

Article

Numerical Study on an RBF-FD Tangent Plane Based Method for Convection–Diffusion Equations on Anisotropic Evolving Surfaces

Nazakat Adil, Xufeng Xiao * and Xinlong Feng

College of Mathematics and System Sciences, Xinjiang University, Urumqi 830017, China; nazakatm@stu.xju.edu.cn (N.A.); fxlmath@xju.edu.cn (X.F.)

* Correspondence: xiaoxufeng111@sina.com

Abstract: In this paper, we present a fully Lagrangian method based on the radial basis function (RBF) finite difference (FD) method for solving convection–diffusion partial differential equations (PDEs) on evolving surfaces. Surface differential operators are discretized by the tangent plane approach using Gaussian RBFs augmented with two-dimensional (2D) polynomials. The main advantage of our method is the simplicity of calculating differentiation weights. Additionally, we couple the method with anisotropic RBFs (ARBFs) to obtain more accurate numerical solutions for the anisotropic growth of surfaces. In the ARBF interpolation, the Euclidean distance is replaced with a suitable metric that matches the anisotropic surface geometry. Therefore, it will lead to a good result on the aspects of stability and accuracy of the RBF-FD method for this type of problem. The performance of this method is shown for various convection–diffusion equations on evolving surfaces, which include the anisotropic growth of surfaces and growth coupled with the solutions of PDEs.

Keywords: convection–diffusion equation; Lagrangian; evolving surface; radial basis function–finite difference; anisotropic radial basis function



Citation: Adil, N.; Xiao, X.; Feng, X. Numerical Study on an RBF-FD Tangent Plane Based Method for Convection–Diffusion Equations on Anisotropic Evolving Surfaces. *Entropy* **2022**, *24*, 857. <https://doi.org/10.3390/e24070857>

Academic Editor: Eun-jin Kim

Received: 22 May 2022

Accepted: 16 June 2022

Published: 22 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Convection–diffusion equations on stationary or evolving surfaces are becoming increasingly popular since they arise in a wide variety of applications [1–4]. For PDEs on stationary surfaces, it is a common practice that the smooth surface is discretized as a triangulated surface, i.e., piecewise linear surface. The popular surface finite element methods [5,6] have been used in this context. A lot of work has been done to derive the finite element methods for solving PDEs on evolving surfaces [3,7,8]. In the finite element framework, the mesh evolving approach is used for the discretization of surface evolutions. However, regenerating a global mesh is expensive for large mesh deformations.

Recently, meshfree methods have become popular due to their flexibility of working on solving surface PDEs. The meshfree methods allow large deformations or topological changes of surfaces to be handled more easily than the mesh-based finite element methods. In the meshfree methods, surfaces are only discretized with scattered points. The RBF method is a high accurate numerical approach in the context of meshfree methods. The global RBF collocation method proposed in [9] provides a high order numerical scheme for solving stationary surface PDEs. The global RBF method has a highly computational cost because it leads to a dense discrete matrix. When it comes to PDEs on evolving surfaces, the dense matrix needs to be recalculated at each time layer. Therefore, the local RBF-FD methods provide a competitive numerical discretization for surface PDEs. The RBF-FD methods have been successfully developed to convection–diffusion and reaction–diffusion PDEs on domains [10,11] and surfaces [12–15]. The RBF-FD methods have also been applied to the evolving surface PDEs [16,17].

In this paper, the surface differential operators are approximated by using the tangent plane method (TPM) in the RBF-FD framework. The TPM was first proposed by Demanet [18]. The key idea of the method is that the surface differential operators are approximated on the tangent space. Recently, the TPM was extended to the context of the meshfree generalized finite difference method and applied to surface PDEs on stationary [19] and evolving surfaces [20]. In [21], the RBF-FD method is combined with TPM to calculate the differential weights for the surface Laplacian. In this paper, we focus on the application of the RBF-FD TPM method on solving evolving surface PDEs. The main advantage of this method comes from the simplicity of the discretization of surface operators.

Furthermore, we found that some anisotropic surface evolutions [3] may introduce the trouble of distortion on the nodes distribution. For example, surfaces evolve in ways that depend linearly on the underlying axes. In this case, the irregular clustering of nodes may lead to results in conditioning problems of the RBF interpolation. To improve the performance of the RBF-FD TPM, we couple the method with ARBF interpolation [22–24] to solve this kind of problem.

The rest of the paper is organized as follows. In Section 2, we give the formulation of a convection–diffusion equation on evolving surfaces. In Section 3, we firstly review the basic notation of ARBF interpolation. Then we show how to calculate the differentiation weights by using the ARBFs within the TPM framework. In Section 4, the PDE is discretized in time. In Section 5, we analyze the accuracy and eigenvalue stability of the approximation for the Laplacian on an ellipsoid using the proposed method. In Section 6, various numerical examples are performed to verify the effectiveness of the proposed method for different surface evolutions. Finally, we end with conclusions and some further issues.

2. Convection–Diffusion Equation on Evolving Surfaces

In this section, we consider the convection–diffusion equation on a time-dependent closed and smooth submanifold $\Gamma(t) \subset R^3$:

$$\frac{d}{dt}u + u\nabla_{\Gamma(t)} \cdot \mathbf{v} - k\Delta_{\Gamma(t)}u = f(t, \mathbf{x}), \quad \text{on } \Gamma(t), \quad t \in (0, T_e], \tag{1}$$

with an initial condition of $u(t, \mathbf{x}) = u_0(\mathbf{x})$. Here, $\frac{d}{dt}u$ is a material derivative, i.e.,

$$\frac{d}{dt}u = u_t + \mathbf{v} \cdot \nabla u,$$

\mathbf{v} is a velocity of the surface $\Gamma(t)$ and k is a diffusion coefficient.

Let \mathbf{n} denote the outward normal vector of the surface $\Gamma(t)$. We assume that the velocity \mathbf{v} contains both normal and tangential components. We denote by $\mathbf{v} = V\mathbf{n} + \mathbf{v}_S$ (with $\mathbf{v}_S \cdot \mathbf{n} = 0$) the velocity of material points on the surface. The tangential velocity \mathbf{v}_S transports u along $\Gamma(t)$.

For a given time t , we denote by \mathbf{P} the projection operator that projects a vector in R^3 at the point \mathbf{x} into the $T_{\mathbf{x}}\Gamma(t)$. Then \mathbf{P} can be written as

$$\mathbf{P} = I - \mathbf{nn}^T.$$

Thus, we can define the surface gradient and Laplacian on $\Gamma(t)$:

$$\nabla_{\Gamma(t)} := \mathbf{P}\nabla, \quad \Delta_{\Gamma(t)} := \nabla_{\Gamma(t)} \cdot \nabla_{\Gamma(t)}.$$

We represent the evolving surface $\Gamma(t)$ by a parameterization approach. The velocity \mathbf{v} determines the movement of nodes on $\Gamma(t)$ by the following form:

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{v}(t, \mathbf{x}(t)). \tag{2}$$

If (2) can be solved analytically, then we can define the parameterization of the $\Gamma(t)$ over the initial surface $\Gamma(0)$:

$$\Gamma(t) = \{\mathbf{x}(t) \mid \mathbf{x}(0) \in \Gamma(0)\}.$$

If not, we use the second order method [20] to numerically solve the ODE system (2):

$$\mathbf{x}^{m+1} = \mathbf{x}^m + \frac{\Delta t}{2}(3\mathbf{v}^m - \mathbf{v}^{m-1}). \tag{3}$$

3. ARBF-FD TPM for the Convection–Diffusion Equation on Evolving Surfaces

3.1. Anisotropic Radial Basis Function Interpolation

In this paper, we use a generalization form of RBF: ARBF. The purpose of using this basis function is that we want to benefit from its ability to handle the anisotropic growth of surfaces.

ARBF can be better understood from the two aspects: one is the RBF interpolant on transformed data, and the other is that using a new distance in the RBF. Let $X = \{\mathbf{x}_i\}_{i=1}^N \subset R^d$ denote an interpolated nodes set, where d is the dimension of space. We change the X into another data set \tilde{X} through a linear transformation or a nonsingular $d \times d$ matrix M :

$$\tilde{X} = \{\tilde{\mathbf{x}}_i \mid \tilde{\mathbf{x}}_i = M\mathbf{x}_i, i = 1, 2, \dots, N\}.$$

Let $T = M^T M$, then T is symmetric positive definite. Results from [24] show that the RBF interpolation on \tilde{X} is equivalent to the ARBF interpolation on X . The corresponding new norm is defined as

$$\|\mathbf{x}\|_T = \sqrt{\mathbf{x}^T T \mathbf{x}}.$$

Now we can define the ARBF interpolant of a target function $s(\mathbf{x})$:

$$I_\phi s(\mathbf{x}) = \sum_{i=1}^N \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|_T) + \sum_{k=1}^{m_p} \beta_k p_k(M\mathbf{x}). \tag{4}$$

It should be noted that the constraints on multivariate polynomials in (4) are

$$\sum_{k=1}^N \alpha_k p_j(M\mathbf{x}_k) = 0, \quad j = 1, 2, 3, \dots, m_p,$$

here, $m_p = \binom{l+d}{l}$ and l is the largest degree of the polynomials. The coefficients $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T, \beta = (\beta_1, \beta_2, \dots, \beta_{m_p})^T$ in (4) are obtained by solving the following linear system:

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} s_X \\ 0 \end{pmatrix},$$

where $A = \{a_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|_T)\}, s_X = (s(\mathbf{x}_1), s(\mathbf{x}_2), \dots, s(\mathbf{x}_N))^T$, and

$$P = \begin{pmatrix} p_1(M\mathbf{x}_1) & p_2(M\mathbf{x}_1) & \dots & p_{m_p}(M\mathbf{x}_1) \\ p_1(M\mathbf{x}_2) & p_2(M\mathbf{x}_2) & \dots & p_{m_p}(M\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ p_1(M\mathbf{x}_N) & p_2(M\mathbf{x}_N) & \dots & p_{m_p}(M\mathbf{x}_N) \end{pmatrix}.$$

In numerical computations, the behaviors of RBF interpolations are usually related to the separation distance and fill distance:

$$q_X = \frac{1}{2} \min_{1 \leq i < j \leq N} \|\mathbf{x}_i - \mathbf{x}_j\|_2, \quad h_X = \sup_{\mathbf{x} \in \Omega} \min_{1 \leq i \leq N} \|\mathbf{x} - \mathbf{x}_i\|_2, \tag{5}$$

where $\Omega \subset R^d$ is a computational domain. The condition number of the RBF interpolation matrix increases with the decrease in the q_X . The fill distance h_X determines the accuracy of the interpolation. The two quantities of a near-uniform nodes set are close to each other.

When M (or T) is an identity matrix, the above ARBF interpolation turns into the RBF interpolation. If the surface does not have an anisotropic structure, the Euclidean distance is sufficient to give good results for the interpolation. For surfaces with a global anisotropic structure, such as ellipsoids, we need to select a matrix M that matches the anisotropic surface geometry. In other words, our goal is to seek a suitable matrix M such that the h_X and q_X of the mapped data \tilde{X} are closer together. In this paper, the basis function is chosen as the anisotropic version of the Gaussian RBF:

$$\phi(\|\mathbf{x} - \mathbf{x}_i\|_T) = \exp(-\epsilon^2\|\mathbf{x} - \mathbf{x}_i\|_T^2),$$

where ϵ is a free shape parameter.

We know that the shape parameter also has an effect on the RBF interpolation. For a fixed node set, it leads to ill-conditioned problems as the shape parameter ϵ becomes small. In this case, RBF becomes flatter, and the local interpolation matrix causes an ill-conditioned linear system. The RBF-QR method [25,26] has been proposed to circumvent this problem. However, our study focuses on the influence of nodes distribution on the RBF interpolation. Therefore, instead of using the RBF-QR method, a suitable shape parameter is selected for each numerical test.

3.2. Differentiation Weights Based on TPM

In this section, we illustrate how to calculate the surface gradient and Laplacian using the ARBF-FD TPM. Without loss of generality, let $S_1 = \{\mathbf{x}_j\}_{j=1}^n \subset X$ denote a stencil, where $\mathbf{x}_j, j = 1, 2, \dots, n$ are the nearest neighbors of the center \mathbf{x}_1 . Let L be a differential operator. The approximation of the $Lu(\mathbf{x})$ at \mathbf{x}_1 can be computed by the following linear combination:

$$Lu(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_1} \approx L_X^1 u(\mathbf{x}) = \sum_{j=1}^n \omega_j^L u(\mathbf{x}_j). \tag{6}$$

Thus the weights ω_j^L are found by solving the system,

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \omega^L \\ \eta \end{pmatrix} = \begin{pmatrix} L\Phi(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_1} \\ L\mathbf{p}(M\mathbf{x})|_{\mathbf{x}=\mathbf{x}_1} \end{pmatrix}, \tag{7}$$

where $\Phi(\mathbf{x}) = (\phi(\|\mathbf{x} - \mathbf{x}_1\|_T), \phi(\|\mathbf{x} - \mathbf{x}_2\|_T), \dots, \phi(\|\mathbf{x} - \mathbf{x}_n\|_T))^T$ and $\mathbf{p}(M\mathbf{x}) = (p_1(M\mathbf{x}), p_2(M\mathbf{x}), \dots, p_{m_p}(M\mathbf{x}))^T$.

A straightforward way is that calculating the weights directly by (7) using a closed-form formulation of surface operators, as proposed in [27,28] (the method is called the RBF-FD direct method hereafter). In this paper, the process of calculating the weights is implemented within the TPM framework. The advantage of using this method lies in the fact that the proposed method does not require a closed-form formulation of surface operators. Thus, it is much easier to implement.

We describe this method in detail below. Let \bar{u} denote the normal extension of u , i.e., $\mathbf{n} \cdot \nabla \bar{u} = 0$. The key idea of TPM is that the problem of computing the surface operator of u is reduced to that of computing the gradient (in R^3) of the \bar{u} . This is due to the fact that

$$\nabla_{\Gamma} u = \nabla \bar{u} - (\mathbf{n} \cdot \nabla \bar{u})\mathbf{n} = \nabla \bar{u}.$$

Notice again that the gradient of \bar{u} in the local orthonormal coordinate system $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{n}\}$ can be written as

$$\nabla \bar{u} = (\mathbf{t}_1 \cdot \nabla \bar{u})\mathbf{t}_1 + (\mathbf{t}_2 \cdot \nabla \bar{u})\mathbf{t}_2 + (\mathbf{n} \cdot \nabla \bar{u})\mathbf{n} = (\mathbf{t}_1 \cdot \nabla \bar{u})\mathbf{t}_1 + (\mathbf{t}_2 \cdot \nabla \bar{u})\mathbf{t}_2.$$

Thus, we only need to compute the $\mathbf{t}_1 \cdot \nabla \bar{u}$ and $\mathbf{t}_2 \cdot \nabla \bar{u}$. Here, \mathbf{t}_1 and \mathbf{t}_2 are the orthogonal basis of the tangent plane $T_{\mathbf{x}_1} \Gamma$. Further, the weights of the surface gradient are computed by

$$\omega_j^{\nabla \Gamma} = \mathbf{t}_1 \omega_j^{\nabla \mathbf{t}_1} + \mathbf{t}_2 \omega_j^{\nabla \mathbf{t}_2}, \quad j = 1, 2, \dots, n. \tag{8}$$

The above differentiation weights $\omega_j^{\nabla \mathbf{t}_1}$ and $\omega_j^{\nabla \mathbf{t}_2}$ correspond to the directional derivatives of a function in the directions of \mathbf{t}_1 and \mathbf{t}_2 , respectively.

For computational simplicity, we project the stencil onto the tangent plane at the center \mathbf{x}_1 . Then we use a rotation $R = (\mathbf{t}_1, \mathbf{t}_2, \mathbf{n})^T$ to transform the projected nodes so that they are now located around a plane that is parallel to the plane $z = 0$. Through the coordinate transformation

$$\mathbf{x}_R = R \mathbf{x}, \tag{9}$$

we can obtain

$$\frac{\partial \bar{u}}{\partial x_R} = \mathbf{t}_1 \cdot \nabla \bar{u}, \quad \frac{\partial \bar{u}}{\partial y_R} = \mathbf{t}_2 \cdot \nabla \bar{u},$$

where x_R and y_R are the first and second components of \mathbf{x}_R . Then we ignore the z -coordinates of \mathbf{x}_R , construct 2D RBFs on the rotated stencil and compute the weights corresponding to $\frac{\partial \bar{u}}{\partial x_R}$ and $\frac{\partial \bar{u}}{\partial y_R}$.

To sum up the above, the calculation of the differentiation weights of the surface gradient requires two steps. The first step is to calculate the 2D weights on the rotated stencil by solving (7). The second step is to transform the weights to the original stencil by using (8).

For the discretization of the surface Laplacian, the RBF-FD TPM only needs the normal vectors. However, the RBF-FD direct method requires reconstructing the curvatures of surfaces. In addition, the RBF-FD TPM uses fewer terms in the polynomial basis in comparison to the RBF-FD direct method. This is because the 2D basis functions are used in the RBF-FD TPM.

We found that the transport scheme on sphere [29] is also essentially a tangent plane method, which uses Householder reflection to implement the process of rotation. However, the Householder reflection can only handle the surface of a sphere. For general surfaces, it is necessary to find \mathbf{t}_1 and \mathbf{t}_2 and construct the matrix R . If Γ is a parametric surface defined by $\mathbf{x} = \mathbf{x}(\psi, \theta)$, then \mathbf{t}_1 and \mathbf{t}_2 can be obtained directly by normalizing \mathbf{x}_ψ and \mathbf{x}_θ .

For the calculation of the surface Laplacian, we use the rotational invariance of the Laplacian or the proof in [19]. Therefore, we can obtain

$$\omega_j^{\Delta \Gamma} = \omega_j^{\Delta T}, \quad j = 1, 2, \dots, n,$$

where Δ_T is the 2D Laplacian on $T_{\mathbf{x}_1} \Gamma$.

4. Discretization in Time

The Lagrangian framework is considered for the numerical discretization of (1). Let time steps $t^m = m\Delta t$, $m = 0, 1, \dots, M_e$, $\Delta t = T_e/M_e$. At each time layer, we update the nodes \mathbf{x}^m through (3). Then we calculate the required differentiation weights for each center node by repeating the procedure presented in Section 3.2. Further, the differentiation matrix D^m is constructed by the differentiation weights. Differentiation weights can be computed in parallel, as the procedure for each point is independent of that for all other points.

The material derivative of u can be simply approximated as

$$\frac{d}{dt} u|_{(t^m, \mathbf{x}^m)} \approx \frac{u(t^m, \mathbf{x}^m) - u(t^{m-1}, \mathbf{x}^{m-1})}{\Delta t}.$$

We use the Crank–Nicolson scheme for time discretization. The fully discrete form is written as

$$(I - \frac{\Delta t}{2}D^m)U^m = (I + \frac{\Delta t}{2}D^{m-1})U^{m-1} + \frac{\Delta t}{2}(F^m + F^{m-1}), \quad m = 1, 2, \dots, M_e, \quad (10)$$

where U_i^m is the approximated value of $u(t^m, \mathbf{x}_i(t^m))$ and $F_i^m = f(t^m, \mathbf{x}_i(t^m))$.

5. Differentiation Accuracy

For the computation of PDEs on stationary or moving surfaces, calculating accurate differentiation weights turned out to be essential in the RBF-FD context. In this section, we show the errors and convergence orders of the proposed method for the approximation of the surface Laplacian. This test is conducted over the ellipsoid given by

$$\Gamma = \{\mathbf{x} \mid \frac{x^2}{9} + \frac{y^2}{9} + z^2 = 1\}. \quad (11)$$

We consider the function

$$u(x, y, z) = \sin(x) \sin(y) \sin(z) + \cos(x) \cos(y) \cos(z). \quad (12)$$

The error is measured by the following form:

$$\text{Error}_{\Delta\Gamma} = \max_{1 \leq i \leq N} |\Delta\Gamma u(\mathbf{x}_i) - L_{\Delta\Gamma, X}^i u(\mathbf{x})|,$$

where $L_{\Delta\Gamma, X}^i$ represents the approximation given by (6) for the surface Laplacian. For the surface, we map ‘minimal energy’ (ME) nodes on the unit sphere to the ellipsoid (11). This node layout is illustrated in Figure 1.

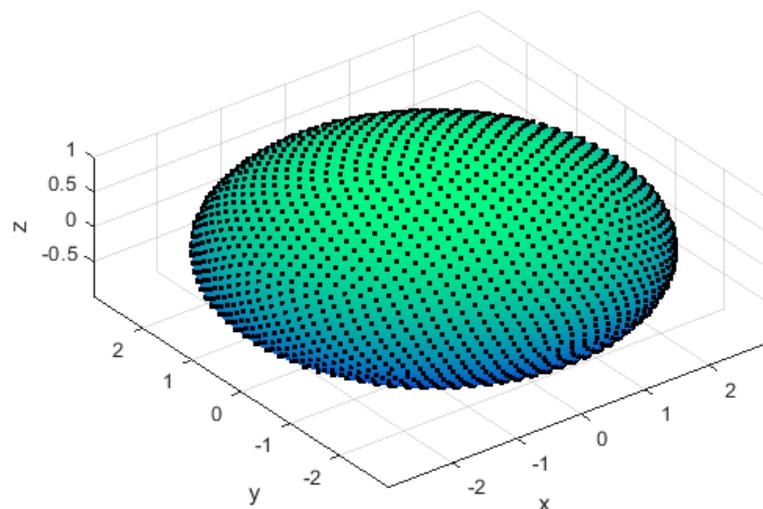


Figure 1. Node layout on the ellipsoid.

5.1. On the Choice of the Matrix M

It is worth pointing out that the using of a suitable M matrix will improve the accuracy and well-conditioning of the interpolation problem (4). Therefore, it leads to better results for the computation of differentiation weights. Note again that the nodes on the ellipsoid are obtained by mapping the ME nodes on the unit sphere. An additional point to note is that the RBFs used in TPM are the 2D basis functions defined on rotated stencils. Therefore, for each stencil, we define a matrix M_2 :

$$M_2 = R_s M_1 R_e^{-1}, \quad (13)$$

where $M_1 = \text{diag}\left(\frac{1}{3}, \frac{1}{3}, 1\right)$, $R_s = (\mathbf{t}_1^s, \mathbf{t}_2^s, \mathbf{n}^s)^T$, $R_e = (\mathbf{t}_1^e, \mathbf{t}_2^e, \mathbf{n}^e)^T$. Here, R_s and R_e consist of the tangent vectors and normal vectors of the unit sphere and the ellipsoid at the center of the stencil, respectively. The role of M_2 is that of the nodes mapped to the unit sphere before rotating through (9) and then performing the rotation step by R_s .

In the implementation, M_2 is only used to construct a 2×2 matrix M by extracting the first two rows and columns of M_2 . We then define the 2D ARBFs using the matrix M and compute differentiation weights using (7). Additionally, we construct an anisotropic type of stencils. That is, the distance between the center and nearest neighbors is measured using the $\|\cdot\|_{T_1}$, where $T_1 = M_1^T M_1$.

5.2. Results on the Ellipsoid

For the choice of parameters, we first choose the degree of polynomials with $l = 1, 2$ and 3. Then, we have set $n = 2m_p + 1$, where $m_p = \binom{l+3}{l}$ and fix $\epsilon = 3$. To explain that the ARBFs would work better than RBFs, we compare the accuracy of the RBF-FD TPM and ARBF-FD TPM. Several observations can be made in Figure 2.

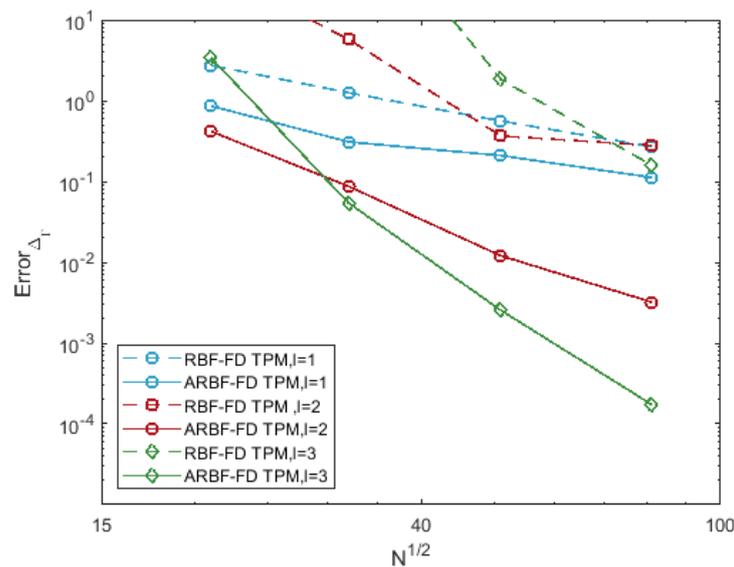


Figure 2. Errors for surface Laplacian of function (12) on the ellipsoid.

1. The errors of the RBF-FD TPM are relatively large. The reason for this is that the irregularity of nodes leads to poor RBF interpolation. The irregularity of nodes is reflected by the difference between the fill distance and separation distance. For convenience, the fill distance in (5) is approximated by

$$\bar{h}_X = \frac{1}{2} \max_{1 \leq j \leq N} \min_{1 \leq i \leq N, i \neq j} \|\mathbf{x}_j - \mathbf{x}_i\|_2.$$

For 6561 nodes on the ellipsoid, $\bar{h}_X = 0.0697$ and $q_X = 0.0214$, the difference between the two quantities is relatively large;

2. The ARBF-FD TPM has smaller errors and faster convergence rates than those of RBF-FD TPM. The M_1 -matrix maps the nodes on the ellipsoid to the near-uniform ME nodes on the unit sphere. The corresponding two distances of the transformed nodes set are $\bar{h}_X = 0.0235$ and $q_X = 0.0210$. We obtained better results by introducing a new distance to reduce the difference between these two quantities. The convergence orders are 1.43, 3.67 and 6.19 for different l , respectively.

In Figure 3, we give the eigenvalues of the RBF-FD TPM and ARBF-FD TPM corresponding to the surface Laplacian differentiation matrix on the ellipsoid, where $l = 2$, $n = 21$, $\epsilon = 3$ and $N = 6561$. Due to the irregular sampling of the nodes on the ellipsoid,

the RBF-FD TPM method has large positive eigenvalues. In the ARBF-FD TPM method, we improved this problem using a new metric. Therefore, the eigenvalues of the ARBF-FD TPM are all in the negative half-plane.

For this test, there are two other ways to improve the results of the RBF-FD TPM. The first one is to use the method proposed in [12], fixing the condition number of local interpolation matrix and choosing stencil-dependent shape parameters. Compared with the ARBF method, choosing different shape parameters in each stencil can be regarded as the isotropic scaling of each stencil. For the evolving surface problems, this strategy increases the computational cost. The second one is to sample near-uniform nodes on the ellipsoid. This can improve the stability of eigenvalues for the ellipsoid. However, if we consider the anisotropic growth of the unit sphere, regenerating the near-uniform nodes at each time layer would be time consuming.

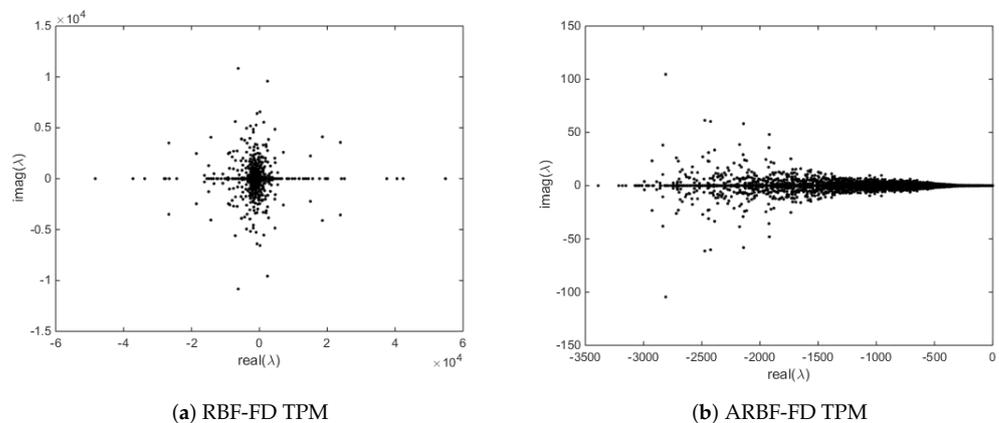


Figure 3. Eigenvalues of the two methods for the surface Laplacian on the ellipsoid.

Summarizing the above discussion, the use of the ARBFs overcomes the ill-conditioned interpolation problem caused by irregular sampling of the nodes without increase of the computation cost. This means that the ARBF-FD TPM method is effective for solving the Poisson or diffusion equations on the stationary ellipsoid. In this paper, we mainly focus on the application of the ARBF-FD TPM for the anisotropic growth of surfaces.

6. Numerical Experiments

6.1. Normal and Tangential Motion on the Unit Sphere

First, we test the accuracy of the proposed method for the Equation (1) on surfaces without anisotropy. In this case, we suggest using isotropic RBFs (i.e., M is the identity) for the numerical simulations. For this, we consider two examples: an expanding sphere and a rotating sphere. The velocity of the expanding sphere has only a normal component, while the velocity of rotating sphere is purely tangential.

6.1.1. Expanding Sphere

For the first example, we consider the expanding sphere,

$$\Gamma(t) = \{\mathbf{x}(t) \mid x^2 + y^2 + z^2 = (1 + 0.5t)^2\}.$$

Then the velocity is $\mathbf{v} = 0.5\mathbf{n}$. In Equation (1), the diffusion coefficient is chosen to be $k = 1$. We assume that the exact solution is given by

$$u(t, \mathbf{x}) = e^{-6t}xy,$$

The time interval of the simulation is set to be $(0, 1]$.

At the initial time, we use ME nodes on the unit sphere. The values of the parameters are given in Table 1. For the two cases listed in the table, we choose $\Delta t = \bar{h}_X/5$ and

$\Delta t = \bar{h}_X^2/30$, respectively. The relative l_2 error in the numerical solution is measured at $T_e = 1$ by

$$l_2 \text{ error} = \left(\frac{\sum_{i=1}^N (U_i^{M_e} - u(T_e, \mathbf{x}_i(T_e)))^2}{\sum_{i=1}^N u(T_e, \mathbf{x}_i(T_e))^2} \right)^{1/2}.$$

Table 1. The values of the parameters.

Case	l	n	ϵ
①	1	9	1
②	2	21	2

The errors of the RBF-FD TPM for the test are given in Figure 4. Second- and fourth-order convergence are obtained for the cases of ① and ②, respectively. It can be found that the nodes remain near-uniformly distributed as the expanding of the sphere. That is, h_X and q_X are relatively close to each other during the surface evolution. For this case, we can see from Figure 4 that higher convergence orders are obtained by the using of the Euclidean distance in RBFs.

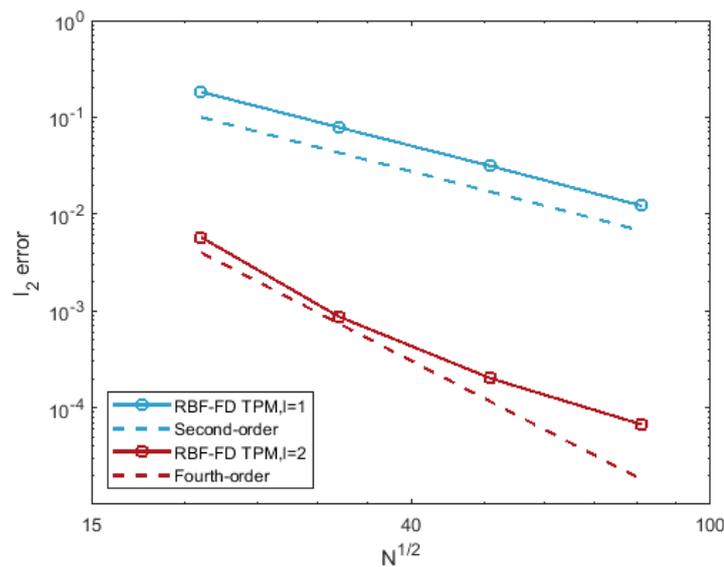


Figure 4. Relative l_2 error of the RBF-FD TPM for the test of expanding sphere at $T_e = 1$.

6.1.2. Rotating Sphere

In this example, we choose a velocity of $\mathbf{v} = (y, -x, 0)^T$, which is a divergence free tangential velocity. Therefore, Equation (1) can be simplified as

$$u_t + \mathbf{v} \cdot \nabla_{\Gamma} u - k \Delta_{\Gamma} u = f.$$

This equation can also be viewed as a convection–diffusion equation on the stationary unit sphere and can be numerically solved in the Eulerian framework. For pure convection or convection-dominated cases, the Eulerian RBF-FD method usually has spurious oscillations. This needs to be rectified with artificial hyperviscosity [15]. The fully Lagrangian method used in this paper does not require the artificial hyperviscosity.

For this example, let $f = 0$ and $k = 10^{-5}$. The initial condition is taken as a cosine bell:

$$u_0(\mathbf{x}) = \begin{cases} \frac{1}{c} \left(1 + \cos\left(\frac{\pi \arccos x}{1/3}\right) \right), & \arccos x < \frac{1}{3}, \\ 0, & \arccos x \geq \frac{1}{3}, \end{cases} \quad (14)$$

where $c = 2$. We simulate the numerical solutions in Figure 5 at times $t = 0, 2\pi$ and 20π . Here we using the parameters of $N = 6561, \Delta t = 0.005$ and ① given in Table 1. There are no spurious oscillations in the numerical solution because the method does not require the discretization of the surface gradient. Numerical results demonstrated the validity of the RBF-FD TPM for the convection-dominated convection–diffusion equations.

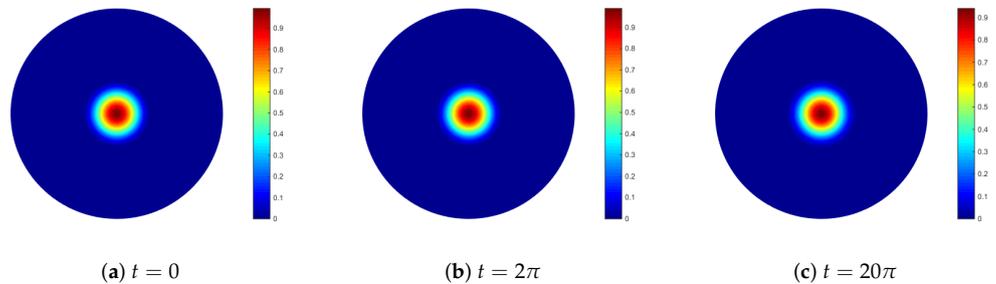


Figure 5. Numerical solution for the test of rotating sphere.

6.2. Evolving Ellipsoid

In this example, we wanted to present the effectiveness of the ARBF-FD TPM method for the anisotropic evolution of surfaces. We consider Equation (1) on an evolving ellipsoid:

$$\Gamma(t) = \{ \mathbf{x}(t) \mid \frac{x^2}{a(t)^2} + \frac{y^2}{b(t)^2} + \frac{z^2}{c(t)^2} = 1 \},$$

where $a(t) = 1 + 4t, b(t) = 1 + 2t$ and $c(t) = 1 + t$. We assume that the nodes move in the following way:

$$x(t) = a(t)x(0), \quad y(t) = b(t)y(0) \quad \text{and} \quad z(t) = c(t)z(0).$$

Then the velocity of the surface is computed by $\mathbf{v}(t, \mathbf{x}) = \left(\frac{a'(t)}{a(t)}x, \frac{b'(t)}{b(t)}y, \frac{c'(t)}{c(t)}z \right)^T$.

We assume that the exact solution is given by

$$u(t, \mathbf{x}) = e^{-t} \sin(x),$$

and the diffusion coefficient $k = 1$. The final time is taken to be $T_e = 1$. The final surface is an ellipsoid which has different growth rates in the x, y and z directions.

The velocity \mathbf{v} leads to an irregular clustering of nodes with the increase in time. In Figure 6, we show how the distribution of nodes changes with the increase in time. \bar{h}_X and q_X corresponding to the nodes set in each subgraph in Figure 6 are given in Table 2. With the anisotropic growth of the surface, these two quantities display differences between themselves.

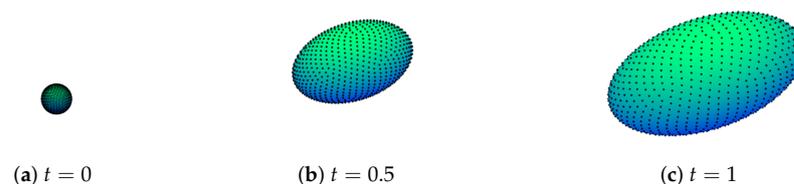


Figure 6. Distribution of nodes on the evolving ellipsoid, where $N = 1024$.

Table 2. The fill and separation distance of the nodes on the ellipsoid, where $N = 1024$.

Distance	$t = 0$	$t = 0.5$	$t = 1$
\bar{h}_X	5.9378×10^{-2}	1.2999×10^{-1}	2.0258×10^{-1}
q_X	5.3421×10^{-2}	8.5483×10^{-2}	1.1430×10^{-1}

The new metric in ARBFs is constructed in the same way as described in Section 5.1. At a given time t , the M_1 -matrix in (13) is chosen by

$$M_1(t) = \text{diag}\left(\frac{1}{a(t)}, \frac{1}{b(t)}, \frac{1}{c(t)}\right).$$

The matrix M_1 maps the nodes to the initial surface (the unit sphere) at each time layer. In more detail, the h_X and q_X of the mapped data are closer together, which will lead to more stable and accurate results. If we consider a more general case of the surface evolution,

$$\mathbf{x}(t) = B(t)\mathbf{x}(0),$$

then we can choose $M_1(t) = B(t)^{-1}$ for this case.

Use the same parameters as the test of the expanding sphere, given in Table 1. For ① and ②, the time step is chosen as $\Delta t = \bar{h}_X/5$ and $\Delta t = \bar{h}_X^2$, respectively. In Figure 7, we give the errors of the RBF-FD TPM and ARBF-FD TPM. For the anisotropic surface evolution, the Euclidean distance is not appropriate. As a result, the RBF-FD TPM method shows eigenvalue instability. However, the proposed ARBF-FD TPM method does not suffer from the irregular clustering of nodes. This is because we map the nodes to the initial surface at each time layer. Further, the values of \bar{h}_X and q_X of the transformed data stay the same as the initial moment. The advantage of the method is that it not only overcomes the instability caused by the irregularity of nodes, but also avoids the loss of accuracy caused by the growth of \bar{h}_X .

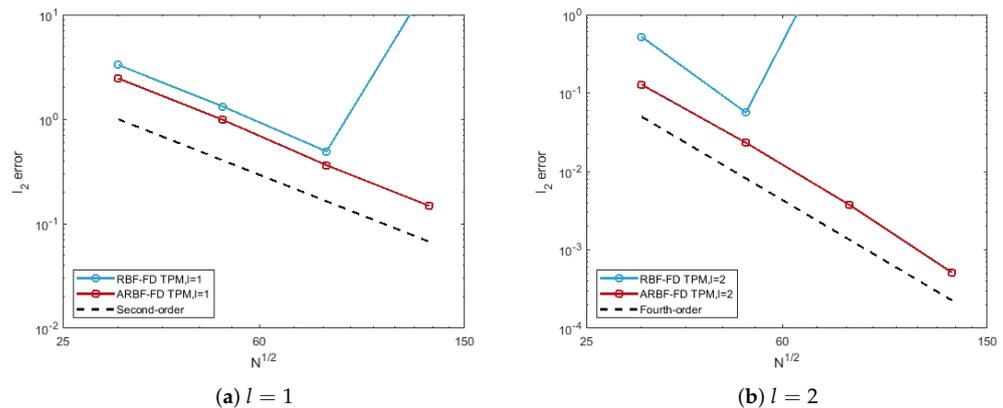


Figure 7. Relative l_2 errors of the RBF-FD TPM and ARBF-FD TPM for the test of evolving ellipsoid at $T_e = 1$.

6.3. Evolving Torus

We consider the anisotropic evolution on the torus:

$$\Gamma(t) = \{\mathbf{x} \mid (x^2 + y^2 + (1 + 2t)^2 z^2 + 0.84)^2 - 4(x^2 + y^2) = 0\}. \tag{15}$$

Thus, the position of the nodes is given by $\mathbf{x}(t) = \left(x(0), y(0), \frac{z(0)}{1+2t}\right)^T$. This surface becomes very thin as it shrinks in the z -direction. It can be seen from Table 3 that the separation distance decreases with time, which leads to an ill-conditioned RBF interpolation. Following the idea of the previous example, the M_1 -matrix is chosen by $M_1(t) = \text{diag}(1, 1, 1 + 2t)$. The matrix transforms the nodes at time t to the initial surface. Then, we replace R_s and R_e in (13) with the rotation matrix corresponding to the torus at the initial time and time t .

Table 3. The fill and separation distance of the nodes on the torus, where $N = 6230$.

Distance	$t = 0$	$t = 1$	$t = 2$
\bar{h}_X	2.8538×10^{-2}	2.7508×10^{-2}	2.7501×10^{-2}
q_X	2.1260×10^{-2}	7.8858×10^{-3}	4.7380×10^{-3}

We simulate the diffusion on the evolving torus with $k = 10^{-1}$. Initial condition is chosen as the cosine bell (14) with $c = 800$. Figure 8 shows the evolution of the surface and the numerical solutions at times $t = 0, t = 1$ and $t = 2$. We use 6230 near-uniform nodes on the initial surface and ② given in Table 1. For the test, let $\Delta t = 0.001$. This surface evolution is anisotropic shrinking, as reflected by the fact that q_X decreases much faster than \bar{h}_X . The mapped nodes (i.e., nodes at $t = 0$) present increased q_X , with consequent reduction of the condition number of the RBF interpolation matrix. This gives a better quality of the ARBF interpolant. Thus, the ARBF-FD TPM method accurately simulates the problem, even when the torus becomes very thin.

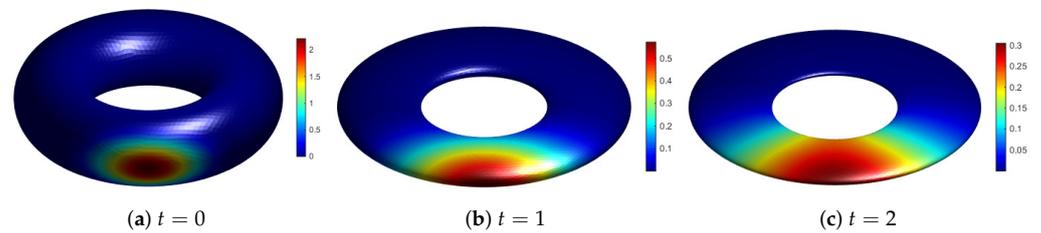


Figure 8. Numerical solution on the evolving torus.

6.4. Solid Tumor Growth

In more general evolutions of surfaces, the velocity is often coupled with the solution of the PDE on the surfaces, such as the model of solid tumor growth [3,4,16,17]. This model is mathematically described by the following reaction–diffusion system [16]:

$$\begin{aligned} \frac{d}{dt}u + u\nabla_{\Gamma(t)} \cdot \mathbf{v} - \Delta_{\Gamma(t)}u &= f_1(u, w), \\ \frac{d}{dt}w + w\nabla_{\Gamma(t)} \cdot \mathbf{v} - d_c\Delta_{\Gamma(t)}w &= f_2(u, w), \end{aligned} \tag{16}$$

where

$$f_1(u, w) = \gamma(a - u + u^2w), \quad \text{and} \quad f_2(u, w) = \gamma(b - u^2w).$$

For the system, we choose $d_c = 10, \gamma = 200, a = 0.1$ and $b = 0.9$. The velocity is chosen to be

$$\mathbf{v} = (-0.01\kappa + 0.4u)\mathbf{n} + \mathbf{v}_a,$$

where $\kappa = \nabla_{\Gamma} \cdot \mathbf{n}$ denotes the mean curvature, and \mathbf{v}_a is a given anisotropic velocity. We observed in the previous articles that the model does not have \mathbf{v}_a . Note that the velocity \mathbf{v} is no longer purely in the normal direction, but includes a tangential component. We compute the surface divergence of the velocity by

$$\nabla_{\Gamma(t)} \cdot \mathbf{v} = (-0.01\kappa + 0.4u)\kappa + \nabla_{\Gamma(t)} \cdot \mathbf{v}_a.$$

For the test, use $N = 6561$ nodes on the unit sphere and $N = 6230$ nodes on the torus $\Gamma(0)$ given by (15) as initial data. The initial conditions are obtained from the simulation of the model (16) in the stationary case (i.e., $\mathbf{v} = 0$) for these two surfaces at $t = 1.7$ and $t = 1$, respectively. In the scheme (10), the nonlinear source term is computed by the following form

$$\frac{1}{2} \left(3f_i(U^{m-1}, W^{m-1}) - f_i(U^{m-2}, W^{m-2}) \right), \quad i = 1, 2.$$

We choose $\mathbf{v}_a = \left(\frac{4}{1+4t}x, \frac{2}{1+2t}y, \frac{1}{1+t}z \right)^T$ for the evolution of the sphere and $\mathbf{v}_a = (0, 0, 0)$ for the torus, respectively. For the test on the sphere, we construct ARBFs in the way illustrated in Section 6.2. For the case on the torus, we use isotropic RBFs.

Note that at each time layer, the normal vector is required for numerical computation. Here, we compute a weighted normal, which is an average (corresponding to the area of each triangle) of all the triangle normals belonging to triangles with the given nodes.

Let $\Delta t = 0.001$. Here, we use the cases of ② and ① (given in Table 1) for the tests on the sphere and the torus, respectively. Numerical approximations of u on the two surfaces are given in Figures 9 and 10. It can be found that the surface expands outwards in regions where the solution u is large. In particular, solid tumor growth on the sphere becomes a bumpy ellipsoid. This is because it contains an artificial anisotropic velocity.

The example demonstrates that our method is effective for solving PDEs on moving point clouds. For this, we only need to calculate the normal vector through the triangular mesh. Although, the velocity \mathbf{v} contains the curvature, we replace this term with the relation $\kappa \mathbf{n} = -\Delta_{\Gamma} \mathbf{x}$. Notice that the velocity also leads to a less uniform distribution of nodes, but the changes of \bar{h}_X and q_X are not significant during the time interval we tested. Some adaptive mesh refinement strategy may be needed if we want to simulate long time evolution.

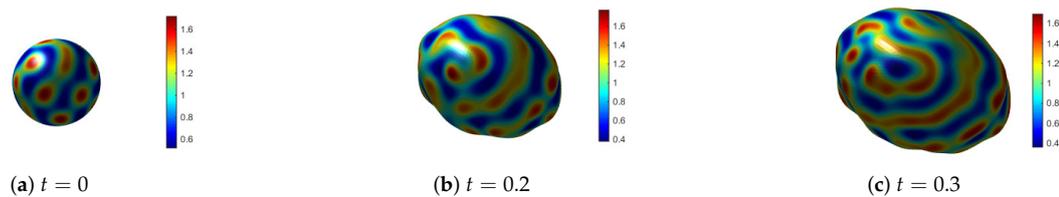


Figure 9. Numerical solution for the solid tumor growth on the sphere.

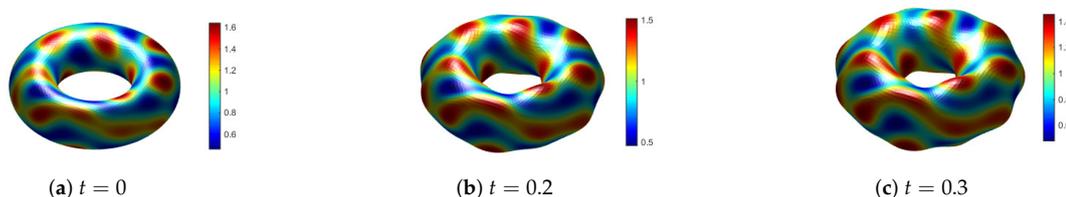


Figure 10. Numerical solution for the solid tumor growth on the torus.

7. Conclusions

In this work, we developed an RBF-FD method that can be employed for practical applications of PDEs on evolving surfaces. This proposed method is based on the tangent plane approach for calculating differentiation weights. The most attractive advantage of the method is that it transforms the problem into computing weights in 2D Euclidean space.

To make the RBF-FD TPM more efficient for anisotropic surface evolutions, we propose the new ARBF-FD TPM. The proposed method gives a significant improvement in numerical experiments. To construct the ARBFs, we define a linear mapping such that the nodes at each time layer are mapped to the initial surface. This approach gives advantages because it is possible to reduce the impact on the irregular clustering of nodes during the anisotropic surface evolution. An extension of the ARBF-FD TPM would be to adapt the method to handle anisotropic diffusion on surfaces. Our method can obtain higher convergence order by increasing the stencil size and degree of polynomials. Furthermore, the proposed ARBF-FD TPM can be applied directly to PDEs on evolving surfaces represented by triangular meshes or point clouds. Therefore, the method is potentially useful for computing PDEs on evolving surfaces.

A limitation of the ARBF-FD TPM is that the current results may be only applicable to the surface evolution in ways that depend linearly on the underlying axes. For evolving

surface problems which are stretched in nonlinear ways, it may be necessary to find a nonlinear mapping. This approach requires defining a more complex basis function than ARBF. Moreover, we will extend the method to handle complex surfaces with large deformations, such as for the topological change of surfaces.

Author Contributions: Conceptualization, X.F. and X.X.; methodology, N.A.; software, N.A.; validation, X.F., X.X.; formal analysis, X.X. and N.A.; investigation, X.X. and N.A.; resources, X.F.; data curation, N.A.; writing—original draft preparation, N.A.; writing—review and editing, X.F., X.X. and N.A.; visualization, N.A.; supervision, X.F.; project administration, X.F. and X.X.; funding acquisition, X.F. and N.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partly supported by the Postgraduate Research and Innovation Program of Xinjiang Province, China (No. XJ2019G011) and Excellent Doctor Innovation Program of Xinjiang University, China (No. XJUBSCX-201930).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bayona, V.; Flyer, N.; Lucas, G.M.; Baumgaertner, A.J. A 3-D RBF-FD solver for modeling the atmospheric global electric circuit with topography (GEC-RBFFD v1.0). *Geosci. Model. Dev.* **2015**, *8*, 3007–3020. [\[CrossRef\]](#)
- Chaplain, M.; Ganesh, M.; Graham, G. Spatio-temporal pattern formation on spherical surfaces: Numerical simulation and application to solid tumour growth. *J. Math. Biol.* **2001**, *42*, 387–423. [\[CrossRef\]](#) [\[PubMed\]](#)
- Barreira, R.; Elliott, C.M.; Madzvamuse, A. The surface finite element method for pattern formation on evolving biological surfaces. *J. Math. Biol.* **2011**, *63*, 1095–1119. [\[CrossRef\]](#) [\[PubMed\]](#)
- Madzvamuse, A.; Barreira, R. Exhibiting cross-diffusion-induced patterns for reaction-diffusion systems on evolving domains and surfaces. *Phys. Rev. E* **2014**, *90*, 043307. [\[CrossRef\]](#) [\[PubMed\]](#)
- Dziuk, G.; Elliott, C.M. Surface finite elements for parabolic equations. *J. Comput. Math.* **2007**, *25*, 385–407.
- Dziuk, G.; Elliott, C.M. Finite element methods for surface PDEs. *Acta. Numer.* **2013**, *22*, 289–396. [\[CrossRef\]](#)
- Dziuk, G.; Elliott, C.M. Finite elements on evolving surfaces. *IMA J. Numer. Anal.* **2007**, *27*, 262–292. [\[CrossRef\]](#)
- Eilks, C.; Elliott, C.M. Numerical simulation of dealloying by surface dissolution via the evolving surface finite element method. *J. Comput. Phys.* **2008**, *227*, 9727–9741.
- Fuselier, E.J.; Wright, G. A high-order kernel method for diffusion and reaction-diffusion equations on surfaces. *J. Sci. Comput.* **2013**, *56*, 535–565. [\[CrossRef\]](#)
- Li, J.; Zhai, S.; Weng, Z.; Feng, X. H-adaptive RBF-FD method for the high-dimensional convection-diffusion equation. *Int. Commun. Heat Mass Transf.* **2017**, *89*, 139–146. [\[CrossRef\]](#)
- Li, J.; Gao, Z.; Feng, X. Method of order reduction for the high-dimensional convection-diffusion-reaction equation with Robin boundary conditions based on MQ RBF-FD. *Int. J. Comp. Methods* **2020**, *17*, 1950058. [\[CrossRef\]](#)
- Shankar, V.; Wright, G.; Kirby, R.M.; Fogelson, A.L. A radial basis function (RBF)—Finite difference (FD) method for diffusion and reaction-diffusion equations on surfaces. *J. Sci. Comput.* **2015**, *63*, 745–768. [\[CrossRef\]](#) [\[PubMed\]](#)
- Sokolov, A.; Davydov, O.; Kuzmin, D.; Westermann, A.; Turek, S. A flux-corrected RBF-FD method for convection dominated problems in domains and on manifolds. *J. Numer. Math.* **2019**, *27*, 253–269. [\[CrossRef\]](#)
- Zhao, F.; Li, J.; Xiao, X.; Feng, X. The characteristic RBF-FD method for the convection-diffusion-reaction equation on implicit surfaces. *Numer. Heat Transf. A-Appl.* **2019**, *75*, 548–559. [\[CrossRef\]](#)
- Shankar, V.; Wright, G.B.; Narayan, A. A robust hyperviscosity formulation for stable RBF-FD discretizations of advection-diffusion-reaction equations on manifolds. *SIAM J. Sci. Comput.* **2020**, *42*, A2371–A2401. [\[CrossRef\]](#)
- Wendland, H.; Künemund, J. Solving partial differential equations on (evolving) surfaces with radial basis functions. *Adv. Comput. Math.* **2020**, *46*, 46–64. [\[CrossRef\]](#)
- Petras, A.; Ling, L.; Piret, C.M.; Ruuth, S.J. A least-squares implicit RBF-FD closest point method and applications to PDEs on moving surfaces. *J. Comput. Phys.* **2019**, *381*, 146–161. [\[CrossRef\]](#)
- Demanet, L. *Painless, Highly Accurate Discretizations of the Laplacian on a Smooth Manifold*; Technical Report; Stanford University: Stanford, CA, USA, 2006.
- Suchde, P.; Kuhnert, J. A meshfree generalized finite difference method for surface PDEs. *Comput. Math. Appl.* **2019**, *78*, 2789–2805. [\[CrossRef\]](#)
- Suchde, P.; Kuhnert, J. A fully Lagrangian meshfree framework for PDEs on evolving surfaces. *J. Comput. Phys.* **2019**, *395*, 38–59. [\[CrossRef\]](#)
- Shaw, S.B. Radial Basis Function Finite Difference Approximations of the Laplace-Beltrami Operator. Master's Thesis, Boise State University, Boise, ID, USA, 2019.
- Casciola, G.; Montefusco, L.B.; Morigi, S. The regularizing properties of anisotropic radial basis functions. *Appl. Math. Comput.* **2007**, *190*, 1050–1062. [\[CrossRef\]](#)

23. Beatson, R.; Davydov, O.; Levesley, J. Error bounds for anisotropic RBF interpolation. *J. Approx. Theory* **2010**, *162*, 512–527. [[CrossRef](#)]
24. Casciola, G.; Lazzaro, D.; Montefusco, L.B.; Morigi, S. Shape preserving surface reconstruction using locally anisotropic radial basis function interpolants. *Comput. Math. Appl.* **2006**, *51*, 1185–1198. [[CrossRef](#)]
25. Fornberg, B.; Piret, C. A stable algorithm for flat radial basis functions on a sphere. *SIAM J. Sci. Comput.* **2007**, *30*, 60–80. [[CrossRef](#)]
26. Larsson, E.; Lehto, E.; Heryudono, A.; Fornberg, B. Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions. *SIAM J. Sci. Comput.* **2013**, *35*, 2096–2119. [[CrossRef](#)]
27. Álvarez, D.; González-Rodríguez, P.; Moscoso, M. A closed-form formula for the RBF-based approximation of the Laplace-Beltrami operator. *J. Sci. Comput.* **2018**, *77*, 1115–1132. [[CrossRef](#)]
28. Álvarez, D.; González-Rodríguez, P.; Kindelan, M. A local radial basis function method for the Laplace-Beltrami operator. *J. Sci. Comput.* **2021**, *86*, 28–48. [[CrossRef](#)]
29. Gunderman, D.; Flyer, N.; Fornberg, B. Transport schemes in spherical geometries using spline-based RBF-FD with polynomials. *J. Comput. Phys.* **2020**, *408*, 109256. [[CrossRef](#)]