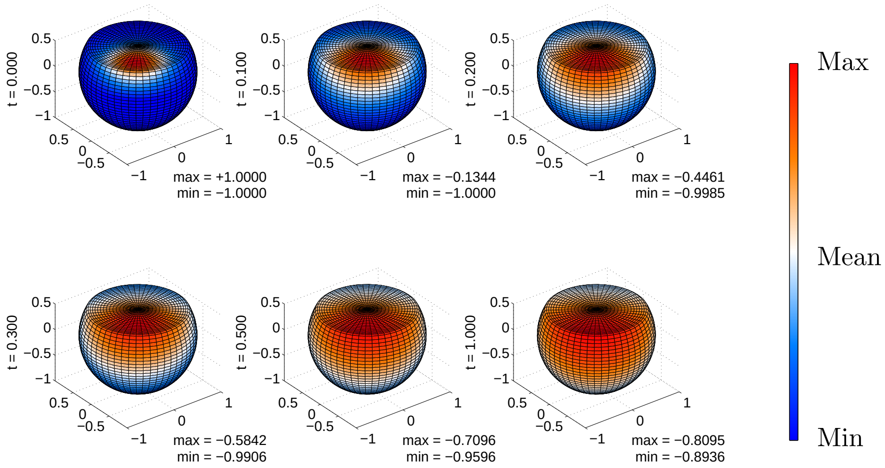


Figure 10: Simulation of diffusion on a flat-top sphere with an off-centered initial hot spot at the top.

References

[1] M. S. Floater, K. Hormann, Surface parameterization: a tutorial and survey, in: In Advances in Multiresolution for Geometric Modelling, Springer, 2005, pp. 157–186.

[2] L. Lui, Y. Wang, T. Chan, Solving PDEs on manifolds with global conformal parameterization, in: N. Paragios, O. Faugeras, T. Chan, C. Schnrr (Eds.), Variational, Geometric, and Level Set Methods in Computer Vision, Vol. 3752 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 307–319.

[3] J. Stam, Flows on surfaces of arbitrary topology, ACM Trans. Graph. 22 (3) (2003) 724–731.

[4] A. Witkin, M. Kass, Reaction-diffusion textures, SIGGRAPH Comput. Graph. 25 (4) (1991) 299–308.

[5] M. Bertalmo, L.-T. Cheng, S. Osher, G. Sapiro, Variational problems and partial differential equations on implicit surfaces, Journal of Computational Physics 174 (2) (2001) 759 – 780.

[6] J. B. Greer, An improvement of a recent Eulerian method for solving PDEs on general geometries, Journal of Scientific Computing 29 (3) (2006) 321–352.

[7] G. Turk, Generating textures on arbitrary surfaces using reaction-diffusion, SIGGRAPH Comput. Graph. 25 (4) (1991) 289–298.

[8] G. Dziuk, Finite elements for the Beltrami operator on arbitrary surfaces, in: S. Hildebrandt, R. Leis (Eds.), Partial Differential Equations and Calculus of Variations, Vol. 1357 of Lecture Notes in Mathematics, Springer Berlin Heidelberg, 1988, pp. 142–155.

[9] G. Dziuk, C. Elliott, Surface finite elements for parabolic equations, Journal of Computational Mathematics 25 (4) (2007) 385 – 407.

[10] N. Flyer, G. B. Wright, A radial basis function method for the shallow water equations on a sphere, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science 465 (2106) (2009) 1949–1976.

[11] E. Fuselier, G. Wright, Scattered data interpolation on embedded submanifolds with restricted positive definite kernels: Sobolev error estimates, SIAM Journal on Numerical Analysis 50 (3) (2012) 1753–1776.

[12] S. J. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, J. Comput. Phys. 227 (3) (2008) 1943–1961.

[13] C. Piret, The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces, Journal of Computational Physics 231 (14) (2012) 4662 – 4675.

[14] E. Larsson, B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions (2003).

[15] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, Computers and Mathematics with Applications 48 (56) (2004) 853 – 867.

[16] E. Larsson, B. Fornberg, A numerical study of some radial basis function based solution methods for elliptic PDEs, Comput. Math. Appl 46 (2003) 891–902.

[17] B. Fornberg, E. Larsson, N. Flyer, Stable computations with Gaussian radial basis functions, SIAM J. Sci. Comput. 33 (2) (2011) 869–892.

[18] E. Larsson, E. Lehto, A. Heryudono, B. Fornberg, Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions, SIAM Journal on Scientific Computing 35 (4) (2013) A2096–A2119.

[19] P. Rockstroh, T. Märx, S. Ruuth, An embedding technique for the solution of reaction-diffusion equations on algebraic surfaces with isolated singularities, ArXiv preprint arXiv:1211.5298.

[20] T. Cecil, J. Qian, S. Osher, Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions, Journal of Computational Physics 196 (2004) 327–347.

[21] M. Li, W. Chen, C. S. Chen, The localized RBFs collocation methods for solving high dimensional PDEs, Eng. Anal. Bound. Elem. 37 (10) (2013) 1300–1304.

[22] C. Shu, H. Ding, K. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 192 (7-8) (2003) 941 – 954.

[23] A. Tolstykh, On using RBF-based differencing formulas for unstructured and mixed structured-unstructured grid calculations, Proceedings of the 16 IMACS World Congress, Lausanne, 2000, pp. 1–6.

[24] G. Wright, Radial basis function interpolation: Numerical and analytical developments, Ph.D. thesis, University of Colorado (2003).

[25] R. Schaback, Direct discretizations with applications to meshless methods for PDEs, Dolomites Research Notes on Approximation 6 (2013) 27–50, proceedings of DWCAA12.

[26] G. F. Fasshauer, Meshfree Approximation Methods with MATLAB, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007.

[27] H. Wendland, Scattered Data Approximation, Cambridge University Press, 2004.

[28] R. Schaback, Error estimates and condition numbers for radial basis function interpolation, Advances in Computational Mathematics 3 (3) (1995) 251–264.

[29] R. Schaback, Multivariate interpolation and approximation by translates of a basis function, World Scientific Publishing Co., Inc, 1995, pp. 491–514.

[30] R. Schaback, A computational tool for comparing all linear PDE solvers, Advances in Computational Mathematics (2014) 1–23.

[31] B. Fornberg, E. Larsson, N. Flyer, Stable computations with Gaussian radial basis functions in 2-D, Tech. Rep. 2009-020, Department of Information Technology, Uppsala University (Aug. 2009).

[32] L. Ling, Pointonsphere, MATLAB Central File Exchange (2005).
URL `http://www.mathworks.com/matlabcentral/fileexchange/6977`