Journal of Computational Physics 479 (2023) 112001

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

The D-RBF-PU method for solving surface PDEs

Reyhaneh Mir[a], Davoud Mirzaei[a,b,*]

[a] Department of Applied Mathematics and Computer Science, Faculty of Mathematics and Statistics, University of Isfahan, 81746-73441
Isfahan, Iran
[b] Division of Scientific Computing, Department of Information Technology, Uppsala University, P O Box 337, S-751 05 Uppsala, Sweden

ARTICLE INFO

Article history:
Received 7 April 2022
Received in revised form 13 December 2022
Accepted 3 February 2023
Available online 9 February 2023

Keywords:
Radial basis function (RBF)
Partial differential equations (PDEs)
Surface PDEs
Partition of unity (PU)
D-RBF-PU method

ABSTRACT

In this paper a new direct RBF partition of unity (D-RBF-PU) method is developed for numerical solution of partial differential equations defined on smooth orientable surfaces which are discretized with sets of scattered nodes and with approximations to normal vectors at each of the nodes. The accuracy, stability and efficiency of the new method are studied through some theoretical and experimental results. This method is a localized RBF based technique, results in a perfectly sparse final linear system, uses only scattered nodes on the surface rather than a connected mesh, and is applicable for a large class of PDEs on manifolds. Applications to some biological and chemical reaction-diffusion models are also given. Results show that the new method outperforms other comparable techniques for surface PDEs.

1. Introduction

Radial basis function (RBF) approximation is a powerful tool for reconstructing an unknown function from its given scattered data values [7,56]. This approximation has become increasingly popular for solving partial differential equations (PDEs) and other applications due to its easy implementation, high-order convergence rates and its ability to handle scattered node layouts on arbitrary domains [16,18]. In this paper we focus on solving surface PDEs using a new technique based on this interesting approach.

Solving PDEs on surfaces have received growing interest due to some applications in meteorology, geometry, biology, image processing, etc. The finite difference (FD) method is a standard technique for solving PDEs in Euclidean domains where the one dimensional FD formulas can be combined to create FD formulas in higher dimensions for partial derivatives. The first limitation of this FD approach for surface PDEs is that the extension of univariate formulas for surface derivatives is not straightforward. The generalized multivariate FD formulas can be instead organized on stencils with scattered nodes, but this approach also raises the question of how the stencil weights should be computed for example in cases in which a polynomial interpolation on a surface is singular. For the special case of spheres the Euclidean polynomials can be replaced by *spherical harmonics* and a theoretical solvability is guaranteed. However, if the space of spherical harmonics is restricted to spherical caps (for local stencils) then the linear system for computing the stencil weights becomes highly ill-conditioned. On the other hand, an alternative to spherical harmonics for other manifolds is not available, directly. These drawbacks can be bypassed if RBF interpolation is used instead to generate the stencil weights in the FD formulas (*e.g.* [17,18,28,36,46,49,50]). This approach is called RBF-FD and results in a sparse and well-conditioned final system contrary to the global RBF

[*] Corresponding author.
E-mail addresses: d.mirzaei@sci.ui.ac.ir, davoud.mirzaei@it.uu.se (D. Mirzaei).

https://doi.org/10.1016/j.jcp.2023.112001

approximation which produces full and ill-conditioned matrices. We note that RBF-FD has been frequently used for solving PDE problems in Euclidean spaces as well; see for example [18] and references therein.

Another approach for localization and evolving RBF into a fast technique applicable on large-scale problems is the RBF *partition of unity* (RBF-PU) method. The first combination of a PU method with RBF interpolation goes back to [55] and the analysis is given in [56]. See also [16]. A fast algorithm based on PU for two-dimensional interpolation of large scattered data sets was presented in [9]. The RBF-PU collocation method for solving transport equations on the unit sphere was developed in [1]. A least squares RBF-PU method as an alternative to the standard collocation method was studied in [26]. RBF-PU has also been shown to be successful for a number of other PDE problems (*e.g.* [10,12,13,40,47,48]). For applications to rational interpolation see [11,15] and for applications to parametric structural and shape optimization see [23,24].

In [30] the new D-RBF-PU method was presented to simplify the implementation and increase the efficiency of the standard RBF-PU method. The direct approximation of PDE operators avoids differentiation against PU weight functions and allows the use of some useful discontinuous weights functions. This method is also related to the RBF-FD method but is much faster for setting up the stiffness and mass matrices. The efficiency of this approach for surface PDEs is even more notable because the computation and implementation of surface derivatives of PU weight functions is a challenging problem. In this paper, we develop the D-RBF-PU method for solving PDEs defined on smooth embedded submanifolds of Euclidean spaces. As a basis function, we use the restriction of Euclidean positive definite kernels on surfaces [20]. On the PU part we apply both smooth and discontinuous weights for blending local RBF approximants and forming a global solution for the PDE problem.

The remainder of this paper is organized as follows. In sections 2 and 3 the continuous differential operators on embedded submanifolds in $\mathbb{R}^d$ and their discrete versions via a specific RBF approximation are reviewed, respectively. In section 4 the well-known RBF-FD method is briefly discussed, and in section 5 the idea of PU approximation on surfaces and the standard approach for solving PDEs are presented. In section 6, the D-RBF-PU method is developed for numerical solution of surface PDEs. Then its advantages over other techniques, its connection to RBF-FD and its error analysis are given. In section 7, some numerical experiments, comparisons with RBF-FD and applications to some reaction-diffusion models are presented.

2. Continuous differential operators on surfaces

Throughout this paper, by terms *surface* or *manifold* we mean a smooth embedded submanifold of codimension 1 in $\mathbb{R}^d$ with no boundary. The specific case $d = 3$ is considered here but straightforward extensions are possible for manifolds of higher codimension, or manifolds of codimension 1 embedded in higher or lower dimensional spaces. We denote a generic manifold by Γ.

The standard way for defining the differential operators on manifolds is through the use of intrinsic coordinates. We instead follow [21,28,46] to formulate these operators entirely in Cartesian (or extrinsic) coordinates to avoid singularities associated with intrinsic coordinate systems. In this methodology the surface gradient of a differentiable function u can be simply formulated as follows. We apply the Euclidean gradient ∇ to u at a point $\boldsymbol{x} = (x, y, z) \in \Gamma$ and then project the resulting vector into the tangent space of Γ at $\boldsymbol{x}$. More precisely

$$\nabla_\Gamma u = \nabla u - \boldsymbol{n}(\boldsymbol{n}\cdot\nabla u) = \nabla u - \boldsymbol{n}\boldsymbol{n}^T(\nabla u) = (I - \boldsymbol{n}\boldsymbol{n}^T)\nabla u$$

where $\boldsymbol{n} = \boldsymbol{n}(\boldsymbol{x})$ is the normal vector on Γ at the point $\boldsymbol{x}$. This shows that the surface gradient operator can be written entirely in Cartesian coordinates as

$$\nabla_\Gamma := \Pi\nabla$$

where $\Pi = I - \boldsymbol{n}\boldsymbol{n}^T$ is the projection operator. In an extensive form for $d = 3$ the surface gradient is written as

$$\nabla_\Gamma = \begin{bmatrix} (1 - n_x n_x)\dfrac{\partial}{\partial x} - n_x n_y \dfrac{\partial}{\partial y} - n_x n_z \dfrac{\partial}{\partial z} \\ -n_x n_y \dfrac{\partial}{\partial x} + (1 - n_y n_y)\dfrac{\partial}{\partial y} - n_y n_z \dfrac{\partial}{\partial z} \\ -n_x n_z \dfrac{\partial}{\partial x} - n_y n_z \dfrac{\partial}{\partial y} + (1 - n_z n_z)\dfrac{\partial}{\partial z} \end{bmatrix} =: \begin{bmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{bmatrix}, \tag{2.1}$$

where $(n^x, n^y, n^z) =: \boldsymbol{n}$. The surface divergence of a smooth vector field $\boldsymbol{f} = (f^x, f^y, f^z) : \Gamma \to \mathbb{R}^3$ at a point $\boldsymbol{x} \in \Gamma$ is expressed as

$$\nabla_\Gamma \cdot \boldsymbol{f} = \mathcal{G}^x f^x + \mathcal{G}^y f^y + \mathcal{G}^z f^z.$$

The Laplace-Beltrami operator Δ_Γ at $\boldsymbol{x} \in \Gamma$ is then defined by

$$\Delta_\Gamma := \nabla_\Gamma \cdot \nabla_\Gamma = (\Pi\nabla)\cdot(\Pi\nabla) = \mathcal{G}^x\mathcal{G}^x + \mathcal{G}^y\mathcal{G}^y + \mathcal{G}^z\mathcal{G}^z. \tag{2.2}$$

Note that all expressions above are in Cartesian coordinates.

3. Discrete differential operators on surfaces

In this section, we present a RBF based method for discretizing the surface gradient and the Laplace-Beltrami operators. First, the RBF interpolation in Euclidean spaces is briefly reviewed and then approximation via basis functions constructed by restricting the RBF on a manifold is discussed. Finally two approaches for approximating the Laplace-Beltrami operator are given. The main part of the section follows the methodology given in [21].

3.1. RBF interpolation

We start with RBF interpolation which is essential for the method we aim to develop in next sections. Assume that $\Omega \subset \mathbb{R}^d$ and $\Phi : \Omega \times \Omega \to \mathbb{R}$ is a kernel with property $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \phi(\|\boldsymbol{x} - \boldsymbol{y}\|)$ for a continuous and even univariate function $\phi : \mathbb{R} \to \mathbb{R}$. Here $\|\cdot\|$ stands for Euclidean norm in $\mathbb{R}^d$. Such kernels are called *radial* functions. We assume that ϕ is positive definite in the following sense.

Definition 3.1. A continuous and even function ϕ is positive definite on $\mathbb{R}^d$ if

$$\sum_{k=1}^{N}\sum_{j=1}^{N} c_k c_j \phi(\|\boldsymbol{x}_k - \boldsymbol{x}_j\|) > 0$$

for any $N \geqslant 1$, any N pairwise distinct points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ and any nonzero vector $\boldsymbol{c} = (c_1, c_2, \ldots, c_N)^T \in \mathbb{R}^N$. If the quadratic form is positive not for all vectors $\boldsymbol{c} \in \mathbb{R}^N$ but for those satisfy

$$\sum_{j=1}^{N} c_j p(\boldsymbol{x}_j) = 0, \quad \forall p \in \mathbb{P}_{m-1}(\mathbb{R}^d)$$

where $\mathbb{P}_{m-1}(\mathbb{R}^d)$ is the polynomial space of degree $m-1$ (order m), then ϕ is called conditionally positive definite of order m. A positive definite function is conditionally positive definite of order $m = 0$ and any order $m > 0$. A radial and (conditionally) positive definite function ϕ is called a radial basis function (RBF).

For a given set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \Omega$ and a continuous function $u : \Omega \to \mathbb{R}$ sampled at points in X we consider the RBF interpolant of u on X of the form

$$I_{\phi,X} u(\boldsymbol{x}) = I_\phi u(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|) + c_{N+1}$$

for a positive definite or first order conditionally positive definite function ϕ. The coefficients c_j are obtained by imposing the interpolation conditions $I_\phi u(\boldsymbol{x}_k) = u(\boldsymbol{x}_k)$, $1 \leqslant k \leqslant N$ and the side condition $c_1 + \cdots + c_N = 0$, or equivalently by solving the linear system

$$\begin{bmatrix} A_X & \boldsymbol{e} \\ \boldsymbol{e}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ c_{N+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \\ 0 \end{bmatrix}$$

where $A_X = (\phi(\|\boldsymbol{x}_k - \boldsymbol{x}_j\|))$, $1 \leqslant k, j \leqslant N$, $\boldsymbol{e} = (1, 1, \ldots, 1)^T$, $\boldsymbol{u} = (u(\boldsymbol{x}_1), \ldots, u(\boldsymbol{x}_N))^T$ and $\boldsymbol{c} = (c_1, \ldots, c_N)^T$. If ϕ is either positive definite or conditionally positive definite of order 1 then the above system is uniquely solvable [56]. The inverse matrix can be obtained as

$$\begin{bmatrix} A_X & \boldsymbol{e} \\ \boldsymbol{e}^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} S_X & \tilde{\boldsymbol{e}} \\ \tilde{\boldsymbol{e}}^T & a \end{bmatrix}$$

where

$$S_X = Z(Z^T A_X Z)^{-1} Z^T \tag{3.1}$$

where $Z \in \mathbb{R}^{N \times (N-1)}$ is any matrix whose columns form a basis for $\ker(\boldsymbol{e}^T)$, $\tilde{\boldsymbol{e}} = \frac{1}{N}(I - S_X A_X)\boldsymbol{e}$ and $a = -\frac{1}{N}\boldsymbol{e}^T A_X \tilde{\boldsymbol{e}}$. For more details about the properties and the inverse of a saddle point matrix of the above form (and even in a more general form) see [15]. We note that $Z^T A_X Z$ is necessarily positive definite on $\mathbb{R}^{N-1}$. Thus, the interpolation coefficients are obtained as

$$\boldsymbol{c} = S_X \boldsymbol{u}, \quad c_{N+1} = \tilde{\boldsymbol{e}}^T \boldsymbol{u}.$$

The interpolant $I_\phi u$ can be rewritten in the Lagrange (nodal) form

$$I_\phi u(\boldsymbol{x}) = \sum_{j=1}^{N} \psi_j(\boldsymbol{x}) u(\boldsymbol{x}_j) \tag{3.2}$$

where the Lagrange functions vector $(\psi_1(\boldsymbol{x}), \ldots, \psi_N(\boldsymbol{x}))^T =: \boldsymbol{\psi}(\boldsymbol{x})$ satisfies

$$\begin{bmatrix} A_X & \boldsymbol{e} \\ \boldsymbol{e}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}(\boldsymbol{x}) \\ \gamma(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x}) \\ 1 \end{bmatrix} \tag{3.3}$$

for $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi(\|\boldsymbol{x}-\boldsymbol{x}_1\|), \ldots, \phi(\|\boldsymbol{x}-\boldsymbol{x}_N\|))^T$. By using the inverse of the saddle point matrix we can write

$$\boldsymbol{\psi}(\boldsymbol{x}) = S_X \boldsymbol{\phi}(\boldsymbol{x}) + \tilde{\boldsymbol{e}}, \quad \gamma(\boldsymbol{x}) = \tilde{\boldsymbol{e}}^T \boldsymbol{\phi}(\boldsymbol{x}) + a.$$

However, the value $\gamma(\boldsymbol{x})$ is unimportant in nodal representation (3.2).

Here, we are interested in RBF interpolations on manifolds but we still use the standard Euclidean distance $\|\cdot\|$ in $\mathbb{R}^d$ instead of geodesic distances intrinsic to surfaces. In fact, we use the restriction of kernel Φ on submanifold Γ. Since Γ is a subset of $\mathbb{R}^d$, the restriction $\Phi|_{\Gamma\times\Gamma}$ inherits the positive definiteness from the original kernel Φ which makes it suited for interpolation problems on Γ.

3.2. Discretizations of surface gradient and Laplacian

Let the RBF ϕ have at least one continuous derivative, $\boldsymbol{x} = (x, y, z) \in \Gamma$, $\boldsymbol{x}_j = (x_j, y_j, z_j) \in \Gamma$, and $r_j(\boldsymbol{x}) = \|\boldsymbol{x}-\boldsymbol{x}_j\|$. Referring to (2.1), the x-component of the surface gradient of translates of ϕ by using the chain rule can be written as

$$\begin{aligned} \mathcal{G}^x \phi(r_j(\boldsymbol{x})) &= (1 - n_x n_x)\frac{\partial}{\partial x}\phi(r_j(\boldsymbol{x})) - n_x n_y \frac{\partial}{\partial y}\phi(r_j(\boldsymbol{x})) - n_x n_z \frac{\partial}{\partial z}\phi(r_j(\boldsymbol{x})) \\ &= \big((1-n_x n_x)(x-x_j) - n_x n_y (y-y_j) - n_x n_z (z-z_j)\big)\frac{\phi'(r_j(\boldsymbol{x}))}{r_j(\boldsymbol{x})}, \end{aligned}$$

where $'$ denotes differentiation with respect to r. Note that the singularity in the above expression will analytically cancel, because ϕ is assumed to be an even function with at least one continuous derivative. Other components of the surface gradient of ϕ can be obtained similarly by

$$\begin{aligned} \mathcal{G}^y \phi(r_j(\boldsymbol{x})) &= \big(-n_x n_y (x-x_j) + (1-n_y n_y)(y-y_j) - n_y n_z (z-z_j)\big)\frac{\phi'(r_j(\boldsymbol{x}))}{r_j(\boldsymbol{x})}, \\ \mathcal{G}^z \phi(r_j(\boldsymbol{x})) &= \big(-n_x n_z (x-x_j) - n_y n_z (y-y_j) + (1-n_z n_z)(z-z_j)\big)\frac{\phi'(r_j(\boldsymbol{x}))}{r_j(\boldsymbol{x})}. \end{aligned}$$

Note that, in the above expressions the normal vector components $n_x = n_x(\boldsymbol{x})$, $n_y = n_y(\boldsymbol{x})$ and $n_z = n_z(\boldsymbol{x})$ depend on variable $\boldsymbol{x}$ although this dependence is ignored in the notation for brevity. The surface gradient of a function $u : \Gamma \to \mathbb{R}$ is then approximated by the surface gradient of its RBF interpolation on Γ, i.e.

$$\begin{aligned} \nabla_\Gamma u(\boldsymbol{x}) \approx \nabla_\Gamma I_\phi u(\boldsymbol{x}) &= \sum_{j=1}^{N} \nabla_\Gamma \psi_j(\boldsymbol{x}) u(\boldsymbol{x}_j) = \nabla_\Gamma \boldsymbol{\psi}^T(\boldsymbol{x})\boldsymbol{u} \\ &= \nabla_\Gamma \boldsymbol{\phi}^T(\boldsymbol{x}) S_X \boldsymbol{u} + \boldsymbol{0} = \begin{bmatrix} \mathcal{G}^x \boldsymbol{\phi}^T(\boldsymbol{x}) \\ \mathcal{G}^y \boldsymbol{\phi}^T(\boldsymbol{x}) \\ \mathcal{G}^z \boldsymbol{\phi}^T(\boldsymbol{x}) \end{bmatrix} S_X \boldsymbol{u} \end{aligned}$$

where the symmetric matrix S_X is defined in (3.1). If we assume $B_X^x = (\mathcal{G}^x \boldsymbol{\phi}^T(\boldsymbol{x}_k))$, $B_X^y = (\mathcal{G}^y \boldsymbol{\phi}^T(\boldsymbol{x}_k))$, $B_X^z = (\mathcal{G}^z \boldsymbol{\phi}^T(\boldsymbol{x}_k))$ all for $1 \leqslant k \leqslant N$ then the approximate surface gradient at points in X is obtained as

$$(\nabla_\Gamma u)|_X \approx (\nabla_\Gamma I_\phi u)|_X = \begin{bmatrix} B_X^x S_X \boldsymbol{u} \\ B_X^y S_X \boldsymbol{u} \\ B_X^z S_X \boldsymbol{u} \end{bmatrix} =: \begin{bmatrix} G_X^x \boldsymbol{u} \\ G_X^y \boldsymbol{u} \\ G_X^z \boldsymbol{u} \end{bmatrix}$$

where $G_X^x = B_X^x S_X$, $G_X^y = B_X^y S_X$ and $G_X^z = B_X^z S_X$ are differential matrices associated to continuous operators $\mathcal{G}^x$, $\mathcal{G}^y$ and $\mathcal{G}^z$, respectively.

To discretize the Laplace-Beltrami operator the standard approach uses

$$\Delta_\Gamma u \approx \Delta_\Gamma I_\phi u = \nabla_\Gamma \cdot \nabla_\Gamma I_\phi u$$

where in the form of (2.2), the gradient operators $\mathcal{G}^x$, $\mathcal{G}^y$ and $\mathcal{G}^z$ need to be applied twice requiring the first partial derivatives of normal vector $\boldsymbol{n} = \boldsymbol{n}(x)$. This makes the approximation rather complicated especially when explicit representation of normal vector is not available or when the manifold is defined by point clouds rather than an explicit representation. On the other hand, second derivatives of basis functions are required. To bypass these difficulties, a *repeated interpolation and differentiation* approach is proposed in [19] as

$$\Delta_\Gamma u \approx \nabla_\Gamma \cdot I_\phi(\nabla_\Gamma I_\phi u).$$

If we imitate the continuous form (2.2) by replacing the differentiation matrices G_X^x, G_X^y and G_X^z by the operators $\mathcal{G}^x$, $\mathcal{G}^y$ and $\mathcal{G}^z$, respectively, then the approximation

$$(\Delta_\Gamma u)\big|_X \approx \big(G_X^x G_X^x + G_X^y G_X^y + G_X^z G_X^z\big)\boldsymbol{u} =: L_X \boldsymbol{u} \tag{3.4}$$

is obtained for the Laplace-Beltrami of u at nodes in X. The N-by-N differentiation matrix L_X approximates the operator Δ_Γ on the set X.

4. RBF-FD method on surfaces

The RBF-FD method arises naturally as a generalization of standard FD scheme where the one dimensional formulas are combined to create FD formulas in higher dimensions for partial derivatives. In a generalized FD method the stencil weights are generated by imposing exactness on (multivariate) polynomials. Due to drawbacks outlined in the introduction section, the RBF-generated formulas generate stencil weights by assuming exactness on the space of shifts of the RBF to scattered points in the stencil.

For a linear surface operator L, the value $(Lu)(\boldsymbol{x}_0)$ can be recovered from values of u at a stencil $\widetilde{X}_0 \subset X$ of points nearing $\boldsymbol{x}_0$. Indeed, we obtain the weights $\xi_{0,j}$ such that

$$Lu(\boldsymbol{x}_0) \approx \sum_{j\in\widetilde{J}_0} \xi_{0,j} u(\boldsymbol{x}_j), \tag{4.1}$$

where $\widetilde{J}_0$ is the set of indices of points in $\widetilde{X}_0$. The weight vector $\boldsymbol{\xi}_0$ will be computed by assuming exactness of the formula for basis functions $u(\boldsymbol{x}) = \phi(\|\boldsymbol{x} - \boldsymbol{x}_j\|)$, $j \in \widetilde{J}_0$. Usually, exactness for constant function $u(\boldsymbol{x}) \equiv 1$ is also imposed. This leads to system

$$\begin{bmatrix} A_{\widetilde{X}_0} & \boldsymbol{e} \\ \boldsymbol{e}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}_0 \\ \gamma_0 \end{bmatrix} = \begin{bmatrix} L\boldsymbol{\phi}_0 \\ 1 \end{bmatrix}$$

where $L\boldsymbol{\phi}_0 = (L\phi(\|\boldsymbol{x}_0 - \boldsymbol{x}_j\|))_{j\in\widetilde{J}_0}$. Comparing with (3.3) we observe that $\boldsymbol{\xi}_0 = L\boldsymbol{\psi}(x_0)$ where $\boldsymbol{\psi}$ is the vector of Lagrange basis functions on set $\widetilde{X}$. If ϕ is conditionally positive definite of order 0 or 1 then the above system is guaranteed to be solvable. For solving a linear PDE problem

$$Lu = f, \text{ on } \Gamma \tag{4.2}$$

by RBF-FD, we consider two sets $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ (trial set) and $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M\}$ (test set) of scattered points on Γ. Then we loop over test points $\boldsymbol{y}_k$ and give the role of $\boldsymbol{x}_0$ to all $\boldsymbol{y}_k$ successively. The stencil weights $\boldsymbol{\xi}_k$ on points $\widetilde{X}_k$ in neighborhood of $\boldsymbol{y}_k$, when interspersed with zeros corresponding to points outside the stencils, form the rows of a differentiation matrix A_L of size $M \times N$ such that

$$\boldsymbol{f} = (Lu)|_Y \approx A_L \boldsymbol{u}$$

where $\boldsymbol{u} = (u(\boldsymbol{x}_1), \ldots, u(\boldsymbol{x}_N))^T$ and $\boldsymbol{f} = (f(\boldsymbol{y}_1), \ldots, f(\boldsymbol{y}_M))^T$. The approximate solution $\widetilde{\boldsymbol{u}}$ of

$$A_L \widetilde{\boldsymbol{u}} = \boldsymbol{f}$$

then gives the approximate nodal values at the trial set X. For postprocessing calculations, the RBF-FD formula can be used again to find the approximate values of u and its derivatives at any evaluation point $\boldsymbol{x} \in \Gamma$.

To tackle the problem of solvability we may use a large enough set Y to have a full rank overdetermined matrix A_L and look for a unique least squares solution $\widehat{\boldsymbol{u}}$ instead. See [42–44] for more details. However, in practice the degeneracy happens in rare situations and usually the case $Y = X$ gives an invertible matrix A_L. For some applications of RBF-FD to surface PDEs see [28,46].

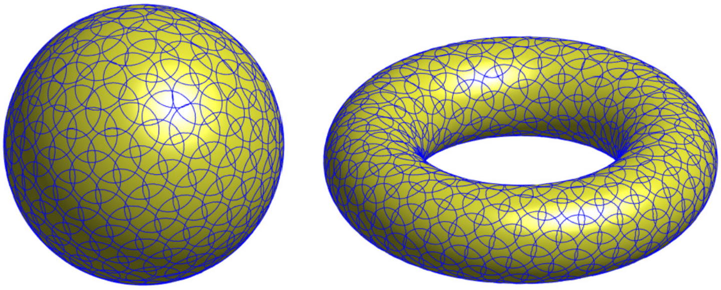

Fig. 1. Overlapping circular coverings on the unit sphere and a torus.

5. Partition of unity approximation

Let $\{\Gamma_\ell\}_{\ell=1}^{N_c}$ be an open and bounded covering of Γ that means all patches Γ_ℓ are open and bounded and $\Gamma = \bigcup_{\ell=1}^{N_c} \Gamma_\ell$. This ensures that every point $\boldsymbol{x} \in \Gamma$ is necessarily covered by at least one patch Γ_ℓ. See Fig. 1 for an example of covering on the unit sphere and on a torus. A family of nonnegative functions $\{w_\ell\}_{\ell=1}^{N_c}$ is called a *partition of unity* (PU) with respect to the covering $\{\Gamma_\ell\}$ if

(1) $\mathrm{supp}(w_\ell) \subseteq \Gamma_\ell,\ 1 \leqslant \ell \leqslant N_c$,

(2) $\displaystyle\sum_{\ell=1}^{N_c} w_\ell(\boldsymbol{x}) = 1,\ \forall \boldsymbol{x} \in \Gamma$,

where $\mathrm{supp}(w) = \overline{\{\boldsymbol{x} \in \mathbb{M} : w(\boldsymbol{x}) \neq 0\}}$. The title 'partition of unity' is induced from the second property.

We start with an overlapping covering $\{\Gamma_\ell\}_{\ell=1}^{N_c}$ of Γ. If we assume V_ℓ is an approximation space on Γ_ℓ and $s_\ell \in V_\ell$ is a local approximant of a function u then

$$s = \sum_{\ell=1}^{N_c} w_\ell s_\ell \tag{5.1}$$

is a global interpolation of u on Γ which is formed by joining the local approximants s_ℓ via PU weights w_ℓ. A PU approximation provides a collection of local approximants (small problems) and blends them via a set of PU weights to form a global approximation. This idea is numerically efficient because of avoiding a large (and possibly unstable) system in favor of several small (and possibly stable) local systems.

The smoothness of the global solution depends on the smoothness of both local approximants and PU weights. A possible choice for w_ℓ are Shepard's weights

$$w_\ell(\boldsymbol{x}) = \frac{\Psi_\ell(\boldsymbol{x})}{\sum_{j=1}^{N_c} \Psi_j(\boldsymbol{x})}, \quad 1 \leqslant \ell \leqslant N_c, \tag{5.2}$$

where Ψ_ℓ are nonnegative, nonvanishing and compactly supported functions on Γ_ℓ.

Assume that the linear PDE problem (4.2) is posed on a manifold Γ. A standard PU method for discretizing this PDE finds the global PU approximation s of u and then approximates Lu by Ls:

$$Lu \approx Ls = \sum_{\ell=1}^{N_c} L(w_\ell s_\ell). \tag{5.3}$$

In this scenario, the surface operator L should act on products $w_\ell s_\ell$ which may result in a complicated computation as it requires a kind of Leibniz's rule for surface derivatives of weight functions. The rational structure of Shepard's weights makes it more challenging especially when higher derivatives are demanded. This may be the reason why the PU approach has been rarely employed for surface PDEs. We refer the reader to [1] as an available implementation on the sphere. In the present paper we suggest an alternative approach which avoids the differentiation against weight functions and allows the use of some discontinuous weights to develop faster algorithms.

6. D-RBF-PU method

Assume that $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ is a set of well distributed scattered points on surface Γ and $\{\Gamma_\ell\}$ is a covering for Γ. Besides, let $X_\ell = X \cap \Gamma_\ell$. When the local approximants s_ℓ are local RBF interpolations on X_ℓ the resulting PU method is called RBF-PU method. This method is an attractive alternative to global RBFs as it performs better in terms of accuracy per computational cost because its resulting differentiation matrix is sparse. On the other side, RBF-PU inherits the advantages of global RBF methods in terms of handling irregular geometries and scattered node layouts. However, RBF-PU when applying on surface PDEs suffer from some disadvantages which were outlined at the end of previous section. To overcome this we follow the D-RBF-PU approach introduced in [30] for solving problems on bounded Euclidean domains and extend it for surface PDEs in this paper. In D-RBF-PU the value Lu is directly approximated by the PU method without any detour via local approximants s_ℓ of u, say

$$Lu \approx \sum_{\ell=1}^{N_c} w_\ell s_\ell^L =: s^L \tag{6.1}$$

where s_ℓ^L are local approximants of Lu on patches Γ_ℓ. The resulting global approximation is denoted by s^L which is clearly different from Ls of the standard approach (5.3). The first notable difference is that in the new method the weight functions are not required to be differentiated. However, these two methods are basically different as they fall into two different discretization categories, namely the *trial space* and the *direct* discretizations highlighted by Schaback in [42,44] in the context of scattered data approximations. See also [31,32] for developments of the second approach in the moving least squares approximation context.

Assume that the set of indices of points from X in patch Γ_ℓ is denoted by J_ℓ, i.e.

$$J_\ell = \{j \in \{1,2,\dots,N\} : \boldsymbol{x}_j \in X_\ell\}.$$

In RBF based PU methods the local approximation spaces on Γ_ℓ are constructed by the shifts of RBFs to points in X_ℓ (possibly) plus a polynomial space. In our case in this paper we assume

$$V_\ell := \operatorname{span}\{\phi(\|\cdot - \boldsymbol{x}_j\|) : j \in J_\ell\} + \operatorname{span}\{1\},$$

and we let $s_\ell \in V_\ell$ be the interpolant function $I_{\phi,X_\ell}u$, i.e.

$$s_\ell(\boldsymbol{x}) = I_{\phi,X_\ell}u(\boldsymbol{x}) = \sum_{j\in J_\ell} \psi_j(\ell;\boldsymbol{x})u(\boldsymbol{x}_j), \quad \boldsymbol{x} \in \Gamma_\ell,$$

where $\psi_j(\ell;\cdot)$ are Lagrange functions associated to patch Γ_ℓ. In D-RBF-PU an obvious choice for local approximants s_ℓ^L in (6.1) is

$$s_\ell^L = Ls_\ell = L(I_{\phi,X_\ell}u) = \sum_{j\in J_\ell} L\psi_j(\ell;\cdot)u(\boldsymbol{x}_j) \tag{6.2}$$

which is used successfully in [30] for different derivative operators L defined on Euclidean domains in $\mathbb{R}^d$. However, as we pointed out in section 3.2 this will cause serious difficulties for handling the derivatives of normal vectors when L is, for example, the Laplace-Beltrami operator on surfaces. An alternative choice for case $L = \Delta_\Gamma$ then is

$$s_\ell^L = \nabla_\Gamma \cdot (I_{\phi,X_\ell}(\nabla_\Gamma(I_{\phi,X_\ell}u))). \tag{6.3}$$

This repeated interpolation and differentiation approach, as described before, avoids differentiation with respect to normal vectors and highly simplifies the computation. By mimicking the relation (3.4) we have

$$(\Delta_\Gamma u)\big|_{X_\ell} \approx (s_\ell^L)|_{X_\ell} = \left(G_{X_\ell}^x G_{X_\ell}^x + G_{X_\ell}^y G_{X_\ell}^y + G_{X_\ell}^z G_{X_\ell}^z\right)\boldsymbol{u}_\ell =: L_{X_\ell}\boldsymbol{u}_\ell,$$

where $\boldsymbol{u}_\ell$ stands for u values at X_ℓ. See [28,46] for the use of this approach in RBF-FD methods on surfaces. This approach gives the approximate Laplacian at all interpolation points in X_ℓ, simultaneously, while the only approximation at the central point is used in RBF-FD methods. In contrast to this, the D-RBF-PU method benefits from the approximation of Δ_Γ at all points in X_ℓ. We note that for the surface gradient ∇_Γ the standard interpolation

$$s_\ell^L = \nabla_\Gamma(I_{\phi,X_\ell}u)$$

will be used. In both cases ($L = \nabla_\Gamma$ and $L = \Delta_\Gamma$) we may expand the local approximant in terms of function values as

$$s_\ell^L = \sum_{j\in J_\ell} \psi_j^L(\ell;\cdot)u(\boldsymbol{x}_j), \tag{6.4}$$

instead of (6.2) to obtain the global PU approximation

$$s^L(\boldsymbol{x}) = \sum_{\ell=1}^{N_c}\sum_{j\in J_\ell} w_\ell(\boldsymbol{x})\psi_j^L(\ell;\boldsymbol{x})u(\boldsymbol{x}_j), \quad \boldsymbol{x} \in X. \tag{6.5}$$

Usually, Shepard's functions (5.2) are used as PU weights. These functions are as smooth as their generating compactly supported functions Ψ_ℓ. For a test point $\boldsymbol{x} \in \Gamma$ let $I(\boldsymbol{x})$ be the set of indices of patches in $\{\Gamma_\ell\}$ which cover the point $\boldsymbol{x}$. A PU approximation at point $\boldsymbol{x}$ is influenced by all trial points in $\cup_{\ell\in I(\boldsymbol{x})} X_\ell$ leading to a denser (but still sparse) differentiation matrix compared with other localized methods such as RBF-FD. Nevertheless, RBF-PU is computationally less expensive than RBF-FD for setting up the differentiation matrix because the number of local linear systems to be solved is highly reduced from N, the number of test points, to N_c, the number of PU patches. In additions, D-RBF-PU allows the use of discontinuous weight functions because in this method the weights are not required to be differentiated. A useful discontinuous weight can be defined by

$$w_\ell(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \Gamma_\ell \text{ is the nearest patch to } \boldsymbol{x}, \\ 0, & \text{otherwise,} \end{cases} \tag{6.6}$$

which gives the total weight 1 to the nearest patch and null weights to the others. To identify the nearest patch we can measure and compare the distance between $\boldsymbol{x}$ and the center of all patches. This weight function results in a final matrix which is as sparse as the RBF-FD matrix because it assigns a single patch per any test point. Thus, the D-RBF-PU method with weight function (6.6) is faster than RBF-FD for setting up the differentiation matrix and it is as fast as RBF-FD for solving the sparse final linear system. We will see in numerical tests that the majority of costs for both methods is subjected to the setting up phase because fast iterative linear algebra solvers are available for solving the final sparse system.

As another remark, D-RBF-PU can be viewed as a RBF-FD method in a PU setting. Comparing (4.1) and (6.4), one can see that for a test point $\boldsymbol{x}_k$ the local approximation $s_\ell^L(\boldsymbol{x}_k)$ is a RBF-FD approximation of $Lu(\boldsymbol{x}_k)$ on the stencil $\widetilde{X}_k = X_\ell$. Since $\boldsymbol{x}_k$ is covered by patches Γ_ℓ for $\ell \in I(\boldsymbol{x}_k)$, all RBF-FD approximants $s_\ell^L(\boldsymbol{x}_k)$, $\ell \in I(\boldsymbol{x}_k)$, are computed and then joined by

$$\sum_{\ell\in I(\boldsymbol{x}_k)} w_\ell(\boldsymbol{x}_k)\, s_\ell^L(\boldsymbol{x}_k)$$

to form the overall approximant $s^L(\boldsymbol{x}_k)$. The important difference is that a local approximation s_ℓ^L is shared with all test points which are covered by patch Γ_ℓ. This is the key property that makes the D-RBF-PU much faster than the standard RBF-FD method. The situation becomes simpler and the algorithm becomes faster when the weight function (6.6) is applied. In this case, each test point $\boldsymbol{x}_k$ is subjected to a single local set $X_\ell = \Gamma_\ell \cap X$ for an index $\ell = \ell_k \in \{1, \ldots, N_c\}$ in which Γ_{ℓ_k} is the closest patch to $\boldsymbol{x}_k$. From (6.5) and (6.6) we have

$$s^L(\boldsymbol{x}_k) = \sum_{\ell=1}^{N_c} \sum_{j\in J_\ell} w_\ell(\boldsymbol{x}_k)\psi_j^L(\ell;\boldsymbol{x}_k)u(\boldsymbol{x}_j) = \sum_{j\in J_{\ell_k}} \underbrace{\psi_j^L(\ell_k;\boldsymbol{x}_k)}_{\xi_{k,j}} u(\boldsymbol{x}_j) \approx Lu(\boldsymbol{x}_k)$$

because $w_\ell(\boldsymbol{x}_k) = 1$ if $\ell = \ell_k$ and 0 otherwise. Again we emphasize that the difference between this approach and the RBF-FD method is that in the new method a set (stencil) X_ℓ is shared with many test points while in the RBF-FD each stencil is associated to a unique test point. This is also connected to the *overlapped* RBF-FD approach presented in [45].

6.1. Some points about error bounds

In our PU setting we assume that the assigned covering $\{\Gamma_\ell\}$ is *regular* which means that there exists a global constant K such that every $\boldsymbol{x} \in \Gamma$ is covered by at most K patches out of $\{\Gamma_\ell\}$, i.e. $|I(\boldsymbol{x})| \leqslant K$ for all $\boldsymbol{x} \in \Gamma$. On the other hand, we assume that for a given discrete set $X \subset \Gamma$ there exists a global constant C such that $\operatorname{diam}(\Gamma_\ell) \leqslant Ch_{X,\Gamma}$ for $1 \leqslant \ell \leqslant N_c$, where $\operatorname{diam}(\Gamma_\ell)$ is the maximum geodesic distance between points in Γ_ℓ. In this case the covering $\{\Gamma_\ell\}$ is called *local* with respect to the discrete set X.

As it is described in [30], the analysis of D-RBF-PU method follows the theory given in [43] for nodal meshless methods. If we are looking for the errors in nodal values $\boldsymbol{u} = (u(\boldsymbol{x}_1), \ldots, u(\boldsymbol{x}_N))^T$, the *consistency* is analyzed by finding an upper bound for $\|A_L\boldsymbol{u} - \boldsymbol{f}\|_p$ where $\|\cdot\|_p$ is the p-norm in $\mathbb{R}^N$. According to the construction, $A_L(k,:)\boldsymbol{u} = s^L(\boldsymbol{x}_k)$ and $f_k = Lu(\boldsymbol{x}_k)$ for $k = 1, \ldots, N$. Here, by $A(k,:)$ we mean the k-th row of A. Thus, for consistency we assume there exist error bounds

$$|s^L(\boldsymbol{x}_k) - Lu(\boldsymbol{x}_k)| \leqslant \varepsilon^L(\boldsymbol{x}_k), \quad k = 1, \ldots, N,$$

to obtain $\|A_L\boldsymbol{u} - \boldsymbol{f}\|_p \leqslant \|\boldsymbol{\varepsilon}^L\|_p$ where $\boldsymbol{\varepsilon}^L = (\varepsilon^L(\boldsymbol{x}_1), \ldots, \varepsilon^L(\boldsymbol{x}_N))^T$. If the final matrix A_L is full rank and admits the global *stability* bound $\|A_L^{-1}\|_p \leqslant C_S$ for a constant C_S independent of spacing distance between points in X, then

$$\|\widetilde{\boldsymbol{u}} - \boldsymbol{u}\|_p \leqslant \|A_L^{-1}\|_p \|A_L\boldsymbol{u} - \boldsymbol{f}\| \leqslant C_S\|\boldsymbol{\varepsilon}^L\|_p.$$

This is a classical error analysis containing the stability and consistency terms on the bound. However, our analysis may leave some parts unproven. The first open problem concerns the invertibility of the square matrix A_L for the D-RBF-PU method. One can bypass this problem by *overtesting* and seeking for a unique least squares solution instead of an exact solution for $A_L\widetilde{\boldsymbol{u}}=\boldsymbol{f}$ (see *e.g.* [44,43]). If the repeated interpolation and differentiation approach is used for approximating $\Delta_\Gamma u$ then the test and trial points should coincide and overtesting is not applicable. One can use the standard approach (6.2) instead but it needs manipulation with derivatives of normal vectors. However, many experiments show that the square matrix A_L is invertible for different distributions of trial (and test) points X. Thus, in almost all practical situations overtesting is not required at all. The existence of the stability constant C_S is the next unproven ingredient not only for the method of this paper but also for all nonsymmetric meshless methods such as RBF-FD and standard RBF-PU methods. In experiments with MATLAB for the case $p=\infty$ we can use the fact that the `condest` command estimates the L_1 condition number which is the L_∞ condition number of the transpose. Thus, we can estimate C_S by

$$\widetilde{C}_S := \frac{\texttt{condest}(A_L^T)}{\|A_L\|_\infty}$$

which is an estimate of the L_∞ norm of A_L^{-1}. This command is very fast for sparse matrices. In case $p=2$ one can compute the smallest singular value of A_L via SVD to measure the 2-norm of the inverse matrix. Experiments show excellent stability bounds for the D-RBF-PU method on various surfaces. See section 7.

The consistency bounds ε_ℓ^L for surface operators ∇_Γ and Δ_Γ can be estimated by modifying the results of [21,33] for local patches $\Gamma_\ell\subset\Gamma$ instead of Γ itself. The theory holds for finitely smooth positive definite RBFs whose Fourier transforms decay algebraically. We assume that $\Phi:\mathbb{R}^d\to\mathbb{R}$ satisfies

$$c_1(1+\|\boldsymbol{\omega}\|^2)^{-\tau}\leqslant\widehat{\Phi}(\boldsymbol{\omega})\leqslant c_2(1+\|\boldsymbol{\omega}\|^2)^{-\tau}, \tag{6.7}$$

for $\tau>d/2$ and constants c_1 and c_2 with $c_1\leqslant c_2$. As examples of RBFs with this property we can mention the compactly supported Wendland's functions $\Phi(\boldsymbol{x})=(1-\|\boldsymbol{x}\|)_+^k p(\|\boldsymbol{x}\|)$ for a proper integer k and a polynomial p, and the Matèrn function $\Phi(\boldsymbol{x})=\|\boldsymbol{x}\|^{\tau-d/2}K_{\tau-d/2}(\|\boldsymbol{x}\|)$ where K_β is the modified Bessel function of second kind of order β. The native space of a positive definite function Φ with property (6.7) on $\mathbb{R}^d$ is the Sobolev space $H^\tau(\mathbb{R}^d)$ while the native space of the restriction of Φ on Γ is the Sobolev space $H^{\tau-1/2}(\Gamma)$. We refer the reader to [41,56] for definitions and more details about native spaces of (conditionally) positive definite functions and to [20] for properties of restricted kernels from $\mathbb{R}^d$ to smooth embedded submanifolds.

If Γ is a smooth compact manifold with no boundary then it was proved in [33] that (see also [20,21])

Theorem 6.1. *Let Γ be a smooth compact 2-dimensional submanifold of $\mathbb{R}^3$ without boundary. Let $X\subset\Gamma$ be a discrete and quasi-uniform set with a sufficiently small fill-distance $h_{X,\Gamma}$. Then*

$$\begin{aligned}
\|u-I_\phi u\|_{L_\infty(\Gamma)} &\leqslant C h_{X,\Gamma}^{\beta-1}\|u\|_{H^\beta(\Gamma)}, \quad 1<\beta\leqslant 2\sigma,\\
\|\nabla_\Gamma u-\nabla_\Gamma I_\phi u\|_{\mathbf{L}_\infty(\Gamma)} &\leqslant C h_{X,\Gamma}^{\beta-2}\|u\|_{H^\beta(\Gamma)}, \quad 2<\beta\leqslant 2\sigma,\\
\|(\Delta_\Gamma u)|_X-L_X\boldsymbol{u}\|_\infty &\leqslant C h_{X,\Gamma}^{\beta-3}\|u\|_{H^\beta(\Gamma)}, \quad 3<\beta\leqslant 2\sigma,
\end{aligned}$$

for all $u\in H^\beta(\Gamma)$ provided that $\Phi=\phi(\|\cdot\|)$ satisfies (6.7) with $\tau>3/2$, $\tau>5/2$ and $\tau>7/2$ for the first, the second and the third bound, respectively, and $\sigma=\tau-1/2$. Here $\mathbf{L}_\infty(\Gamma)=(L_\infty(\Gamma))^3$.

The above error bounds can be simply updated if we replace the compact manifold Γ by open and bounded coverings $\Gamma_\ell\subset\Gamma$ and X by $X_\ell=X\cap\Gamma_\ell$. For a fixed ℓ, since Γ_ℓ is a *paracompact* submanifold of $\mathbb{R}^3$ there exists an atlas $\widetilde{\mathcal{A}}=\{(\widetilde{\Psi}_k,\widetilde{U}_k)\}$, $k=1,\ldots,K$ of finitely slice charts for Γ_ℓ with an associated intrinsic atlas $\mathcal{A}=\{(\Psi_k,U_k)\}$ [27]. The sets $\widetilde{U}_k\subset\mathbb{R}^3$ are open in $\mathbb{R}^3$ and cover Γ_ℓ, and each $\widetilde{\Psi}_k:\widetilde{U}_k\to B(0,1)$ is a one-to-one smooth map. For the intrinsic atlas the sets $U_k\subset\Gamma_\ell$ are open and $\Psi_k:U_k\to B'(0,1)=\{\boldsymbol{x}=(x,y,z)\in B(0,1):z=0\}$ are considered as copies of the open ball in $\mathbb{R}^2$. Now, let $\{\chi_k\}$ be a partition of unity subordinate to $\{U_k\}$. For a function u defined on Γ_ℓ, the projections $\pi_k(u):\mathbb{R}^2\to\mathbb{R}$ are defined by

$$\pi_k(u)(\boldsymbol{x}):=\begin{cases}\chi_k u(\Psi_k^{-1}(\boldsymbol{x})), & \boldsymbol{x}\in B'(0,1),\\ 0, & \text{otherwise.}\end{cases}$$

Sobolev spaces $H^\sigma(\Gamma_\ell)$ are defined by the norms

$$\|u\|_{H^\sigma(\Gamma_\ell)}:=\left(\sum_{k=1}^K\|\pi_k(u)\|^2_{H^\sigma(\mathbb{R}^2)}\right)^{1/2},$$

where K is the number of charts in the atlas. It is obvious that the norm depends on the special choice of atlas $\widetilde{\mathcal{A}}$ and partition of unity $\{\chi_k\}$. But, the same space arises by different collections of these objects and the norms are equivalent.

Using this construction, the theory of [21,33] can be adapted to conclude a similar theorem with Γ and X being replaced by Γ_ℓ and X_ℓ, respectively. But since the periodicity is lost on local surfaces $\Gamma_\ell\subset\Gamma$, the duality trick of [33] can not be applied to double the orders for smoother functions in H^β for values of β up to 2σ. The error bounds read as

$$\begin{aligned}\|u - I_\phi u\|_{L_\infty(\Gamma_\ell)} &\leqslant C h_{X_\ell,\Gamma_\ell}^{\beta-1}\|u\|_{H^\beta(\Gamma_\ell)}, \quad 1<\beta\leqslant\sigma,\\ \|\nabla_\Gamma u - \nabla_\Gamma I_\phi u\|_{\mathbf{L}_\infty(\Gamma_\ell)} &\leqslant C h_{X_\ell,\Gamma_\ell}^{\beta-2}\|u\|_{H^\beta(\Gamma_\ell)}, \quad 2<\beta\leqslant\sigma,\\ \|(\Delta_\Gamma u)|_{X_\ell} - L_{X_\ell}\boldsymbol{u}_\ell\|_\infty &\leqslant C h_{X_\ell,\Gamma_\ell}^{\beta-3}\|u\|_{H^\beta(\Gamma_\ell)}, \quad 3<\beta\leqslant\sigma,\end{aligned} \tag{6.8}$$

for all $u\in H^\beta(\Gamma)$. Finally we have the following theorem.

Theorem 6.2. *Let Γ be a smooth 2-dimensional submanifold of $\mathbb{R}^3$ and $X\subset\Gamma$ be a discrete and quasi-uniform set of points with sufficiently small fill-distance $h_{X,\Gamma}$. Let $\{\Gamma_\ell\}_{\ell=1}^{N_c}$ be an open, bounded, regular and local (with respect to X) covering for Γ with PU functions $\{w_\ell\}$. If s^L is the direct PU approximation of Lu for $L=Id,\nabla_\Gamma,\Delta_\Gamma$, then*

$$|Lu(\boldsymbol{x}) - s^L(\boldsymbol{x})| \leqslant C h_{X,\Gamma}^{\beta-k-1}\|u\|_{H^\beta(\Gamma)}, \quad k+1<\beta\leqslant\sigma, \tag{6.9}$$

for all $u\in H^\beta(\Gamma)$ and all $\boldsymbol{x}\in X$ provided that $\Phi=\phi(\|\cdot\|_2)$ satisfies (6.7) with $\tau>k+3/2$ and $\sigma=\tau-1/2$ where $k=0,1,2$ for $L=Id,\nabla_\Gamma,\Delta_\Gamma$, respectively.

Proof. Since $\{w_\ell\}$ forms a partition of unity, we have for any $\boldsymbol{x}\in X$

$$\begin{aligned}|Lu(\boldsymbol{x}) - s^L(\boldsymbol{x})| &= \sum_{\ell\in I(\boldsymbol{x})} w_\ell(\boldsymbol{x})\,|Lu(\boldsymbol{x}) - s_\ell^L(\boldsymbol{x})|\\ &\leqslant \sum_{\ell\in I(\boldsymbol{x})} w_\ell(\boldsymbol{x})\,\|(Lu)|_X - (s_\ell^L)|_X\|_\infty.\end{aligned}$$

If we assume that $\|(Lu)|_X - (s_\ell^L)|_X\|_\infty \leqslant \varepsilon_\ell^L$ and $\varepsilon^L(\boldsymbol{x}) := \max_{\ell\in I(\boldsymbol{x})}\varepsilon_\ell^L$ then we have

$$|Lu(\boldsymbol{x}) - s^L(\boldsymbol{x})| \leqslant \varepsilon^L(\boldsymbol{x}), \quad \boldsymbol{x}\in X.$$

It remains to find the local bounds ε_ℓ^L for $\ell=1,\ldots,N_c$, which are obtainable from (6.8) because we have $(s_\ell^L)|_{X_\ell} = (I_\phi u)|_{X_\ell}$ for $L=Id$, $(s_\ell^L)|_{X_\ell} = (\nabla_\Gamma I_\phi u)|_{X_\ell}$ for $L=\nabla_\Gamma$ and $(s_\ell^L)|_{X_\ell} = L_{X_\ell}\boldsymbol{u}_\ell$ for $L=\Delta_\Gamma$. Finally, since $h_{X_\ell,\Gamma_\ell}\leqslant h_{X,\Gamma}$ and $\|u\|_{H^\beta(\Gamma_\ell)}\leqslant\|u\|_{H^\beta(\Gamma)}$ for all $\ell=1,\ldots,N_c$, the final bound (6.9) results. □

6.2. Cost comparison

The total computational cost of a localized RBF-based method consists of (i) the cost of setting up the final differentiation matrix, and (ii) the cost of solving the final linear system. Usually, the final system of such a method is highly sparse and can be effectively inverted using the available iterative linear algebra solvers for sparse and large systems. Thus, the majority of the computational costs arises in the first phase where on one hand a data structure should be performed to collect the indices of local sets, and on the other hand lots of local linear systems should be solved for calculating the stencil weights or constructing the local approximants.

Assume that we are given N quasi-uniform trial and N test points together with N_c number of PU patches on surface Γ. According to the construction, N_c is much smaller than N. Besides, the number of trial points in each patch is independent of $h_{X,\Gamma}$ (and thus independent of N) because the covering is assumed to be both local and regular. This means that the number of points in each patch remains unchanged, approximately, when the discretization is refined. Thus if $n_\ell := |X_\ell|$ then there exists a global constant n_P independent of the discretization distance $h_{X,\Gamma}$ such that $n_\ell\leqslant n_P$. This is true for the size of RBF-FD stencils as well and we can assume $|\widetilde{X}_k|\leqslant n_P$ for $1\leqslant k\leqslant N$. Assume that the set of indices of trial points in each local domain is known in advance. In D-RBF-PU N_c and in RBF-FD N local linear systems should be solved to set up the final matrix. In fact, in RBF-FD the stencils are changed per test points while in D-RBF-PU a local patch is shared with many test points in a certain neighborhood. Consequently, the total cost for solving all local systems is of order $n_P^3 N_c$ for the D-RBF-PU method and it is of order $n_P^3 N$ for the RBF-FD method. As in the PU approximation N_c is a small fraction of N, say $N_c\approx\frac{N}{k}$, the D-RBF-PU is k times faster than RBF-FD for solving all local systems. We will observe this in numerical experiments of section 7.

The cost of collecting the indices of points among all N trial points in each patch or stencil is ignored in the above cost analysis. This can be done in a preprocessing step using a boxing strategy to avoid collecting local indices for each patch or each stencil separately. The overall complexity would be of order $2N$ for RBF-FD and of order $N+N_c$ for D-RBF-PU. As a more relevant strategy we can use a kd-tree algorithm that takes $\mathcal{O}(N\log N)$ time for building a tree on X and $\mathcal{O}(N^{1-1/d})$ time for range searchings.

Finally, in D-RBF-PU the complexity of evaluating the final solution at an evaluation set X_e is dominated by the cost of building another data structure on X_e to collect the indices of evaluation points in each patch Γ_ℓ. However, in RBF-FD new stencils should be formed per each evaluation point and new local systems should be solved.

Altogether, both methods possess the same complexity rate but D-RBF-PU is approximately k times faster than RBF-FD where N is approximately k times N_c.

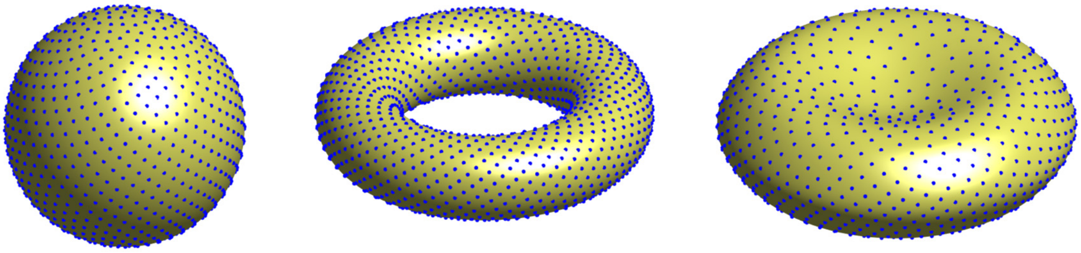

Fig. 2. Phyllotaxis spiral nodes on sphere, torus, and red blood cell (RBC).

7. Numerical experiments

The first part of this section is devoted to some numerical experiments of the D-RBF-PU method and its comparison with the RBF-FD method for a PDE of the form

$$\Delta_\Gamma u(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x}) \cdot \nabla_\Gamma u(\boldsymbol{x}) + c(\boldsymbol{x}) u(\boldsymbol{x}) = f(\boldsymbol{x})$$

on some smooth embedded submanifolds $\Gamma \subset \mathbb{R}^3$ of dimension 2. Here, $\boldsymbol{b} \in \mathbb{R}^3$ and $c \in \mathbb{R}$ are sufficiently smooth functions of spatial variable $\boldsymbol{x} \in \Gamma$, and f is a given source function. In numerical experiments of this part the unit sphere, a torus and a red blood cell (RBC) are used as computational domains. In Cartesian coordinates these surfaces are defined as

$$\Gamma = \left\{ (x, y, z) \in \mathbb{R}^3 : \, x^2 + y^2 + z^2 = 1 \right\},$$

$$\Gamma = \left\{ (x, y, z) \in \mathbb{R}^3 : \, \left(c_1 - \sqrt{x^2 + y^2}\right)^2 + z^2 = r_1^2 \right\},$$

$$\Gamma = \left\{ (x, y, z) \in \mathbb{R}^3 : \, \left(1 - \frac{x^2+y^2}{r_2^2}\right)\left(c_0 + c_2\left(\frac{x^2+y^2}{r_2^2}\right) + c_4\left(\frac{x^2+y^2}{r_2^2}\right)^2\right)^2 - 4z^2 = 0 \right\}.$$

We set $c_1 = 1$, $r_1 = \frac{1}{3}$, $c_0 = 0.25$, $c_2 = 2$, $c_4 = -1.5$ and $r_2 = 1$. For trial points and patch centers we use phyllotaxis spiral nodes based on the idea given in [38] in all experiments. These nodes are also available for download from [22]. The points on RBC are obtained by mapping the points on the unit sphere using the parametric equations of this surface. Example node sets are presented in Fig. 2.

A PU covering $\{\Gamma_\ell\}$ is simply constructed by letting $\Gamma_\ell = B(\boldsymbol{\omega}_\ell, \rho) \cap \Gamma$ for N_c quasi-uniform centers $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{N_c}\} =: X_c \subset \Gamma$. Here $B(\boldsymbol{\omega}, \rho)$ is the open ball in $\mathbb{R}^3$ of radius ρ and center $\boldsymbol{\omega}$. Patch centers $\boldsymbol{\omega}_\ell$ on Γ are obtained by the same procedure as described for trial points. For quasi-uniform points we set $N_c = \lceil N/16 \rceil$ to have $h_{X_c,\Gamma} \approx 4h_{X,\Gamma}$. The patch radii are set to be proportional to $h_{X_c,\Gamma}$, say $\rho = C_{ovlp} h_{X_c,\Gamma}$, to obtain a local and regular covering. The overlapping constant $C_{ovlp} > 0$ should be large enough to ensure both the inclusion $\Gamma \subset \cup \Gamma_\ell$ and the accuracy of local approximants. On the other side, it should be as small as permissible to get a reasonably sparse final linear system. We will illustrate some experiments to come up with a proper constant C_{ovlp}.

As a smooth PU weight, the function

$$\Psi_\ell = \psi\left(\| \cdot - \boldsymbol{\omega}_\ell \|_2 / \rho_\ell\right), \quad \psi(r) = (1-r)_+^6 (35r^2 + 18r + 3),$$

is employed in (5.2) where $\psi(r)$ is the C^4 compactly supported Wendland's function. However, we will apply the constant-generated PU weight function (6.6) in most of our experiments.

In this study the restrictions of kernels

$$\begin{aligned} &\text{Matérn (MA):} && \phi(r) = (\varepsilon r)^{\tau - \frac{d}{2}} K_{\tau - \frac{d}{2}}(\varepsilon r), \ \tau > d/2, \\ &\text{Wendland (WE):} && \phi(r) = (1 - \varepsilon r)_+^8 \left(32(\varepsilon r)^3 + 25(\varepsilon r)^2 + 8\varepsilon r + 1\right) \end{aligned}$$

on different surfaces are used as basis functions. The first kernel is positive definite on $\mathbb{R}^d$ for $d < 2\tau$ while the second one is positive definite on $\mathbb{R}^d$ for $d \leqslant 3$. The native space of the MA kernel on the all above 2 dimensional surfaces is $H^{\tau - 1/2}(\Gamma)$ while it is $H^{4.5}(\Gamma)$ for the WE kernel. The parameter $\tau = 5$ is used in all experiments. In both cases the shape parameter $\varepsilon > 0$ is tuned such that a reasonable accuracy is obtained without encountering any serious instability. We set $\varepsilon = 3$ and $\varepsilon = 1/3$ for MA and WE, respectively.

In all cases we use the function

$$u(x, y, z) := \cos(10z + 3)\, e^{-x^2} - \sin(4x + 5y),$$

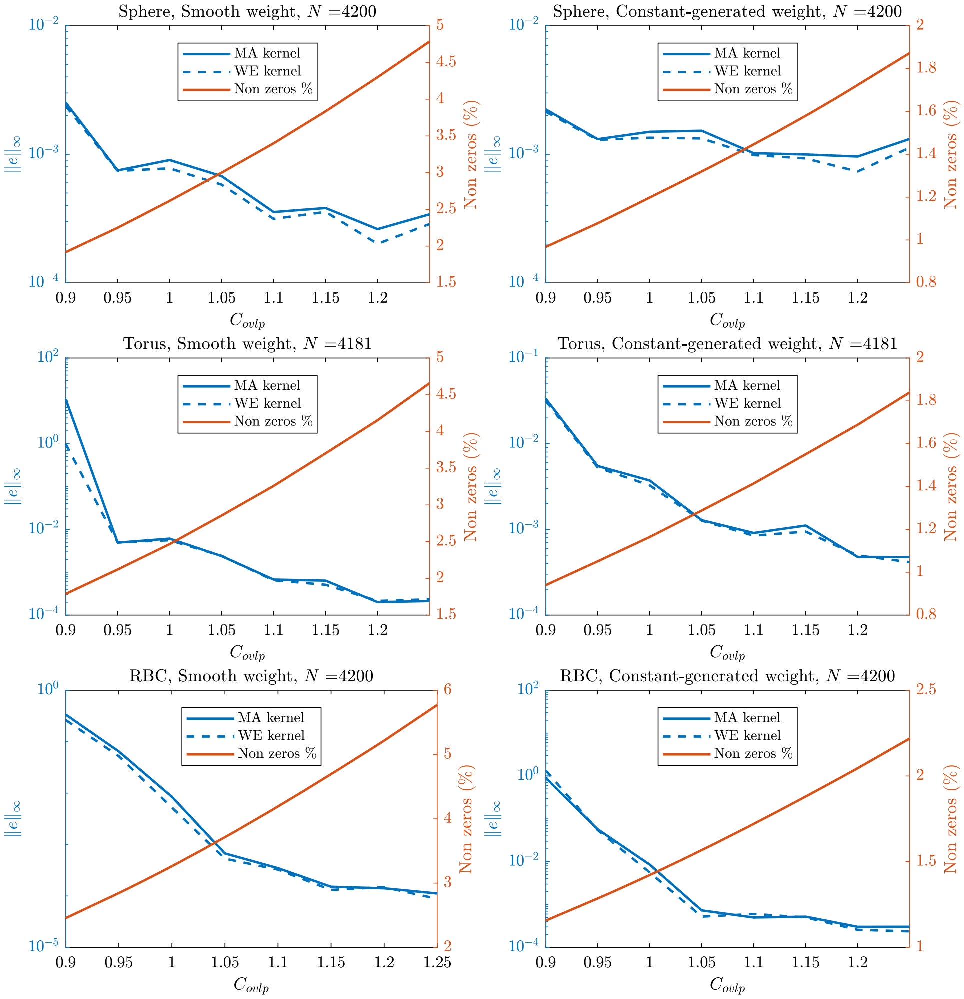

Fig. 3. Accuracy (left vertical axis) and sparsity (right vertical axis) with respect to the overlap constant C_{ovlp} in D-RBF-PU method. First row: sphere, second row: torus, third row: RBC. Left column: the smooth PU weight, right column: the constant-generated PU weight.

as a true solution of the PDE. The velocity vector $\boldsymbol{b}(\boldsymbol{x}) \equiv (1, 2, -1)$ and $c(\boldsymbol{x}) \equiv 1$ are used in all experiments. In each case, the right hand side function f is calculated, accordingly.

To adjust a good overlapping constant C_{ovlp} we test the accuracy vs. amount of sparsity for different surfaces with both MA and WE kernels. Some results are given in Fig. 3 on sphere, torus and RBC surfaces. Relative errors (left vertical axis) and percentage of nonzero elements of the final matrix (right vertical axis) are shown in terms of the overlap constant C_{ovlp}. The smooth PU weight is applied for the left panels and the constant-generated PU weight for the right ones. According to these results and other experiments with different numbers of points which are not presented here for the sake of brevity, we set $C_{ovlp} = 1.1$ to have a balance between accuracy and sparsity. We note that in this experiment the iterative approach is applied for discretizing the Laplace-Beltrami operator. While not presented here, a similar behavior is observed with the standard approach. We further observe from Fig. 3 that the D-RBF-PU method with the constant-generated weight is about 2.5 times sparser than the case with the smooth weight function.

In Fig. 4 errors and convergence orders of D-RBF-PU with smooth and constant-generated weights and comparison with RBF-FD on three surfaces using WE and MA kernels are depicted. For results on this figure the Laplace-Beltrami operator is discretized based on the repeated interpolation and differentiation approach. The same results for standard approach are given in Fig. 5. In all plots computational convergence orders are obtained by the linear least squares fitting to error values and are written alongside the figure legends. To make the comparison fair the RBF-FD stencils are chosen to be $\widetilde{X}_k = B(\boldsymbol{x}_k, \rho) \cap X$ where ρ is the patch radius in D-RBF-PU. Both methods possess the same theoretical order although in this experiment the D-RBF-PU method with both smooth and discontinuous weights gives more accurate results and higher convergence orders. However, it seems possible to scale both methods in different ways to obtain better accuracies and convergence rates.

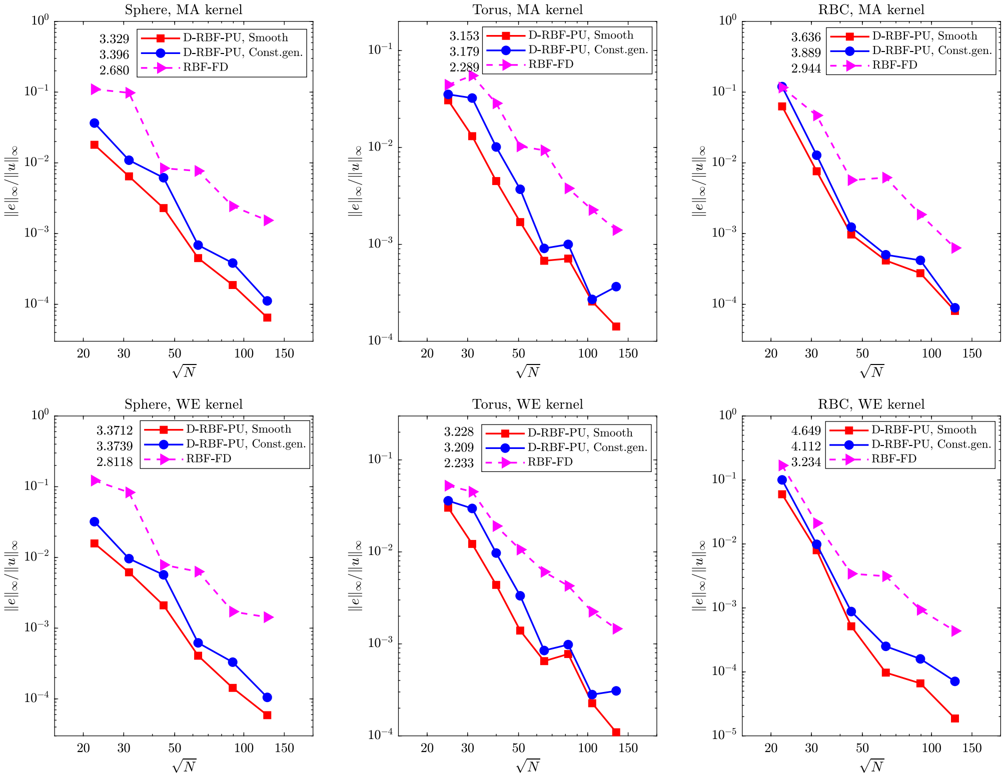

Fig. 4. Errors and convergence orders of D-RBF-PU with smooth and constant-generated weights and comparison with RBF-FD on sphere, torus and RBC surfaces using Mátern (first row) and Wendland (second row) kernels. The repeated interpolation and differentiation approach is applied to discretize the Laplace-Beltrami operator. Computational convergence orders are written alongside the figure legends.

A selection of plots for the stability constant is presented in Fig. 6 for the three surfaces with WE kernel. While not presented here, the same behavior is observed for experiments with the MA kernel. The final matrix is extremely stable as $\widetilde{C}_S \approx 1$ in almost all cases. We emphasize that the stability constant $\widetilde{C}_S$ measures the conditioning of the final sparse differentiation matrix. It is of course different from the conditioning of local RBF systems on PU patches. The problem of conditioning of local systems at small values of the shape parameter ε might be treated by some tricky techniques such as Contour-Padé or RBF-RA algorithms [57,58]. Equipping the method with such stabilization techniques allows to run the algorithm for higher values of N to obtain a better accuracy. We did not pursue this issue in this paper, instead we tuned the shape parameter to obtain a reasonable accuracy without encountering any instability.

As we pointed out before, D-RBF-PU is computationally cheaper than RBF-FD. In Fig. 7 the CPU times used for setting up and for solving the final linear systems together with the total CPU times are compared. Since these systems are sparse and well-conditioned, the solving time is very small compared to the setting up time. Looking at the total time plot, although the rates are the same, D-RBF-PU is about 10 times faster than RBF-FD because in D-RBF-PU much fewer number of local systems should be solved to form the final matrix. While not presented here, cost plots for other surfaces obey the same behavior.

7.1. Application to reaction-diffusion equations

Among others, reaction-diffusion equations arise from mathematical models to describe Turing pattern formations in some biological applications and spiral waves in an excitable chemical media [3,14,34,35,51,53]. The solutions of reaction-diffusion equations can exhibit pattern forming from random initial conditions. However, most results reported in the literature concern these phenomena on Euclidean domains. Growing attention has been recently paid for such models on general surfaces to investigate the effect of curvature to pattern and wave formations. Applications on simple surfaces such as spheres and cones could be found in [37,54,59] and applications on general surfaces may be found in [5,6,8,21,25,29,39,52]. We consider the system of reaction-diffusion equations

$$\begin{aligned} \frac{\partial u}{\partial t} &= f(u,v) + \delta_1 \Delta_\Gamma u \\ \frac{\partial v}{\partial t} &= g(u,v) + \delta_2 \Delta_\Gamma v \end{aligned} \tag{7.1}$$

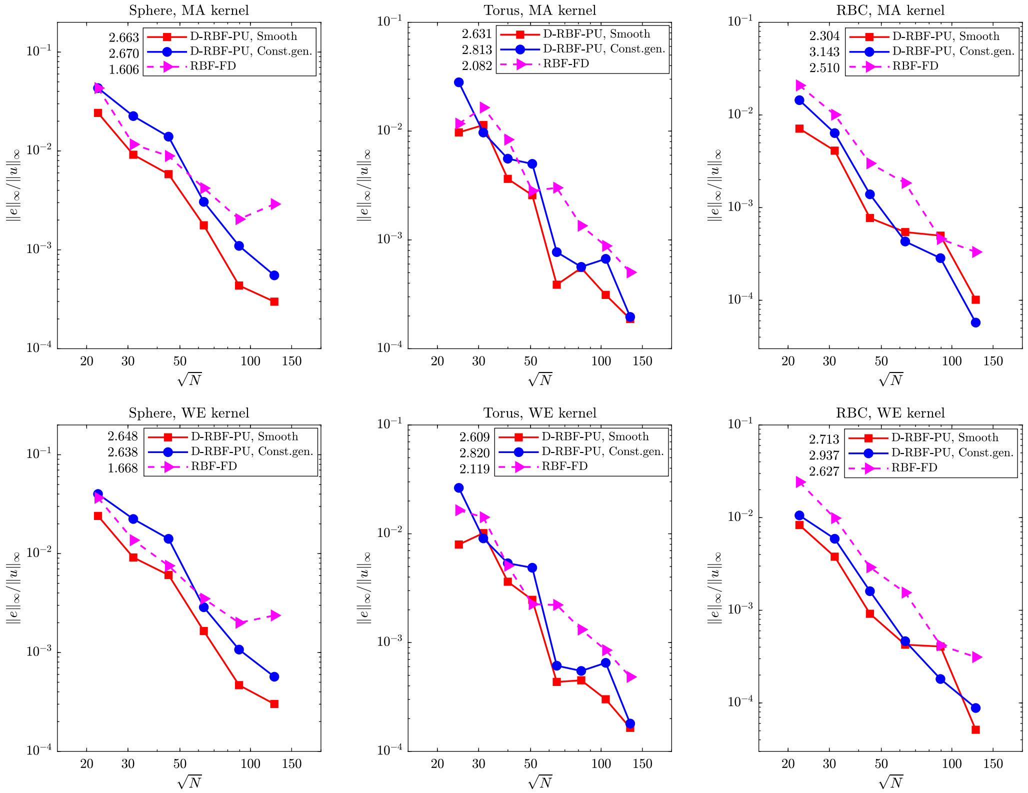

Fig. 5. Errors and convergence orders of D-RBF-PU with smooth and constant-generated weights and comparison with RBF-FD on sphere, torus and RBC surfaces using Mátern (first row) and Wendland (second row) kernels. The standard approach is applied to discretize the Laplace-Beltrami operator. Computational convergence orders are written alongside the figure legends.

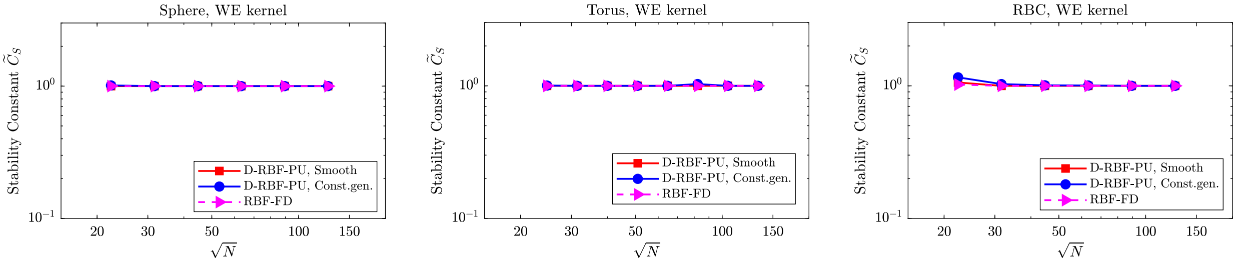

Fig. 6. The behavior of the stability constant $\widetilde{C}_S$ for D-RBF-PU and RBF-FD methods on different surfaces. Results show a perfectly stable final linear system in each case.

where f and g are nonlinear reaction terms and δ_1 and δ_2 are diffusion parameters. Here $u = u(\boldsymbol{x}, t)$ and $v = v(\boldsymbol{x}, t)$ are chemical unknowns where $\boldsymbol{x}$ and t stand for spatial and time variables, respectively. The initial conditions are imposed by $u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x})$ and $v(\boldsymbol{x}, 0) = v_0(\boldsymbol{x})$ for $\boldsymbol{x} \in \Gamma$ and known functions u_0 and v_0.

In this section we consider two examples of this time-dependent system on different surfaces. Additions to surfaces defined in the previous subsection, a Dupin's Cyclide, a tooth shape surface and a frog will be used in our experiments. The surfaces and example node sets are presented in Fig. 8. The first two surfaces are respectively defined as

$$\Gamma = \{(x, y, z) \in \mathbb{R}^3 : (x^2 + y^2 + z^2 + c_2^2 - c_4^2)^2 - 4(c_1 x - c_3 c_4)^2 - 4c_2^2 y^2 = 0\}$$
$$\Gamma = \{(x, y, z) \in \mathbb{R}^3 : x^8 + y^8 + z^8 - (x^2 + y^2 + z^2) = 0\}.$$

The parameters for Dupin's Cyclide are $c_1 = 2$, $c_2 = 1.9$, $c_3 = \sqrt{0.39}$ and $c_4 = 1$. These two surfaces are used for example in [28] to treat the same problem using a compact RBF-FD method. For the case of frog we are only given a cloud X containing 5900 points and approximate normals at each point. A subset of 530 points out of X is used as patch centers and a varying patch radius strategy is applied so that each patch contains exactly 41 interpolation points.

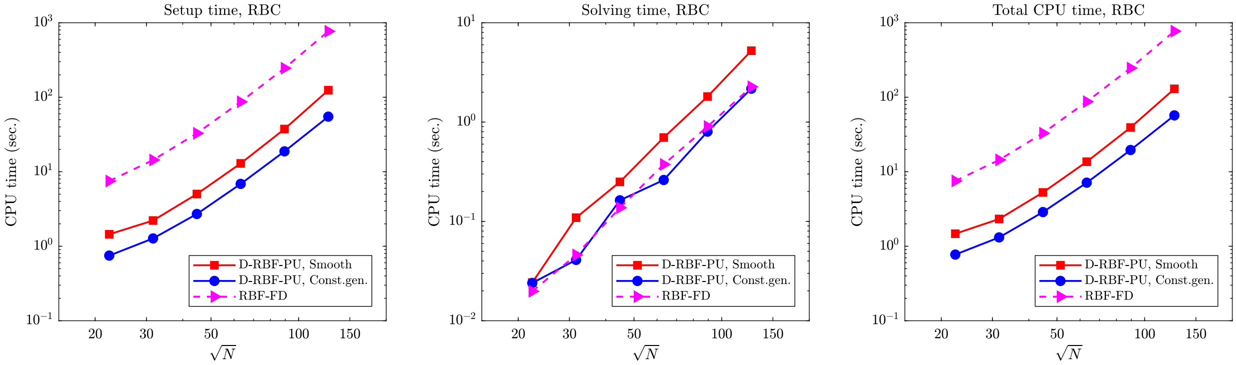

Fig. 7. The CPU times used for setting up and solving the final linear system on RBC surface at different numbers of points. The rates are the same but D-RBF-PU is approximately 10 times faster.

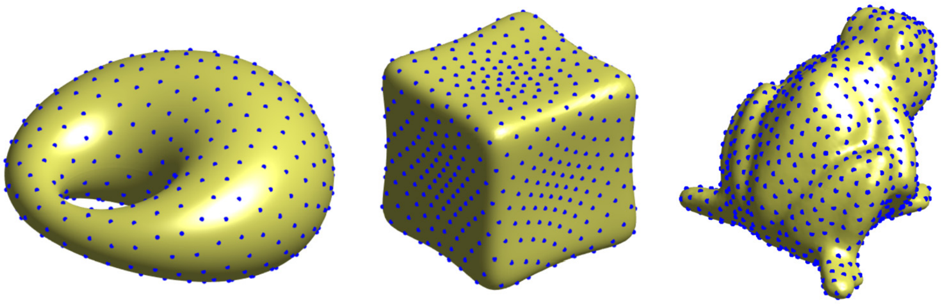

Fig. 8. Example node sets on Dupin's Cyclide, tooth shape and frog surfaces.

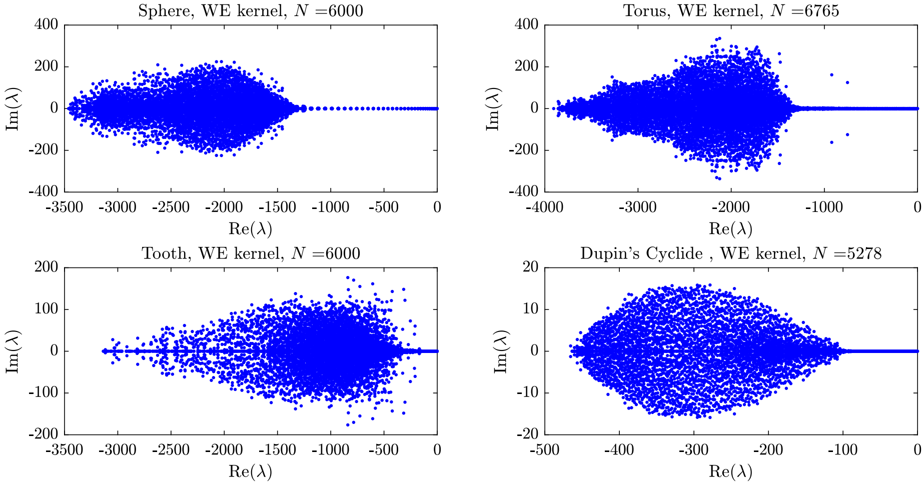

Fig. 9. The spectrum of the discrete Laplace-Beltrami with D-RBF-PU method and discontinuous PU weight on different surfaces. All eigenvalues fall on the left half plane.

The D-RBF-PU method is applied for discretizing the spatial variables and the semi-implicit backward differential formula of order two (SBDF2) [2] is employed for time integration. At the first time step SBDF1 is used to bootstrap SBDF2. The final linear system was efficiently solved by the GMRES algorithm with a standard incomplete LU preconditioner. It is known that as a necessary condition for time stability the spectrum of the Laplace-Beltrami differentiation matrix should fall in the left half plane. Experiments show that if the fill distance $h_{X,\Gamma}$ is small enough then it appears that all of the eigenvalues do in fact lie in the left half plane. In few cases, small positive real parts of order $10^{-16} - 10^{-8}$ appear in the spectrum. This might be bypassed by applying a small diagonal increment on the discrete Laplacian matrix. Fig. 9 shows the eigenvalue patterns on different surfaces for some specific number of nodes which are used for numerical simulations in the following. The WE kernel is used in this and all experiments from here on.

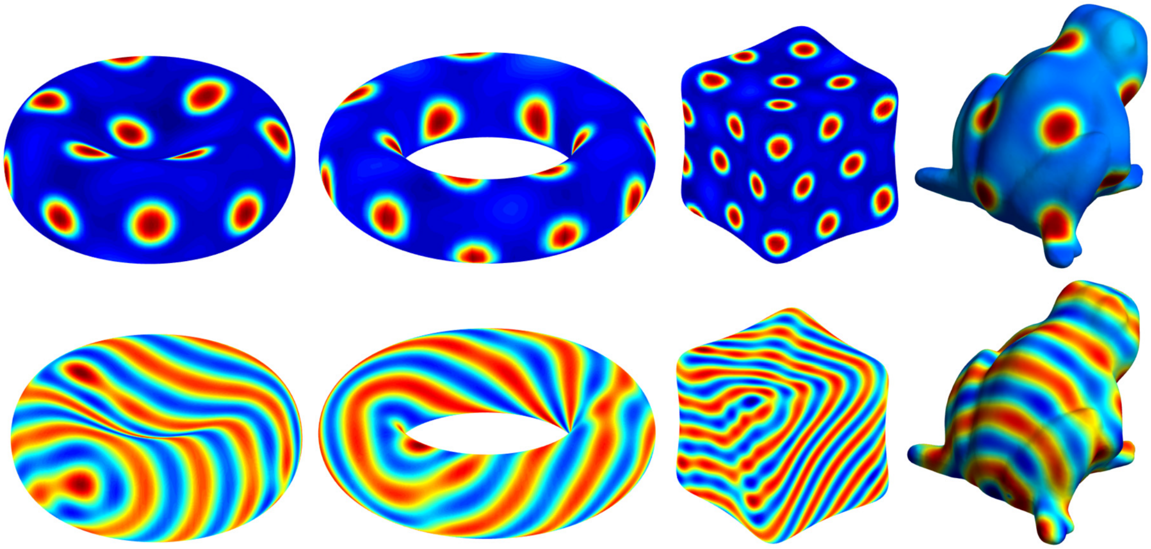

Fig. 10. Quasi steady Turing spots and stripe patterns resulting from solving the Brusselator equation. In all plots, red corresponds to a high concentration of u and blue to a low concentration. (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

To model the Turing patterns we consider a particular form of (7.1) with

$$f(u,v) = \alpha u(1-\tau_1 v^2) + v(1-\tau_2 u),$$
$$g(u,v) = \beta v\left(1+\frac{\alpha\tau_1}{\beta}uv\right) + u(\gamma+\tau_2 v)$$

for known parameters $\alpha, \beta, \tau_1, \tau_2$ and γ. The system (7.1) with the above reaction terms is known as the Brusselator equation where $(u,v)=(0,0)$ is a unique steady state for $\gamma=-\alpha$ [4,8,21,54]. In the absence of diffusion terms ($\delta_1=\delta_2=0$) and with certain parameter values the steady state $(0,0)$ is linearly stable while in the presence of diffusion it is unstable [51]. Depending on the values of other coefficients this instability exhibits different types of patterns when the steady state is reached [4]. Here we illustrate spots and stripes patterns.

In our experiments the initial condition is given by $u=v=0$ except on a bound of width 0.1 around the equator of each surface where the values of u and v are chosen to be uniformly randomly distributed in $[-0.5, 0.5]$. We use the `RandStream` function in MATLAB to generate such values [19]. We set $\Delta t = 0.01$ in SBDF2 and execute the simulation until a steady state solution is reached. To obtain spot patterns we set $\delta_2 = 0.0045$, $\delta_1 = 0.516\delta_2$, $\tau_1 = 0.02$, $\tau_2 = 0.2$, $\alpha = 0.899$, $\beta = -0.91$, and $\gamma = -\alpha$. For stripe patterns we use $\delta_2 = 0.0021$, $\tau_1 = 3.5$, $\tau_2 = 0$ and keep the values of other parameters unchanged. These values are all motivated from the values used in [8,19]. In Fig. 10 the steady state values of u, the resulting spot and stripe patterns, on RBC, torus, tooth and frog are shown. These results are in good agreement with those calculated in [19] and other sources.

Our next application of the D-RBF-PU method for solving a reaction-diffusion equation on surfaces concerns the equation (7.1) with reaction terms

$$f(u,v) = \frac{1}{\alpha}u(1-u)\left(u-\frac{v+b}{a}\right),$$
$$g(u,v) = u - v.$$

This is a Fitzhugh-Nagumo type system which is a simple model for dynamics of many excitable media. Here, u and v can be considered as some chemical concentrations or as membrane potential and current. The known parameters α, a and b govern the reaction kinetics. Small values of parameter α result in two separated regions on the surface each with values $u=0$ and $u=1$ with a thin interface separating these two regions. The solution of this system is a spiral wave evolving on the surface. Here we present some results on the sphere and on the Dupin's Cyclide. Results are presented in Fig. 11 at different time levels with initial conditions

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

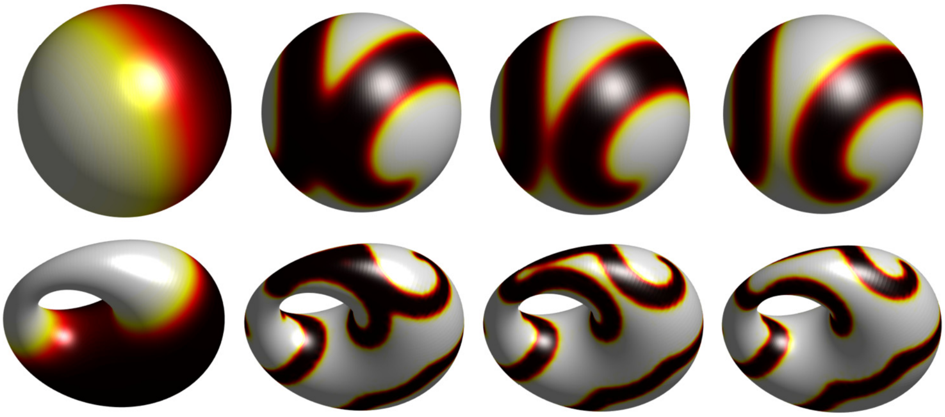

Fig. 11. Spiral wave patterns resulting from solving the Fitzhugh–Nagumo equation. The first row shows the excitation variable u on the sphere at times $t = 0, 15, 18$ and 21 (from left to right). The second row shows u on Dupin's Cyclide at times $t = 0, 42, 45$ and 48 (from left to right). In all plots, black corresponds to a high concentration of u and white to a low concentration.

on both surfaces. Motivating by [21], parameter values $a = 0.75$, $b = 0.02$, $\alpha = 0.02$, $\delta_1 = 2.5(2\pi/50)^2$ and $\delta_2 = 0$ are assigned. We observe from Fig. 11 that the initial density u evolves into approximately symmetric spiral waves that remain intact but meander around the surfaces throughout the simulation time.

8. Conclusion

In this work we have presented a novel localized RBF method based on partition of unity for discretizing a large class of PDE problems on manifolds. The new D-RBF-PU method yields a numerical algorithm that is more efficient, more accurate and faster than standard RBF-PU and RBF-FD methods. It can be easily applied to any smooth orientable surface that is discretized with a set of scattered nodes and with approximations to the normal to the surface at each of the nodes. As an important property we can mention that in the new method surface PDE operators are not required to be applied on rational PU weight functions. This property simplifies the algorithm, highly, and allows the use of some discontinuous weights to develop a sparser (than RBF-PU) and faster (than RBF-FD) localized RBF technique. The consistency bounds were obtained for finitely smooth basis functions whose Fourier transforms decay only algebraically. Moreover, the stability, accuracy and efficiency properties of the scheme were tested on a general PDE defined on different surfaces.

In additions, we have shown how the new method can be easily adaptable to reaction-diffusion equations on surfaces. In particular, the method was applied on the Brusselator equation and on a Fitzhugh-Nagumo type system to obtain the formation of Turing patterns and spiral waves in an excitable media, respectively. For advancing in time, the semi-implicit BDF2 scheme was employed for time discretization. The final linear system was efficiently solved by the GMRES algorithm with a standard incomplete LU preconditioner. Experiments showed that the final matrices on different manifolds satisfy the conditions that ensure eigenvalue stability for temporal integration.

Finally, we stress that the method can be used to solve various types of PDEs on surfaces. An extension to convection-diffusion problems is left for a future work. Moreover, the proposed method can be extended for solving a boundary value problem defined on a surface subdomain with a sufficiently smooth boundary. Imposing the boundary condition using the proposed RBF based method is straightforward because the boundary can be viewed as another submanifold with co-dimension 2.

CRediT authorship contribution statement

Reyhaneh Mir: Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Davoud Mirzaei:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Supervision, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Davoud Mirzaei reports financial support was provided by Institute for Research in Fundamental Sciences. Davoud Mirzaei reports financial support was provided by Iran National Science Foundation.

Data availability

Data will be made available on request.

Acknowledgements

The second author was in part supported by a grant from IPM, No. 1400650417 and by a grant form INSF, No. 98012657. We greatly acknowledge the helpful comments of anonymous reviewers.

References

[1] K. Aiton, A radial basis function partition of unity method for transport on the sphere, Master's thesis, Boise State University, Boise, Idaho, 2014.
[2] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit-explicit methods for time-dependent PDEs, SIAM J. Numer. Anal. 32 (1997) 797–823.
[3] R.E. Baker, E.A. Gaffney, P.K. Maini, Partial differential equations for self-organization in cellular and developmental biology, Nonlinearity 21 (2008) 251–290.
[4] R. Barrio, C. Varea, J. Aragón, P. Maini, A two-dimensional numerical study of spatial pattern formation in interacting Turing systems, Bull. Math. Biol. 61 (1999) 483–505.
[5] M. Bergdorf, I. Sbalzarini, P. Koumoutsakos, A Lagrangian particle method for reaction-diffusion systems on deforming surfaces, J. Math. Biol. 61 (2010) 649–663.
[6] M. Bertalmio, L.-T. Cheng, S. Osher, G. Sapiro, Variational problems and partial differential equations on implicit surfaces, J. Comput. Phys. 174 (2001) 759–780.
[7] M. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, Cambridge, New York, 2003.
[8] D.A. Calhoun, C. Helzel, A finite volume method for solving parabolic equations on logically Cartesian curved surface meshes, SIAM J. Sci. Comput. 31 (6) (2010) 4066–4099.
[9] R. Cavoretto, A. De Rossi, E. Perracchione, Partition of unity interpolation on multivariate convex domains, Int. J. Model. Simul. Sci. Comput. 6 (04) (2015) 1550034.
[10] M.R. Ahmadi Darani, The RBF partition of unity method for solving the Klein-Gordon equation, Eng. Comput. 37 (2021) 795.
[11] S. De Marchi, A. Martínez, E. Perracchione, Fast and stable rational RBF-based partition of unity interpolation, J. Comput. Appl. Math. 349 (2019) 331–343.
[12] S. De Marchi, A. Martínez, E. Perracchione, M. Rossini, RBF-based partition of unity methods for elliptic PDEs: adaptivity and stability issues via variably scaled kernels, J. Sci. Comput. 79 (1) (2019) 321–344.
[13] K.P. Deake, E.J. Fuselier, G.B. Wright, A partition of unity method for divergence-free or curl-free radial basis function approximation, SIAM J. Sci. Comput. 43 (2021) A1950–A1974.
[14] I.R. Epstein, J.A. Pojman, An Introduction to Nonlinear Chemical Dynamics, Oxford University Press, Oxford, 1998.
[15] E. Farazandeh, D. Mirzaei, A rational RBF interpolation with conditionally positive definite kernels, Adv. Comput. Math. 47 (2021) 74.
[16] G.E. Fasshauer, Meshfree Approximation Methods with Matlab, World Scientific, 2007.
[17] N. Flyer, E. Lehto, S. Blaise, G.B. Wright, A. St-Cyr, A guide to RBF-generated finite differences for nonlinear transport: shallow water simulations on a sphere, J. Comput. Phys. 231 (11) (2012) 4078–4095.
[18] B. Fornberg, N. Flyer, Solving PDEs with radial basis functions, Acta Numer. (2015) 215–258.
[19] E. Fuselier, T. Hangelbroek, F.J. Narcowich, J.D. Ward, G.B. Wright, Localized bases for kernel spaces on the unit sphere, SIAM J. Numer. Anal. 51 (2013) 2538–2562.
[20] E. Fuselier, G.B. Wright, Scattered data interpolation on embedded submanifolds with restricted positive definite kernels: Sobolev error estimates, SIAM J. Numer. Anal. 50 (2012) 1753–1776.
[21] E. Fuselier, G.B. Wright, A high-order kernel method for diffusion and reaction-diffusion equations on surfaces, J. Sci. Comput. 56 (2013) 535–565.
[22] E.J. Fuselier, Nodes used in order-preserving approximation of derivatives with periodic radial basis functions, accessed 2012, http://math.highpoint.edu~efuselier/OrderPreservingData/.
[23] H.S. Ho, B. Lui, X. Xing, M.Y. Wang, Parametric shape and topology optimization with radial basis functions and partition of unity method, in: AIP Conference Proceedings, vol. 1233, American Institute of Physics, 2010, pp. 276–281.
[24] H.S. Ho, B.F. Lui, M.Y. Wang, Parametric structural optimization with radial basis functions and partition of unity method, Optim. Methods Softw. 26 (4–5) (2011) 533–553.
[25] H. Kim, A. Yun, S. Yoon, C. Lee, J. Park, J. Kim, Pattern formation in reaction-diffusion systems on evolving surfaces, Comput. Math. Appl. 80 (2020) 2019–2028.
[26] E. Larsson, V. Shcherbakov, A. Heryudono, A least squares radial basis function partition of unity method for solving PDEs, SIAM J. Sci. Comput. 39 (2017) A2538–A2563.
[27] J.M. Lee, Introduction to Smooth Manifolds, Graduate Texts in Mathematics, vol. 218, Springer, New York, 2003.
[28] E. Lehto, V. Schankar, G.B. Wright, A radial basis function (RBF) compact finite difference (FD) scheme for reaction-diffusion equations on surfaces, SIAM J. Sci. Comput. 39 (2017) A2129–A2151.
[29] C.B. Macdonald, S.J. Ruuth, The implicit closest point method for the numerical solution of partial differential equations on surfaces, SIAM J. Sci. Comput. 31 (6) (2010) 4330–4350.
[30] D. Mirzaei, The direct radial basis function partition of unity (D-RBF-PU) method for solving PDEs, SIAM J. Sci. Comput. 43 (2021) A54–A83.
[31] D. Mirzaei, R. Schaback, Direct Meshless Local Petrov-Galerkin (DMLPG) method: a generalized MLS approximation, Appl. Numer. Math. 33 (2013) 73–82.
[32] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives, IMA J. Numer. Anal. 32 (2012) 983–1000.
[33] V. Mohammadi, D. Mirzaei, M. Dehghan, Numerical simulation and error estimation of the time-dependent Allen-Cahn equation on surfaces with radial basis functions, J. Sci. Comput. 79 (2019).
[34] J.D. Murray, Mathematical Biology I: An Introduction, 3rd edition, Springer, New York, 2007.
[35] J.D. Murray, Mathematical Biology II: Spatial Models and Biomedical Applications, 3rd edition, Springer, New York, 2008.
[36] A. Petras, L. Ling, S.J. Ruuth, An RBF-FD closest point method for solving PDEs on surfaces, J. Comput. Phys. 370 (2018) 43–57.
[37] R.G. Plaza, F. Sánchez-Garduno, P. Padilla, R.A. Barrio, P.K. Maini, The effect of growth and curvature on pattern formation, J. Dyn. Differ. Equ. 16 (2004) 1093–1121.
[38] J.N. Ridley, Ideal phyllotaxis on general surfaces of revolution, Math. Biosci. 79 (1986) 1–24.
[39] S.J. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, J. Comput. Phys. 227 (2008) 1943–1961.
[40] A. Safdari-Vaighani, A. Heryudono, E. Larsson, A radial basis function partition of unity collocation method for convection—diffusion equations arising in financial applications, J. Sci. Comput. 64 (2) (2015) 341–367.
[41] R. Schaback, A unified theory of radial basis functions. Native Hilbert spaces for radial basis functions. II, J. Comput. Appl. Math. 121 (1–2) (2000) 165–177.

[42] R. Schaback, Direct discretization with application to meshless methods for PDEs, in: Dolomites Research Notes on Approximation, Proceedings of DWCAA12, vol. 6, 2013, pp. 37–51.

[43] R. Schaback, Error analysis of nodal meshless methods, in: Meshfree Methods for Partial Differential Equations VIII, Bonn, 2015, in: Lecture Notes in Computational Science and Engineering, vol. 115, Springer, 2017, pp. 117–143.

[44] R. Schaback, All well–posed problems have uniformly stable and convergent discretizations, Numer. Math. 132 (2016) 597–630.

[45] V. Shankar, The overlapped radial basis function-finite difference (RBF-FD) method: a generalization of RBF-FD, J. Comput. Phys. 342 (2017) 211–228.

[46] V. Shankar, G.B. Wright, R.M. Kirby, A.L. Fogelson, A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction-diffusion equations on surfaces, J. Sci. Comput. 63 (2015) 745–768.

[47] V. Shcherbakov, Radial basis function partition of unity operator splitting method for pricing multi-asset American options, BIT Numer. Math. 56 (2016) 1401–1423.

[48] V. Shcherbakov, E. Larsson, Radial basis function partition of unity methods for pricing Vanilla basket options, Comput. Math. Appl. 71 (1) (2016) 185–200.

[49] A. Sokolov, O. Davydov, D. Kuzmin, A. Westermann, S. Turek, A flux-corrected RBF-FD method for convection dominated problems in domains and on manifolds, J. Numer. Math. 27 (4) (2019) 253–269.

[50] P. Suchde, J. Kuhnert, A meshfree generalized finite difference method for surface PDEs, Comput. Math. Appl. 78 (8) (2019) 2789–2805.

[51] A. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. B 237 (1952) 37–72.

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

[53] J.J. Tyson, J.P. Keener, Singular perturbation theory of traveling waves in excitable media (a review), Physica D 32 (1988) 327–361.

[54] C. Varea, J. Aragon, R.A. Barrio, Turing patterns on a sphere, Phys. Rev. E 60 (1999) 4588–4592.

[55] H. Wendland, Fast evaluation of radial basis functions: methods based on partition of unity, in: Approximation Theory, X: Wavelets, Splines, and Applications, Vanderbilt University Press, Nashville, TN, 2002, pp. 473–483.

[56] H. Wendland, Scattered Data Approximation, Cambridge University Press, 2005.

[57] G.B. Wright, Radial basis function interpolation: numerical and analytical developments, PhD thesis, University of Colorado, Boulder, 2003.

[58] G.B. Wright, B. Fornberg, Stable computations with flat radial basis functions using vector-valued rational approximations, J. Comput. Phys. 331 (2017) 137–156.

[59] V.S. Zykov, A.S. Mikhailov, S.C. Müller, Controlling spiral waves in confined geometries by global feedback, Phys. Rev. Lett. 78 (1997) 3398.