

No. 579

November 2017

**Numerical study of the RBF-FD level set based
method for partial differential equations on
evolving-in-time surfaces**

A. Sokolov, O. Davydov, S. Turek

Numerical study of the RBF-FD level set based method for partial differential equations on evolving-in-time surfaces

Andriy Sokolov* Oleg Davydov† Stefan Turek*

November 30, 2017

Abstract

In this article we present a Radial Basis Function (RBF)-Finite Difference (FD) level set based method for numerical solution of partial differential equations (PDEs) of the reaction-diffusion-convection type on an evolving-in-time hypersurface $\Gamma(t)$. In a series of numerical experiments we study the accuracy and robustness of the proposed scheme and demonstrate that the method is applicable to practical models.

Key words Radial basis functions, finite differences, evolving surfaces, level set, surface PDEs

1 Introduction

Numerical simulation of partial differential equations posed on an evolving-in-time hypersurface $\Gamma(t)$ is a rapidly growing branch of numerical mathematics, which finds its applications in many industrial tasks. During the last decade many profound finite-element-based methods for surfaces-defined PDEs were developed: parametric methods [8, 9, 10], bulk-layer methods of the phase-field [21] and level-set [9, 23] types, the trace FEM [19] and the space-time FEM [20], etc. All these methods are of the finite element nature, meaning that one has to construct a mesh before any numerical simulation begins. Very often, some largely CPU- and time-consuming work has to be done with or related to the mesh during the simulation process.

On the other hand, kernel methods based on radial basis functions are becoming increasingly popular for the numerical simulation of partial differential equations due to their flexibility of working with scattered data nodes, high accuracy, and significantly simpler implementation. These methods demonstrated promising results for various problems of PDEs in two- and three-dimensional domains, see, e.g. [1, 2, 3, 13].

In the recently appeared works of G. Wright et al. [12, 22] the RBF-FD methodology was applied to the simulation of surface PDEs of reaction-diffusion type on stationary

*Institute for Applied Mathematics, TU Dortmund, Email addresses: asokolow@math.uni-dortmund.de, ture@featflow.de.

†Department of Mathematics, University of Giessen, Email address: oleg.davydov@math.uni-giessen.de.

manifolds. In the current paper, with the help of the level set technique, we extend the RBF-FD method to reaction-diffusion-convection partial differential equations on evolving-in-time surfaces.

2 PDE on evolving hypersurface

2.1 Problem formulation

We consider the following reaction-diffusion-convection equation

$$\frac{\partial^* u}{\partial t} + \mathbf{w} \cdot \nabla_{\Gamma(t)} u = D \Delta_{\Gamma(t)} u + g(u) \text{ on } \Gamma(t) \times T, \quad (1)$$

where $\Gamma(t)$ is a compact, smooth, connected and closed hypersurface in \mathbb{R}^d , $d = 2, 3$. Then, $\frac{\partial^* u}{\partial t}$ is a time-derivative, which takes into account the evolution of $\Gamma(t)$ and will be explained below, $\Delta_{\Gamma(t)} u$ is the Laplace-Beltrami term, \mathbf{w} is some vector field which transports u along $\Gamma(t)$ and $g(\cdot)$ is a kinetic term. The corresponding initial and boundary (if any) conditions for u have to be provided. We adopt the notation by writing vector fields in bold letters, i.e., $\mathbf{c} = (c_1, \dots, c_n)^T$. We assume that the solution u of (1) can be (naturally) extended from $\Gamma(t)$ to an ϵ -band $\Omega_\epsilon(t)$, see Figure 1. The domain of interest or also the calculational domain is $\Omega = \Omega_{in} \cup \Omega_{out} \cup \Gamma$. For the sake of simplicity, we also assume that $\Gamma(t) \subset \Omega_\epsilon(t) \subset \Omega$ during the whole simulation time $t \in [0, T]$.

The surface derivative

$$\frac{\partial^* u}{\partial t} = \partial_t^\bullet u + u \nabla_{\Gamma(t)} \cdot \mathbf{v}$$

can be obtained by the Leibniz formula

$$\frac{d}{dt} \int_{\Gamma(t)} u = \int_{\Gamma(t)} \partial_t^\bullet u + u \nabla_{\Gamma(t)} \cdot \mathbf{v}.$$

By $\partial_t^\bullet u = \partial_t u + \mathbf{v} \cdot \nabla u$ one denotes the covariant or advective surface material derivative. The surface velocity $\mathbf{v} = V \mathbf{n} + \mathbf{v}_S$ can be decomposed into velocity components in the normal direction $V \mathbf{n}$, with \mathbf{n} to be a surface outward normal vector, and in the tangential direction \mathbf{v}_S . Using the relation

$$\begin{aligned} \nabla_\Gamma \cdot \mathbf{v} &= \nabla_\Gamma V \cdot \mathbf{n} + V \nabla_\Gamma \cdot \mathbf{n} + \nabla_\Gamma \cdot \mathbf{v}_S = V \nabla_\Gamma \cdot \mathbf{n} + \nabla_\Gamma \cdot \mathbf{v}_S = \\ &= -VH + \nabla_\Gamma \cdot \mathbf{v}_S, \end{aligned}$$

and therefore

$$\mathbf{v} \cdot \nabla u = V \mathbf{n} \cdot \nabla u + \mathbf{v}_S \cdot \nabla u = V \frac{\partial u}{\partial \mathbf{n}} + \mathbf{v}_S \cdot \nabla u,$$

where H is a mean curvature, we can rewrite (1) as

$$\partial_t u + \mathbf{v}_S \cdot \nabla u - VH u + V \frac{\partial u}{\partial \mathbf{n}} + u \nabla_{\Gamma(t)} \cdot \mathbf{v}_S + \mathbf{w} \cdot \nabla_{\Gamma(t)} u = D \Delta_{\Gamma(t)} u + g(u),$$

or, in terms of the surface material derivative, as

$$\partial_t^\bullet u + u \nabla_\Gamma \cdot \mathbf{v} + \mathbf{w} \cdot \nabla_{\Gamma(t)} u = D \Delta_{\Gamma(t)} u + g(u). \quad (2)$$

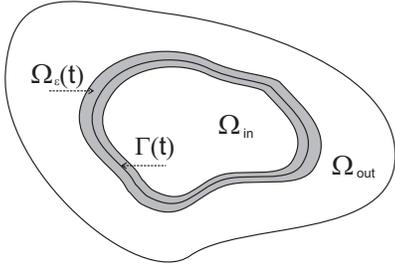


Figure 1: Geometric illustration

2.2 Level set method

For the implicit prescription of a compact, smoothly connected and oriented hypersurface $\Gamma(t) \subset \Omega$ we introduce a smooth level set function

$$\phi(t, \mathbf{x}) = \begin{cases} < 0, & \text{if } \mathbf{x} \text{ is inside } \Gamma(t), \\ = 0, & \text{if } \mathbf{x} \in \Gamma(t), \\ > 0, & \text{if } \mathbf{x} \text{ is outside } \Gamma(t), \end{cases} \quad (3)$$

such that $|\nabla\phi| \neq 0$. Then, an outward normal to $\Gamma(t)$ at the point \mathbf{x} is

$$\mathbf{n}(\mathbf{x}) = (n^1, n^2, \dots, n^d)^T = \nabla\phi(\mathbf{x})/|\nabla\phi(\mathbf{x})| \quad (4)$$

and the matrix

$$\mathcal{P}_\Gamma = I - \mathbf{n}\mathbf{n}^T = (\delta_{ij} - n^i n^j)_{i,j=1}^d \quad (5)$$

is the projection onto the tangent space $\mathcal{T}_\mathbf{x}\Gamma(t)$. For a scalar function η and a tangential vector field $\boldsymbol{\eta} = (\eta^1, \eta^2, \dots, \eta^d)^T$ on Γ extended into Ω we can define

$$\nabla_\Gamma \eta := (\mathcal{P}_\Gamma \nabla) \eta = \left\{ \frac{\partial \eta}{\partial x_i} - \sum_{j=1}^d n^i n^j \frac{\partial \eta}{\partial x^j} \right\}_{i=1}^d, \quad (6)$$

$$\nabla_\Gamma \cdot \boldsymbol{\eta} = \sum_{i=1}^d \left(\frac{\partial \eta^i}{\partial x^i} - \sum_{j=1}^d n^i n^j \frac{\partial \eta^i}{\partial x^j} \right), \quad (7)$$

the surface gradient ∇_Γ and the surface divergence $\nabla_\Gamma \cdot$ operators, respectively. Using this notation, the Laplace-Beltrami operator can be written as

$$\Delta_\Gamma \eta = \nabla_\Gamma \cdot \nabla_\Gamma \eta = \mathcal{P}_\Gamma \nabla \cdot \mathcal{P}_\Gamma \nabla \eta. \quad (8)$$

2.3 Discretization in time

For the discretisation in time of the surface PDE

$$\partial_t u + \mathbf{v} \cdot \nabla u + u \nabla_{\Gamma(t)} \cdot \mathbf{v} + \mathbf{w} \cdot \nabla_{\Gamma(t)} u = D \Delta_{\Gamma(t)} u + g(u), \quad (9)$$

we use the θ -scheme method. Given u^n and the time step $\Delta t = t_{n+1} - t_n$, solve for $u = u^{n+1}$ (for the sake of simplicity we omit the index $\{n+1\}$ there it is possible, e.g. $t = t^{n+1}$)

$$\begin{aligned} \frac{u - u^n}{\Delta t} &+ \theta (\mathbf{v} \cdot \nabla u + u \nabla_{\Gamma(t)} \cdot \mathbf{v} + \mathbf{w} \cdot \nabla_{\Gamma(t)} u - D \Delta_{\Gamma(t)} u + g(u)) \\ &= -(1 - \theta) (\mathbf{v}^n \cdot \nabla u^n + u^n \nabla_{\Gamma(t^n)} \cdot \mathbf{v}^n \\ &+ \mathbf{w}^n \cdot \nabla_{\Gamma(t^n)} u^n - D \Delta_{\Gamma(t^n)} u^n + g(u^n)). \end{aligned} \quad (10)$$

If we denote corresponding discrete operators, whose RBF-FD construction will be described in Section 3, by

$$L(t, \Gamma(t)) \mathbf{u} \approx -\Delta_{\Gamma(t)} u|_X, \quad (11)$$

$$\tilde{K}(t, \mathbf{v}) \mathbf{u} \approx -\mathbf{v} \cdot \nabla u|_X, \quad (12)$$

$$\tilde{\tilde{K}}(t, \mathbf{w}, \Gamma(t)) \mathbf{u} \approx -\mathbf{w} \cdot \nabla_{\Gamma(t)} u|_X, \quad (13)$$

$$G(t, \Gamma(t)) \mathbf{u} \approx u \nabla_{\Gamma(t)} \cdot \mathbf{v}|_X, \quad (14)$$

where $\mathbf{u} = (u_1, u_2, \dots, u_N)^T \approx u|_X = (u(\mathbf{x}_1), u(\mathbf{x}_2), \dots, u(\mathbf{x}_N))^T$, where $X = \{\mathbf{x}_j\}_{j=1}^N \subset \Omega$, then the semi-discrete equation (10) can be rewritten in the following matrix form:

$$\begin{aligned} & [I + \theta \Delta t \{ -\tilde{K}(t, \mathbf{v}) - \tilde{K}(t, \mathbf{w}, \Gamma) + L(t, \Gamma) + G(t, \Gamma) \}] \mathbf{u} \\ &= [I - (1 - \theta) \Delta t \{ -\tilde{K}(t^n, \mathbf{v}^n) - \tilde{K}(t^n, \mathbf{w}^n, \Gamma^n) \\ &\quad + L(t^n, \Gamma^n) + G(t^n, \Gamma^n) \}] \mathbf{u}^n \\ &\quad + \theta \Delta t g(\mathbf{u}) + (1 - \theta) \Delta t g(\mathbf{u}^n), \end{aligned} \quad (15)$$

For our numerical simulations we take either the Implicit-Euler scheme, which corresponds to $\theta = 1$, or the Crank-Nicolson scheme, which is obtained from (15) by setting $\theta = \frac{1}{2}$.

3 RBF-FD for PDEs on evolving-in-time surfaces

3.1 Kernel interpolation and operator approximation

Given a set of scattered nodes $X = \{\mathbf{x}_j\}_{j=1}^N \subset \Omega$ we are looking for a continuous function $u : \Omega \rightarrow \mathbb{R}$ as a kernel interpolant, those general form is

$$I_\phi u(\mathbf{x}) = \sum_{j=1}^N c_j \Phi(\mathbf{x}, \mathbf{x}_j), \quad \mathbf{x} \in \Omega, \quad (16)$$

such that its restriction $u|_{\Gamma(t)}$ is a solution of equation (2). Here, Φ is a positive definite kernel called a *radial basis function* (RBF) with the property $\Phi(\mathbf{x}, \mathbf{y}) = \varphi(\|\mathbf{x} - \mathbf{y}\|)$. Denoting $\mathbf{r}_j(\mathbf{x}_i) = \|\mathbf{x}_i - \mathbf{x}_j\|$, the interpolation coefficients $\{c_j\}_{j=1}^N$ are determined by enforcing $I_\phi u|_X = u|_X$ as the following linear system:

$$A_X c_X = u_X, \quad (17)$$

where

$$A_X = \begin{bmatrix} \varphi(\mathbf{r}_1(\mathbf{x}_1)) & \varphi(\mathbf{r}_2(\mathbf{x}_1)) & \dots & \varphi(\mathbf{r}_N(\mathbf{x}_1)) \\ \varphi(\mathbf{r}_1(\mathbf{x}_2)) & \varphi(\mathbf{r}_2(\mathbf{x}_2)) & \dots & \varphi(\mathbf{r}_N(\mathbf{x}_2)) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\mathbf{r}_1(\mathbf{x}_N)) & \varphi(\mathbf{r}_2(\mathbf{x}_N)) & \dots & \varphi(\mathbf{r}_N(\mathbf{x}_N)) \end{bmatrix}, \quad c_X = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}, \quad u_X = \begin{bmatrix} u(\mathbf{x}_1) \\ u(\mathbf{x}_2) \\ \vdots \\ u(\mathbf{x}_N) \end{bmatrix}.$$

For a positive definite φ , this system is positive definite and hence solvable.

In the following we use the radial basis function finite difference (RBF-FD) method for approximation of all linear differential operators, which arise through our derivations. Let \mathcal{L} be one of these linear operators. Then the approximation of $\mathcal{L}u$ at the point ζ is sought as a weighted sum of function values $u(\xi_j)$ at the points $\Xi = \Xi_\zeta = \{\xi_1, \xi_2, \dots, \xi_K\}$ neighboring to ζ :

$$\mathcal{L}u(\zeta) \approx \sum_{j=1}^K \omega_j u(\xi_j), \quad \xi_j \in \Xi, \quad (18)$$

where the approximation weights $\boldsymbol{\omega} = (\omega_1, \omega_2, \dots, \omega_K)^T$ can be computed by solving the linear system

$$A_{\Xi} \boldsymbol{\omega} = [\mathcal{L}\varphi(\mathbf{r}_j(\boldsymbol{\zeta}))]_{j=1}^K \quad \text{with} \quad A_{\Xi} := [\varphi(\mathbf{r}_j(\boldsymbol{\xi}_i))]_{i,j=1}^K. \quad (19)$$

In general, good choice of stencil points $\boldsymbol{\xi}_i$ for the accurate approximation of $\mathcal{L}u(\boldsymbol{\zeta})$ is a nontrivial task which requires additional analysis [4, 3, 6]. In this article, the set $\Xi_{\boldsymbol{\zeta}}$ consists of the $K = 9$ points nearest to $\boldsymbol{\zeta}$ in the Euclidean distance, including $\boldsymbol{\zeta}$ itself. Either Gaussian $\varphi(r) = \exp(-\epsilon^2 r^2)$ with $\epsilon > 0$ close to zero, or the polyharmonic radial basis function $\varphi(r) = r^\gamma$ with $\gamma = 5$ are used in all presented numerical simulations. In the case of Gaussian we use a QR preconditioning technique that allows stable computation of the weights for any value of the shape parameter ϵ [14, 5, 18]. Polyharmonic RBF is only conditionally positive definite and therefore the interpolant (16) is extended in this case by a polynomial term of degree $\lfloor \gamma/2 \rfloor$, see [13, 11] for details.

In the case of a vector-valued operator \mathcal{L} the weights ω_j are vectors, and $\boldsymbol{\omega}$ is a matrix. In particular, (18) is replaced by

$$\nabla u(\boldsymbol{\zeta}) \approx \boldsymbol{\omega}_{\nabla}(\boldsymbol{\zeta}, \Xi)^T u_{\Xi} \quad (20)$$

for the gradient operator ∇ , where each column of the matrix $\boldsymbol{\omega}_{\nabla}(\boldsymbol{\zeta}, \Xi) \in \mathbb{R}^{K \times d}$ is obtained by solving (19) for the corresponding partial derivative operator. Clearly, a gradient-type operator $\mathcal{L}_{\text{grad}}^A u = A \nabla u$ with components $\sum_{j=1}^d a_{ij} \frac{\partial u}{\partial x_j}$, $i = 1, \dots, d$, where $A : \Omega \rightarrow \mathbb{R}^{d \times d}$, can be discretized as

$$\mathcal{L}_{\text{grad}}^A u(\boldsymbol{\zeta}) \approx \left[\sum_{i=1}^K \omega_{ij} u(\boldsymbol{\xi}_i) \right]_{j=1}^d = A(\boldsymbol{\zeta}) \boldsymbol{\omega}_{\nabla}^T(\boldsymbol{\zeta}, \Xi) u_{\Xi}, \quad (21)$$

where $\boldsymbol{\omega} := \boldsymbol{\omega}_{\nabla}(\boldsymbol{\zeta}, \Xi) A^T(\boldsymbol{\zeta})$. A simple calculation shows that the same weight matrix $\boldsymbol{\omega} = \boldsymbol{\omega}_{\nabla}(\boldsymbol{\zeta}, \Xi) A^T(\boldsymbol{\zeta})$ gives a discretization

$$\mathcal{L}_{\text{div}}^A \mathbf{u}(\boldsymbol{\zeta}) \approx \sum_{i=1}^K \sum_{j=1}^d \omega_{ij} u_j(\boldsymbol{\xi}_i) = \text{trace} (A(\boldsymbol{\zeta}) \boldsymbol{\omega}_{\nabla}^T(\boldsymbol{\zeta}, \Xi) \mathbf{u}_{\Xi}) \quad (22)$$

of the divergence-type operator $\mathcal{L}_{\text{div}}^A \mathbf{u} = A \nabla \cdot \mathbf{u} := \sum_{i,j=1}^d a_{ij} \frac{\partial u_i}{\partial x_j}$, where $\mathbf{u} = (u_1, \dots, u_d)^T$ is a vector-function, and $\mathbf{u}_{\Xi} = [u_j(\boldsymbol{\xi}_i)]_{i,j=1}^{K,d}$.

Formulas (21) and (22) can be combined to obtain an approximation of the anisotropic diffusion operator

$$\Delta^{A,B} u := A \nabla \cdot B \nabla u = \mathcal{L}_{\text{div}}^A \mathcal{L}_{\text{grad}}^B u, \quad A, B : \Omega \rightarrow \mathbb{R}^{d \times d}.$$

To this end, an auxiliary set of points $\Gamma = \{\gamma_1, \dots, \gamma_L\}$ is chosen in the neighborhood of $\boldsymbol{\zeta}$, an approximation of the vector

$$\mathbf{u}(\gamma_s) := \mathcal{L}_{\text{grad}}^B u(\gamma_s) \approx \left[\sum_{i=1}^K \omega_{ij}(\gamma_s) u(\boldsymbol{\xi}_i) \right]_{j=1}^d, \quad \boldsymbol{\omega}(\gamma_s) := \boldsymbol{\omega}_{\nabla}(\gamma_s, \Xi) B^T(\gamma_s),$$

is obtained by (21) for each $s = 1, \dots, L$, and inserted into (22), where Γ is used instead of Ξ . Setting $\tilde{\boldsymbol{\omega}} := \boldsymbol{\omega}_{\nabla}(\boldsymbol{\zeta}, \Gamma) A^T(\boldsymbol{\zeta})$, we arrive at

$$\Delta^{A,B} u(\boldsymbol{\zeta}) \approx \sum_{i=1}^K \omega_i u(\boldsymbol{\xi}_i), \quad \omega_i = \sum_{s=1}^L \sum_{j=1}^d \tilde{\omega}_{sj} \omega_{ij}(\gamma_s), \quad (23)$$

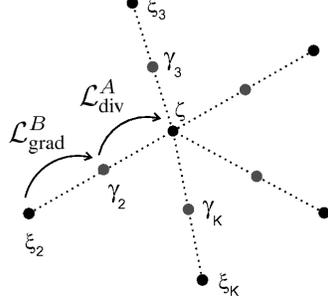


Figure 2: Discretization of the anisotropic diffusion.

that is

$$\omega_i = \text{trace}(\tilde{\omega} [\omega_{ij}(\gamma_s)]_{j,s=1}^{d,L}), \quad i = 1, \dots, K.$$

In the case when $A = B$ and $\zeta \in \Gamma = \Xi$ the formulas for ω_i in (23) can be simplified since $\tilde{\omega}$ coincides with one of the matrices $\omega(\gamma_s)$, see [12, 22]. We however prefer to choose Γ closer to ζ , in order to obtain more reliable numerical differentiation formulas for $\mathcal{L}_{\text{grad}}^B u(\gamma_s)$. In this paper we use

$$\gamma_j = (\zeta + \xi_j)/2, \quad j = 1, \dots, K, \quad (24)$$

see Figure 2, where $\xi_1 = \zeta$.

3.2 RBF-FD discretization in space

We now describe the discrete operators in (11)–(14). After choosing a set of nodes $X = \{\mathbf{x}_j\}_{j=1}^N \subset \Omega$, we select for each $\zeta \in X$ a set of neighbors $\Xi_\zeta \subset X$.

Thanks to (8), the value of the Laplace-Betrami operator $-\Delta_{\Gamma(t)} u(\zeta)$ can be approximated according to (23) with $-A = B = \mathcal{P}_{\Gamma(t)}$, and the weights ω_i of this formula become the nonzero entries of the ζ -row of the matrix $L(t, \Gamma(t))$ in (11).

For the generalized RBF-FD approximation of convection operators $\mathbf{v} \cdot \nabla u$ and $\mathbf{w} \cdot \nabla_{\Gamma(t)} u$ we make an assumption that both vector fields \mathbf{v} and \mathbf{w} can be extended outside of $\Gamma(t)$ to the whole domain Ω . In the case of the level set framework this extension of the surface velocity \mathbf{v} is straightforward as a velocity field of the corresponding level set. Then by (6), $\nabla_{\Gamma(t)} u = \mathcal{P}_{\Gamma(t)} \nabla u$, and hence for example

$$(\mathbf{w} \cdot \nabla_{\Gamma(t)} u)(\zeta) \approx \mathbf{w}^T(\zeta) \mathcal{P}_{\Gamma(t)}(\zeta) \omega_{\nabla}^T(\zeta, \Xi_\zeta) u_{\Xi_\zeta}$$

as in (21), leading to the weights for the ζ -row of $\tilde{K}(t, \mathbf{w}, \Gamma(t))$ in (13). Note that for convection dominated flows this approximation cannot be used as it is because of the stability issues: dominated convection terms may lead to the non-positiveness of a given numerical scheme and in such a way cause the appearance of negative values and give rise to non-physical oscillations in the numerical solution. In this article though we do not discuss this issue.

Construction of the RBF-FD approximation of the term $u \nabla_{\Gamma(t)} \cdot \mathbf{v}$ in (14) is done by (22) in the form

$$(u \nabla_{\Gamma(t)} \cdot \mathbf{v})(\zeta) \approx \omega_\zeta u(\zeta), \quad \omega_\zeta = \text{trace}(\mathcal{P}_{\Gamma(t)}(\zeta) \omega_{\nabla}^T(\zeta, \Xi_\zeta) \mathbf{v}_{\Xi_\zeta}).$$

Hence, $G(t, \Gamma(t))$ in (14) is a diagonal matrix with the numbers ω_ζ on the diagonal.

4 Numerical results

Here we demonstrate the applicability of the proposed RBF-FD scheme. In the following subsections we validate the spatial convergence of our scheme by considering an example of a heat equation on a curve. In next subsections we show that the scheme can be applied not only to the surface evolution in the normal directions but also in the tangential one. In the last example we apply the RBF-FD approximation to convection dominated problems to demonstrate that additional stabilization techniques are required in this case.

4.1 Example 1

In the first test case we will validate the performance of the scheme and measure its accuracy by comparing with a given analytical solution. We solve the following equation

$$\frac{\partial^* u(\mathbf{x}, t)}{\partial t} = D\Delta_{\Gamma(t)}u(\mathbf{x}, t) + g(\mathbf{x}, t) \quad \text{on } \Gamma(t), \quad (25)$$

where $\Gamma(t)$ is prescribed as the zero level set of the function

$$\phi(\mathbf{x}, t) = |\mathbf{x}| - 1.0 + \sin(4t)(|\mathbf{x}| - 0.5)(1.0 - |\mathbf{x}|). \quad (26)$$

As a domain we choose $\Omega = \{\mathbf{x} \in \mathbb{R}^2 : 0.5 \leq |\mathbf{x}| \leq 1.0\}$. The boundary of the domain $\partial\Omega$ is aligned with a curve from the family Γ_r . The analytical solution is chosen to be

$$u(\mathbf{x}, t) = e^{-t/|\mathbf{x}|^2} \frac{x_1}{|\mathbf{x}|}. \quad (27)$$

Since $\Gamma(t)$ is time-dependent, the equation (25) transforms into

$$\partial_t u + \mathbf{v}_S \cdot \nabla u + V \frac{\partial u}{\partial \mathbf{n}} - VHu + u\nabla_\Gamma \cdot \mathbf{v}_S - \Delta_\Gamma u = g, \quad (28)$$

where H is the mean curvature of $\Gamma(t)$ and therefore $H = -1/|\mathbf{x}|$. Substituting $\mathbf{v}_S = 0$ into (28) we get

$$\partial_t u + V \frac{\partial u}{\partial \mathbf{n}} - VHu - \Delta_\Gamma u = g. \quad (29)$$

The function $u(\mathbf{x}, t)$ from (27) solves

$$\partial_t u - \Delta_\Gamma u = 0.$$

Therefore, one finds that

$$g = V \frac{\partial u}{\partial \mathbf{n}} - VHu = Vu \left(\frac{2t}{|\mathbf{x}|^3} + \frac{1}{|\mathbf{x}|} \right).$$

As the initial condition we set $u_{\text{init}} = u(\mathbf{x}, t = 0)$. Here, we calculate numerical solutions by the implicit scheme, $\theta = 1$ in (10), and the Crank-Nicolson schemes, $\theta = 1/2$ in (10). The corresponding mesh, as well as initial condition and analytical and numerical solutions are shown in Figures 3(a)-3(d). Starting from $t = 0$, we calculate until the time point $T = 0.1$ with the time step $\Delta t \approx h^2$ by the Implicit-Euler scheme, $\theta = 1$ in (15), and $\Delta t \approx h$ by the second-order Crank-Nicolson scheme, $\theta = 1/2$ in (15).

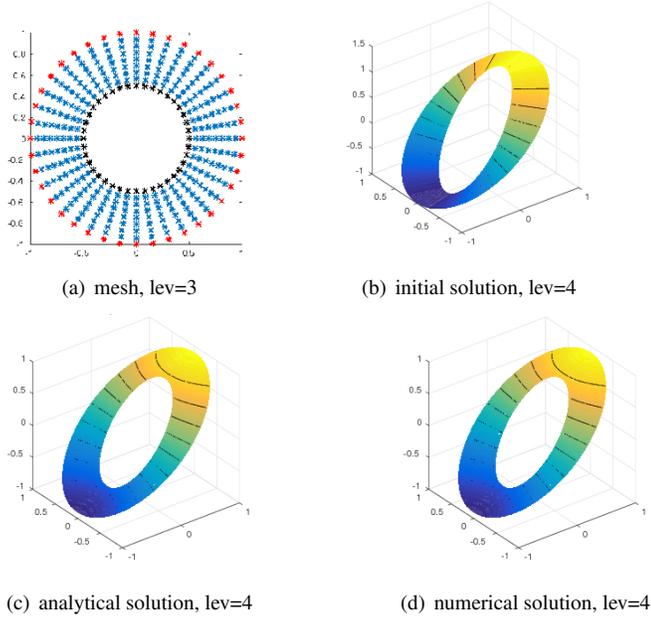


Figure 3: Mesh, initial, analytical and numerical solutions.

In Table 1 we measure the difference between the analytical and numerical solutions and obtain orders of convergence for the Implicit-Euler and Crank-Nicolson schemes. The corresponding error is defined as (cf. [7])

$$l_2(\Omega)\text{-error} = \left(\frac{1}{|d.o.f.}| \sum_{\mathbf{x}_i \in \Omega} |u_{\text{analyt}}(\mathbf{x}_i, T) - u_{\text{num}}(\mathbf{x}_i, T)|^2 \right)^{\frac{1}{2}},$$

| lev. | d.o.f | num. of time steps | $l_2(\Omega)$ -error | order |
|---|-------|--------------------|----------------------|-------|
| Implicit scheme, $\Delta t \approx h^2$ | | | | |
| 1 | 30 | 3 | 0.035854 | – |
| 2 | 100 | 10 | 0.009567 | 1.905 |
| 3 | 360 | 40 | 0.002602 | 1.878 |
| 4 | 1360 | 160 | 0.000748 | 1.798 |
| 5 | 5280 | 640 | 0.000213 | 1.812 |
| Crank-Nicolson, $\Delta t \approx h$ | | | | |
| 1 | 30 | 5 | 0.040218 | – |
| 2 | 100 | 10 | 0.09203 | 2.127 |
| 3 | 360 | 20 | 0.002367 | 1.959 |
| 4 | 1360 | 40 | 0.000673 | 1.814 |
| 5 | 5280 | 80 | 0.000192 | 1.809 |

Table 1: Convergence of the Implicit-Euler and Crank-Nicolson schemes.

One observes that the Crank-Nicolson scheme requires much fewer time steps in order to reach accuracy of the second order as the Implicit-Euler scheme.

4.2 Example 2

As our second test case we take Example 2 from [7]: we solve the equation (25) in the domain $\Omega = \{\mathbf{x} \in \mathbb{R}^2 : 0.5 \leq |\mathbf{x}| \leq 1.0\}$ on the stationary level sets of

$$\phi(\mathbf{x}, t) = |\mathbf{x}| - 0.75.$$

Here, the initial solution is $u_0(\mathbf{x}) = \sin(4\gamma)$, where $\gamma \in [0, 2\pi)$ is the polar angle and the tangential velocity of the surface Γ is $\mathbf{v}_S = 0$. Since $\gamma_t = 0$, the normal component of the surface velocity V is also zero. The mean value of u_0 vanishes on every level set Γ_r , hence the solution tends to zero as time tends to infinity. But this occurs at a rate which depends on the radius of the circle because of the different diffusion coefficients on the different circles. Numerical solutions at successive time instances are presented in Figure 4. Here, the Implicit-Euler and Crank-Nicolson schemes deliver the same numerical results, with the difference that the Crank-Nicolson scheme requires much fewer time steps.

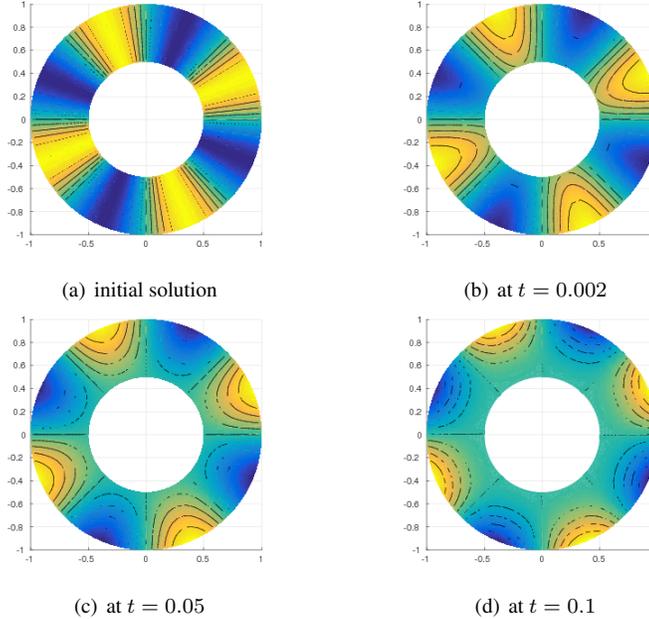


Figure 4: Solution at various time instances, $\Delta t = 0.0001$.

4.3 Example 3

In this test case we keep everything similar to the previous example in Section 4.2, but the tangential velocity of the surface is defined as

$$\mathbf{v}_S = 10 \frac{(-\phi_{x_2}, \phi_{x_1})}{|\nabla \phi|}. \quad (30)$$

Numerical results at some instances of time intervals are shown in Figure 5. This example demonstrates that our approach is able to treat PDEs on time-dependent surfaces which move not only in the normal, but also in the tangential direction.

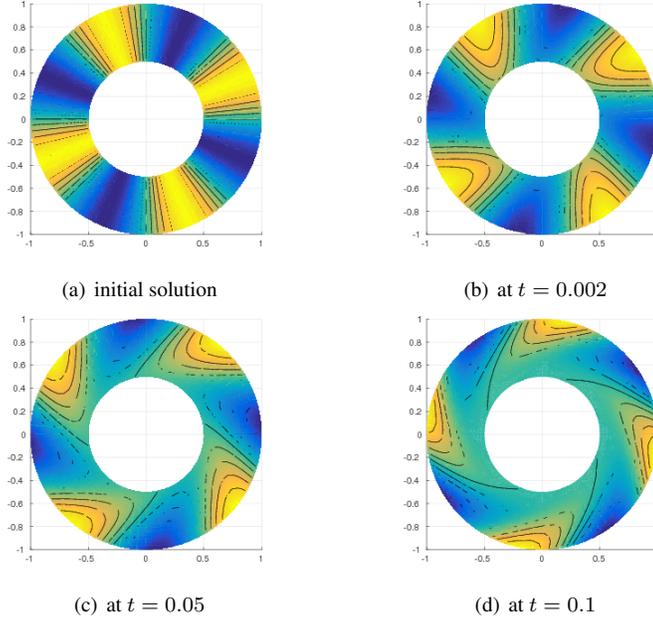


Figure 5: Solution at various time instances, $\Delta t = 0.0001$.

4.4 Example 4

The RBF-FD approximations of operator make it possible to perform numerical simulation for the pure transport equation:

$$\partial_t u + \mathbf{v} \cdot \nabla u = 0, \quad \text{in } \Omega = [0, 1]^2, \quad (31)$$

where $\mathbf{v} = (-y, x)$. As a simulation setting we choose $\Delta t = 0.001$ and $T = 3.0$. We place 6561 nodes in a Cartesian equidistant way inside Ω , which corresponds to the 81-by-81 refinement of the unit square. Initial conditions are taken from the solid-body rotation benchmark [16, 15, 17] and are shown in Figure 6(a).

The pure RBF-FD discretization for the transport problem does not guarantee mass conservation and does not keep numerical solution nonnegative. As a result, the nonphysical negative values of the numerical solution grow rapidly as time evolves, which might lead to an abnormal termination of the simulation run. In figure 6(b) we demonstrate the corresponding numerical solution at the time instance $t = 1.0$.

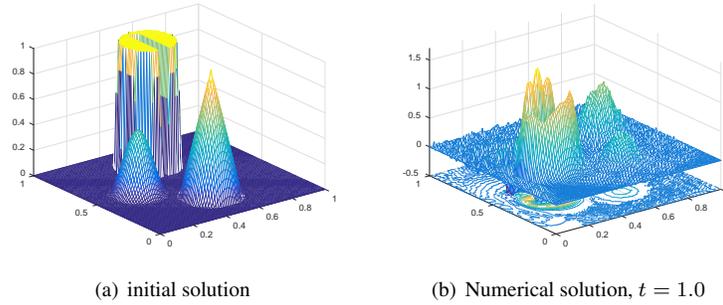


Figure 6: The pure RBF-FD scheme for the transport problem.

One can try to avoid this problem in many ways: to add nodes into those parts of the domain, where gradients of the numerical solution are large, to make the time-step smaller, to add some hyperviscosity into the model, or to use other stabilization techniques for the RBF-FD scheme. In figures 7(b)–7(d) the numerical results which are obtained by using FCT techniques from works of Kuzmin et al. [15, 16, 17] are presented. The discussion about the applicability of the FCT methods for RBF-FD schemes remain out of the scope of this work and will be discussed elsewhere.

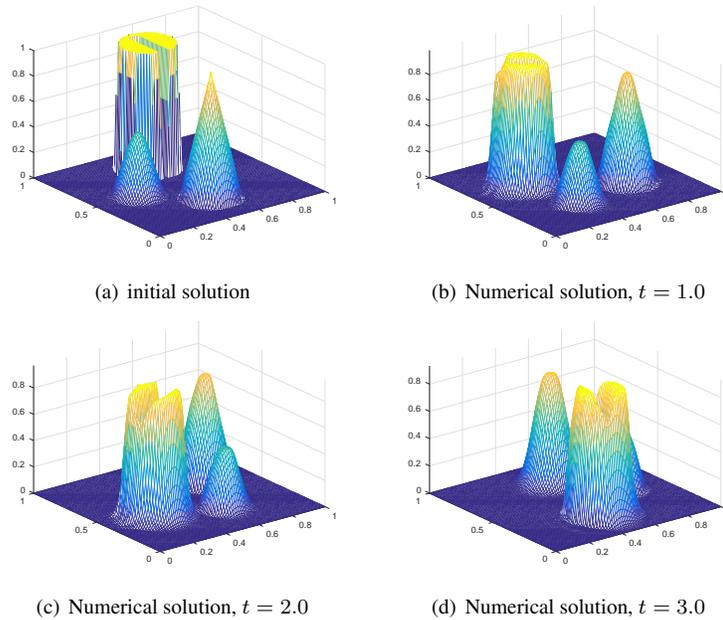


Figure 7: The FCT stabilization technique for the RBF-FD scheme of the transport problem.

4.5 Example 5

We test the following PDE on surfaces around a narrow band for Implicit-Euler and Crank-Nicolson schemes

$$\frac{\partial^* u}{\partial t} + \alpha u = D\Delta_{\Gamma(t)}u \quad \text{on } \Gamma(t), \quad (32)$$

where

$$\frac{\partial^* u}{\partial t} = \partial_t u + \mathbf{v} \cdot \nabla u + u \nabla_{\Gamma(t)} \cdot \mathbf{v}. \quad (33)$$

For test purposes we set $\alpha = 0.2$. Let us choose the initial prescription of the level set function as

$$\phi(\mathbf{x}, t) = |\mathbf{x}| - (1.0 + bt \sin(5\gamma))$$

with $b = 10$ and γ is the polar angle.

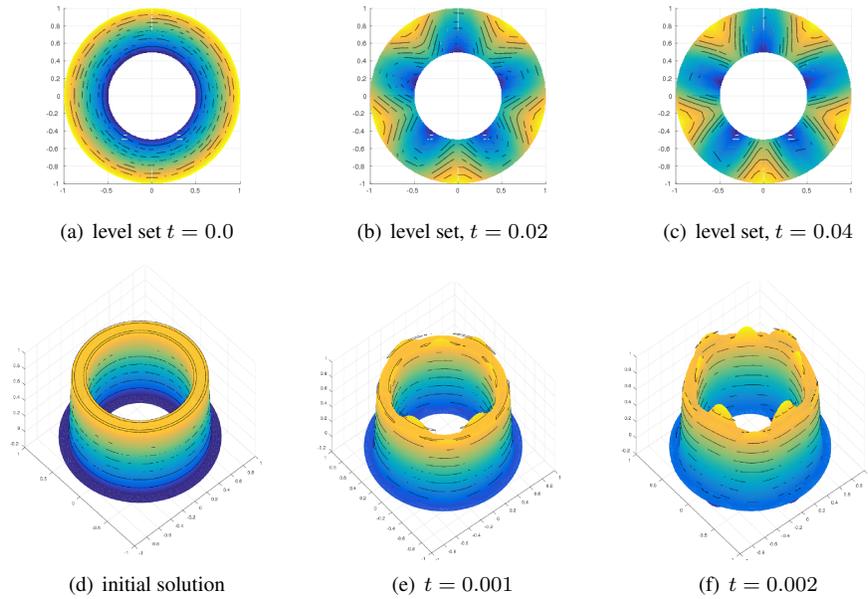


Figure 8: Evolution of the level set and numerical solution.

The computational domain is $\Omega = \{\mathbf{x} \in \mathbb{R}^2 : 0.5 \leq |\mathbf{x}| \leq 1.5\}$ and the initial condition u_0 is

$$u_0 = \begin{cases} 0.75 & \text{if } 0.65 \leq |\mathbf{x}| \leq 0.85, \\ 0.0 & \text{otherwise.} \end{cases}$$

Figures 8(a)-8(c) show the evolution of the level set at different time instances. Figure 8(d) is the initial solution and corresponding results in Figures 8(e) and 8(e) are obtained through the Crank-Nicolson scheme. Here, we use the mesh at the 4th level of refinement (which corresponds to 1360 d.o.f). The time-step is chosen to be $\Delta t = 0.0001$. We observe that as time evolves, numerical oscillations appear near vicinities of the steep gradient of u . These nonphysical negative values grow rapidly as time evolves, as a result some additional treatment of this effect is required.

5 Conclusion

In the current article we presented some methodology that allows the extension of the Radial Basis Function (RBF)-Finite Difference (FD) scheme to the numerical solution of partial differential equations (PDEs) of the reaction-diffusion type on an evolving-in-time hypersurface $\Gamma(t)$. Our numerical results confirm the reliability of the proposed computational framework in terms of numerical convergence and capturing of typical/expected solution profiles. We have thus developed an RBF-FD approach that can be employed for practical applications that involve PDEs on evolving surfaces.

The framework has a straightforward extension to three dimensional models which is mandatory when considering real-life applications, though some computational and code optimization are required, since the computational and analytical complexity significantly increases in three dimensional case. Detailed numerical investigations are subject of forthcoming work.

We also demonstrated that for convection dominated problems additional implementation of some stabilization technique is required to guarantee positivity preservation and non-oscillatory behavior of a numerical solution. In a follow-up paper we demonstrate that it is possible to efficiently adapt the Flux-corrected transport (FCT) technique to the proposed RBF-FD numerical scheme.

References

- [1] G. A. Barnett, N. Flyer, L. J. Wicker, *An RBF-FD polynomial method based on polyharmonic splines for the Navier-Stokes equations: Comparisons on different node layouts*, Journal of Computational Physics, arXiv:1509.02615, (2015).
- [2] T. Cecil, J. Qian and S. Osher, *Numerical methods for high dimensional Hamilton-Jacobi equations using radial basis functions*, Journal of Computational Physics, **196** (2004), pp. 327–347.
- [3] Dang Thi Oanh, Oleg Davydov and Hoang Xuan Phu, *Adaptive RBF-FD Method for Elliptic Problems with Point Singularities in 2D*, Applied Mathematics and Computation, **313** (2017), pp. 474–497.
- [4] Oleg Davydov and Dang Thi Oanh, *Adaptive meshless centres and RBF stencils for Poisson equation*, J. Comput. Phys., **230** (2011), pp. 287–304.
- [5] Oleg Davydov and Dang Thi Oanh, *On optimal shape parameter for Gaussian RBF-FD approximation of Poisson equation*, Comput. Math. Appl., **62** (2011), pp. 2143–2161.
- [6] O. Davydov and R. Schaback, *Error Bounds for Kernel-Based Numerical Differentiation*, Numer. Math., **132** (2016), pp. 243–269.
- [7] G. Dziuk and C.M. Elliott, *An Eulerian approach to transport and diffusion on evolving implicit surfaces*, Comput Visual Sci., **13** (2010), pp. 17–28.
- [8] G. Dziuk, C.M. Elliott, *A fully discrete evolving surface finite element method*, SIAM J. Numer. Anal., **50**, no. 5 (2012), pp. 2677–2694.
- [9] G. Dziuk and C.M. Elliott, *Finite element method for surface PDEs*, Acta Numerica, **50**, vol. 22 (2013), pp. 289–396.

- [10] C. M. Elliott, B. Stinner and C. Venkataraman, *Modelling cell motility and chemotaxis with evolving surface finite elements*, J. R. Soc. Interface., **9**, no. 76 (2012), pp. 3027–44.
- [11] Natasha Flyer, Bengt Fornberg, Victor Bayona, Gregory A. Barnett, *On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy*, Journal of Computational Physics, **321** (2016), pp. 21–38.
- [12] Edward J. Fuselier, Grady B. Wright, *A High-Order Kernel Method for Diffusion and Reaction-Diffusion Equations on Surfaces*, arXiv:1206.0047, (2012).
- [13] G. E. Fasshauer, *Meshfree Approximation Methods with Matlab*, Interdisciplinary Mathematical Sciences - Vol. 6. World Scientific Publishers, Singapore (2007).
- [14] B. Fornberg, E. Larsson, and N. Flyer, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput., **33**, no. 2 (2011), pp. 869–892.
- [15] D. Kuzmin and S. Turek, *Flux correction tools for finite elements*, J. Comput. Phys., **175** (2002), pp. 525–558.
- [16] D. Kuzmin and M. Möller, *Algebraic flux correction I. Scalar conservation laws*, in: D. KUZMIN, R. LÖHNER, S. TUREK (Eds.), *Flux-Corrected Transport: Principles, Algorithms, and Applications*, Springer, Berlin, (2005), pp. 155–206.
- [17] D. Kuzmin, *Explicit and implicit FEM-FCT algorithms with flux linearization*, J. Comput. Phys., **228**, (2009), pp. 2517–2534.
- [18] Elisabeth Larsson, Erik Lehto, Alfa Heryudono, and Bengt Fornberg *Stable Computation of Differentiation Matrices and Scattered Node Stencils Based on Gaussian Radial Basis Functions*, SIAM Journal on Scientific Computing, **35**, no. 4 (2013), pp. A2096-A2119.
- [19] M. A. Olshanskii, A. Reusken and J. Grande, *A Finite Element method for elliptic equations on surfaces*, SIAM J. Numer. Anal., **47**, no. 5 (2009), pp. 3339–3358.
- [20] M. A. Olshanskii, A. Reusken and X. Xu, *An Eulerian space-time Finite Element method for diffusion problems on evolving surfaces*, SIAM J. Numer. Anal., **52**, no. 3 (2014), pp. 1354–1377.
- [21] A. Rätz and A. Voigt, *PDE’s on surfaces — a diffuse interface approach*, Comm. Math. Sci., **4**, no. 3 (2006), pp. 575–590.
- [22] Varun Shankar, Grady B. Wright, Robert M. Kirby and Aaron L. Fogelson, *A Radial Basis Function (RBF)-Finite Difference (FD) Method for Diffusion and Reaction-Diffusion Equations on Surfaces*, Journal of Scientific Computing, **63**, Issue 3 (2015), pp. 745-768.
- [23] A. Sokolov, R. Ali and Stefan Turek, *An AFC-stabilized implicit finite element method for partial differential equations on evolving-in-time surfaces*, Journal of Computational and Applied Mathematics, **289**, (2006), pp. 101–115.
- [24] G. Wright, *Radial Basis Function: Numerical and Analytical Developments*, Phd thesis, University of Colorado (2003).