2-1-2013

A High-Order Kernel Method for Diffusion and Reaction-Diffusion Equations on Surfaces

Edward J. Fuselier
High Point University

Grady B. Wright
Boise State University

A High-Order Kernel Method for Diffusion and Reaction-Diffusion Equations on Surfaces

Edward J. Fuselier · Grady B. Wright

Abstract In this paper we present a high-order kernel method for numerically solving diffusion and reaction-diffusion partial differential equations (PDEs) on smooth, closed surfaces embedded in $\mathbb{R}^d$. For two-dimensional surfaces embedded in $\mathbb{R}^3$, these types of problems have received growing interest in biology, chemistry, and computer graphics to model such things as diffusion of chemicals on biological cells or membranes, pattern formations in biology, nonlinear chemical oscillators in excitable media, and texture mappings. Our kernel method is based on radial basis functions (RBFs) and uses a semi-discrete approach (or the method-of-lines) in which the surface derivative operators that appear in the PDEs are approximated using collocation. The method only requires nodes at "scattered" locations on the surface and the corresponding normal vectors to the surface. Additionally, it does not rely on any surface-based metrics and avoids any intrinsic coordinate systems, and thus does not suffer from any coordinate distortions or singularities. We provide error estimates for the kernel-based approximate surface derivative operators and numerically study the accuracy and stability of the method. Applications to different non-linear systems of PDEs that arise in biology and chemistry are also presented.

Edward J. Fuselier
Department of Mathematics and Computer Science
High Point University
High Point, NC 27262
E-mail: efuselie@highpoint.edu

Grady B. Wright (corresponding author)
Department of Mathematics
Boise State University
Boise, ID 83725-1555
E-mail: gradywright@boisestate.edu

1 Introduction

Kernel methods such as those based on radial basis functions (RBFs) are becoming increasingly popular for numerically solving partial differential equations (PDEs) because they are geometrically flexible, algorithmically accessible, and can be highly accurate. Since Kansa's pioneering work [42], there have been many successful applications of kernel methods to various types of PDEs defined on planar regions in two and higher dimensions (see, for example, [20, Ch. 38–45] and the references therein). More recently, these methods have been developed and applied to PDEs defined on the surface of a sphere [23, 25, 26, 30, 37].

In this paper, we present a high-order kernel method using RBFs for numerically solving certain PDEs defined on more general surfaces. In particular, we focus on diffusion and reaction-diffusion equations on smooth, closed embedded submanifolds $\mathbb{M} \subset \mathbb{R}^d$. In the case of two species, the prototypical form of the latter type of these PDEs is

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \delta_u \Delta_{\mathbb{M}} u + f_u(t, u, v), \\
\frac{\partial v}{\partial t} &= \delta_v \Delta_{\mathbb{M}} v + f_v(t, u, v),
\end{aligned}
\tag{1}
$$

where $u, v : \mathbb{M} \longrightarrow \mathbb{R}$, $\delta_u, \delta_v \geq 0$, f_u, f_v are (possibly non-linear) scalar functions, and $\Delta_{\mathbb{M}}$ is the Laplace-Beltrami operator for the surface. For two dimensional surfaces embedded in three dimensional space, systems like (1) have received growing interest to model such things as diffusion of chemicals on biological cells or membranes [59,60], pattern formations in biology [12,67], nonlinear chemical oscillators in excitable media [13,40,52], and texture mappings in computer graphics [65].

In the past decade, many methods have been developed for PDEs like (1) on surfaces. Nearly all of these techniques can be classified into two types, *intrinsic methods* and *embedded, narrow-band methods*. The former methods use coordinates intrinsic to the surface and a surface-based mesh to discretize the differential operators (e.g. [11,15–17,45,47,63,71]), while the latter methods extend the entire PDE in $\mathbb{R}^3$ in a narrow band around the surface and then modify the differential operators so that the solution is restricted to the surface (e.g. [2,8,9,14,39,49,50,57–60]). Intrinsic methods have the benefit that the resulting discretization scheme is consistent with the dimension of the original problem. However, properly dealing with the inherent coordinate distortions or singularities that arise in the metric terms of the surface differential operators can be difficult, and these methods are generally limited to low orders of accuracy. Embedded, narrow-band methods have the benefit that the surface differential operators are posed in extrinsic coordinates so that all coordinate singularities can be avoided and standard methods such as finite differences on 3D Cartesian grids, or finite elements on 3D unstructured meshes can be used. Additionally, the surface can be naturally represented using standard level-set methods, and thus quite topologically complicated surfaces can be handled. However, these methods require consistent extensions of the initial data on the surface to the embedding space, which can be non-trivial. Also, some of these methods (e.g. [9]) lead to degenerate surface differential operators since, for example, they allow diffusion to occur only in directions tangential to the surface. This makes it difficult to use implicit discretization techniques for time-dependent problems. Additionally, the need to implement artificial boundary conditions in

the embedding space can lead to accuracy degradations [39]. Finally, all embedded methods have an added computational expense since they solve the equations in a dimension at least one greater than that in which the equations are posed. This added expense can grow quite significantly depending on the surface, the order of the differential operators, and the order of accuracy of the method [49].

The present kernel method combines many of the benefits of both the intrinsic and narrow band methods. It uses a semi-discrete approach (or the method-of-lines) in which the surface derivative operators (e.g. surface Laplacian) that appear in the PDEs are approximated using collocation. These approximations are made using RBF interpolation on surfaces [35]. In this way, the method is similar to the ones used in [23, 25, 26, 30] for the surface of a sphere. However, the new method applies to more general surfaces, including those that can be defined parametrically or implicitly (as the level-set of a function). Since the method is based on RBFs, it allows the computational nodes to be placed at "scattered" locations on the surface (see Figure 1 for examples). Furthermore, it does not rely on any surface-based metrics or intrinsic coordinate systems, thus avoiding any coordinate distortions or singularities. In this regard, our kernel method is similar to the embedded, narrow-band methods discussed above. The difference is that our method approximates the surface differential operators directly on the surface instead of having to extend quantities off the surface and do the approximations in $\mathbb{R}^3$. Thus, like the intrinsic methods, our method is posed in the same dimension as the original problem. This ability to directly approximate the differential operator also means that no degeneracies arise from having to modify the equations so that the solutions remain restricted to the surface. Furthermore, computational efficiencies are gained by not having to extend into the full three dimensional space to solve the problems.

Recently, a different kernel method based on RBFs was introduced by Piret for solving PDEs on surfaces, which is called the Orthogonal Gradients (OGr) method [57]. This method shares many similarities to the embedded, narrow band method known as the Closest Point Method [49, 50, 58] and is different from our method in a number of ways. For example, the OGr method requires expanding into the embedding space ($\mathbb{R}^3$) to obtain discrete surface differential operators, which increases the computational complexity. Further, a differentiation matrix is constructed by enforcing that the first and second normal derivatives to the surface of a certain RBF interpolant vanish. This means in particular that one must analytically calculate the derivatives of the normal vectors to the surface, which requires that a smooth representation of the surfaces is known. In contrast, our method only requires nodes at scattered locations on the surface and the corresponding normal vectors to the surface, and does not require expanding into $\mathbb{R}^3$. Furthermore, the discrete operators for the present method are constructed from the inversion of a single traditional positive-definite kernel interpolation matrix, which is guaranteed to be invertible. Additionally, we provide a theory for the convergence of the discrete surface differential operators.

The application of kernel methods (including RBFs) to problems on general surfaces is still in its infancy. The goal of this paper is thus to investigate some of the theoretical and applied aspects of our kernel method and to establish its applicability. As a result, we (a) present error estimates for approximating various surface differential operators; (b) numerically investigate the stability and accuracy of the method, showing how solutions can be computed to very high orders

(a) Unit sphere   (b) Red blood cell   (c) "Bumpy sphere"

(d) Torus   (e) Dupin's cyclide   (f) Bretzel2
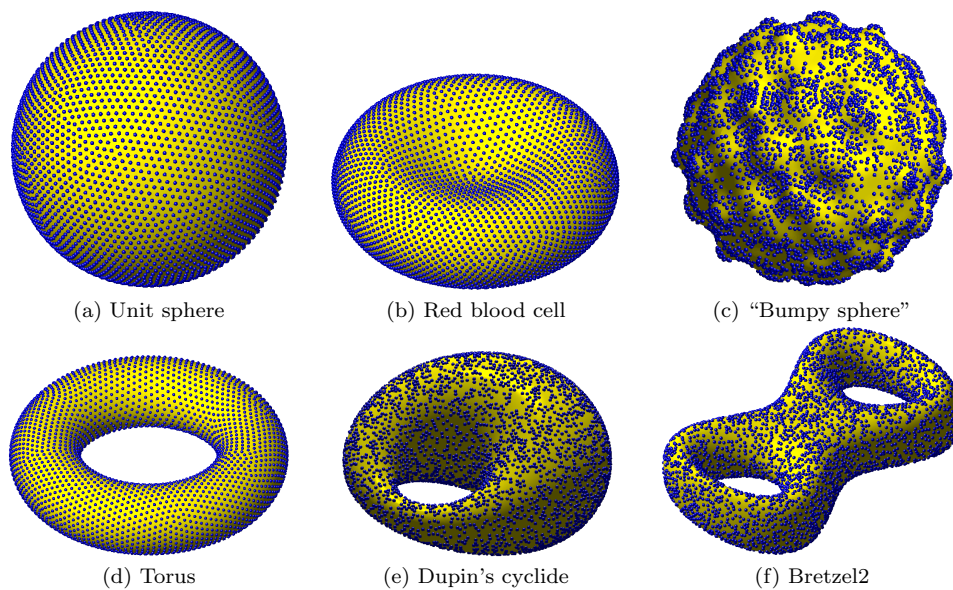
Fig. 1 Example surfaces and "scattered" nodes (represented as small solid spheres) used in the numerical experiments. A description of the surfaces and the node sets is given in Appendix B.

of accuracy; and (c) apply the method to two relevant problems in biology and chemistry.

The remainder of the paper is organized as follows. In next section, we briefly review kernel interpolation with RBFs. We follow this with a short presentation in Section 3 on how to formulate surface differential operators in Cartesian coordinates. Section 4 describes how these surface differential operators are discretized in the form of differentiation matrices and presents the kernel method in method-of-lines form. Making heavy use of the results from [35], we provide error estimates for the discrete surface derivatives in Section 5 (and proofs in Appendix A). Section 6 numerically investigates the eigenvalue stability of the kernel method. We numerically demonstrate the convergence of the method for the forced scalar diffusion equation on two different surfaces in Section 7. Applications of the method to simulations of pattern formations in a Turing system and to spiral waves in excitable media are presented in Section 8. We conclude with some comments on potential future enhancements of the method in Section 9.

Before continuing to the main body of the paper, we pause to make some relevant remarks on what follows. While there are many types of classifications of surfaces or manifolds, in our work, we use the term surface or manifold to always refer to a smooth embedded submanifold of $\mathbb{R}^d$ with no boundary. We use the notation $\mathbb{M}$ to denote the manifold. For simplicity, we present our method for manifolds of dimension 2 embedded in $\mathbb{R}^3$. However, it can be naturally extended to manifolds of codimension 1 embedded in $\mathbb{R}^d$. Although not pursued here, more technical, but straightforward extensions are possible for manifolds of higher codimension.

2 Brief overview of kernel interpolation

Fundamental to the construction of our approximate surface differential operators are kernel interpolants. Let $\Omega \subseteq \mathbb{R}^d$ and $\phi : \Omega \times \Omega \to \mathbb{R}$ be a continuous function, which we will refer to as a *kernel*. Given a set of "scattered" nodes $X = \{\mathbf{x}_j\}_{j=1}^N \subset \Omega$ and a continuous target function $f : \mathbb{R}^d \to \mathbb{R}$ sampled at X, a kernel interpolant takes the general form

$$I_\phi f(\mathbf{x}) = \sum_{j=1}^N c_j \phi(\mathbf{x}, \mathbf{x}_j), \ \mathbf{x} \in \Omega, \tag{2}$$

where c_j are determined by requiring $I_\phi f\big|_X = f\big|_X$. It is well-known that there are many kernels ϕ for which a unique solution to this problem exists. In particular, if for any non-zero vector $b \in \mathbb{R}^N$, ϕ satisfies

$$\sum_{i=1}^N \sum_{j=1}^N b_i \phi(\mathbf{x}_i, \mathbf{x}_j) b_j > 0, \tag{3}$$

then there exists a unique solution to (2). We call ϕ that satisfy (3) for all finite node sets $X \subset \Omega$ *positive definite kernels* on Ω.

In this study we focus on the popular subclass of positive definite kernels called *radial basis functions* (RBFs), which have the property that $\phi(\mathbf{x}, \mathbf{y}) = \phi(\|\mathbf{x} - \mathbf{y}\|)$, where $\| \cdot \|$ is the standard Euclidean norm. (Strictly speaking, there are RBFs that are only conditionally positive definite [20, Ch. 7], but our focus will be on the positive definite ones). For these radial kernels (2) becomes

$$I_\phi f(\mathbf{x}) = \sum_{j=1}^N c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) \tag{4}$$

and the interpolation constraints can be expressed as the following linear system:

$$\left[\ A_X \ \right] \underbrace{\begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix}}_{c_f} = \underbrace{\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix}}_{f_X}, \tag{5}$$

where $(A_X)_{i,j} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$. For a positive definite ϕ, this system is positive definite and hence non-singular.

We will use the following two popular RBFs in this study:

$$\underline{\text{Matérn:}} \quad \phi_\nu(r) = C_\nu (\varepsilon r)^{\nu - d/2} K_{\nu - d/2}(\varepsilon r), \ \nu > d/2, \ \varepsilon > 0, \tag{6}$$

$$\underline{\text{IMQ:}} \quad \phi(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}}, \ \varepsilon > 0, \tag{7}$$

which are positive definite on $\mathbb{R}^d$. In (6), $C_\nu = 2^{1-(\nu-d/2)}/\Gamma(\nu - d/2)$ and K_β is the modified Bessel function of the second kind of order β. The Matérn family of kernels are piecewise smooth, with the smoothness controlled by the parameter ν. For example, with $\nu = \frac{d+3}{2}$, $\phi_\nu \in C^2(\mathbb{R}^d)$ and with $\nu = \frac{d+5}{2}$, $\phi_\nu \in C^4(\mathbb{R}^d)$.

In contrast, the inverse multiquadric (IMQ) kernel is $C^\infty(\mathbb{R}^d)$. Infinitely smooth kernels like the IMQ are used often in solving PDEs, and we have included results for the Matérn family to illustrate the different convergence rates that are possible with finitely smooth kernels.

Typically, kernel, and in particular RBF, interpolation is applied to problems where the data are scattered in $\mathbb{R}^d$ or on the surface of the sphere. The present authors have recently proved that favorable error estimates can be achieved when RBF interpolation is applied to reconstruction problems on more general manifolds $\mathbb{M}$ of $\mathbb{R}^d$ [35]. It is worth noting, that the interpolants for these reconstruction problems are not modified from their original form (4), i.e. distances are still measured as straight line distances, and not as distances intrinsic to the manifold. Thus, the interpolation method needs only limited knowledge of the underlying manifold; it only requires computing the solution to the linear system (5). In what follows, we will use (4) to approximate the surface gradient and surface divergence over a given set of nodes $X \subset \mathbb{M}$ on the manifold, and combine these to obtain an approximation for the surface Laplacian. For this construction, we will only need a point set $X \subset \mathbb{M}$ and the normals to $\mathbb{M}$ at each point in X.

We conclude with some remarks on the parameter ε in (6) and (7), which is called the "shape parameter" since it can be used for changing the kernels from peaked (large ε) to flat (small ε). In general, better accuracy is obtained for smaller values of ε [20,32, Ch. 16–17]. However, the standard way of computing the interpolant by solving (5) becomes increasingly ill-conditioned. While several types of stable algorithms have been developed for bypassing this ill-conditioning [21,27, 29,31], these algorithms generally break down when the nodes X fall on a lower dimensional manifold of $\mathbb{R}^d$, unless the surface topology is taken into account in the algorithm (as done on the sphere in [29]). Modifications to these algorithms are underway to address this issue [46], but they are not yet generally available. As a result, in this study we will not give much attention to issues of the shape parameter, but will instead choose it in the applications to ensure the linear system (5) is reasonably well-conditioned.

3 Continuous surface differential operators

Here we assume that $\mathbb{M}$ is a smooth embedded submanifold of $\mathbb{R}^d$ with no boundary and $\dim(\mathbb{M}) = d - 1$. In the descriptions of the surface differential operators on $\mathbb{M}$ that follows, we will focus on the case of $d = 3$ for notational simplicity. The extension to other d should be obvious. Additionally, all expressions are given in extrinsic (or Cartesian) coordinates.

For any point $\mathbf{x} = (x, y, z)$ on $\mathbb{M}$, we denote the normal vector to $\mathbb{M}$ at $\mathbf{x}$ as $\mathbf{n} = (n^x, n^y, n^z)$ and the vector space of tangent vectors to $\mathbb{M}$ at $\mathbf{x}$ as $T_\mathbf{x}\mathbb{M}$. The surface gradient on $\mathbb{M}$ at $\mathbf{x}$ is then given as

$$\nabla_\mathbb{M} := \mathbf{P}\nabla = \left(\mathbf{I} - \mathbf{n}\mathbf{n}^T\right)\nabla \qquad (8)$$

Here the 3-by-3 matrix $\mathbf{P}$ projects vectors in $\mathbb{R}^3$ to $T_\mathbf{x}\mathbb{M}$. Letting $\mathbf{e}^x$, $\mathbf{e}^y$, $\mathbf{e}^z$, be the standard unit vectors in x, y, and z directions in $\mathbb{R}^3$, we can re-write (8) in

component form as

$$\nabla_{\mathbb{M}} := \begin{bmatrix} (\mathbf{e}^x \cdot \mathbf{P})\,\nabla \\ (\mathbf{e}^y \cdot \mathbf{P})\,\nabla \\ (\mathbf{e}^z \cdot \mathbf{P})\,\nabla \end{bmatrix} = \begin{bmatrix} (\mathbf{e}^x - n^x\mathbf{n}) \cdot \nabla \\ (\mathbf{e}^y - n^y\mathbf{n}) \cdot \nabla \\ (\mathbf{e}^z - n^z\mathbf{n}) \cdot \nabla \end{bmatrix} = \begin{bmatrix} \mathbf{p}^x \cdot \nabla \\ \mathbf{p}^y \cdot \nabla \\ \mathbf{p}^z \cdot \nabla \end{bmatrix} = \begin{bmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{bmatrix}. \tag{9}$$

With this notation, the surface divergence of a smooth vector field $\mathbf{f} = (f^x, f^y, f^z) : \mathbb{M} \to \mathbb{R}^3$ at a point $\mathbf{x} \in \mathbb{M}$ can be expressed as

$$\nabla_{\mathbb{M}} \cdot \mathbf{f} := (\mathbf{P}\nabla) \cdot \mathbf{f} = \mathcal{G}^x f^x + \mathcal{G}^y f^y + \mathcal{G}^z f^z. \tag{10}$$

Finally, the Laplace-Beltrami operator on $\mathbb{M}$ at $\mathbf{x}$ can be written using (9) and (10) as:

$$\Delta_{\mathbb{M}} := \nabla_{\mathbb{M}} \cdot \nabla_{\mathbb{M}} = (\mathbf{P}\nabla) \cdot (\mathbf{P}\nabla) = \mathcal{G}^x\mathcal{G}^x + \mathcal{G}^y\mathcal{G}^y + \mathcal{G}^z\mathcal{G}^z. \tag{11}$$

In what follows, we will approximate the Laplace-Beltrami operator in the form of (11) by replacing the continuous operator (9) with a discrete version based on the kernel interpolant (4).

4 Kernel collocation technique in method-of-lines form

The discrete approximations to the surface differential operators that will be used in the method-of-lines (MOL) are based on applying the projected gradient (9) to the kernel interpolant (4). We illustrate how to construct these approximations by first applying the projected gradient to a radial kernel. We then show how to express the discrete operators as *differentiation matrices*. This is followed by a description of the MOL formulation with these discrete operators for the diffusion equation. As in the previous section, we formulate everything in $\mathbb{R}^3$, with an obvious extension to other dimensions.

4.1 Projected gradient of a radial kernel

Let $\mathbf{x}$ and $\mathbf{x}_j$ be points on $\mathbb{M}$, and let $r_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_j\|$ denote the Euclidean distance from $\mathbf{x}$ to $\mathbf{x}_j$, i.e. $r_j(\mathbf{x}) = \sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2}$. With this notation, a radial kernel $\phi : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ centered at $\mathbf{x}_j$ is given by $\phi(r_j(\mathbf{x}))$. Under the assumption that $\phi(r_j(\mathbf{x}))$ is differentiable at $\mathbf{x}$, a simple application of the chain rule gives the gradient of $\phi(r_j(\mathbf{x}))$ as

$$\nabla\phi(r_j(\mathbf{x})) = \begin{bmatrix} \dfrac{\partial}{\partial x}\phi(r_j(\mathbf{x})) \\ \dfrac{\partial}{\partial y}\phi(r_j(\mathbf{x})) \\ \dfrac{\partial}{\partial z}\phi(r_j(\mathbf{x})) \end{bmatrix} = \begin{bmatrix} (x - x_j) \\ (y - y_j) \\ (z - z_j) \end{bmatrix} \frac{\phi'(r_j(\mathbf{x}))}{r_j(\mathbf{x})} = (\mathbf{x} - \mathbf{x}_j)\frac{\phi'(r_j(\mathbf{x}))}{r_j(\mathbf{x})}, \tag{12}$$

where we have used $'$ to denote differentiation with respect to r. Note that the apparent singularity in the above formula when $r_j(\mathbf{x}_j) = 0$ will analytically cancel since ϕ is assumed to be an even function with at least one continuous derivative.

The surface gradient of $\phi(r_j(\mathbf{x}))$ can then be obtained from (9). After applying (9), the components of the surface gradient of $\phi(r_j(\mathbf{x}))$ are

$$\mathcal{G}^x \phi(r_j(\mathbf{x})) = ((1 - n^x n^x)(x - x_j) - n^x n^y (y - y_j) - n^x n^z (z - z_j)) \frac{\phi'(r_j(\mathbf{x}))}{r_j(\mathbf{x})}, \quad (13)$$

$$\mathcal{G}^y \phi(r_j(\mathbf{x})) = ((1 - n^y n^y)(y - y_j) - n^x n^y (x - x_j) - n^y n^z (z - z_j)) \frac{\phi'(r_j(\mathbf{x}))}{r_j(\mathbf{x})}, \quad (14)$$

$$\mathcal{G}^z \phi(r_j(\mathbf{x})) = ((1 - n^z n^z)(z - z_j) - n^x n^z (x - x_j) - n^y n^z (y - y_j)) \frac{\phi'(r_j(\mathbf{x}))}{r_j(\mathbf{x})}. \quad (15)$$

We now have all the components necessary to build the RBF approximation of the surface gradient $\mathbf{P}\nabla$ of a scalar quantity defined on $\mathbb{M}$.

4.2 Discrete operators in terms of differentiation matrices

Let $f : \mathbb{M} \to \mathbb{R}$ be some differentiable target function known at a set of "scattered" node locations $X = \{\mathbf{x}_j\}_{j=1}^N \subset \mathbb{M}$. The first step in constructing a discrete approximation to the surface gradient of f on X is to construct an interpolant to f of the form (4). In the notation of this section, the interpolant takes the form

$$I_\phi f(\mathbf{x}) = \sum_{j=1}^N c_j \phi(r_j(\mathbf{x})), \quad (16)$$

where the coefficients c_j are determined by collocation. We next apply the projected gradient operator (9) to $I_\phi f$ and evaluate at the nodes in X. Focusing on the $\mathcal{G}^x$ component of this operator and using (13), we obtain

$$\left(\mathcal{G}^x I_\phi f(\mathbf{x})\right)\big|_{\mathbf{x}=\mathbf{x}_i} = \sum_{j=1}^N c_j \left(\mathcal{G}^x \phi(r_j(\mathbf{x}))\right)\big|_{\mathbf{x}=\mathbf{x}_i},$$

$$= \sum_{j=1}^N c_j \underbrace{\left((1 - n_i^x n_i^x)(x_i - x_j) - n_i^x n_i^y (y_i - y_j) - n_i^x n_i^z (z_i - z_j)\right) \frac{\phi'(r_j(\mathbf{x}_i))}{r_j(\mathbf{x}_i)}}_{(B_X^x)_{i,j}},$$

where $i = 1, \ldots, N$ and (n_i^x, n_i^y, n_i^z) is the normal vector at $\mathbf{x}_i$. We can write this approximation in terms of a matrix vector product using the notation from the linear system (5) as follows:

$$\left(\mathcal{G}^x I_\phi f\right)\big|_X = B_X^x c_f = (B_X^x A_X^{-1}) f_X = G_X^x f_X. \quad (17)$$

The matrix G_X^x is an N-by-N differentiation matrix that represents the discrete RBF approximation to the x-component of the projected gradient operator over the set of nodes in X. We can similarly obtain the discrete approximations to the y- and z-components of the projected gradient operator as follows:

$$\left(\mathcal{G}^y I_\phi f\right)\big|_X = (B_X^y A_X^{-1}) f_X = G_X^y f_X, \quad (18)$$

$$\left(\mathcal{G}^z I_\phi f\right)\big|_X = (B_X^z A_X^{-1}) f_X = G_X^z f_X, \quad (19)$$

where the entries in B_X^y and B_X^z are given by

$$(B_X^y)_{i,j} = \left((1 - n_i^y n_i^y)(y_i - y_j) - n_i^x n_i^y (x_i - x_j) - n_i^y n_i^z (z_i - z_j)\right) \frac{\phi'(r_j(\mathbf{x}_i))}{r_j(\mathbf{x}_i)},$$

$$(B_X^z)_{i,j} = \left((1 - n_i^z n_i^z)(z_i - z_j) - n_i^x n_i^z (x_i - x_j) - n_i^y n_i^z (y_i - y_j)\right) \frac{\phi'(r_j(\mathbf{x}_i))}{r_j(\mathbf{x}_i)},$$

for $i,j = 1,\ldots,N$. The full discrete approximation to the surface gradient operator $\nabla_{\mathbb{M}}$ is given in terms of (17)–(19) as the $3N$-by-N matrix

$$G_X = \begin{bmatrix} G_X^x \\ G_X^y \\ G_X^z \end{bmatrix}. \tag{20}$$

A discrete approximation to the surface divergence operator (10) can also be constructed from (17)–(19). Let $\mathbf{f} = (f^x, f^y, f^z) : \mathbb{M} \to \mathbb{R}^3$ be a smooth vector field and $\mathbf{f}_X = \begin{bmatrix} f_X^x & f_X^y & f_X^z \end{bmatrix}^T$ denote the $3N$-by-1 vector of samples of $\mathbf{f}$ on X. Then the discrete approximation of the surface divergence of $\mathbf{f}$ on X is given by

$$D_X \mathbf{f}_X = \begin{bmatrix} G_X^x & G_X^y & G_X^z \end{bmatrix} \mathbf{f}_X = G_X^x f_X^x + G_X^y f_X^y + G_X^z f_X^z. \tag{21}$$

Finally we can combine (20) and (21) to obtain a discrete approximation to the Laplace-Beltrami operator (11):

$$L_X = D_X G_X = G_X^x G_X^x + G_X^y G_X^y + G_X^z G_X^z. \tag{22}$$

The N-by-N differentiation matrix L_X represents our discrete RBF approximation to the surface Laplacian and is what we will use in the applications that follow. It is important to note that the construction of L_X only requires a node set $X \subset \mathbb{M}$ and the normal vectors to $\mathbb{M}$. Additionally, L_X is constructed using extrinsic coordinates and is thus free from any coordinate singularities (e.g. the pole singularity for the unit sphere $\mathbb{S}^2$). In the case of $\mathbb{M} = \mathbb{S}^2$, other studies have taken a different approach at approximating the surface Laplacian using kernel collocation (e.g. [37,70]). These studies apply the surface Laplacian directly instead of approximating it with projected gradient as we have done. For more general manifolds, it is non-trivial to work out the surface Laplacian in closed form, which is why we are advocating the use of the discrete approximation (22).

In Section 5, we show that our discrete approximations (20)–(22) converge to the continuous surface derivatives (9)–(11), respectively, as the nodes in X increases and "fill" the manifold $\mathbb{M}$.

Although the computational cost of constructing the discrete surface differential operators is $\mathcal{O}(N^3)$, these constructions are a preprocessing step and can be done once and stored. Applying these surface differential operators to a vector requires $\mathcal{O}(N^2)$ operations. In Section 9, we make some remarks on how these costs may be reduced.

4.3 MOL formulation

To describe the MOL formulation of our kernel method, we use the equation for diffusion of a scalar quantity u on a surface with a (non-linear) forcing term:

$$\frac{\partial u}{\partial t} = \delta \Delta_{\mathbb{M}} u + f(t, u). \tag{23}$$

Here $\delta > 0$ is the diffusion coefficient, $f(t, u)$ is the forcing term, and an initial value of u at time $t = 0$ is given. Letting $X = \{\mathbf{x}_j\}_{j=1}^{N} \subset \mathbb{M}$ and $u_X \in \mathbb{R}^N$ denote the vector containing the samples of u at the points in X, our method for (23) takes the form

$$\frac{d}{dt} u_X = \delta L_X u_X + f(t, u_X), \tag{24}$$

where L_X is the differentiation matrix (22) corresponding to our discrete approximation of $\Delta_{\mathbb{M}}$. This is a system of N coupled ODEs and, provided it is stable, can be advanced in time with a suitably chosen time-integration method (see the Sections 7 and 8 for examples). In Section 6, we numerically investigate the eigenvalue stability of the method.

   We conclude by noting that Appendix C contains example MATLAB code for generating the discrete surface Laplacian for the red blood cell surface according to the above method. It also simulates a coupled reaction-diffusion system that is described in more detail in Section 8.

5 Convergence results for the discrete surface derivatives

In this section we present the convergence results for our discrete differential operators. Given that these operators are essentially obtained by differentiating interpolants, our error estimates will rely on the approximation power of kernel interpolants. Recently, Sobolev error estimates for a wide variety of kernel interpolants (constructed from RBF kernels restricted to surfaces) were made available for smooth, embedded submanifolds of $\mathbb{R}^d$ [35], and these results can be used to derive the bounds in this section. Precisely stated theorems and their proofs are given in Appendix A for the interested reader. A summary of the results will be presented after we establish the necessary notation.

5.1 Definitions and notation

As is typical in scattered data approximation, the error estimates are given in terms of the *mesh norm* h and *separation radius* q of the point set X on $\mathbb{M}$, which are given by

$$h := \sup_{\mathbf{x} \in \mathbb{M}} \min_{\mathbf{x}_j \in X} d_{\mathbb{M}}(\mathbf{x}, \mathbf{x}_j) \qquad q := \frac{1}{2} \min_{\substack{\mathbf{x}_j \in X \\ \mathbf{x}_j \neq \mathbf{x}_i}} d_{\mathbb{M}}(\mathbf{x}_i, \mathbf{x}_j),$$

where $d_{\mathbb{M}}(\mathbf{x}, \mathbf{x}_j)$ is the geodesic distance between $\mathbf{x}$ and $\mathbf{x}_j$ on $\mathbb{M}$, i.e. it is the arclength of the shortest curve on $\mathbb{M}$ connecting $\mathbf{x}$ and $\mathbf{x}_j$. The uniformity of the data is also important - this is measured by the *mesh ratio* $\rho := h/q$.

Every positive definite kernel is the reproducing kernel for a space of continuous functions, often called the *native space* (see [20, Chapter 13] or [68, Chapter 10] for details). We will denote the native space of a scalar kernel ϕ with domain $\Omega \subseteq \mathbb{R}^d$ by $\mathcal{N}_\phi(\Omega)$. When determining the approximation power of a given kernel, there is a rich theory to draw from when the target function lies within $\mathcal{N}_\phi(\Omega)$.

The native spaces associated with all of the kernels we consider here are subsets of, and sometimes equal to, Sobolev spaces. We denote by $H^t(\mathbb{R}^d)$ the space of functions whose distributional derivatives up to order t are all in $L_2(\mathbb{R}^d)$, with the usual norm. We will ultimately consider Sobolev spaces on a manifold $\mathbb{M}$, where $\mathbb{M}$ is a 2-dimensional smooth embedded submanifold of $\mathbb{R}^3$. We denote by $H^t(\mathbb{M})$ the space of functions on $\mathbb{M}$ whose projections onto $\mathbb{R}^2$ by coordinate charts are all in the Sobolev space $H^t(\mathbb{R}^2)$ (see [35, Section 2]). Since our manifold is embedded in $\mathbb{R}^3$, a natural choice for vector-valued functions on $\mathbb{M}$ is $\mathbf{H}^t(\mathbb{M}) := (H^t(\mathbb{M}))^3$. The norm for a function $\mathbf{f} = (f^x, f^y, f^z) \in \mathbf{H}^t(\mathbb{M})$ is given by

$$\|\mathbf{f}\|_{\mathbf{H}^t(\mathbb{M})} := \left(\|f^x\|_{H^t(\mathbb{M})}^2 + \|f^y\|_{H^t(\mathbb{M})}^2 + \|f^z\|_{H^t(\mathbb{M})}^2 \right)^{1/2}.$$

The kernels that we use throughout the paper give rise to spaces of continuous Sobolev functions. Thus from this point on we assume that for some $\tau > 3/2$ the kernel ϕ satisfies

$$\widehat{\phi}(\omega) \le C(1 + \|\omega\|^2)^{-\tau}, \tag{25}$$

where $\widehat{\phi}$ is the Fourier transform of ϕ on $\mathbb{R}^3$ and C is a constant.[1] Radial kernels whose Fourier transforms decay exponentially, such as the Gaussian or IMQ, obviously satisfy this. For finitely smooth kernels, we will require the stronger condition that

$$c(1 + \|\omega\|^2)^{-\tau} \le \widehat{\phi}(\omega) \le C(1 + \|\omega\|^2)^{-\tau}, \tag{26}$$

where c and C are positive constants. Note that in particular the Matérn kernels (6) satisfy (26) since [68, p.133]

$$\widehat{\phi}_\nu(\omega) = (1 + \|\omega\|^2)^{-\nu}.$$

It can be shown that kernels satisfying (25) obey $\mathcal{N}_\phi(\mathbb{M}) \subseteq H^s(\mathbb{M})$, and that those satisfying (26) give $\mathcal{N}_\phi(\mathbb{M}) = H^s(\mathbb{M})$ [35, Theorem 3.3], where $s = \tau - 1/2$. The parameter s governs the smoothness of the native spaces intrinsic to the surface $\mathbb{M}$, so the error estimates that follow will often be given in terms of s.

5.2 Summary of convergence results

We need to define continuous analogues to our discrete differential operators. Recall that the operators G_X and D_X are designed by applying surface differential operators to an RBF interpolant. This prompts the following definitions:

$$G_\mathbb{M}f := \nabla_\mathbb{M} I_\phi f, \qquad D_\mathbb{M}\mathbf{f} := \nabla_\mathbb{M} \cdot I_\Phi \mathbf{f}, \qquad L_\mathbb{M}f := D_\mathbb{M}G_\mathbb{M}f,$$

---

[1] The condition $\tau > 3/2$ ensures that functions within the kernel's native space are continuous on $\mathbb{R}^3$.

where $I_\Phi \mathbf{f} = (I_\phi f^x, I_\phi f^y, I_\phi f^z)$ is the vector-valued kernel interpolant of $\mathbf{f}$. For any continuous functions f and $\mathbf{f}$, the functions above sampled on X agree with the associated discrete differential operator acting on the vector f_X or $\mathbf{f}_X$, e.g. $(L_\mathbb{M} f)|_X = L_X f_X$. To determine how well L_X approximates $\Delta_\mathbb{M}$, for example, we will derive estimates for the error $\|L_\mathbb{M} f - \Delta_\mathbb{M} f\|$ over the entire domain.

The error will be measured in the $L_2(\mathbb{M})$ and $L_\infty(\mathbb{M})$ norms, and we do this for a few reasons. Even though error is often measured empirically with an ∞-type norm, the current theoretical ∞-norm error estimates for many kernel approximation problems typically give slower rates than those observed experimentally. The observed ∞-norm rates tend to be closer to the L_2-type convergence rates, which are often optimal. Thus one can probably expect the faster $L_2(\mathbb{M})$ convergence rates in practice. With that, here is our first approximation result.

Result 1: First-order discrete operators. Suppose the kernel ϕ satisfies (25) with $\tau > 3/2 + 1$ and define $s = \tau - 1/2$. Then the following holds for all $f \in N_\phi(\mathbb{M})$ and $\mathbf{f} \in (N_\phi(\mathbb{M}))^3$.

$$\|G_\mathbb{M} f - \nabla_\mathbb{M} f\|_{L_2(\mathbb{M})} = \mathcal{O}(h^{s-1}), \quad \|G_\mathbb{M} f - \nabla_\mathbb{M} f\|_{L_\infty(\mathbb{M})} = \mathcal{O}(h^{s-2}),$$
$$\|D_\mathbb{M} \mathbf{f} - \nabla_\mathbb{M} \cdot \mathbf{f}\|_{L_2(\mathbb{M})} = \mathcal{O}(h^{s-1}), \quad \|D_\mathbb{M} \mathbf{f} - \nabla_\mathbb{M} \cdot \mathbf{f}\|_{L_\infty(\mathbb{M})} = \mathcal{O}(h^{s-2}).$$

For a proof, see Proposition 1, Appendix A. Note that for kernels such as the Gaussian and IMQ, which satisfy (25) for all $\tau > 0$, this implies convergence faster than any fixed algebraic order for target functions within the native space.

A key assumption in the above result is that the target function is within $\mathcal{N}_\phi(\mathbb{M})$. When the kernel is less smooth, we have the ability to provide error rates for many target functions *not* within $\mathcal{N}_\phi(\mathbb{M})$ - the only requirement is that the targets be in a Sobolev space smooth enough so that first-order derivatives are continuous (since $\dim(\mathbb{M}) = 2$, this means that $f \in H^\beta(\mathbb{M})$ for some $\beta > 2$). In this case the uniformity of the data sites, measured by the mesh ratio ρ, enters the error estimates. Proposition 2 in Appendix A implies the following.

Result 2: First-order discrete operators with finitely smooth kernels. Suppose ϕ satisfies (26) with $\tau > 3/2 + 1$, define $s = \tau - 1/2$, and let β satisfy $s \geq \beta > 2$. Then the following holds for all $f \in H^\beta(\mathbb{M})$ and $\mathbf{f} \in \mathbf{H}^\beta(\mathbb{M})$:

$$\|G_\mathbb{M} f - \nabla_\mathbb{M} f\|_{L_2(\mathbb{M})} = \mathcal{O}(h^{\beta-1}\rho^{s-\beta}), \quad \|G_\mathbb{M} f - \nabla_\mathbb{M} f\|_{L_\infty(\mathbb{M})} = \mathcal{O}(h^{\beta-2}\rho^{s-\beta}),$$
$$\|D_\mathbb{M} \mathbf{f} - \nabla_\mathbb{M} \cdot \mathbf{f}\|_{L_2(\mathbb{M})} = \mathcal{O}(h^{\beta-1}\rho^{s-\beta}), \quad \|D_\mathbb{M} \mathbf{f} - \nabla_\mathbb{M} \cdot \mathbf{f}\|_{L_\infty(\mathbb{M})} = \mathcal{O}(h^{\beta-2}\rho^{s-\beta}).$$

In addition, for very smooth target functions[2] these kernels give the following error rates:

$$\|G_\mathbb{M} f - \nabla_\mathbb{M} f\|_{L_2(\mathbb{M})} = \mathcal{O}(h^{2s-1}), \quad \|G_\mathbb{M} f - \nabla_\mathbb{M} f\|_{L_\infty(\mathbb{M})} = \mathcal{O}(h^{2s-2}),$$
$$\|D_\mathbb{M} \mathbf{f} - \nabla_\mathbb{M} \cdot \mathbf{f}\|_{L_2(\mathbb{M})} = \mathcal{O}(h^{2s-1}), \quad \|D_\mathbb{M} \mathbf{f} - \nabla_\mathbb{M} \cdot \mathbf{f}\|_{L_\infty(\mathbb{M})} = \mathcal{O}(h^{2s-2}).$$

Finding bounds on the discrete Laplacian is less straightforward - the repeated process of "interpolate, then differentiate" complicates matters. This will ultimately result in more factors of ρ in the estimates, even in the case of very smooth

---

[2] See the discussion preceeding Proposition 2 in Appendix A for details.

target functions. Unfortunately error bounds for the discrete Laplacian constructed from infinitely smooth kernels are currently unavailable. However, we will be able to provide high-order convergence rates in the discrete Laplacian for any kernel satisfying (26). The following comes from Theorems 1 and 2, Appendix A.

Result 3: Discrete Laplacian with finitely smooth kernels. Suppose ϕ satisfies (26) with $\tau > 3/2 + 2$, define $s = \tau - 1/2$, and let β satisfy $s \geq \beta > 3$. Then the following holds for all $f \in H^{\beta}(\mathbb{M})$ and $\mathbf{f} \in \mathbf{H}^{\beta}(\mathbb{M})$:

$$\|L_{\mathbb{M}}f - \Delta_{\mathbb{M}}f\|_{L_2(\mathbb{M})} = \mathcal{O}(h^{\beta-2}\rho^{2(s-\beta)+1}), \ \|L_{\mathbb{M}}f - \Delta_{\mathbb{M}}f\|_{L_{\infty}(\mathbb{M})} = \mathcal{O}(h^{\beta-3}\rho^{2(s-\beta)+1}).$$

In addition, for very smooth target functions[3] these kernels give the following error rates:

$$\|L_{\mathbb{M}}f - \Delta_{\mathbb{M}}f\|_{L_2(\mathbb{M})} = \mathcal{O}(h^{2s-2}\rho), \quad \|L_{\mathbb{M}}f - \Delta_{\mathbb{M}}f\|_{L_{\infty}(\mathbb{M})} = \mathcal{O}(h^{2s-3}\rho).$$
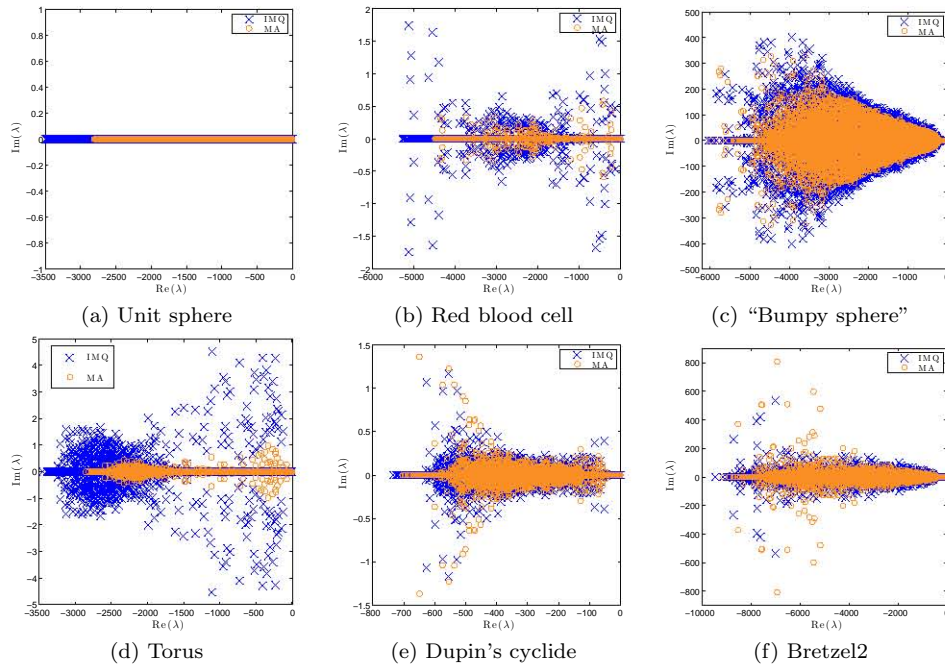
For solving time-dependent PDEs, one often is only concerned with approximating derivatives at the nodes since typically these are the only locations the numerical solution is known. If our point set X is always chosen in such a way that ρ is bounded by a fixed constant and f is any target function in a Sobolev space smooth enough to guarantee continuous second-order derivatives (that is, $f \in H^{\beta}(\mathbb{M})$ with $\beta > 3$), Result 3 guarantees that

$$\|L_X f_X - (\Delta_{\mathbb{M}}f)|_X\|_{l_{\infty}} \leq \|L_{\mathbb{M}}f - \Delta_{\mathbb{M}}f\|_{L_{\infty}(\mathbb{M})} = \mathcal{O}(h^{\beta-3}) \to 0.$$

Now that we have established spatial accuracy, we need to discuss the stability of numerical PDE solvers based on these discrete differential operators.

6 Eigenvalue stability

A necessary condition for stability of the MOL approach described in Section 4 is that the eigenvalues of the differentiation matrices L_X in (24) must be in the stability domain of the ODE solver used for advancing the system in time. We propose using either fully-implicit backward differentiation formulae (BDF) methods in the case where there is no forcing terms, and semi-implicit BDF methods (SBDF) in the case of forcing terms (see Sections 7 and 8 for details). This means that, at the very least, all eigenvalues must be in the left half plane. Unfortunately, the construction of the discrete approximation of $\Delta_{\mathbb{M}}$ does not guarantee that this property will hold for L_X. Fortunately however, we have found from numerous numerical experiments that provided the mesh norm h is small enough (i.e. the surface is well-discretized), then it appears that all of the eigenvalues do in fact lie in the left half plane. To illustrate these observations, in Figures 2(a)–(f) the eigenvalues of L_X are plotted for the surfaces and node sets depicted in Figure 1. Results are shown for both the IMQ kernel and the $\nu = 7$ Matérn kernel, which is $C^{10}(\mathbb{R}^3)$. While not presented here, similar results were obtained for the Matérn kernels with $\nu < 7$. The shape parameters, which are listed in the figure, have been chosen for this experiment so that the condition numbers of (5) are around 10^{10}. We find that even when the nodes are quite scattered (as in the case of the

(a) Unit sphere   (b) Red blood cell   (c) "Bumpy sphere"

(d) Torus   (e) Dupin's cyclide   (f) Bretzel2

Fig. 2 Eigenvalues of the discrete Laplacian for the surfaces and node sets shown in Figure 1. ×'s correspond to the IMQ kernel with shape parameters chosen as (a) $\varepsilon = 2.8$, (b) $\varepsilon = 4$, (c) $\varepsilon = 5$, (d) $\varepsilon = 2.8$, (e) $\varepsilon = 2$, and (f) $\varepsilon = 6.5$, while ○'s correspond to the $\nu = 7$ Matérn (MA) kernel with shape parameters chosen as (a) $\varepsilon = 8$, (b) $\varepsilon = 12$, (c) $\varepsilon = 15$, (d) $\varepsilon = 8$, (e) $\varepsilon = 6$, and (f) $\varepsilon = 16$.

Dupin's cyclide and Bretzel2) the eigenvalues all lie in the left half plane in these examples.

For node sets that are not sufficiently dense, we have found cases where some eigenvalues of the differentiation matrix shift over to the right half-plane. To illustrate this, we repeat the above experiment for the "bumpy sphere" with only $N = 1035$ nodes instead of $N = 5256$. Figure 3(a) shows this reduced node set on the "bumpy sphere", while part (b) of this figure shows the corresponding eigenvalues of the discrete Laplacian for both the $\nu = 7$ Matérn and IMQ kernels. The shape parameters here were chosen so that the condition numbers of (5) are similar to those from the $N = 5256$ experiments. We can see from this latter figure that the majority of the eigenvalues fall in the left half plane for both kernels, but that there are now a few that fall in the right half plane. Fortunately, these "unstable" eigenvalues appear to correspond to high frequency eigenvectors as illustrated in part (c) of Figure 3 for the Matérn kernel (a similar result holds for the IMQ kernel and the results are thus omitted). Unphysical eigenmodes of this type also appear in other contexts of kernel methods for PDEs [25, 26, 28]. For these applications, Fornberg and Lehto [28] describe a global hyperviscosity stabilization procedure for selectively shifting the unstable eigenvalues to the left half plane without a reduction in accuracy. Given the similarity of the unstable

---

[3] See Theorem 2 for details.
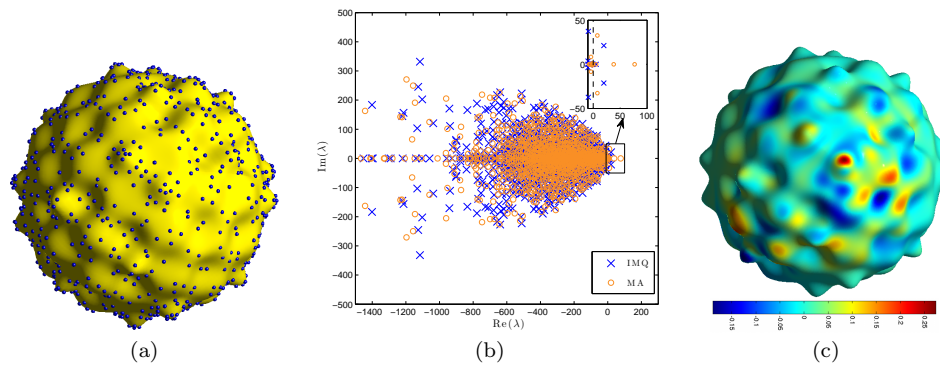
Fig. 3 (a) "Bumpy sphere" with $N = 1035$ nodes. (b) Eigenvalues of the discrete Laplacian corresponding to part (a). ×'s correspond to the IMQ kernel with shape parameter $\varepsilon = 2$, while ○'s correspond to the $\nu = 7$ Matérn (MA) kernel with shape parameter $\varepsilon = 6$. (c) Eigenvector corresponding to the eigenvalue of the discrete Laplacian with the largest real part shown in part (b) using the $\nu = 7$ Matérn kernel.

eigenmodes in the current problem, it seems reasonable to expect this hyperviscosity procedure to also be applicable in the present context. Alternatively, it is possible filter these unstable eigenmodes by doing a full eigenvalue decomposition of the differentiation matrix, zeroing out the eigenvalues with positive real part, and then reassembling the (modified) matrix. However, this procedure is more computationally expensive than the hyperviscosity approach (but still only needs to be done once as a preprocessing step). In the numerical experiments of this study that follow in the next section, all eigenvalues of discrete Laplacian lie in the left half plane. We leave exploration of stabilization strategies to a separate study.

7 Numerical convergence results

In this section we demonstrate the convergence of our numerical method for approximating the scalar diffusion equation with forcing (23) on two surfaces: the unit sphere $\mathbb{S}^2$ and the torus shown in Figure 1(d). In all the tests we use the standard fourth-order backward differentiation formulae (BDF4) method for advancing the approximate solution in time. The time-step is set to $\Delta t = 10^{-4}$ to ensure that spatial errors dominate over temporal errors.

Results for three kernels of varying smoothness are presented to illustrate the different convergence rates that are possible. The first two are the $\nu = 4$ and $\nu = 6$ kernels from the Matérn family (6), which are $C^4(\mathbb{R}^3)$ and $C^8(\mathbb{R}^3)$, respectively. The third kernel is the infinitely smooth IMQ (7). For the Matérn kernels, the analysis from Section 5 applies so we expect an algebraic rate of decay in the approximate solutions. While we don't have estimates on the approximation of the surface Laplacian for the IMQ kernel, the estimates from Result 1 in Section 5 on the discrete surface gradient and divergence suggest that this infinitely smooth kernel should give rates of approximation that are faster than any polynomial order (provided the underlying target is sufficiently smooth). We fix the shape parameters of the kernels in all the tests as follows: $\varepsilon = 4$ for $\nu = 4$ Matérn, $\varepsilon = 8$

for $\nu = 8$ Matérn, and $\varepsilon = 3$ for IMQ. These values have not been optimized and are chosen simply to illustrate the convergence rates of our method.

Finally, we measure the errors in the numerical solutions using both the discrete two-norm and max-norm. To define the discrete two-norm we use an approximation of the continuous $L_2(\mathbb{M})$-norm based on quadrature rules defined at the collocation nodes. We use the following abuse of notation to denote our discrete two-norm:

$$\|f\|_{L_2(\mathbb{M})} := \left(\int_{\mathbb{M}} [f(\mathbf{x})]^2 d\mathbf{x} \right)^{1/2} \approx \left(\sum_{i=1}^{N} w_i [f(\mathbf{x}_i)]^2 \right)^{1/2} := \|f\|_{\ell_2},$$

where $\{w_i\}_{i=1}^{N}$ are quadrature weights for the surface integral at the collocation nodes $\{\mathbf{x}_i\}_{i=1}^{N}$ on the manifold. These weights are computed using a second-order accurate midpoint rule approximation on the surface. We use the standard definition for discrete max-norm and denote it with the standard notation of ℓ_∞.
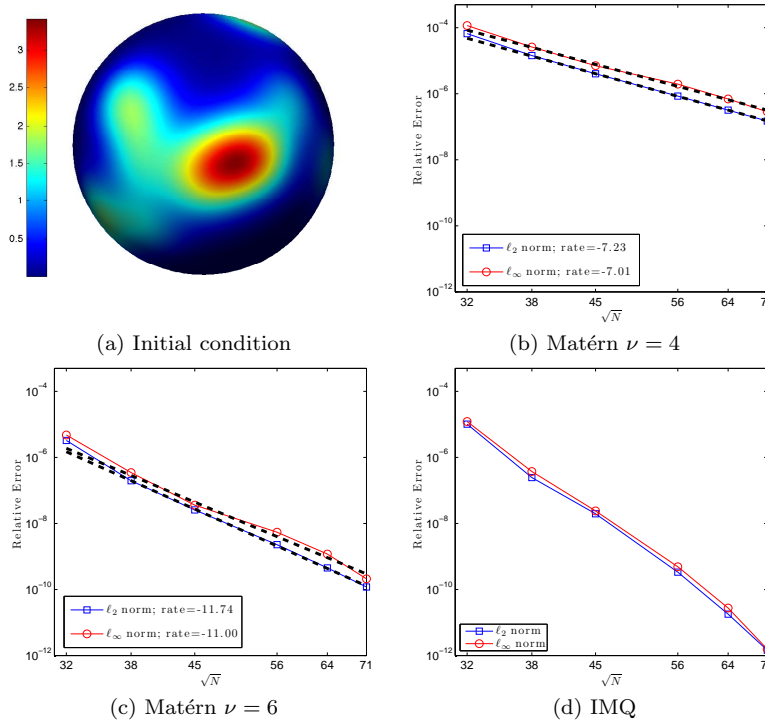
7.1 Diffusion on the unit sphere with forcing

For the first example, we use a test problem first presented in [11]. For this test, an artificial solution to (23) is specified (with $\delta = 1$) and the forcing function f is chosen so that this solution is maintained for all time. The solution is given by

$$u(t, \mathbf{x}) = \exp(-5t) \sum_{k=1}^{23} \exp\left(-10 \arccos(\boldsymbol{\xi}_k \cdot \mathbf{x})\right), \qquad (27)$$

where $\boldsymbol{\xi}_k$, $k = 1, \ldots, 23$ are randomly placed points on the surface of the sphere. The term in the sum is a Gaussian centered at $\boldsymbol{\xi}_k$ with the distance measured using the geodesic distance. The solution is clearly $C^\infty(\mathbb{S}^2)$ and its value at $t = 0$ is displayed in Figure 4(a). In our results, the forcing function corresponding to (27) is computed analytically and evaluated implicitly in time.

The test calls for comparing the errors in the approximate solution of (23) at time $t = 0.2$ at various spatial resolutions. For our spatial resolutions, we use minimum energy (ME) node sets on the unit sphere discussed in Appendix B.1. The different sizes of the node sets are $N = 1024, 1444, 2025, 3136, 4096$, and 5041, which are all perfect squares. Figures 4(b)–(d) display the results of the convergence study for these node sets and the three respective radial kernels. Since the mesh norm for the ME nodes satisfies $h \sim 1/\sqrt{N}$, we plot the relative errors vs. $\sqrt{N}$. All three convergence plots are on a log-log scale.

Figures 4(b) and (c) illustrate the algebraic convergence rate of the approximate solution for the two Matérn kernels. The black dashed line in these figures shows the lines of best fit to the data and the slope of the different lines is given in the legend. Since the solution is $C^\infty(\mathbb{S}^2)$, we expect the estimates from Result 3 to apply to the approximations of the surface Laplacian in this test problem. Note that node sets we are using are quasi-uniform, so the mesh ratio that appears in Result 3 will have little or no effect here. For the $\nu = 4$ Matérn kernel this result predicts the approximate surface Laplacian to converge like $\mathcal{O}(h^5)$ in the ℓ_2-norm and $\mathcal{O}(h^4)$ for the ℓ_∞-norm, while for the $\nu = 6$ Matérn kernel these convergence rates should be $\mathcal{O}(h^9)$ and $\mathcal{O}(h^8)$. We see from Figures 4(b) and (c), however, that the measured rates of convergence for the numerical solutions to the forced

Fig. 4 (a) Initial condition for the forced diffusion problem on the unit sphere. (b)–(c) Convergence results for the forced diffusion problem for three different radial kernels. These last three plots are displayed on a log-log scale with the horizontal axis being the square root of the number of collocation nodes N, which satisfies $\sqrt{N} \sim 1/h$. Black dashed lines in (b) and (c) are the lines of best fit to the last 4 values and the slopes of the lines are given in the legends.

diffusion problem (23) are much higher than the predicted rates for the surface Laplacian. One reason for these higher rates may be that the estimates from Result 3 apply to the entire manifold, while we are only measuring the errors at the collocation nodes (which are the only locations where the approximate solution is known). We may therefore be benefitting from some form of super-convergence, which is known to occur at the nodes in certain periodic spline methods [62]. A more detailed study of this phenomenon is given in [36].

Figure 4(d) displays the results for the IMQ kernel. The results in this plot suggest that the convergence of the approximate solution is converging at a rate faster than any polynomial order. Further numerical investigations not presented here seem to indicate that the solution is converging at an exponential rate. This type of convergence has also been observed for collocation methods based on infinitely smooth kernels for the scalar transport and nonlinear shallow water wave equations on the surface of the sphere [25, 26].

7.2 Diffusion on a torus with forcing

For the second example, we again consider computing the solution of (23), but with the torus shown in Figure 1(d) and described in Appendix B.4. Like the previous test, we specify the solution and then choose the forcing function so that this solution is maintained for all time. The solution is given by

$$u(t, \mathbf{x}) = \frac{1}{8} \exp(-5t) x \left(x^4 - 10x^2 y^2 + 5y^4 \right) \left(x^2 + y^2 - 60z^2 \right), \qquad (28)$$

where $\mathbf{x}$ is a point on the Torus; see Figure 5(a) for a plot of $u(0, \mathbf{x})$. We compute the forcing function analytically and evaluate it implicitly in the BDF4 scheme. A key ingredient to computing the forcing function is determining the surface Laplacian of $u(t, \mathbf{x})$. For convenience, we provide the equation for this below:

$$\Delta_{\mathbb{M}} u(t, \mathbf{x}) = -\frac{3}{8\rho^2} \exp(-5t) \left[x(x^4 - 10x^2 y^2 + 5y^4) \right.$$
$$\left. (10248\rho^4 - 34335\rho^3 + 41359\rho^2 - 21320\rho + 4000) \right], \quad (29)$$

where $\rho = \sqrt{x^2 + y^2}$.

As in the previous test, we compare the errors in the approximate solution of (23) at time $t = 0.2$ at various spatial resolutions. We also use ME node sets on the torus for the spatial discretization as discussed in Appendix B.4. The sizes of the different node sets are $N = 500, 750, 1000, 2000, 3000,$ and 4000. The results of the convergence study for these node sets and the three radial kernels are displayed in Figures 5(b)–(c). As in the previous test, the mesh norm for the ME nodes on the torus satisfies $h \sim 1/\sqrt{N}$, so we again plot the relative errors vs. $\sqrt{N}$.

Figures 5(b) and (c) display the results for the $\nu = 4$ and $\nu = 6$ Matérn kernels, respectively. Like the previous test, we clearly see the algebraic rates of convergence for these kernels, and the measured rates of convergence of the approximate solutions for both kernels are much higher than the predicted rates of convergence for the discrete Laplacian from Result 3. We again believe these higher rates may be related to our restriction of measuring the errors at the collocation nodes. Finally, we note that the measured convergence rates for the current test problem and the previous one are comparable, with only a slightly higher observed rate for the current one.

Figure 5(c) displays the results for the IMQ kernel. The results indicate the approximate solution is converging at a rate higher then any polynomial degree, which is entirely inline with the previous results. While not presented here, a further analysis of the numerical results indicate that the rate of convergence is again exponential for this kernel.

8 Applications

In the applications we consider systems of PDEs of the form (1). We use the third-order, semi-implicit, backward differentiation formulae (SBDF3) method discussed and analyzed in [4] for advancing the semi-discrete systems in time. This scheme treats the diffusion terms implicitly and the (non-linear) reaction terms explicitly. We bootstrap the simulation with one step of the first order SBDF followed by
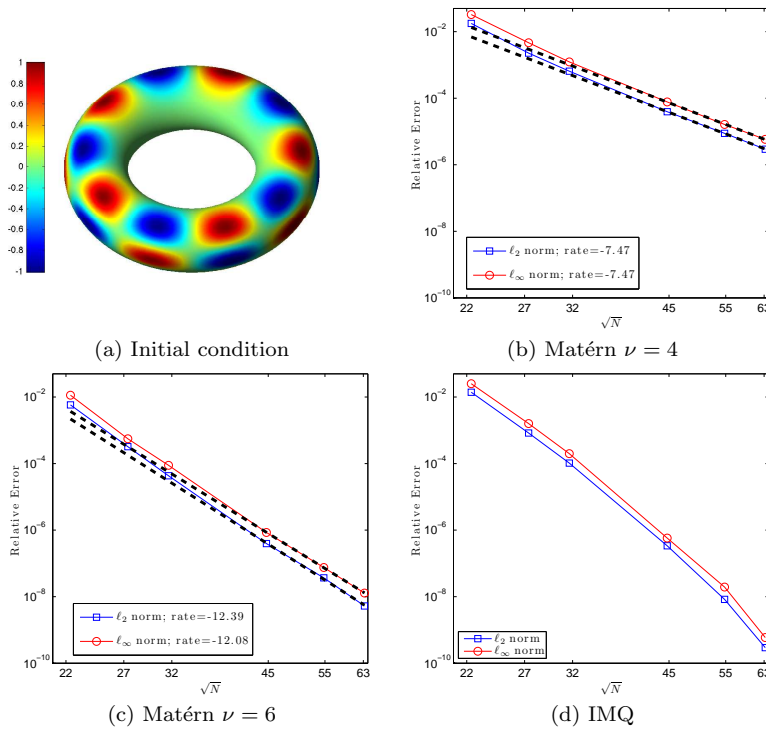
Fig. 5 (a) Initial condition for the forced diffusion problem on a torus. (b)–(c) Convergence results for the forced diffusion problem for three different radial kernels. These last three plots are displayed on a log-log scale with the horizontal axis being the square root of the number of collocation nodes N, which satisfies $\sqrt{N} \sim 1/h$. Black dashed lines in (b) and (c) are the lines of best fit to the last 3 values and the slopes of the lines are given in the legends.

one step of the second order SBDF (see [4] for details on these schemes). Upon initial LU decompositions of the linear systems that result from the implicit discretization, each time-step requires $\mathcal{O}(N^2)$ operations.

We only present results for the IMQ kernel using the same shape parameters as used in the stability tests (refer to the caption of Figure 2 for these values). However, similar results were obtained when using other shape parameters and when using the Matérn kernels.

8.1 Turing patterns

Since Turing's classical paper [64] that suggested how certain non-linear models of reaction and diffusion can lead to stable, heterogeneous pattern formations, there has been an explosion of research in reaction-diffusion-type models for various kinds of morphogenesis [44, 51, 53]. Most of the theoretical and numerical investigations of these models have been restricted to flat planar domains, but there is growing interest in studying these models on more general surfaces to understand how curvature may effect pattern formation. Additionally, there has been

interest in the computer graphics community for using Turing models to generate interesting textures and patterns on surfaces [65].

We illustrate how our method can be applied to these types of pattern formation problems by applying it to the Turing system (a linearized Brusselator model) from [7,67]:

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \delta_u \Delta_{\mathbb{M}} u + \underbrace{\alpha u(1 - \tau_1 v^2) + v(1 - \tau_2 u)}_{f_u(u,v)}, \\
\frac{\partial v}{\partial t} &= \delta_v \Delta_{\mathbb{M}} v + \underbrace{\beta v \left(1 + \frac{\alpha \tau_1}{\beta} uv\right) + u(\gamma + \tau_2 v)}_{f_v(u,v)}.
\end{aligned}
\tag{30}
$$

Here u and v are morphogens with u the "activator" and v is the "inhibitor". If $\alpha = -\gamma$ then $(u,v) = (0,0)$ is a unique equilibrium point of this system. By changing the diffusivity rates of u and v an instability can form that leads to different pattern formations. The cubic coupling parameter τ_1 favors the formation of stripes, while the quadratic coupling parameter τ_2 favors the formation spots [7]. The spot pattern formations are more robust than stripes and take far less time to reach "steady-state". The model (30) and similar models have been previously used to illustrate the applicability of some of the other numerical methods for PDEs on surfaces [8,9,11,50,57,58,65].

Figure 6 shows the results from our numerical solutions of (30) on various surfaces using parameters that lead to both spot and stripe patterns. Similar to the experiments on the surface of the sphere in [67], we set the initial values of u and v to random values between -0.5 and 0.5 in a thin strip around the "equator" of each surface and $u = v = 0$ elsewhere. Values for all parameters were also motivated from the values used in [67] and are listed in Table 1. A time-step of $\Delta t = 0.01$ was used in all the simulations and the numerical solutions were computed until steady-state was reached. The resulting stripe and spot patterns in Figure 6 are qualitatively similar to those obtained from other numerical methods [8,9,11,50, 58,65].

Appendix C contains MATLAB code for generating a spot pattern on the red blood cell similar to that shown in the top right picture of Figure 6.

| Surface/Pattern	δ_v	α	β	γ	τ_1	τ_2
Red blood cell/spots	$4.5 \cdot 10^{-3}$	0.899	-0.91	-0.899	0.02	0.2
Red blood cell/stripes	$2.1 \cdot 10^{-3}$	0.899	-0.91	-0.899	3.5	0
Bumpy sphere/spots	$4.5 \cdot 10^{-3}$	0.899	-0.91	-0.899	0.02	0.2
Bumpy sphere/stripes	$2.1 \cdot 10^{-3}$	0.899	-0.91	-0.899	3.5	0
Dupin's cyclide/spots	$4.5 \cdot 10^{-2}$	0.899	-0.91	-0.899	0.02	0.2
Dupin's cyclide/stripes	$1.89 \cdot 10^{-2}$	0.899	-0.91	-0.899	3.5	0
Bretzel2/spots	$2.1 \cdot 10^{-3}$	0.899	-0.91	-0.899	0.02	0.2
Bretzel2/stripes	$8.87 \cdot 10^{-4}$	0.899	-0.91	-0.899	3.5	0

Table 1 Values of the parameters of (30) used in the numerical experiments shown in Figure 6. In all cases $\delta_u = 0.516\delta_v$.
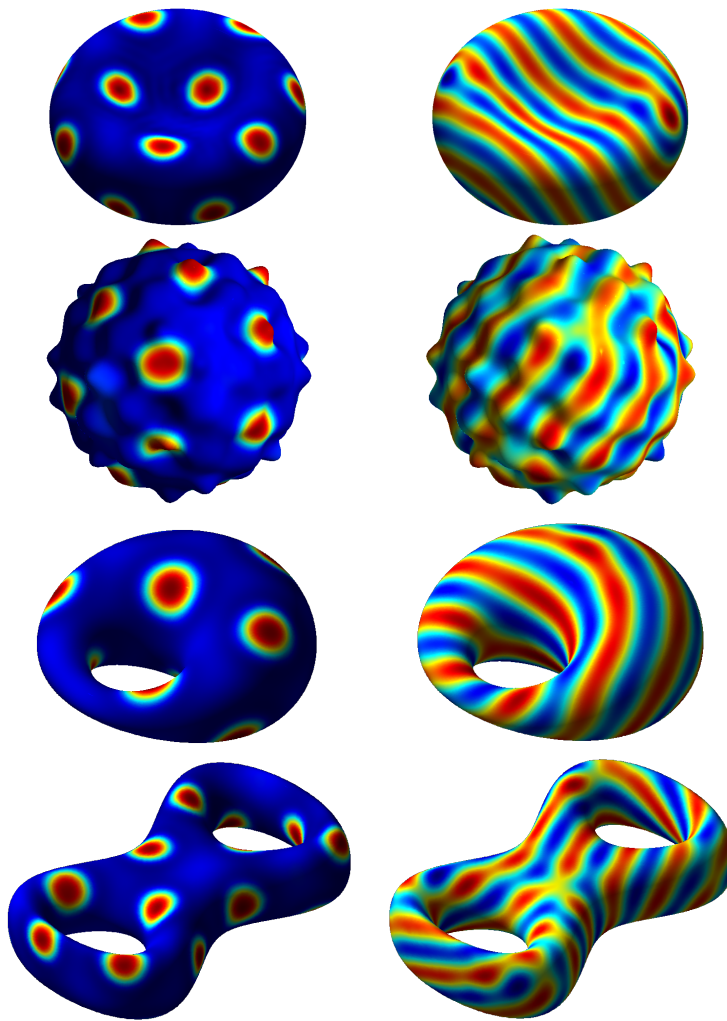
Fig. 6 Turing spot and stripe patterns computed from the model (30) on various surfaces. The pseudocolor plots are for the activator u once steady state is reached. In all plots red corresponds to a high concentration of u and blue to a low concentration. For all simulations the time-step was set to $\Delta t = 0.01$; parameters used to get the different patterns are given in Table 1.

8.2 Spiral waves in excitable media

Spiral waves can be observed in many excitable chemical, biological, and physical media [18, 43, 66]. Important examples include Belousov-Zhabotinsky chemical reactions and electrical activity in the membranes of organisms. While numerical methods have been developed for these models in planar domains (see [5, 6] and the corresponding software EZ-Spiral), there has been growing interest in studying these models on non-planar surfaces since most physically relevant problems occur on curved surfaces and curvatrue can effect the wave processes [13, 40, 52].
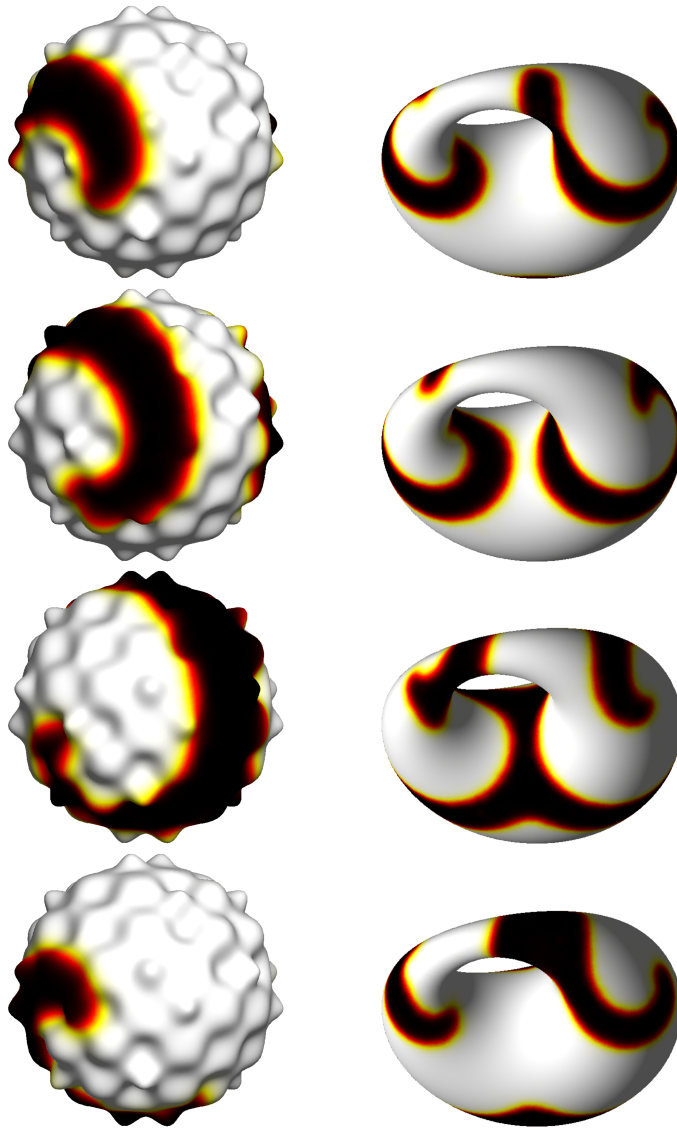
Fig. 7 Snap shots of spiral wave patterns computed from the Fitzhugh-Nagumo model (31) at different times. The left column shows the excitation variable u on the bumpy sphere at times $t = 42.24, 42.80, 43.76$ and 44.8. The right column shows u on Dupin's cyclide at times $t = 42.76, 43.48, 44.40$ and 45.24. The node sets used in the simulations are shown in Figure 1 (c) and (e), respectively, and the time-step was set to $\Delta t = 0.02$. The colors in the pseudocolor plots change linearly from white ($u = 0$) to black ($u = 1$)..

To illustrate how our kernel method may be applied to these models, we focus on

a simple two-variable reaction-diffusion model used in [5]:

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \delta_u \Delta_{\mathbb{M}} u + \underbrace{\frac{1}{\alpha} u \left(1 - u\right) \left(u - \frac{v + b}{a}\right)}_{f_u(u,v)}, \\
\frac{\partial v}{\partial t} &= \delta_v \Delta_{\mathbb{M}} v + \underbrace{u - v}_{f_v(u,v)},
\end{aligned}
\tag{31}
$$

where u and v can be viewed as some chemical concentrations or as membrane potential and current. The parameters a, b, and α govern the reaction kinetics and δ_u and δ_v are the diffusivities of the u and v species respectively. The parameter α is chosen as $\alpha << 1$ so that the u field takes on the values $u = 0$ or $u = 1$ almost everywhere, with a thin interface (or reaction zone) separating these two regions. The system (31) is of Fitzhugh-Nagumo type [22] and provides a simple model for dynamics of many excitable media. These types of systems have also been studied on the surface of the sphere [38,72].

Figure 7 displays snap shots from the numerical solution of (31) on the bumpy sphere and Dupin's cyclide at different times of the simulation. For all of these experiments we set the initial conditions to

$$
\begin{aligned}
u(0, \mathbf{x}) &= \frac{1}{2} \left[1 + \tanh(2x + y)\right], \\
v(0, \mathbf{x}) &= \frac{1}{2} \left[1 - \tanh(3z)\right],
\end{aligned}
$$

and $\Delta t = 0.02$. Values for the parameters that remain fixed for the two surfaces are as follows: $a = 0.75$, $b = 0.02$, $\alpha = 0.02$, and $\delta_v = 0$. The values of δ_u vary depending on the surface as follows: $\delta_u = 1.5(2\pi/50)^2$ for the bumpy sphere and $\delta_u = 2.5(2\pi/50)^2$. These values are similar to those used in one of the experiments from [6] on a 2-D planar domain. The two quasi-periodic, counter-rotating spiral waves seen on both surfaces in Figure 7 are qualitatively similar to those observed on the sphere in [38]. These spiral waves remain intact but meander around the surfaces throughout the simulation time.

9 Concluding remarks

In this paper, we have introduced a new kernel method based on collocation with RBFs for constructing discrete approximations to differential operators on surfaces, with a focus on surfaces of dimension 2 embedded in $\mathbb{R}^3$. The method is different than other methods for approximating surface differential operators in that it only requires "scattered" nodes on the surface and the corresponding normal vectors, it does not require expanding into the $\mathbb{R}^3$ embedding space, and it can give high-orders of accuracy. We have studied the approximation power of the method and have provided error estimates for target functions of various smoothness. We have used the method to construct discrete approximations to the surface Laplacian and combined it in a method-of-lines approach to numerically solve diffusion and reaction-diffusion equations on surfaces. We presented numerical results for two forced diffusion equations, illustrating the convergence of the method for

kernels with finite and infinite smoothness. The numerical results for the finitely smooth kernels indicate that the convergence rates are faster than our theory predicts, which may be related to some form of super-convergence common to spline methods. Finally, we have illustrated the flexibility of our method by successfully applying it to two important problems in biology and chemistry.

The primary shortcoming of the method is the $\mathcal{O}(N^2)$ computational cost for applying these operators. In a follow up study, we intend to address this issue by using a local RBF finite-difference (RBF-FD) method for constructing the approximate surface differential operators. This method has been used successfully in [24, 28] to overcome the computational cost of the global method [25, 26] for the shallow water wave equations on the sphere and has recently been adapted to multi-GPU architectures in [10]. We expect the cost of constructing and applying these discrete operators based on RBF-FD to be reduced to $\mathcal{O}(N)$, with a minor decrease in the order of accuracy. These enhancements may make the method competitive for solving diffusion and reaction-diffusion equations on evolving surfaces, which is also important in many biological applications.

A Convergence Results

In this section we present the convergence results for our discrete differential operators. The results presented here are for the special case when $\mathbb{M} \subset \mathbb{R}^3$ is a two dimensional surface, although similar results hold in higher dimensions. The arguments will depend heavily on error estimates for kernel interpolants on smooth, embedded submanifolds of $\mathbb{R}^d$ given recently in [35]. We will frequently reference results from that paper throughout this section. Sobolev error estimates for the surface gradient and divergence operators will immediately follow the results in [35]. However, the estimates for the discrete surface Laplacian will require more work.

Proposition 1 *Let $1 \leq q \leq \infty$, ϕ be a positive definite kernel satisfying (25) with $\tau > 3/2+1$, and define $s = \tau - 1/2$. Then there is a constant $h_{\mathbb{M}}$ depending only on $\mathbb{M}$ such that if a finite node set $X \subset \mathbb{M}$ satisfies $h \leq h_{\mathbb{M}}$, for all $f \in \mathcal{N}_\phi(\mathbb{M})$ and $\mathbf{f} \in (\mathcal{N}_\phi(\mathbb{M}))^3$ we have*

$$\|G_{\mathbb{M}}f - \nabla_{\mathbb{M}}f\|_{\mathbf{L}_q(\mathbb{M})} \leq Ch^{s-1-2(1/2-1/q)_+}\|f\|_{\mathcal{N}_\phi(\mathbb{M})},$$

$$\|D_{\mathbb{M}}\mathbf{f} - \nabla_{\mathbb{M}} \cdot \mathbf{f}\|_{L_q(\mathbb{M})} \leq Ch^{s-1-2(1/2-1/q)_+}\|\mathbf{f}\|_{\mathcal{N}_\phi(\mathbb{M})},$$

where $(x)_+ = x$ if $x \geq 0$ and is zero otherwise.

Proof We will focus on the discrete gradient operator. We have

$$\|G_{\mathbb{M}}f - \nabla_{\mathbb{M}}f\|_{\mathbf{L}_q(\mathbb{M})} = \|\nabla_{\mathbb{M}}I_\phi f - \nabla_{\mathbb{M}}f\|_{\mathbf{L}_q(\mathbb{M})} \leq C\|I_\phi f - f\|_{W_q^1(\mathbb{M})},$$

where $W_q^1(\mathbb{M})$ is the L_q Sobolev space of order 1 (see Section 2, [35]). The assumptions on τ allow us to apply Theorem 4.6 in [35], and the results follow. The error bounds in Theorem 4.6 in [35] carry over to the vector-valued case, thus bounds for the divergence operator error are obtained in a similar manner.

When ϕ satisfies (26), i.e. ϕ has finite smoothness, the estimates come in two types: the first concerns targets that may be too rough to be within the native space, and the second

applies to functions with additional smoothness. This additional smoothness is measured with the inverse of the integral operator[4]

$$Tf(x) := \int_{\mathbb{M}} \phi(y,x) f(y) d\mu(y).$$

We denote a vector version of this operator by $\mathbf{T}$, which simply applies T to each component of a 3-dimensional vector field. Similar to the proof above, the result below follows from Theorem 4.12 and Corollary 4.10 in [35].

Proposition 2 *Let* $1 \le q \le \infty$, *and let* ϕ *be a positive definite kernel satisfying* (26) *with* $\tau > 3/2 + 1$, *and define* $s = \tau - 1/2$. *Then there is a constant* $h_{\mathbb{M}}$ *depending only on* $\mathbb{M}$ *such that if a finite node set* $X \subset \mathbb{M}$ *satisfies* $h \le h_{\mathbb{M}}$, *we have the following:*

1. **Rough target functions**. *Let* β *be such that* $s \ge \beta > 2$. *Then for all* $f \in H^\beta(\mathbb{M})$ *and* $\mathbf{f} \in \mathbf{H}^\beta(\mathbb{M})$ *we have*

$$\|G_{\mathbb{M}} f - \nabla_{\mathbb{M}} f\|_{\mathbf{L}_q(\mathbb{M})} \le C h^{\beta - 1 - 2(1/2 - 1/q)_+} \rho^{s-\beta} \|f\|_{H^\beta(\mathbb{M})},$$

$$\|D_{\mathbb{M}} \mathbf{f} - \nabla_{\mathbb{M}} \cdot \mathbf{f}\|_{L_q(\mathbb{M})} \le C h^{\beta - 1 - 2(1/2 - 1/q)_+} \rho^{s-\beta} \|\mathbf{f}\|_{\mathbf{H}^\beta(\mathbb{M})}.$$

2. **Smooth target functions**. *For all* $f \in L_2(\mathbb{M})$ *such that* $T^{-1} f \in L_2(\mathbb{M})$ *and* $\mathbf{f} \in \mathbf{L}_2(\mathbb{M})$ *such that* $T^{-1} \mathbf{f} \in \mathbf{L}_2(\mathbb{M})$, *we have*

$$\|G_{\mathbb{M}} f - \nabla_{\mathbb{M}} f\|_{\mathbf{L}_q(\mathbb{M})} \le C h^{2s - 1 - 2(1/2 - 1/q)_+} \|T^{-1} f\|_{L_2(\mathbb{M})},$$

$$\|D_{\mathbb{M}} \mathbf{f} - \nabla_{\mathbb{M}} \cdot \mathbf{f}\|_{\mathbf{L}_q(\mathbb{M})} \le C h^{2s - 1 - 2(1/2 - 1/q)_+} \|\mathbf{T}^{-1} \mathbf{f}\|_{\mathbf{L}_2(\mathbb{M})}.$$

Obtaining convergence rates for the discrete Laplace operator is not as straightforward, and will require extra tools. The first is the "zeros lemma" from Narcowich, Ward and Wendland [54]. We will use the manifold version stated in [35, Lemma 4.5], with parameters adapted to our situation.

Proposition 3 *Let* $1 \le q \le \infty$, $t \in \mathbb{R}$ *with* $t > 1$. *Let* $\mu \in \mathbb{N}$ *satisfy* $0 \le \mu \le \lceil t - 2(1/2 - 1/q)_+ \rceil - 1$. *Also, let* $X \subset \mathbb{M}$ *be a discrete set with mesh norm* $h_{X,\mathbb{M}}$. *Then there is a constant depending only on* $\mathbb{M}$ *such that if* $h_{X,\mathbb{M}} \le C_{\mathbb{M}}$ *and if* $u \in H^t(\mathbb{M})$ *satisfies* $u|_X = 0$, *then*

$$|u|_{W_q^\mu(\mathbb{M})} \le C h^{t - \mu - 2(1/2 - 1/q)_+} |u|_{H^t(\mathbb{M})}.$$

Proposition 3 also holds with full Sobolev norms, and a similar result holds for vector fields.

Since the operator $L_{\mathbb{M}}$ is a mixture of differential and interpolation operators, it will be beneficial to bound the norm of the interpolation operator and its associated error in several different Sobolev spaces. In particular, we will use the following.

Lemma 1 *Let* ϕ *satisfy* (26) *with* $\tau > 3/2$, *and define* $s = \tau - 1/2$. *Let* β *be such that* $s \ge \beta > 1$. *Then there is a constant* $h_{\mathbb{M}}$ *depending only on* $\mathbb{M}$ *such that if a finite node set* $X \subset \mathbb{M}$ *satisfies* $h \le h_{\mathbb{M}}$ *then for all* $f \in H^\beta(\mathbb{M})$ *we have*

$$\|I_\phi f - f\|_{H^\beta(\mathbb{M})} \le C \rho^{s-\beta} \|f\|_{H^\beta(\mathbb{M})} \qquad and \qquad \|I_\phi f\|_{H^\beta(\mathbb{M})} \le C \rho^{s-\beta} \|f\|_{H^\beta(\mathbb{M})}.$$

Proof The last estimate in the proof of Theorem 4.12 in [35] is

$$\|f - I_\phi f\|_{H^\beta(\mathbb{M})} \le C \rho^{s-\beta} \|f\|_{H^\beta(\mathbb{M})}.$$

Applying a triangle inequality and observing that $1 \le \rho$, we get

$$\|I_\phi f\|_{H^\beta(\mathbb{M})} \le \|f - I_\phi f\|_{H^\beta(\mathbb{M})} + \|f\|_{H^\beta(\mathbb{M})} \le C(\rho^{s-\beta} + 1) \|f\|_{H^\beta(\mathbb{M})} \le C \rho^{s-\beta} \|f\|_{H^\beta(\mathbb{M})}.$$

---

[4] With this integral operator comes the pseudodifferential operators T^{-r}, $r > 0$. A function f is in the native space if and only if $T^{-1/2} f \in L_2(\mathbb{M})$ [35, Proposition 4.9], so we expect functions such that $T^{-1} f \in L_2(\mathbb{M})$ to be twice as smooth.

Now we are poised to present the following theorem.

Theorem 1 *Let $1 \le q \le \infty$, ϕ satisfy (26) with $\tau > 3/2 + 2$, and define $s = \tau - 1/2$. Let β be such that $s \ge \beta > 3$. Then there is a constant $h_{\mathbb{M}}$ depending only on $\mathbb{M}$ such that if a finite node set $X \subset \mathbb{M}$ satisfies $h \le h_{\mathbb{M}}$, then for all $f \in H^\beta(\mathbb{M})$ we have the estimate*

$$\|L_{\mathbb{M}} f - \Delta_{\mathbb{M}} f\|_{L_q(\mathbb{M})} \le C h^{\beta - 2 - 2(1/2 - 1/q)_+} \rho^{2(s-\beta)+1} \|f\|_{H^\beta(\mathbb{M})}. \tag{32}$$

Proof First, we have

$$\|L_{\mathbb{M}} f - \Delta_{\mathbb{M}} f\|_{L_q(\mathbb{M})} \le \|L_{\mathbb{M}} f - \Delta_{\mathbb{M}} I_\phi f\|_{L_q(\mathbb{M})} + \|\Delta_{\mathbb{M}} f - \Delta_{\mathbb{M}} I_\phi f\|_{L_q(\mathbb{M})}. \tag{33}$$

For the rightmost term, the assumption $\beta > 3$ allows us to use [35, Theorem 4.12] to get:

$$\|\Delta_{\mathbb{M}} f - \Delta_{\mathbb{M}} I_\phi f\|_{L_q(\mathbb{M})} \le C\|f - I_\phi f\|_{W_q^2(\mathbb{M})} \le C h^{\beta - 2 - 2(1/2 - 1/q)_+} \rho^{s-\beta} \|f\|_{H^\beta(\mathbb{M})}.$$

For the other term in (33), we have

$$\begin{aligned}
\|L_{\mathbb{M}} f - \Delta_{\mathbb{M}} I_\phi f\|_{L_q(\mathbb{M})} &= \|\nabla_{\mathbb{M}} \cdot (I_\Phi(\nabla_{\mathbb{M}} I_\phi f)) - \Delta_{\mathbb{M}} I_\phi f\|_{L_q(\mathbb{M})} \\
&= \|\nabla_{\mathbb{M}} \cdot (I_\Phi(\nabla_{\mathbb{M}} I_\phi f)) - \nabla_{\mathbb{M}} \cdot (\nabla_{\mathbb{M}} I_\phi f)\|_{L_q(\mathbb{M})} \\
&\le C\|I_\Phi(\nabla_{\mathbb{M}} I_\phi f) - \nabla_{\mathbb{M}} I_\phi f\|_{\mathbf{W}_q^1(\mathbb{M})}.
\end{aligned} \tag{34}$$

Note that the last estimate is the interpolation error for the target function $\mathbf{g} := \nabla_{\mathbb{M}} I_\phi f$.

We claim that $\mathbf{g}$ is in $\mathbf{H}^t(\mathbb{M})$ for all $t < 2s - 2$. Since ϕ satisfies (26), $I_\phi f \in H^\nu(\mathbb{R}^3)$ for all $\nu < 2\tau - 3/2$. By the trace theorem restricting to $\mathbb{M}$ puts $I_\phi f$ in $H^{\nu - 1/2}(\mathbb{M})$. Thus $\mathbf{g} \in \mathbf{H}^{\nu - 3/2}(\mathbb{M})$ for all $\nu < 2\tau - 3/2$, which is equivalent to $\mathbf{g} \in \mathbf{H}^t(\mathbb{M})$ for all $t < 2s - 2$. In particular, $\mathbf{g} \in \mathbf{H}^{\beta - 1}(\mathbb{M})$. Also, since $\beta > 2$ then we have

$$1 \le \lceil (\beta - 1) - 2(1/2 - 1/q)_+ \rceil - 1,$$

and that $\beta - 1 > 1$. This allows us to estimate (34) with Proposition 3 (with parameter $\mu = 1$ and target smoothness $t = \beta - 1$) to get

$$\begin{aligned}
\|I_\Phi(\nabla_{\mathbb{M}} I_\phi f) - \nabla_{\mathbb{M}} I_\phi f\|_{\mathbf{W}_q^1(\mathbb{M})} &\le C h^{(\beta - 1) - 1 - 2(1/2 - 1/q)_+} \|I_\Phi \mathbf{g} - \mathbf{g}\|_{\mathbf{H}^{\beta - 1}(\mathbb{M})}. \\
&= C h^{\beta - 2 - 2(1/2 - 1/q)_+} \|I_\Phi \mathbf{g} - \mathbf{g}\|_{\mathbf{H}^{\beta - 1}(\mathbb{M})}.
\end{aligned}$$

Further, since $\beta - 1 > 1$, we can also apply Lemma 1, and with two applications of it we get

$$\begin{aligned}
\|I_\Phi \mathbf{g} - \mathbf{g}\|_{\mathbf{H}^{\beta - 1}(\mathbb{M})} &\le C\rho^{s - \beta + 1} \|\mathbf{g}\|_{\mathbf{H}^{\beta - 1}(\mathbb{M})} = C\rho^{s - \beta + 1} \|\nabla_{\mathbb{M}} I_\phi f\|_{\mathbf{H}^{\beta - 1}(\mathbb{M})} \tag{35} \\
&\le C\rho^{s - \beta + 1} \|I_\phi f\|_{H^\beta(\mathbb{M})} \le C\rho^{2(s-\beta)+1} \|f\|_{H^\beta(\mathbb{M})}.
\end{aligned}$$

This completes the proof. $\qquad\blacksquare$

Continuing with our error analysis, we now shift our attention to target functions that are very smooth. First we will need a lemma.

Lemma 2 *Let ϕ satisfy (26) with $\tau > 3/2$, and let $s = \tau - 1/2$. Then for all $f \in L_2(\mathbb{M})$ such that $T^{-1} f \in L_2(\mathbb{M})$ we have the following estimate*

$$\|f - I_\phi f\|_{H^s(\mathbb{M})} \le C h^s \|T^{-1} f\|_{L_2(\mathbb{M})}.$$

Also, for all $\mathbf{f} \in \mathbf{L}_2(\mathbb{M})$ such that $\mathbf{T}^{-1} \mathbf{f} \in \mathbf{L}_2(\mathbb{M})$ we have

$$\|\mathbf{f} - I_\Phi \mathbf{f}\|_{\mathbf{H}^s(\mathbb{M})} \le C h^s \|\mathbf{T}^{-1} \mathbf{f}\|_{\mathbf{L}_2(\mathbb{M})}.$$

Proof The first estimate is established in the proof of Corollary 4.10 in [35]. By working component-wise the second estimate follows from the first.

Theorem 2 *Let $1 \le q \le \infty$, ϕ satisfy (26) with $\tau > 3/2 + 2$, and define $s = \tau - 1/2$. Let $f \in L_2(\mathbb{M})$ be such that $T^{-1}f \in L_2(\mathbb{M})$ and $\mathbf{T}^{-1}\nabla_\mathbb{M} f \in \mathbf{L}_2(\mathbb{M})$. Then there is a constant $h_\mathbb{M}$ depending only on $\mathbb{M}$ such that if a finite node set $X \subset \mathbb{M}$ satisfies $h \le h_\mathbb{M}$, we have the estimate*

$$\|L_\mathbb{M} f - \Delta_\mathbb{M} f\|_{L_q(\mathbb{M})} \le Ch^{2s-2-2(1/2-1/q)_+}\rho(\|T^{-1}f\|_{L_2(\mathbb{M})} + \|\mathbf{T}^{-1}\nabla_\mathbb{M} f\|_{\mathbf{L}_2(\mathbb{M})}).$$

Proof We begin as in the proof of the previous theorem. We have

$$\|L_\mathbb{M} f - \Delta_\mathbb{M} f\|_{L_q(\mathbb{M})} \le \|L_\mathbb{M} f - \Delta_\mathbb{M} I_\phi f\|_{L_q(\mathbb{M})} + \|\Delta_\mathbb{M} f - \Delta_\mathbb{M} I_\phi f\|_{L_q(\mathbb{M})}. \tag{36}$$

To estimate the rightmost term we can use the error estimates in [35, Corollary 4.10] to get:

$$\|\Delta_\mathbb{M} f - \Delta_\mathbb{M} I_\phi f\|_{L_q(\mathbb{M})} \le C\|f - I_\phi f\|_{W_q^2(\mathbb{M})} \le Ch^{2s-2-2(1/2-1/q)_+}\|T^{-1}f\|_{L_2(\mathbb{M})}.$$

For the other term in (36), we proceed as in the proof of Theorem 1 to get

$$\|L_\mathbb{M} f - \Delta_\mathbb{M} I_\phi f\|_{L_q(\mathbb{M})} \le C\|I_\Phi(\mathbf{g}) - \mathbf{g}\|_{\mathbf{W}_q^1(\mathbb{M})}, \tag{37}$$

where $\mathbf{g} = \nabla_\mathbb{M} I_\phi f$, and as before we know that this function is in $\mathbf{H}^t(\mathbb{M})$ for all $t < 2s - 2$. In particular, $\mathbf{g} \in \mathbf{H}^{s-1}(\mathbb{M})$ and we can estimate (37) with Proposition 3 (with $t = s - 1$ and $\mu = 1$) to get

$$\|I_\Phi(\mathbf{g}) - \mathbf{g}\|_{\mathbf{W}_q^1(\mathbb{M})} \le Ch^{s-2-2(1/2-1/q)_+}\|I_\Phi \mathbf{g} - \mathbf{g}\|_{\mathbf{H}^{s-1}(\mathbb{M})}.$$

This is where the proof detours from that of the previous theorem. To estimate $\|I_\Phi \mathbf{g} - \mathbf{g}\|_{\mathbf{H}^{s-1}(\mathbb{M})}$, note that it is bounded it by following quantity:

$$\underbrace{\|I_\Phi(\nabla_\mathbb{M} I_\phi f) - I_\Phi(\nabla_\mathbb{M} f)\|_{\mathbf{H}^{s-1}(\mathbb{M})}}_{I} + \underbrace{\|I_\Phi(\nabla_\mathbb{M} f) - \nabla_\mathbb{M} f\|_{\mathbf{H}^{s-1}(\mathbb{M})}}_{II} + \underbrace{\|\nabla_\mathbb{M} f - \nabla_\mathbb{M} I_\phi f\|_{\mathbf{H}^{s-1}(\mathbb{M})}}_{III}.$$

We will bound each term individually. First we concentrate on I. An application of Lemma 1, which we may apply since $s - 1 > 1$, gives us

$$\|I_\Phi(\nabla_\mathbb{M} I_\phi f) - I_\Phi(\nabla_\mathbb{M} f)\|_{\mathbf{H}^{s-1}(\mathbb{M})} = \|I_\Phi(\nabla_\mathbb{M} I_\phi f - \nabla_\mathbb{M} f)\|_{\mathbf{H}^{s-1}(\mathbb{M})}$$
$$\le C\rho\|\nabla_\mathbb{M} I_\phi f - \nabla_\mathbb{M} f\|_{\mathbf{H}^{s-1}(\mathbb{M})} = C\rho(III).$$

Thus a bound for III will result in a bound for I. To bound III, we can apply Lemma 2 to get

$$\|\nabla_\mathbb{M} I_\phi f - \nabla_\mathbb{M} f\|_{\mathbf{H}^{s-1}(\mathbb{M})} \le C\|I_\phi f - f\|_{\mathbf{H}^s(\mathbb{M})} \le h^s\|T^{-1}f\|_{L_2(\mathbb{M})}.$$

To bound II, we again employ Lemma 2:

$$\|I_\Phi(\nabla_\mathbb{M} f) - \nabla_\mathbb{M} f\|_{\mathbf{H}^{s-1}(\mathbb{M})} \le \|I_\Phi(\nabla_\mathbb{M} f) - \nabla_\mathbb{M} f\|_{\mathbf{H}^s(\mathbb{M})} \le Ch^s\|\mathbf{T}^{-1}\nabla_\mathbb{M} f\|_{\mathbf{L}_2(\mathbb{M})}.$$

This completes the proof.

B Surfaces and node sets from the numerical experiments

B.1 Unit sphere

This manifold is, of course, described implicitly by

$$\mathbb{M} = \left\{ \mathbf{x} = (x, y, z) \in \mathbb{R}^3 \;\middle|\; x^2 + y^2 + z^2 = 1 \right\}. \tag{38}$$

The node sets we use for discretizing the unit sphere are the minimum energy (ME) node sets of Womersley and Sloan [69]. These node sets are approximately uniformly distributed over the surface of the sphere and have the nice property that the mesh norm h and the separation radius q decrease uniformly like the inverse of the square root of the number of nodes N, i.e. $h, q \sim \frac{1}{\sqrt{N}}$. Additionally, the nodes in these sets are not oriented along any vertices or lines, which emphasizes the ability of our method to handle arbitrary node layouts. They have been used quite successfully in many other RBF applications, e.g. [25, 26, 28, 33, 34, 55, 70].

B.2 Red blood cell

This manifold is a mathematical model for human red blood cells in static equilibrium conditions. The model was first proposed in [19] and has been used in many subsequent studies (e.g. [48]). The model can be described parametrically as follows:

$$\mathbb{M} = \left\{ (x, y, z) \in \mathbb{R}^3 \;\middle|\; x = r_0 \cos\lambda \cos\theta, \; y = r_0 \sin\lambda \cos\theta, \; z = \frac{1}{2}\sin\theta \left(c_0 + c_2 \cos^2\theta + c_4 \cos^4\theta\right) \right\},$$

(39)

where $-\pi/2 \le \theta \le \pi/2$, $-\pi \le \lambda < \pi$, $r_0 = 3.91/3.39$, $c_0 = 0.81/3.39$, $c_2 = 7.83/3.39$, and $c_4 = -4.39/3.39$. The node sets we used for discretizing this manifold were obtained using a radial projection of the ME points for the surface of the sphere described above.

B.3 Bumpy sphere

This manifold is constructed from the "bumpy sphere" surface [3] using the following procedure. First, $N = 5256$ points on the bumpy sphere surface were obtained from `http://shapes.aimatshape.net`. This surface is homeomorphic to the unit sphere, so spherical coordinates for each of the $N = 5256$ points on the bumpy sphere were obtained. Upon normalizing the original nodes by the maximum distance of any of the nodes from the origin, a parametric model for the surface was constructed using the RBF geometric modeling technique from [61]. The original (normalized) $N = 5256$ points on the bumpy sphere are used as the computational nodes in all the numerical experiments and the parametric model is used for computing the normal vectors to the surface.

B.4 Torus

This manifold is given by the implicit equation:

$$\mathbb{M} = \left\{ \mathbf{x} = (x, y, z) \in \mathbb{R}^3 \;\middle|\; \left(1 - \sqrt{x^2 + y^2}\right)^2 + z^2 - \frac{1}{9} = 0 \right\}.$$

(40)

The node sets used for discretizing this manifold were obtained by arranging the nodes so that their Reisz energy (with a power of 2) is near minimal as described in Hardin and Saff's seminal article [41]. As with the ME sphere nodes, these ME torus nodes sets provide a near uniform discretization of the manifold with an approximately uniform decrease in the mesh norm h and the separation radius q. The node sets used in the experiments were kindly given to us by Drs. Douglas Hardin and Edward Saff and Ms. Ayla Gafni from Vanderbilt University.

B.5 Dupin's cyclide

This manifold is given by the implicit equation:

$$\mathbb{M} = \left\{ \mathbf{x} = (x, y, z) \in \mathbb{R}^3 \;\middle|\; \left(x^2 + y^2 + z^2 - d^2 + b^2\right)^2 - 4\left(ax + cd\right)^2 - 4b^2 y^2 = 0 \right\},$$

(41)

where $a = 2$, $b = 1.9$, $d = 1$, and $c^2 = a^2 - b^2$. The node set for discretizing this manifold was obtained from the algorithm of Palais, Palais, and Karcher (PPK) [56], which will be included in a future version of the amazing mathematical visualization software package 3-D-XplorMath [1]. This algorithm generates "uniformly random" point clouds for surfaces and works for both implicit and parametrically defined surfaces. To obtain the node set displayed in Figure 1(e) we used the PPK algorithm to generate a point cloud consisting of 9562 points. We then thinned this point set by removing points that were too close together in order to increase the separation radius. The final node set ended up at $N = 4948$. Both choices for the starting and ending number of nodes were chosen somewhat arbitrarily.

B.6 Bretzel2

This manifold is given by the implicit equation:

$$\mathbb{M} = \left\{ \mathbf{x} = (x, y, z) \in \mathbb{R}^3 \ \middle| \ \left(x^2(1-x^2) - y^2\right)^2 + \frac{1}{2}z^2 = \frac{1}{40} \right\}. \tag{42}$$

Unlike the the other surfaces, this one does not have an explicit parameterization. The node set for this domain was also obtained from the PPK algorithm. In this case we started with a point cloud of 10278 points and thinned it to end up with a node set of $N = 5041$ nodes.

C Example Matlab code

The code below simulates the Turing system (30) on the red blood cell surface using the parameters for spots in Table 1. The results should be similar to those shown in the top right plot of Figure 6. The code below uses SBDF2 as the numerical time integration method to simplify the presentation. For the code to work, the $N = 4096$ ME node set for the sphere must be downloaded from [69], or from the second authors' webpage `http://math.boisestate.edu/~wright/research/`.

```matlab
%% Setup surface, nodes, and radial kernel
% Red blood cell (RBC) surface
r0=3.39; c0=0.81/r0; c2=7.83/r0; c4=-4.39/r0; a=3.91/r0;
rbc = @(la,th) [a*cos(la).*cos(th) a*sin(la).*cos(th) ...
   0.5*sin(th).*(c0+c2*(cos(th)).^2+c4*(cos(th)).^4)];
syms la th; % Compute the surface normal vectors symbolically
nr=inline(cross(diff(rbc(la,th),la),diff(rbc(la,th),th)));
% Use minimum energy nodes projected to the RBC; need to download these
x=load('me63.4096'); x=x(:,1:3); N=size(x,1);
[la,th]=cart2sph(x(:,1),x(:,2),x(:,3)); x=rbc(la,th);
nr=nr(la,th); nr=nr./repmat(sqrt(sum(nr.^2,2)),[1 3]);
% Radial kernel
ep=4;  % Shape parameter
phi=@(r2) 1./sqrt(1+ep^2*r2);          % IMQ
dphi=@(r2) -ep^2./sqrt(1+ep^2*r2).^3; % Derivative of IMQ over r

%%  Compute the Surface Laplacian - not specific to RBC surface
xij=repmat(x(:,1),[1 N]); xij=xij-xij.'; nxi=repmat(nr(:,1),[1 N]);
yij=repmat(x(:,2),[1 N]); yij=yij-yij.'; nyi=repmat(nr(:,2),[1 N]);
zij=repmat(x(:,3),[1 N]); zij=zij-zij.'; nzi=repmat(nr(:,3),[1 N]);
r2=xij.^2 + yij.^2 + zij.^2; A=dphi(r2);
DPx=((1-nxi.^2).*xij - nxi.*nyi.*yij - nxi.*nzi.*zij).*A;
DPy=(-nxi.*nyi.*xij + (1-nyi.^2).*yij - nyi.*nzi.*zij).*A;
DPz=(-nxi.*nzi.*xij - nyi.*nzi.*yij + (1-nzi.^2).*zij).*A;
A=chol(phi(r2)); DPx=(DPx/A)/A.'; DPy=(DPy/A)/A.'; DPz=(DPz/A)/A.';
Lap=DPx*DPx+DPy*DPy+DPz*DPz;  % Surface Laplacian

%% Turing spot pattern example using SBDF2
del=0.0045; d=0.516; tau1=0.02; tau2=0.2; alp=0.899; bet=-0.91; gam=-alp;
ufun=@(u,v) alp*u.*(1-tau1*v.^2)+v.*(1-tau2*u);
vfun=@(u,v) bet*v.*(1+alp*tau1/bet*u.*v)+u.*(gam+tau2*v);
% Initial condition
stream=RandStream('mrg32k3a','seed',7122005); u=rand(stream,N,2)-0.5;
id=find(abs(x(:,3))>=0.1); u(id,:)=0; v=u(:,2); u=u(:,1);
% Implicit systems for SBDF2
dt=0.05; tfinal=400;
Du=1.5*eye(N)-del*d*dt*(Lap); [Lu,Uu,pu]=lu(Du,'vector');
Dv=1.5*eye(N)-del*dt*(Lap);   [Lv,Uv,pv]=lu(Dv,'vector');
optsu.UT=true; optsl.LT=true;
```

```matlab
% One step of SBDF1 to bootstrap SBDF2
rhsu=u+dt*ufun(u,v); rhsv=v+dt*vfun(u,v);
u0=u; u=(eye(N)-del*d*dt*Lap)\rhsu;
v0=v; v=(eye(N)-del*dt*Lap)\rhsv;
for j=1:tfinal/dt    % SBDF2
   rhsu=2*u-0.5*u0+dt*(2*ufun(u,v)-ufun(u0,v0));
   rhsv=2*v-0.5*v0+dt*(2*vfun(u,v)-vfun(u0,v0));
   u0=u; u=linsolve(Uu,linsolve(Lu,rhsu(pu),optsl),optsu);
   v0=v; v=linsolve(Uv,linsolve(Lv,rhsv(pv),optsl),optsu);
end

%% Interpolate the solution to a grid on the RBC surface using RBFs and plot
sz=[101 201]; M = prod(sz);  % surface grid parameters
[ll,tt]=meshgrid(linspace(-pi,pi,sz(2)),linspace(-pi/2,pi/2,sz(1)));
xx=rbc(ll(:),tt(:));
% Interpolate to the grid
re2=(repmat(xx(:,1),[1 N])-repmat(x(:,1).',[M 1])).^2;
re2=re2+(repmat(xx(:,2),[1 N])-repmat(x(:,2).',[M 1])).^2;
re2=re2+(repmat(xx(:,3),[1 N])-repmat(x(:,3).',[M 1])).^2;
uu=reshape(phi(re2)*(A\(A.'\u)),sz);
% Plot the results
yy=reshape(xx(:,2),sz); zz=reshape(xx(:,3),sz); xx=reshape(xx(:,1),sz);
surf(xx,yy,zz,uu); shading interp; daspect([1 1 1]); axis tight;
```

References

1. 3DXM Consortium: 3D-XplorMath. Website. http://3D-XplorMath.org
2. Adalsteinsson, D., Sethian, J.A.: Transport and diffusion of material quantities on propagating interfaces via level set methods. J. Comput. Phys. **185**(1), 271–288 (2003). DOI http://dx.doi.org/10.1016/S0021-9991(02)00057-8
3. AIM@SHAPE: "Bumpy sphere" from AIM@SHAPE shape repository. Website. http://shapes.aimatshape.net. 2011.
4. Ascher, U.M., Ruuth, S.J., Wetton, B.T.R.: Implicit-explicit methods for time-dependent partial differential equations. SIAM J. Numer. Anal. **32**, 797–823 (1995)
5. Barkley, D.: A model for fast computer simulation of waves in excitable media. Physica D **49**, 61–70 (1991)
6. Barkley, D.: Barkley model. Scholarpedia **3**(10), 1877 (2008)
7. Barrio, R., Varea, C., Aragón, J., Maini, P.: A two-dimensional numerical study of spatial pattern formation in interacting Turing systems. Bull. Math. Biol. **61**, 483–505 (1999)
8. Bergdorf, M., Sbalzarini, I., Koumoutsakos, P.: A Lagrangian particle method for reaction-diffusion systems on deforming surfaces. Journal of Mathematical Biology **61**, 649–663 (2010). URL http://dx.doi.org/10.1007/s00285-009-0315-2. 10.1007/s00285-009-0315-2
9. Bertalmío, M., Cheng, L., Osher, S., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces. J. Comput. Phys. **174**, 759–780 (2001)
10. Bollig, E.F., Flyer, N., Erlebacher, G.: Using radial basis function finite difference (RBF-FD) for PDE solutions on the GPU (2011). Submitted.
11. Calhoun, D.A., Helzel, C.: A finite volume method for solving parabolic equations on logically Cartesian curved surface meshes. SIAM Journal on Scientific Computing **31**(6), 4066–4099 (2009). DOI 10.1137/08073322X. URL http://link.aip.org/link/?SCE/31/4066/1
12. Chaplain, M.A., Ganesh, M., Graham, I.G.: Spatio-temporal pattern formation on spherical surfaces: numerical simulation and application to solid tumour growth. J. Math. Biol. **42**, 387–423 (2001). DOI 10.1007/s002850000067
13. Davydov, V.A., Manz, N., Steinbock, O., Zykov, V.S., Müller, S.C.: Excitation fronts on a periodically modulated curved surface. Phys. Rev. Lett. **85**, 868–871 (2000). DOI 10.1103/PhysRevLett.85.868. URL http://link.aps.org/doi/10.1103/PhysRevLett.85.868
14. Deckelnick, K., Dziuk, G., Elliott, C.M., Heine, C.: An *h*-narrow band finite-element method for elliptic equations on implicit surfaces. IMA Journal of Numerical Analysis **30**(2), 351–376 (2010). DOI 10.1093/imanum/drn049

15. Dziuk, G.: Finite elements for the Beltrami operator on arbitrary surfaces. In: S. Hilde-brandt, R. Leis (eds.) Partial Differential Equations and Calculus of Variations, Lecture Notes in Mathematics, vol. 1357. Springer, Berlin (1988)
16. Dziuk, G., Elliot, C.: Finite elements on evolving surfaces. IMA J. Num. Anal. **27**, 262–292 (2007)
17. Dziuk, G., Elliot, C.M.: Surface finite elements for parabolic equations. J. Comp. Math. **25**, 385–407 (2007)
18. Epstein, I.R., Pojman, J.A.: An Introduction to Nonlinear Chemical Dynamics. Oxford Univ. Press, Oxford (1998)
19. Evans, E., Fung, Y.C.: Improved measurements of the erythrocyte geometry. Microvascular Research **4**(4), 335 – 347 (1972). DOI DOI:10.1016/0026-2862(72)90069-6. URL `http://www.sciencedirect.com/science/article/pii/0026286272900696`
20. Fasshauer, G.E.: Meshless Approximation Methods with MATLAB. Interdisciplinary Mathematical Sciences - Vol. 6. World Scientific Publishers, Singapore (2007)
21. Fasshauer, G.E., McCourt, M.J.: Stable evaluation of Gaussian radial basis function interpolants. SIAM Journal on Scientific Computing **34**(2), A737–A762 (2012). DOI 10.1137/110824784. URL `http://link.aip.org/link/?SCE/34/A737/1`
22. FitzHugh, R.: Fitzhugh-nagumo simplified cardiac action potential model. Biophys. J. **1**, 445–466 (1961)
23. Flyer, N., Lehto, E.: Rotational transport on a sphere: Local node refinement with radial basis functions. J. Comput. Phys. **229**, 1954–1969 (2010)
24. Flyer, N., Lehto, E., Blaise, S., Wright, G.B., St-Cyr, A.: A guide to RBF-generated finite differences for nonlinear transport: shallow water simulations on a sphere. J. Comput. Phys. **231**, 4078–4095 (2012)
25. Flyer, N., Wright, G.B.: Transport schemes on a sphere using radial basis functions. J. Comp. Phys. **226**, 1059–1084 (2007)
26. Flyer, N., Wright, G.B.: A radial basis function method for the shallow water equations on a sphere. Proc. Roy. Soc. A **465**, 1949–1976 (2009)
27. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial functions. SIAM J. Sci. Comput. **33**, 869–892 (2011)
28. Fornberg, B., Lehto, E.: Stabilization of RBF-generated finite difference methods for convective PDEs. J. Comput. Phys. **230**, 2270–2285 (2011)
29. Fornberg, B., Piret, C.: A stable algorithm for flat radial basis functions on a sphere. SIAM J. Sci. Comput. **200**, 178–192 (2007)
30. Fornberg, B., Piret, C.: On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere. J. Comput. Phys. **227**, 2758–2780 (2008)
31. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. Comput. Math. Appl. **48**, 853–867 (2004)
32. Fornberg, B., Wright, G., Larsson, E.: Some observations regarding interpolants in the limit of flat radial basis functions. Comput. Math. Appl. **47**, 37–55 (2004)
33. Fuselier, E., Narcowich, F., Ward, J., Wright, G.: Error and stability estimates for divergence-free RBF interpolants on the sphere. Math. Comp. **78**, 2157–2186 (2009)
34. Fuselier, E.J., Wright, G.B.: Stability and error estimates for vector field interpolation and decomposition on the sphere with rbfs. SIAM J. Numer. Anal. **47**(5), 3213–3239 (2009). DOI http://dx.doi.org/10.1137/080730901
35. Fuselier, E.J., Wright, G.B.: Scattered Data Interpolation on Embedded Submanifolds with Restricted Positive Definite Kernels: Sobolev Error Estimates. SIAM Journal on Numerical Analysis **50**(3), 1753–1776 (2012). DOI 10.1137/110821846. URL `http://epubs.siam.org/doi/abs/10.1137/110821846`
36. Fuselier, E.J., Wright, G.B.: Order preserving approximations of derivatives with periodic radial basis functions (2013). Submitted
37. Gia, Q.T.L.: Approximation of parabolic PDEs on spheres using spherical basis functions. Adv. Comput. Math. **22**, 377–397 (2005)
38. Gomatam, J., Amdjadi, F.: Reaction-diffusion equations on a sphere: meandering of spiral waves. Phys. Rev. E **56**, 3913–3919 (1997). DOI 10.1103/PhysRevE.56.3913. URL `http://link.aps.org/doi/10.1103/PhysRevE.56.3913`
39. Greer, J.B.: An improvement of a recent Eulerian method for solving PDEs on general geometries. J. Sci. Comput. **29**, 321–352 (2006)
40. Grindrod, P., Lewis, M.A., Murray, J.D.: A geometrical approach to wave-type solutions of excitable reaction-diffusion systems. Proc. Roy. Soc. A **433**(1887), 151–164 (1991). DOI 10.1098/rspa.1991.0040

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

42. Kansa, E.J.: Multiquadrics – a scattered data approximation scheme with applications to computational fluid-dynamics – ii: Solutions to parabolic, hyperbolic and elliptic partial differential equations. Comput. Math. Appl. **19**, 147–161 (1990)

43. Keener, J., Sneyd, J.: Mathematical physiology. Springer, New York (1998)

44. Kondo, S., Miura, T.: Reaction-diffusion model as a framework for understanding biological pattern formation. Science **329**(5999), 1616–1620 (2010). DOI 10.1126/science.1179047

45. Landsberg, C., Voigt, A.: A multigrid finite element method for reaction-diffusion systems on surfaces. Comput. Visual. Sci. **13**, 177–185 (2010)

46. Larsson, E., Lehto, E., Heryudono, A., Fornberg, B.: Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions. Report No. 2012-020, Dept. of Information Technology, Uppsala Univ., Uppsala, Sweden, 2012

47. Liu, D., Xu, G., Zhang, Q.: A discrete scheme of Laplace-Beltrami operator and its convergence over quadrilateral meshes. Comput. Math. Appl. **55**, 1081–1093 (2008)

48. Liu, Y., Liu, W.K.: Rheology of red blood cell aggregation by computer simulation. J. Comp. Phys. **220**, 139–154 (2006)

49. MacDonald, C.B., Ruuth, S.J.: Level set equations on surfaces via the closest point method. J. Sci. Comput. **35**, 219–240 (2008)

50. MacDonald, C.B., Ruuth, S.J.: The implicit closest point method for the numerical solution of partial differential equations on surfaces. SIAM J. Sci. Comput. **31**, 4330–4350 (2009)

51. Maini, P.K., Othmer, H.G. (eds.): Mathematical models for biological pattern formation. Springer, New York (2000)

52. Manz, N., Davydov, V.A., Müller, S.C., Bär, M.: Dependence of the spiral rotation frequency on the surface curvature of reaction-diffusion systems. Physics Letters A **316**(5), 311 – 316 (2003). DOI 10.1016/S0375-9601(03)01148-4. URL http://www.sciencedirect.com/science/article/pii/S0375960103011484

53. Murray, J.: Mathematical biology II: Spatial Models and Biomedical Applications. Springer, New York (2003)

54. Narcowich, F.J., Ward, J.D., Wendland, H.: Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting. Math. Comp. **74**(250), 743–763 (electronic) (2005)

55. Narcowich, F.J., Ward, J.D., Wright, G.B.: Divergence-free RBFs on surfaces. J. Fourier Anal. Appl. **13**(6), 643–663 (2007)

56. Palais, R., Palais, B., Karcher, H.: Point clouds: distributing points unifomly on a surface. Technical report (2011)

57. Piret, C.: The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces. J. Comput. Phys. **231**(20), 4662–4675 (2012)

58. Ruuth, S.J., Merriman, B.: A simple embedding method for solving partial differential equations on surfaces. J. Comput. Phys. **227**, 1943–1961 (2008)

59. Sbalzarini, I.F., Hayer, A., Helenius, A., Koumoutsakos, P.: Simulations of (an)isotropic diffusion on curved biological surfaces. Biophys. J. **90**, 878–885 (2006)

60. Schwartz, P., Adalsteinsson, D., Colella, P., Arkin, A.P., Onsum, M.: Numerical computation of diffusion on a surface. Proc. Nat. Acad. Sci. **102**(32), 11,151–11,156 (2005)

61. Shankar, V., Wright, G.B., Fogelson, A.L., Kirby, R.M.: A study of different modeling choices for simulating platelets within the immersed boundary method. Submitted (2011)

62. Shelley, M.J., Baker, G.R.: Order-preserving approximations to successive derivatives of periodic functions by iterated splines. SIAM J. Numer. Anal. **25**(6), 1442–1452 (1988). DOI 10.1137/0725084. URL http://dx.doi.org/10.1137/0725084

63. Stam, J.: Flows on surfaces of arbitrary topology. In: SIGGRAPH '03: ACM SIGGRAPH 2003 Papers, pp. 724–731. ACM, New York, NY, USA (2003). DOI http://doi.acm.org/10.1145/1201775.882338

64. Turing, A.: The chemical basis of morphogenesis. Philos. Trans. R. Soc. B **237**, 37–52 (1952)

65. Turk, G.: Generating textures on arbitrary surfaces using reaction-diffusion. In: SIG-GRAPH '91: Proceedings of the 18st Annual Conference on Computer Graphics and Interactive Techniques, pp. 289–298. ACM, New York, NY, USA (1991)

66. Tyson, J.J., Keener, J.P.: Singular perturbation theory of traveling waves in excitable media (a review). Physica D **32**(3), 327–361 (1988). DOI 10.1016/0167-2789(88)90062-0. URL http://www.sciencedirect.com/science/article/pii/0167278988900620

67. Varea, C., Aragon, J., Barrio, R.: Turing patterns on a sphere. Phys. Rev. E **60**, 4588–4592 (1999)
68. Wendland, H.: Scattered data approximation, *Cambridge Monographs on Applied and Computational Mathematics*, vol. 17. Cambridge University Press, Cambridge (2005)
69. Womersley, R.S., Sloan, I.H.: Interpolation and cubature on the sphere. Website. `http://web.maths.unsw.edu.au/~rsw/Sphere/`
70. Wright, G.B., Flyer, N., Yuen, D.: A hybrid radial basis function - pseudospectral method for thermal convection in a 3D spherical shell. Geochem. Geophys. Geosyst. **11**, Q07,003 (2010)
71. Xu, G.: Discrete Laplace-Beltrami operators and their convergence. Computer Aided Geometric Design **21**, 767–784 (2004)
72. Yagisita, H., Mimura, M., Yamada, M.: Spiral wave behaviors in an excitable reaction-diffusion system on a sphere. Physica D **124**(1–3), 126–136 (1998). DOI 10.1016/S0167-2789(98)00182-1. URL `http://www.sciencedirect.com/science/article/pii/S0167278998001821`