Journal of Scientific Computing (2025) 103:2
https://doi.org/10.1007/s10915-024-02772-2

A Narrow Band Numerical Method for a Surface Reaction-Diffusion System Coupled with Surface Motion

Song Lu[1,2] · Xianmin Xu[2]

Received: 31 May 2023 / Revised: 9 April 2024 / Accepted: 13 December 2024

Abstract

Reaction-diffusion equations on surfaces are widely used for modeling various phenomena in biology. This paper presents a novel numerical method for solving a surface reaction-diffusion system coupled with the evolution of the surface. The coupled system has been used to model the growth of hard tumors. A stabilized trace finite element method is used to discretize the reaction-diffusion system on evolving surfaces. The surface motion is computed using a diffusion-generated method for the level-set function, which involves solving a heat equation in each time step followed by a redistance operation. Both the trace finite element space for the reaction-diffusion system and the finite element space for the level-set function are defined in a narrow band region near the surface on a bulk mesh. The method is fully decoupled and allows for easy handling of topology changes. Numerical experiments demonstrate the efficiency of the proposed method for solving this complex problem.

Keywords Reaction diffusion systems · Evolving surfaces · Trace finite element method · Diffusion generated motion

1 Introduction

Reaction diffusion systems have plenty of applications in biology. One of the most well-known models is that for Turing pattern suggested by Alan Turing [1]. This model illustrates that chemical species may form a patterned distribution of concentrations in space due to diffusion-driven instability. Furthermore, it has been discovered that the stripes of tigers and zebras, spots of cheetahs, sea shell patterns, fish patterns, and intricate patterns on animal feathers are all related to a similar mechanism [2–7]. In addition, the reaction-diffusion system can also be utilized to model and simulate solid tumor growth, where the system is defined on an evolving surface [8–10].

✉ Xianmin Xu
xmxu@lsec.cc.ac.cn

Song Lu
lusong@lsec.cc.ac.cn

[1] School of Mathematical Sciences, University of Chinese Academy of Sciences, 100049 Beijing, China

[2] LSEC, ICMSEC, NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 100190 Beijing, China

Published online: 12 February 2025

In the community of numerical analysis, there has been much interest in solving partial differential equations on evolving surfaces recently. As a result, numerous numerical methods have been developed (c.f. [11–21, 21, 22]). A Lagrangian framework-based approach, utilizing a surface finite element method, is a natural method to handle the evolution of the surface [23–25]. In this approach the transport properties of the finite element space are used to deal with the convection term in the system. When the surface undergoes significant deformation, the meshes may have poor qualities, and an arbitrary Eulerian-Langrangian method can be used instead [26]. In addition to the surface finite element methods, Eulerian framework-based numerical methods have also been developed, such as the trace finite element method [15, 27–29], among some others.

The above mentioned methods have also been applied to surface reaction-diffusion systems. Madzvamuse et al. developed a lumped-mass method for pattern formation on stationary surfaces [30, 31]. For more general problems on evolving surfaces, Barreira et al. proposed a surface finite element method for the phase-field model [32], while the trace finite element method has been used to study the Allen–Cahn equation on evolving surfaces in [33].

The focus of this paper is on numerical methods for surface reaction-diffusion systems coupled with surface evolution. As an example, we consider the Brusselator model of tumor growth, which consists of a reaction-diffusion system with two species (activator and inhibitor) defined on an evolving surface [8–10]. The surface growth depends on the activator distribution and the local mean curvature. The model has been solved numerically by a surface finite element method in [32].

We propose a novel numerical approach for the Brusselator model of tumor growth. We decouple the entire system by solving the reaction-diffusion equations first, and then moving the interface. To solve the reaction-diffusion system, we utilize a stabilized trace finite element method. For surface evolution, we develop a diffusion-generated method that efficiently solves mean curvature flow. Our method has a first-order accuracy with respect to the time step size. The numerical results are consistent with previous calculations. One key advantage of the present method is that we can deal with the topological change of the surface easily in the Eulerian framework. We present various numerical examples to further highlight the efficiency of our approach.

The structure of the paper is as follows: In Sect. 2, we provide a brief introduction to the mathematical model. In Sect. 3, we introduce the semi-discrete in time scheme and the weak formulae. The fully discrete method is described in details in Sect. 4. In Sect. 5, we present numerical results and we conclude the paper in the last section.

2 The Mathematical Model

Consider a closed tumor surface $\Gamma(t) \in \Omega$ changing its shape over time, with outward normal direction $\mathbf{n}(\mathbf{x}, t)$. For convenience of notation, we often omit the indicators t and $\mathbf{x}$. There are two chemical species on the tumor surface, one promoting its growth and the other inhibiting it, with concentrations u and v, respectively. Assume $\mathbf{w}$ is the velocity of the evolving surface, which satisfies

$$\mathbf{w} = (\lambda u - \epsilon H)\mathbf{n}, \tag{1}$$

where λ represents the growth rate, ϵ characterizes the surface tension of the tumor, and H is the mean curvature.[1] If the surface is convex at a point, the mean curvature there is positive, otherwise it is negative. The concentrations u and v satisfy the following reaction-diffusion

[1] We define the mean curvature as the sum of the principal curvatures rather than the arithmetic mean.

system on the tumor surface,

$$\dot{u} + u\nabla_\Gamma \cdot \mathbf{w} = D_1 \Delta_\Gamma u + f_1(u, v), \quad \text{on } \Gamma(t), \tag{2}$$

$$\dot{v} + v\nabla_\Gamma \cdot \mathbf{w} = D_2 \Delta_\Gamma v + f_2(u, v), \quad \text{on } \Gamma(t). \tag{3}$$

Here ∇_Γ is the surface gradient and Δ_Γ is the Laplace-Beltrami operator. D_1 and D_2 are diffusion coefficients. The reaction terms are given by the Brusselator model [9],

$$(f_1, f_2) = (\gamma(a - u + u^2 v), \gamma(b - u^2 v)), \tag{4}$$

where γ, a and b are given parameters. This model strongly couples the evolution of a curved surface with the reaction-diffusion system on the surface.

We know that the formation of a Turing pattern for a reaction-diffusion system, e.g. (2)–(3), must meet certain conditions [8]. Firstly, the reaction terms in the system must exhibit a certain type of nonlinearity, like that in Eq. (4), known as an activator-inhibitor dynamics. This implies that the chemical reactions between the species must involve a positive feedback loop, where the concentration of one species promotes the production of the other, while the second species inhibits the production of the first. Secondly, the diffusivities of the two species must be sufficiently different, i.e. D_1 and D_2 are not close to each other, so that one species diffuse more slowly than the other. This creates a spatial instability in the system, where small perturbations in the concentrations of the species can grow and develop into stable spatial patterns.

2.1 Reformulation of the PDE System

We will use the level-set method to capture the motion of the interface. The method works well even when the interface evolves with topological changes. Assume Γ can be represented implicitly by a level set function ϕ, i.e.

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

Here we simply assume that ϕ is negative inside Γ and positive outside. For any $\mathbf{x}$ on Γ, we know the material derivative $\dot{\phi}(\mathbf{x}) = 0$, i.e.

$$\phi_t + \mathbf{w} \cdot \nabla\phi = 0.$$

By Eq. (1) and $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$, the equation can be rewritten as

$$\phi_t + (\lambda u - \epsilon H)|\nabla\phi| = 0, \qquad \text{on } \Gamma(t).$$

The equation can be extended in a neighboring region of $\Gamma(t)$ as

$$\phi_t + (\lambda u^e - \epsilon H)|\nabla\phi| = 0. \tag{5}$$

Here u^e denotes a smooth extension of u from Γ to the neighboring region, defined as $u^e(x) = u(\boldsymbol{p}(x))$, where $\boldsymbol{p}(x) = x - d(x)\boldsymbol{n}$ is the projection of x to Γ. Notice that the mean curvature H can be written as a function of ϕ by

$$H = \operatorname{div} \frac{\nabla\phi}{|\nabla\phi|}. \tag{6}$$

With the definition of u^e, the reaction diffusion Eq. (2) can be rewritten as

$$\frac{\partial u^e}{\partial t} + \mathbf{w} \cdot \nabla u^e + u\nabla_\Gamma \cdot \mathbf{w} = D_1 \Delta_\Gamma u + f_1(u, v), \quad \text{on } \Gamma(t).$$

Notice the extension u^e satisfies $\boldsymbol{n} \cdot \nabla u^e = 0$. By the definition of the velocity in Eq. (1), the equation can be further simplified to

$$\frac{\partial u^e}{\partial t} + u(\lambda u - \epsilon H)H = D_1 \Delta_\Gamma u + f_1(u, v), \qquad \text{on } \Gamma(t). \tag{7}$$

Here we have used the fact that $\mathbf{w} \cdot \nabla u^e = (\lambda u - \epsilon H)\mathbf{n} \cdot \nabla u^e = 0$ and $\nabla_\Gamma \cdot \mathbf{w} = \nabla_\Gamma \cdot ((\lambda u - \epsilon H)\mathbf{n}) = (\lambda u - \epsilon H)\nabla_\Gamma \cdot \mathbf{n} = (\lambda u - \epsilon H)H$. Similarly, the Eq. (3) can be rewritten as

$$\frac{\partial v^e}{\partial t} + v(\lambda u - \epsilon H)H = D_2 \Delta_\Gamma v + f_2(u, v), \qquad \text{on } \Gamma(t). \tag{8}$$

In summary, the mathematical model we will study in this paper is a coupled system consist of Eqs. (5)–(8).

3 A Decoupled Semi-discrete Scheme

In this section, we will do discretization in time and introduce a decoupled semi-discrete scheme to the system (5)–(8) described in the previous section.

We solve the system (5)–(8) in a time interval $[0, T]$. We partition the interval uniformly as

$$0 = t_0 < \cdots < t_{n-1} < t_n < \cdots < t_N = T,$$

with the time step $\tau = \frac{T}{N}$. Suppose that we already have the solution of the system at time t_{n-1}, that is given by $(\phi^{n-1}, u^{n-1}, v^{n-1})$. We will compute the solution (ϕ^n, u^n, v^n) by a decoupled scheme.

3.1 A Diffusion Generated Method for the Level-Set Equation

We first consider the time dicretization to Eq. (5) which is a nonlinear partial differential equation. In general, it is very difficult to solve the equation numerically. Here we introduce an efficient numerical scheme based on an operator splitting technique. Notice that when ϕ is a sign distance function at time t_{n-1} defined as follows,

$$\phi(\mathbf{x}, t_{n-1}) = \begin{cases} -\text{dist}(\mathbf{x}, \Gamma^{n-1}), & \mathbf{x} \text{ is inside of } \Gamma^{n-1}, \\ \ \ \text{dist}(\mathbf{x}, \Gamma^{n-1}), & \qquad \text{otherwise}. \end{cases} \tag{9}$$

When τ is small, we assume that $|\nabla \phi| \approx 1$. Equation (5) can be approximated well by a heat equation

$$\phi_t - \epsilon \Delta \phi = -\lambda u^{n-1,e}, \tag{10}$$

at time t_{n-1}. This motivates us to consider a diffusion-generated motion method for Eq. (5) in each time step, as stated below.

Algorithm 1(One step evolution of the surface)

Step 1. Suppose ϕ^{n-1} and u^{n-1} are given, such that $|\nabla \phi^{n-1}| = 1$. Solve a heat Eq. (6) for one step by a backward Euler scheme,

$$\frac{\varphi - \phi^{n-1}}{\tau} - \epsilon \Delta \varphi = -\lambda u^{n-1,e}. \tag{11}$$

Step 2. We do redistance φ to obtain a signed distance function ϕ^n, i.e.

$$\phi^n := \mathrm{redist}(\varphi),$$

so that ϕ^n has the same zero level set as φ and satisfies $|\nabla \phi^n| = 1$.

The diffusion generated method originates from the MBO methold [34] for characteristic functions. The method can be derived either by a Taylor expansion [35] or by using the Onsager variational principle as an approximation tool [36]. The advantage of the method is that we need only to solve a linear diffusion equation for one time step plus a standard re-distance operation. This is much easier than solving a nonlinear geometric PDE for mean curvature flow.

3.2 A Finite Difference Scheme for the Surface Reaction-Diffusion System

Now we consider the time discretization of the reaction-diffusion system (7)–(8). We simply use a backward Euler scheme and decouple the system as follows,

$$\frac{u^n - u^{n-1,e}}{\tau} + (\lambda u^{n-1,e} - \epsilon H^n) H^n u^n - D_1 \Delta_\Gamma u^n = \tilde{f}_1(u^{n-1,e}, v^{n-1,e}, u^n), \quad \text{on } \Gamma^n, \tag{12}$$

$$\frac{v^n - v^{n-1,e}}{\tau} + (\lambda u^{n-1,e} - \epsilon H^n) H^n v^n - D_2 \Delta_\Gamma v^n = \tilde{f}_2(u^n, v^n), \quad \text{on } \Gamma^n. \tag{13}$$

Here $\Gamma^n = \{\mathbf{x} : \phi^n(\mathbf{x}) = 1\}$ and the mean curvature is calculated by

$$H^n = \Delta \phi^n,$$

where ϕ^n is obtained from Algorithm 1. $\tilde{f}_i$ are linearized forms to the functions f_i, given by

$$\tilde{f}_1(u^{n-1,e}, v^{n-1,e}, u^n) = \gamma(a - u^n + u^{n-1,e} v^{n-1,e} u^n),$$
$$\tilde{f}_2(u^n, v^n) = \gamma(b - (u^n)^2 v^n)).$$

We choose the semi-implicit scheme which is more stable than an explicit Euler scheme and easier to solve than a fully implicit scheme. Notice that the above system (12)–(13) is fully decoupled. We first solve the equation for u^n and then the equation for v^n in each time step. Both of them are linear equations.

3.3 Weak Formulae

Before we introduce the fully discrete scheme, we give the weak formulae for the Eqs. (11)–(13). We first consider the weak formulae to Eq. (11). Notice that we are interested only the evolution of the zero level-set of ϕ in Algorithm 1. Therefore, we can solve Eq. (11) in a narrow band region near Γ^{n-1}. Let $\mathcal{O}(\Gamma^{n-1})$ be a neighboring region of Γ^{n-1} defined as

$$\mathcal{O}(\Gamma^{n-1}) := \{\mathbf{x} \in \Omega : \mathrm{dist}(\mathbf{x}, \Gamma^{n-1}) < \delta_{n-1}\},$$

where $\delta_{n-1} > 0$ is a positive number making sure that $\Gamma^n \in \mathcal{O}(\Gamma^{n-1})$. Let $\mathcal{W}_{n-1} = H^1(\mathcal{O}(\Gamma^{n-1}))$. The weak form for Eq. (11) is to find $\varphi \in \mathcal{W}_{n-1}$, such that

$$\int_{\mathcal{O}(\Gamma^{n-1})} \varphi\psi + \epsilon\tau \int_{\mathcal{O}(\Gamma^{n-1})} \nabla\varphi \cdot \nabla\psi = \int_{\mathcal{O}(\Gamma^{n-1})} \phi^{n-1}\psi - \lambda\tau \int_{\mathcal{O}(\Gamma^{n-1})} u^{n-1,e}\psi, \quad \forall \psi \in \mathcal{W}_{n-1}. \tag{14}$$

We now derive weak formulae to the Eqs. (12)–(13). We first introduce some functional spaces as in [29]. Define $\mathcal{V}_n := \overline{\mathcal{V}_n^{\star}}^{\|\cdot\|_{\mathcal{V}_n}}$, with

$$\mathcal{V}_n^{\star} := \{v \in C^2(\mathcal{O}(\Gamma^n)) : \nabla v \cdot \nabla \phi^n = 0\},$$

$$\|v\|_{\mathcal{V}_n} := \left(\|v\|^2_{H^1(\Gamma^n)} + \|\nabla v \cdot \nabla \phi^n\|^2_{L^2(\mathcal{O}(\Gamma^n))}\right)^{\frac{1}{2}}.$$

Suppose that u^{n-1} and v^{n-1} belong to $\mathcal{V}_{n-1}^{*}$ so that their extensions are defined implicitly. Then the weak forms for (12)–(13) are given by,

$$\int_{\Gamma^n} \left[\gamma + \frac{1}{\tau} + (\lambda u^{n-1} - \epsilon H^n)H^n - \gamma u^{n-1}v^{n-1}\right] u^n w + D_1 \int_{\Gamma^n} \nabla_\Gamma u^n \cdot \nabla_\Gamma w$$
$$+ \rho_n \int_{\mathcal{O}(\Gamma^n)} (\boldsymbol{n} \cdot \nabla u^n)(\boldsymbol{n} \cdot \nabla w) = a\gamma \int_{\Gamma^n} w + \int_{\Gamma^n} \frac{u^{n-1}}{\tau} w, \qquad \forall w \in \mathcal{V}_n^{\star}, \tag{15}$$

and

$$\int_{\Gamma^n} \left[\frac{1}{\tau} + (\lambda u^n - \epsilon H^n)H^n + \gamma (u^n)^2\right] v^n w + D_2 \int_{\Gamma^n} \nabla_\Gamma v^n \cdot \nabla_\Gamma w$$
$$+ \rho_n \int_{\mathcal{O}(\Gamma^n)} (\boldsymbol{n} \cdot \nabla v^n)(\boldsymbol{n} \cdot \nabla w) = b\gamma \int_{\Gamma^n} w + \int_{\Gamma^n} \frac{v^{n-1}}{\tau} w, \qquad \forall w \in \mathcal{V}_n^{\star}. \tag{16}$$

In the above equations, the integral terms in $\mathcal{O}(\Gamma^n)$ act as a stabilization term and also define an extension in the neighborhood to the function u^n and v^n, where $\rho_n > 0$ is a stabilization parameter. The choice of δ_n and ρ_n will be given in the next section.

4 The Fully Discrete Method

In this section, We introduce a fully discrete numerical scheme by discretizing the Eqs. (14)–(16). We will use a standard finite element method to approximate the level-set function and a trace finite element method to discretize the surface reaction diffusion system. The discrete scheme is based on an Eulerian framework and enables us to solve the system easily even when the surface changes its topology [28].

4.1 The Finite Element Space in a Narrow Band

Let $\Omega \subset \mathbb{R}^3$ such that $\Gamma(t) \subset \Omega$, for all $t \in (0, T)$. Let $\mathcal{T}_h$ be a regular triangulation of Ω with mesh size h, and W_h^k be the standard conforming kth order finite element space in Ω, i.e.

$$W_h^k := \{v_h \in C(\Omega) : v_h|_K \in P_k, \forall K \in \mathcal{T}_h\}.$$

If ϕ_h^{n-1} is given, we define the discrete surface

$$\Gamma_h^{n-1} := \left\{\mathbf{x} \in \Omega : \phi_h^{n-1}(\mathbf{x}) = 0\right\} = \bigcup_{K \in \mathcal{T}_h^\Gamma} \Gamma_K, \quad \Gamma_K := \Gamma_h^{n-1} \cap K. \tag{17}$$

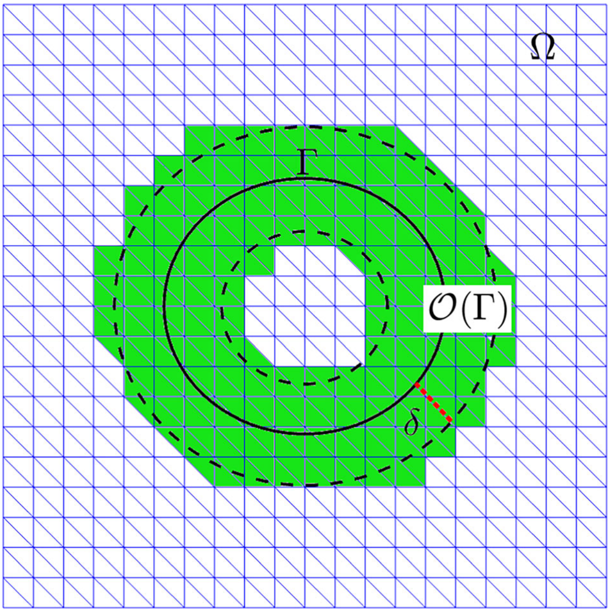

Fig. 1 Definition of a neighboring region around an interface Γ

Here $\mathcal{T}_H^\Gamma := \{K \in \mathcal{T}_h : \Gamma_h^{n-1} \cap K \neq \emptyset\}$. We assume that Γ_h^{n-1} is a good approximation to Γ^{n-1}. Introduce a narrow band region near the surface Γ_h^{n-1},

$$\mathcal{N}_{\delta_{n-1}}(\Gamma_h^{n-1}) := \{\mathbf{x} \in \mathbb{R}^3 \mid \mathrm{dist}(\mathbf{x}, \Gamma_h^{n-1}) < \delta_{n-1}\}. \tag{18}$$

Then a neighborhood of Γ_h^{n-1} consist of all discrete elements intersecting with $\mathcal{N}_{\delta_{n-1}}(\Gamma_h^{n-1})$ is given by

$$\mathcal{S}(\Gamma_h^{n-1}) := \{K \in \mathcal{T}_h : K \cap \mathcal{N}_{\delta_{n-1}}(\Gamma_h^{n-1}) \neq \emptyset\}, \qquad \mathcal{O}(\Gamma_h^{n-1}) := \mathrm{int}(\cup_{K \in \mathcal{S}(\Gamma_h)} K).$$

Figure 1 illustrates the definition of a neighboring region near an interface Γ. Assume that $\Gamma_h^n \subset \mathcal{O}(\Gamma^{n-1})$ always holds for all $n > 1$. We define a finite element space in the neighborhood as

$$V_h^k(\mathcal{O}(\Gamma_h^{n-1})) := \{v_h \in C(\mathcal{O}(\Gamma_h^{n-1})) : v_h|_K \in P_k, \forall K \in \mathcal{S}(\Gamma_h^{n-1})\}.$$

We will introduce the fully discrete problem for the system (14)–(16) below.

4.2 Finite Element Discretization of the Level-Set Equation

We first consider the level-set Eq. (14). We will use a quadratic finite element method to discretize the level set function. We do not choose a linear finite element space since we need to calculate the mean curvature of the interface which is related to the second order derivatives of the level set function. We consider the discretization of Eq. (14) in $V_h^2(\mathcal{O}(\Gamma_h^{n-1}))$. This is to find a piecewisely quadratic continuous function $\varphi_h \in V_h^2(\mathcal{O}(\Gamma_h^{n-1}))$ such that

$$\int_{\mathcal{O}(\Gamma_h^{n-1})} \varphi_h \psi_h + \epsilon\tau \int_{\mathcal{O}(\Gamma_h^{n-1})} \nabla\varphi_h \cdot \nabla\psi_h = \int_{\mathcal{O}(\Gamma_h^{n-1})} \phi_h^{n-1}\psi_h - \lambda\tau \int_{\mathcal{O}(\Gamma_h^{n-1})} u_h^{n-1,e}\psi_h,$$
$$\forall\psi \in V_h^2(\mathcal{O}(\Gamma_h^{n-1})). \tag{19}$$

We then do re-distance to φ_h to compute ϕ_h^n, which satisfies $|\nabla\phi_h^n| \approx 1$ and has (almost) the same zero level-set with φ_h. In applications, the re-distance operation is done in the neighbouring region of Γ_h^n

$$\phi_h^n = \mathrm{redist}(\varphi_h), \qquad \text{in } \mathcal{O}(\Gamma_h^n),$$

by some standard techniques, e.g. the fast marching method ([37, 38]).

We briefly introduce the fast marching method below. We introduce some notations. Notice that both φ_h and ϕ_h^n are continuous P_2 finite element functions. For each tetrahedron $K \in \mathcal{S}(\Gamma_h^n)$, the freedoms are defined on the four vertexes and the middle points of all edges of K. Then we can divide the tetrahedron into four sub-tetrahedra naturally by connecting the middle points. After division for all the tetrahedra in $\mathcal{S}(\Gamma_h^n)$, we obtain a refined mesh $\mathcal{T}_h'$. We denote by $I(\varphi_h)$ the piecewise linear interpolation of φ_h on $\mathcal{T}_h'$. The zero level-set of $I(\varphi_h)$ is set to be Γ_h'. Now we are ready to give the fast marching method. We compute the values of ϕ_h on all the vertexes of $\mathcal{T}_h'$ in two steps: initialization and extension. In the initialization step, we set the value of ϕ_h in a band region $\mathcal{T}_{\Gamma_h'}$ consist of all the tetrahedra intersecting with Γ_h' by directly computing the distance of the vertex to Γ_h'. In the extension step, we set the value of ϕ_h in the outer region by a greedy algorithm. We first extend the approximate distance to the neighbours of $\mathcal{T}_{\Gamma_h'}$ and then to neighbours of neighbours. For more details on the fast marching method for the finite element functions, we refer to Algorithm 7.4.1 in Sect. 7.4 in [38].

4.3 The Trace Finite Element Method for the Surface Reaction Diffusion System

We will use a trace finite element method to discretize the surface reaction diffusion system. For simplicity, we use linear finite element method for both u and v.

Once ϕ_h^n is given, we have $\Gamma_h^n = \{\mathbf{x} : \phi_h^n(\mathbf{x}) = 0\}$. We first compute the mean curvature by

$$H_h^n = \nabla \cdot \frac{G\nabla\phi_h^n}{|G\nabla\phi_h^n|},$$

where $G\nabla\phi_h^n$ is a higher order approximation of $\nabla\phi_h^n$ obtained by applying a gradient recovery technique [39, 40]. We briefly introduce this technique below. Suppose $G\nabla\phi_h^n \in (W_h^2)^3$ is a second order vector finite element function. We need only determine the value of $G\nabla\phi_h^n$ on all the vertexes and the middle points of all edges in $\mathcal{S}(\Gamma_h^n)$. Let x_i be a vertex of a tetrahedron in $\mathcal{S}(\Gamma_h^n)$ and $\omega(x_i) := \{K | x_i \in K\}$ be the neighbouring region of x_i consists of all the tetrahedra which has x_i as a vertex. We fit a polynomial p_{x_i} of degree 3 to the values in all nodes of $\omega(x_i)$ by a least square approach

$$\sum_{z \in \mathcal{N}(\omega(x_i))} |\phi_h(z) - p_{x_i}(z)|^2 = \min_{p \in P_3} \sum_{z \in \mathcal{N}(\omega(x_i))} |\phi_h(z) - p(z)|^2.$$

Here $\mathcal{N}(\omega(x_i))$ consist of all the vertexes and middle points of all edges in $\omega(x_i)$. We assume there is a unique solution p_{x_i}. (Notice that we may include more neighbours into $\omega(x_i)$ if the solution is not unique.) We define $G\nabla\phi_h^n(x_i) = \nabla p_{x_i}(x_i)$. For x_{ij} which is a middle point of an edge with nodes x_i and x_j, we simply set $G\nabla\phi_h^n(x_{ij}) = \frac{1}{2}(\nabla p_{x_i}(x_i) + \nabla p_{x_i}(x_j))$.

Then we can derive a fully discrete problem for (15)–(16) in $V_h^1(\mathcal{O}(\Gamma_h^n))$ by using the trace finite element method [28, 29]. The methods reads as follows.

We first find a $u_h^n \in V_h^1(\mathcal{O}(\Gamma_h^n))$ such that

$$\begin{aligned}&\int_{\Gamma_h^n}\left[\gamma+\frac{1}{\tau}+(\lambda u_h^{n-1}-\epsilon H_h^n)H_h^n-\gamma u_h^{n-1}v_h^{n-1}\right]u_h^n w_h + D_1\int_{\Gamma_h^n}\nabla_\Gamma u_h^n\cdot\nabla_\Gamma w_h\\&+\rho_n\int_{\mathcal{O}(\Gamma_h^n)}(\boldsymbol{n}_h^n\cdot\nabla u_h^n)(\boldsymbol{n}_h^n\cdot\nabla w_h)=a\gamma\int_{\Gamma_h^n}w_h+\int_{\Gamma_h^n}\frac{u_h^{n-1}}{\tau}w_h,\qquad\forall w_h\in\mathcal{V}_n^\star.\qquad(20)\end{aligned}$$

Then we find a discrete function $v_h^n \in V_h^1(\mathcal{O}(\Gamma_h^n))$ such that

$$\begin{aligned}&\int_{\Gamma_h^n}\left[\frac{1}{\tau}+(\lambda u_h^n-\epsilon H_h^n)H_h^n+\gamma(u_h^n)^2\right]v_h^n w_h + D_2\int_{\Gamma_h^n}\nabla_\Gamma v_h^n\cdot\nabla_\Gamma w_h\\&+\rho_n\int_{\mathcal{O}(\Gamma_h^n)}(\boldsymbol{n}_h^n\cdot\nabla v_h^n)(\boldsymbol{n}_h^n\cdot\nabla w_h)=b\gamma\int_{\Gamma_h^n}w_h+\int_{\Gamma_h^n}\frac{v_h^{n-1}}{\tau}w_h,\qquad\forall w_h\in\mathcal{V}_n^\star.\qquad(21)\end{aligned}$$

Here $\boldsymbol{n}_h^n=\nabla\phi_h^n$. Notice that in both equations, we solve the problem in a narrow band region $\mathcal{O}(\Gamma_h^n)$. Most terms are defined on Γ_h^n, where the trace of the finite element space is used to calculate integrals. That is why the method is called the trace finite element method in [15]. Finally, we would like to remark that in implementation we actually calculate the integration in an approximate surface $(\Gamma_h^n)'$, which is the zero level set of $I(\phi_h^n)$. This does not affect the convergence rate for the linear trace finite element method [15, 41].

4.4 The Fully Discrete Algorithm

In summary, we derive a fully decoupled algorithm to solve the mathematical model described in Sect. 2 as follows.

Algorithm 2

Step 0. Given Ω, make a partition to get $\mathcal{T}_h$. Given the initial level set function ϕ^0 that implicitly defines the initial surface, compute its P_2 interpolation ϕ_h^0. Prepare the initial value functions u_h^0 and v_h^0. Set the time step τ and the end time T. Set $n=1$.
Step 1. For given ϕ_h^{n-1}, solve Eq. (19), to get φ_h. Compute $\phi_h^n=\text{redist}(\varphi_h)$ in $\mathcal{O}(\Gamma_h^n)$ by a fast marching method.
Step 2. Solve the reaction-diffusion Eqs. (20)–(21) on $\mathcal{O}(\Gamma_h^n)$ to get u_h^n and v_h^n.
Step 4. Set $n=n+1$. If $n\tau>T$, stop. Otherwise, return **Step 2**.

In Step 2, we need determine the parameters δ_n and ρ_n. We should first choose δ_n large enough so that $\Gamma_h^n\in\mathcal{O}(\Gamma_h^{n-1})$. We estimate the normal velocity by computing $w_\infty := \max(\lambda u_h^{n-1}-\epsilon H_h^{n-1})$ and let $\delta_n=c_0(w_\infty\tau+h)$ where $c_0>1$ is a constant. The choice of ρ_n has been discussed in [29]. In general we can choose $\rho_n=c_1(\delta_n+h)^{-1}$ with c_1 is a constant. Such a choice guarantees both the accuracy and stability of the stabilized trace finite element method. In our numerical experiments, we simply set $c_0=3$ and $c_1=1$.

Remark 1 There are some difficulties to do error estimates to the present numerical method. Firstly, the surface reaction diffusion system is nonlinear and non-monotone. The diffusion coefficients are not equal. This induces instability corresponding to emergence of patterns of the solutions [42]. Secondly, the coupling between the surface reaction diffusion system and the evolution of the surface make the numerical analysis even more difficult. To address these difficulties, we may first consider a reaction-diffusion system on evolving surface with given velocity. Then we consider a simpler coupling between the surface PDEs and the geometric motion of the surface[43]. We will leave the numerical analysis for future work.

5 Numerical Experiments

In this section, we present some numerical experiments obtained by Algorithm 2. We first show some convergence tests for the algorithm and then present some interesting examples on stationary and evolving surfaces.

5.1 Accuracy Check

Since the problem does not have an analytical solution, we consider a simple case in which the solution can be solved numerically in high accuracy. We then use the solution as a reference to compute the numerical errors. Here we take $D_1 = D_2$ which avoid the generation of Turing patterns. Initially, Γ is the spherical surface with radius R_0 and both u_0 and v_0 are constants. In this setup, the solution $u(t)$ and $v(t)$ will always be constant and do not depend on the position of the spherical surface. In this case, the system (2)–(3) and Eq. (1) can be simplified to

$$\begin{cases} \frac{\partial u}{\partial t} + u(\lambda u - \epsilon \frac{2}{R})\frac{2}{R} = f_1(u, v), \\ \frac{\partial v}{\partial t} + v(\lambda u - \epsilon \frac{2}{R})\frac{2}{R} = f_2(u, v), \\ \frac{\partial R}{\partial t} = (\lambda u - \epsilon \frac{2}{R}). \end{cases} \tag{22}$$

This is a system of ordinary differential equations(ODEs) which can be solved in high accuracy by some standard ODE solvers. We solve the ODE system by an implicit Euler scheme under sufficiently small time step and regard the numerical solutions $(u(t), v(t))$ as a reference. Since the solution is homogeneous in space, we test only the convergence behavior with respect to the time step size. For simplicity, we consider only the errors for u, which in L^∞ norm is defined by

$$Err(\Delta t) = \max_{i \in \{1,2,\cdots,T/\Delta t\}} \{\|u_h(x, t_i) - \hat{u}(t_i)\|_{L^\infty(\Gamma_h(t))}\}. \tag{23}$$

Here $t_i = i\Delta t$, $u_h(x, t_i)$ represents the numerical solution at time t_i calculated by Algorithm 2. Δt represents the time step size in numerical experiments. The other parameters as chosen as $D_1 = D_2 = 10$, $u_0 = 2$, $v_0 = 0$, $R_0 = 1$, $\gamma = 10$, $a = 0.1$, $b = 0.9$.

We set the bulk region $\Omega = (-2, 2)^3$ which is discretized uniformly with mesh size $h = 1/16$. We first test the case with a stationary surface by setting $\lambda = \epsilon = 0$. The time evolving of the constant solution $u_h(t)$ and the errors are shown in Fig. 2. We could see that when the time step size become smaller, the numerical solution converges to the reference solution. The convergence rate in L^∞ norm is shown in the left column in Table 1. We can see that it is of the first order convergence with respect to the time step. This is reasonable for the fully decoupled scheme(Algorithm 2), where the backward Euler scheme is used.

We also do numerical tests for the case with evolving surface. The numerical results are shown in Fig. 3. Similarly to the stationary surface case, the numerical solution for u converges to the reference solution for all time $t \in (0, T)$.

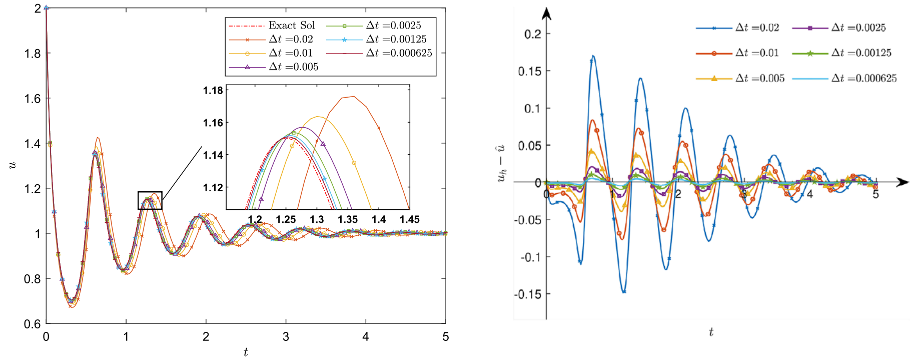

Fig. 2 Numerical solutions of Eq. (22) on a stable surface ($\lambda = 0, \epsilon = 0$) compared to exact solutions (left) and the difference between numerical and exact solutions (right) at different time steps. The termination time is $T = 5$

Table 1 Convergence rate of numerical results for Eq. (22) on stable and evolving surfaces

	Stationary surface			Evolving surface	
Δt	Err	Rate	Δt	Err	Rate
0.02	0.170	–	0.02	0.180	–
0.01	0.084	1.019	0.01	0.095	0.92
0.005	0.042	1.015	0.005	0.053	0.84
0.0025	0.021	1.010	0.0025	0.030	0.83
0.00125	0.010	1.005	0.00125	0.017	0.805

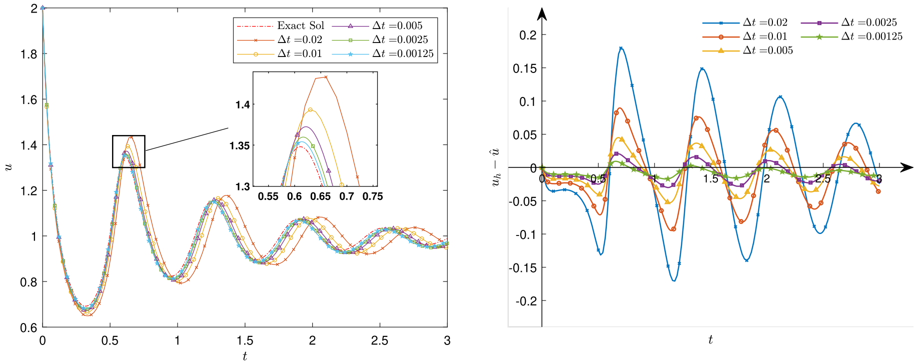

Fig. 3 Numerical solutions of Eq. (22) to exact one on evolving surfaces ($\lambda = 0.05, \epsilon = 0.005, \hat{R}(T) = 1.135$) (left) and the difference between the numerical solution and the exact one (right) at different time steps. The termination time of the solution $T = 3$

The errors and the convergence rate are shown in the right column in Table 1. We can see that it is close to the first order convergence with respect to the time step. This is reasonable for the fully decoupled scheme(Algorithm 2), where the backward Euler scheme is used for the reaction diffusion system and a diffusion generated method is used for evolution of the surface. The convergence rate is slightly worse for the evolving surface than the stationary surfaces.

Table 2 Convergence rate with respect to the mesh size for the system (24)–(25)

h	Stationary surface Err	Rate	h	Evolving surface Err	Rate
0.25	0.0379650	–	0.5	0.124676	–
0.125	0.0175934	1.11	0.25	0.0642842	0.96
0.0625	0.00592568	1.57	0.125	0.0203089	1.66
0.03125	0.00126699	2.23	0.0625	0.0101490	1.00

To further test the convergence of the method with respect to the spacial mesh size, we consider a different problem. We set $\lambda = 0$ and the motion of the surface is given by a mean curvature flow $v_n = H$. Suppose initially the surface is a sphere with radius R_0. Then the surface will be spherical with surface $R(t)$ determined by

$$\frac{dR}{dt} = \frac{-2\epsilon}{R}.$$

The solution is given by $R(t) = \sqrt{R_0^2 - 4\epsilon t}$. In this case, the original system (2)–(3) is reduced to

$$\begin{aligned}\frac{\partial u}{\partial t} - \frac{4\epsilon}{R^2}u &= D_1\Delta_\Gamma u + f_1(u, v),\\ \frac{\partial v}{\partial t} - \frac{4\epsilon}{R^2}v &= D_2\Delta_\Gamma v + f_1(u, v).\end{aligned}$$

In this setup, it is still not easy to find an analytical solution to the problem. Hence we slightly change the problem by adding extra right hand side terms to the problem. Suppose that $(u, v) = (\tilde{u}, \tilde{v})$ are the solutions of

$$\frac{\partial u}{\partial t} - \frac{4\epsilon}{R^2}u = D_1\Delta_\Gamma u + f_1(u, v) + F_1(x, y, t), \tag{24}$$

$$\frac{\partial v}{\partial t} - \frac{4\epsilon}{R^2}v = D_2\Delta_\Gamma v + f_1(u, v) + F_2(x, y, t). \tag{25}$$

Here we set $F_1(x, y, t) = \frac{\partial \tilde{u}}{\partial t} - \frac{4\epsilon}{R^2}\tilde{u} - D_1\Delta_\Gamma\tilde{u} - f_1(\tilde{u}, \tilde{v})$ and $F_2(x, y, t) = \frac{\partial \tilde{v}}{\partial t} - \frac{4\epsilon}{R^2}\tilde{v} - D_1\Delta_\Gamma\tilde{v} - f_1(\tilde{u}, \tilde{v})$. In our numerical tests, we set $\tilde{u} = x$ and $\tilde{v} = y$. The parameters are chosen as $R_0 = 1$, $D_1 = D_2 = 1$, $\epsilon = 0.05$ and $T = 0.036$. In the formulae of f_1 and f_2, we set $\gamma = 1$, $a = 0.1$ and $b = 0.1$. We set $\Delta t = 5 \times 10^{-4}$ so that the time step is small enough. In Table 2, we show the errors for u in L^2 norm. We test two cases with evolving surfaces or stationary surfaces($\epsilon = 0$). It seems that the convergence rate is about $O(h)$ for the evolving surface case and about $O(h^2)$ for the stationary case. It seems that the convergence rate is sub-optimal in the evolving surface case. The might be due to the fact that the mean curavature is not computed accurately enough in the present method.

5.2 Turing Pattern on Stationary Surfaces

In this subsection, we present some interesting numerical examples for stationary surfaces. For our experiments, we have chosen parameters $D_1 = 1$, $D_2 = 10$, $a = 0.1$, $b = 0.9$, $u_0 = 1$, and $w_0 = 0.9$. It is worth noting that D_1 and D_2 are not equal, and this difference in diffusion constants may induce instability in the reaction-diffusion system, leading to the

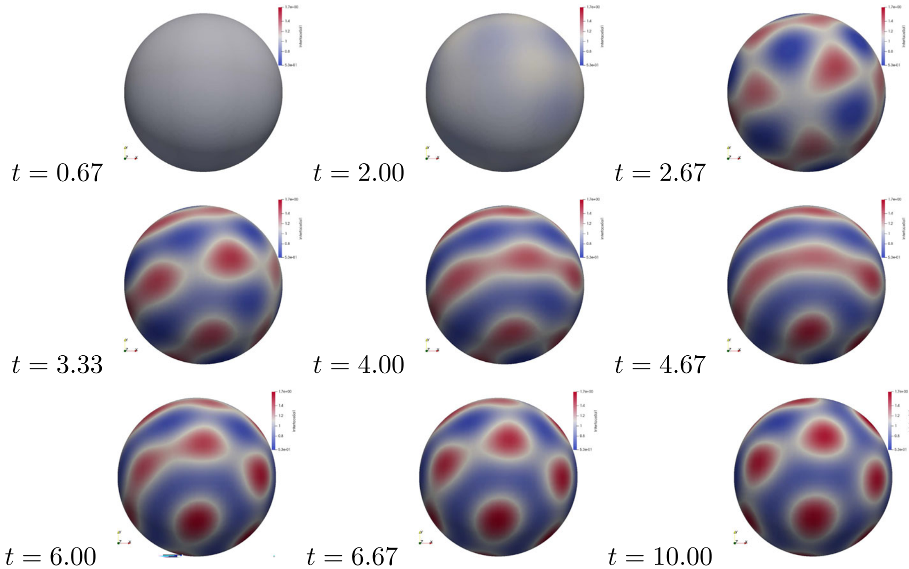

Fig. 4 Pattern formation on stable unit sphere. Parameters:$\gamma = 200$, $\Delta t = 0.001$

emergence of Turing patterns. Additionally, we set $\lambda = \epsilon = 0$ so that the surface remains stationary and does not change with time. The choice of other parameters may differ for different numerical examples.

Example 1. We first simulate the formation of a Turing pattern on a stationary unit sphere. The numerical results are displayed in Fig. 4. The level-set function of the unit sphere is expressed as:

$$\Phi(\mathbf{x}) = |\mathbf{x}|_2 - 1.$$

The bulk domain is $\Omega = (-2, 2)^3$, which is discretized uniformly with a mesh size of $h = 1/32$. Initially, the solution is almost constant. However, due to the instability induced by cross-diffusion and reaction, the solutions are not stable, and patterns gradually emerge over time. In the final states, we observe a dotted pattern for u. The numerical results are consistent with previous simulations, such as those in [32].

Example 2. The setup of the experiment is almost the same as in the previous example. We change the shape of the substrate and consider a dumbbell-shaped surface, which is given by the following level-set function,

$$\Phi_{dumbbell}(\mathbf{x}) = \left(\frac{25x_1^2}{4} + \frac{441x_2^2}{25} + \frac{25x_3^2}{4} + \frac{9}{10}\right)^2 - 64x_2^2 - \frac{13}{10}.$$

The numerical examples are displayed in Fig. 5. As observed, the numerical results are similar to those obtained on a spherical surface. The constant functions are unstable, and patterns gradually emerge over time. However, it appears that the patterns emerge later than those observed on the unit sphere since we have chosen a smaller γ for these experiments.

Example 3. Our third example involves a stationary “tooth"-like surface represented by the level-set function given by:

$$\Phi_{tooth}(\mathbf{x}) = 256x_1^4 - 16x_1^2 + 256x_2^4 - 16x_2^2 + 256x_3^4 - 16x_3^2.$$

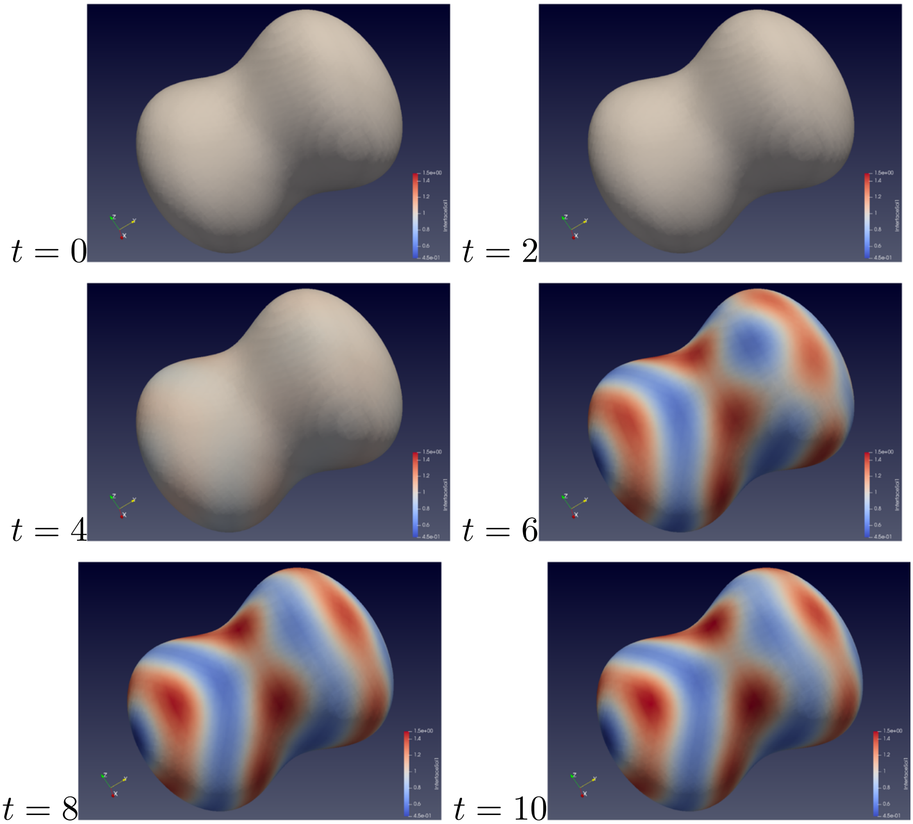

Fig. 5 Pattern formation on stationary “dumbell" surface. Parameters: $\gamma = 100$, $\Delta t = 0.01$

We will use the same parameters as in Example 2. The shape of the surface and the numerical results are displayed in Fig. 6, which clearly shows the emergence of Turing patterns on the surface.

5.3 Numerical Simulations for Evolving Surfaces

We will now consider the fully coupled system (1)–(3), which we will solve using Algorithm 2. We will use some parameters that were chosen for the stationary surface, i.e., $D_1 = 1$, $D_2 = 10$, $a = 0.1$, $b = 0.9$, $u_0 = 1$ and $w_0 = 0.9$. Additionally, we will set $\lambda = 0.1$ and $\epsilon = 0.01$ to ensure that the surface evolves over time. We have selected ϵ to be much smaller than λ to ensure that the concentration of the activator u mainly controls the surface's normal velocity. We will conduct experiments for various examples.

Example 4. We begin by considering a spherical surface with an initial radius of $R_0 = 1$, and the numerical results are displayed in Fig. 7. As observed, Turing patterns gradually emerge due to the instability of the reaction-diffusion system. In the early stages, the surface undergoes minimal changes. However, as the Turing pattern emerges, the concentration of u increases, leading to faster growth of the surface in areas where the density of u is higher. Consequently, the surface's shape is no longer spherical, and the enclosed volume increases

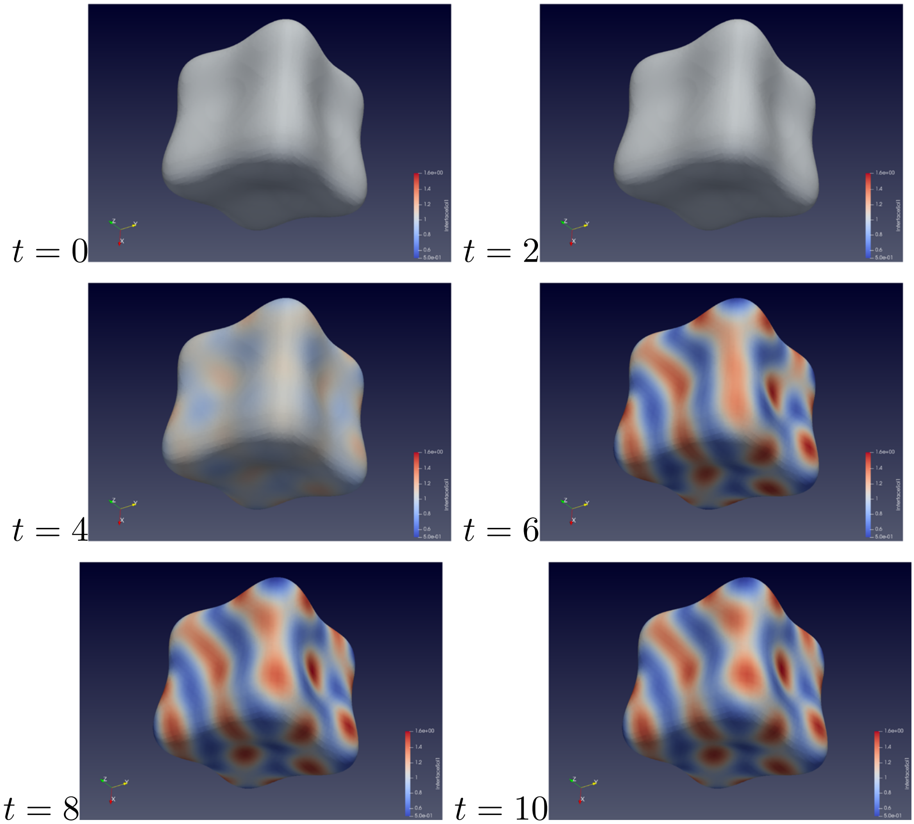

Fig. 6 Pattern formation on stationary “tooth" surface. Parameters: $\gamma = 100$, $\Delta t = 0.01$

dramatically in later stages, while the pattern remains similar. These numerical results are consistent with those presented in [32], which used a similar approach to model solid tumor growth. In comparison with the method used in [32], the advantage of our method is that we can deal with problems with topology changes easily, while this is difficult for the evolving finite element method. This will be shown in the next example. We would also like to remark that the degree of freedoms in our method are usually larger than that in the evolving FEM with the same mesh size h. However, the solution of the surface-reaction systems are of the same order since our method is defined in a narrow band region with width of order h.

Example 5. In the last example, we examine the evolving surface problem with an initial surface represented by a level-set function

$$\Phi_{donut}(\mathbf{x}) = \sqrt{\left(\sqrt{x_1^2 + x_2^2} - 1\right)^2 + x_3^2} - \frac{3}{5}.$$

The initial surface has a “donut" shape. The other parameters used in this example are similar to those in the previous one. Figure 8 displays the numerical results, which indicate that the phenomena observed are similar to those with an initial spherical surface. However, the difference is that topology changes occur as the surface expands. This topology change can not be computed directly by the previous Lagrangian type numerical methods, where

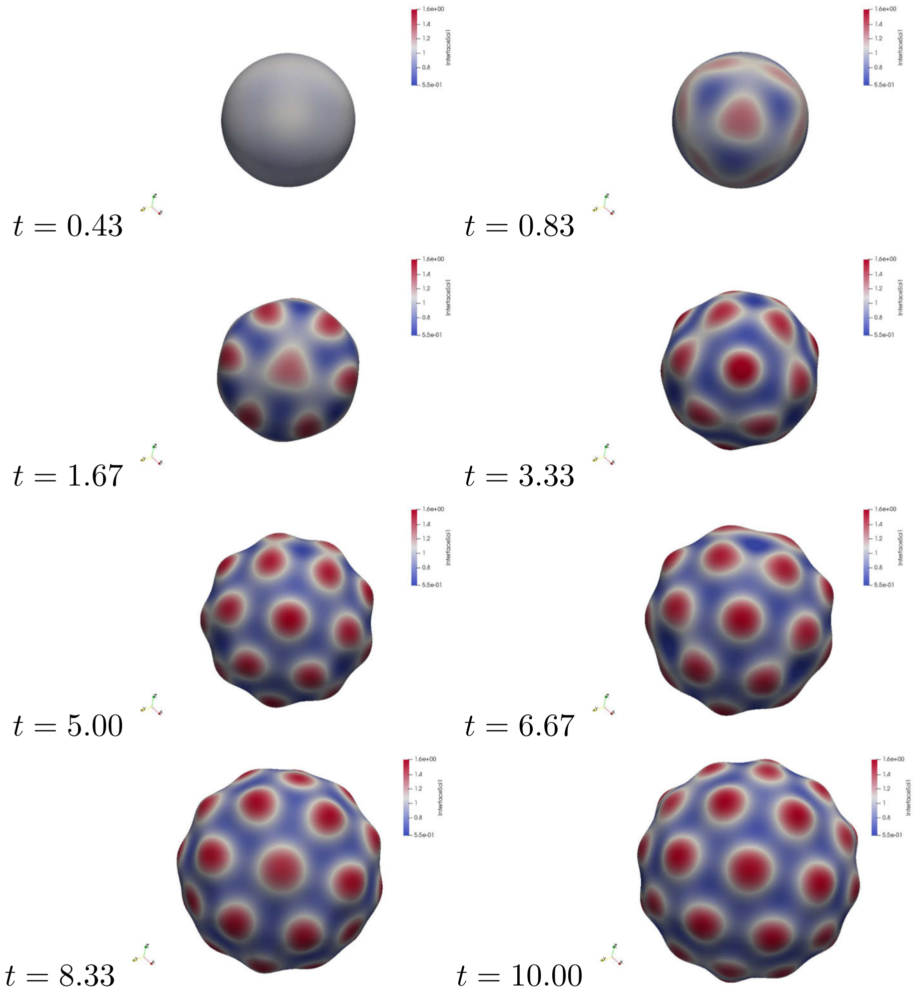

Fig. 7 The evolution and pattern formation of a surface that is initially a unit sphere. Parameters: $\gamma = 200$, $\Delta t = 0.005$

the surface triangulation moves in accordance with the surface motion. Nevertheless, our algorithm, which utilizes an Eulerian type method, where the surface is represented by a level-set function and its evolution is obtained by a diffusion-generated numerical scheme, is not affected by the topology changes, as demonstrated by the numerical results.

6 Conclusion

We have developed a numerical method for a surface reaction-diffusion system that is coupled with the evolution of the surface. Our approach uses the trace finite element method to discretize the surface partial differential equations and applies a diffusion-generated method for the evolution of the surface, which is represented by a level-set function. Our numerical

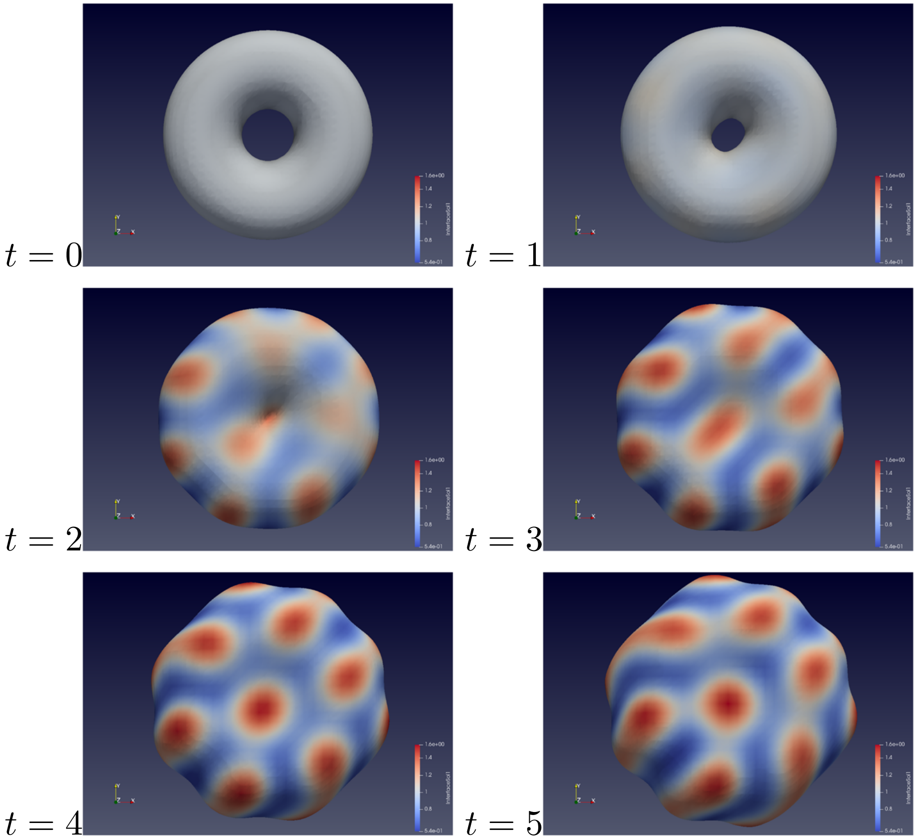

Fig. 8 Pattern formation on evolving initial “donut" surface. Parameters: $\gamma = 100$, $\Delta t = 0.01$

scheme is fully decoupled, and we only need to solve some linear equations in a narrow band region in each time step. Numerical simulations have shown that the method has an first-order convergence rate with respect to the time step size and second order convergence with respect to the spacial mesh size. We have also presented several examples that demonstrate some key features of the algorithm. Firstly, both the surface reaction-diffusion system and the level-set functions are discretized on the same bulk mesh. The coupling between them can be handled without any difficulty, and there is no need to do interpolation among different meshes. Secondly, the algorithm is based on an Eulerian framework, which makes it easy to deal with the large deformation and topology changes of the surface. The diffusion-generated method for the evolution of the surface avoids solving a nonlinear geometric PDE for the surface, making the method more efficient.

In the future, we plan to perform numerical analysis on the proposed algorithm. We will study the stability and error estimates of the fully discrete algorithm. It would also be interesting to develop some higher-order numerical schemes for the coupled system. For that purpose, we should consider higher order trace-finite element method [39, 44, 45] and higher order diffusion generated method [46]. With these methods, we can simulate some interesting phenomena on dynamic propagation for Turing patterns on surfaces [47].

Acknowledgements We thank Maxim A. Olshanskii for helpful discussions. We are also grateful to the anonymous referees whose comments have served to greatly improve various parts of the paper.

Funding This work is partially supported by the National Natural Science Foundation of China (11971469 and 12371415) and by Beijing Natural Science Foundation (Grant No. Z240001).

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work.

References

1. Turing, A.M.: The chemical basis of morphogenesis. Bull. Math. Biol. **52**(1), 153–197 (1990)
2. Meinhardt, H.: The Algorithmic Beauty of Sea Shells. Springer, Cham (2009)
3. Sanderson, A.R., Kirby, R.M., Johnson, C.R., Yang, L.: Advanced reaction-diffusion models for texture synthesis. J. Graph. Tools **11**(3), 47–71 (2006)
4. Sardet, C., Roegiers, F., Dumollard, R., Rouviere, C., McDougall, A.: Calcium waves and oscillations in eggs. Biophys. Chem. **72**(1–2), 131–140 (1998)
5. Watanabe, M., Iwashita, M., Ishii, M., Kurachi, Y., Kawakami, A., Kondo, S., Okada, N.: Spot pattern of leopard danio is caused by mutation in the zebrafish connexin41.8 gene. EMBO Rep. **7**(9), 893–897 (2006)
6. Economou, A.D., Ohazama, A., Porntaveetus, T., Sharpe, P.T., Kondo, S., Basson, M.A., Gritli-Linde, A., Cobourne, M.T., Green, J.: Periodic stripe formation by a turing mechanism operating at growth zones in the mammalian palate. Nat. Genet. **44**(3), 348–351 (2012)
7. Murray, J.D.: Mathematical Biology: I & II. Springer, Cham (2002)
8. Gierer, A., Meinhardt, H.: A theory of biological pattern formation. Kybernetik **12**(1), 30–39 (1972)
9. Schnakenberg, J.: Simple chemical reaction systems with limit cycle behaviour. J. Theor. Biol. **81**(3), 389–400 (1979)
10. Chaplain, M.A., Ganesh, M., Graham, I.G.: Spatio-temporal pattern formation on spherical surfaces: numerical simulation and application to solid tumour growth. J. Math. Biol. **42**(5), 387–423 (2001)
11. Dziuk, G., Elliott, C.M.: Finite element methods for surface PDEs. Acta Numer. **22**, 289 (2013)
12. Bonito, A., Demlow, A., Nochetto, R.H.: Finite element methods for the Laplace–Beltrami operator. In: Handbook of Numerical Analysis, vol. 21, pp. 1–103. Elsevier, Amsterdam (2020)
13. Xu, J.-J., Zhao, H.-K.: An Eulerian formulation for solving partial differential equations along a moving interface. J. Sci. Comput. **19**, 573–594 (2003)
14. Du, Q., Gunzburger, M.D., Ju, L.: Voronoi-based finite volume methods, optimal Voronoi meshes, and PDEs on the sphere. Comput. Methods Appl. Mech. Eng. **192**(35–36), 3933–3957 (2003)
15. Olshanskii, M.A., Reusken, A., Grande, J.: A finite element method for elliptic equations on surfaces. SIAM J. Numer. Anal. **47**(5), 3339–3358 (2009)
16. Leung, S., Lowengrub, J., Zhao, H.: A grid based particle method for solving partial differential equations on evolving surfaces and modeling high order geometrical motion. J. Comput. Phys. **230**(7), 2540–2561 (2011)
17. Liang, J., Zhao, H.: Solving partial differential equations on point clouds. SIAM J. Sci. Comput. **35**(3), A1461–A1486 (2013)
18. Dedè, L., Quarteroni, A.: Isogeometric analysis for second order partial differential equations on surfaces. Comput. Methods Appl. Mech. Eng. **284**, 807–834 (2015)
19. Li, Z., Shi, Z.: A convergent point integral method for isotropic elliptic equations on a point cloud. Multiscale Model. Simul. **14**(2), 874–905 (2016)
20. MacDonald, G., Mackenzie, J.A., Nolan, M., Insall, R.H.: A computational method for the coupled solution of reaction-diffusion equations on evolving domains and manifolds: Application to a model of cell migration and chemotaxis. J. Comput. Phys. **309**, 207–226 (2016)
21. Lehto, E., Shankar, V., Wright, G.B.: A radial basis function (RBF) compact finite difference (FD) scheme for reaction-diffusion equations on surfaces. SIAM J. Sci. Comput. **39**(5), A2129–A2151 (2017)
22. Bachini, E., Farthing, M.W., Putti, M.: Intrinsic finite element method for advection-diffusion-reaction equations on surfaces. J. Comput. Phys. **424**, 109827 (2021)

23. Dziuk, G., Elliott, C.M.: Finite elements on evolving surfaces. IMA J. Numer. Anal. **27**(2), 262–292 (2007)
24. Dziuk, Gerhard, Elliott, Charles M.: Surface finite elements for parabolic equations. J. Comput. Math. **25**, 385–407 (2007)
25. Elliott, C.M., Stinner, B., Venkataraman, C.: Modelling cell motility and chemotaxis with evolving surface finite elements. J. R. Soc. Interface **9**(76), 3027–3044 (2012)
26. Elliott, C.M., Styles, V.: An ALE ESFEM for solving PDEs on evolving surfaces. Milan J. Math. **80**(2), 469–501 (2012)
27. Olshanskii, M.A., Reusken, A., Xu, X.: An Eulerian space-time finite element method for diffusion problems on evolving surfaces. SIAM J. Numer. Anal. **52**(3), 1354–1377 (2014)
28. Olshanskii, M.A., Xu, X.: A trace finite element method for PDEs on evolving surfaces. SIAM J. Sci. Comput. **39**(4), A1301–A1319 (2017)
29. Lehrenfeld, C., Olshanskii, M.A., Xu, X.: A stabilized trace finite element method for partial differential equations on evolving surfaces. SIAM J. Numer. Anal. **56**(3), 1643–1672 (2018)
30. Frittelli, M., Madzvamuse, A., Sgura, I., Venkataraman, C.: Lumped finite elements for reaction-cross-diffusion systems on stationary surfaces. Comput. Math. Appl. **74**(12), 3008–3023 (2017)
31. Frittelli, M., Madzvamuse, A., Sgura, I., Venkataraman, C.: Preserving invariance properties of reaction-diffusion systems on stationary surfaces. IMA J. Numer. Anal. **39**(1), 235–270 (2019)
32. Barreira, R., Elliott, C.M., Madzvamuse, A.: The surface finite element method for pattern formation on evolving biological surfaces. J. Math. Biol. **63**(6), 1095–1119 (2011)
33. Olshanskii, M., Xu, X., Yushutin, V.: A finite element method for Allen-Cahn equation on deforming surface. Comput. Math. Appl. **90**, 148–158 (2021)
34. Merriman, B., Bence, J.K., Osher, S.J.: Motion of multiple junctions: a level set approach. J. Comput. Phys. **112**(2), 334–363 (1994)
35. Esedoglu, S., Ruuth, S., Tsai, R.: Diffusion generated motion using signed distance functions. J. Comput. Phys. **229**(4), 1017–1042 (2010)
36. Lu, S., Xu, X.: An efficient diffusion generated motion method for wetting dynamics. J. Comput. Phys. **441**, 110476 (2021)
37. Sethian, J.A.: Level Set Methods and Fast Marching Methods. Cambridge University Press, Cambridge (1999)
38. Gross, S., Reusken, A.: Numerical Methods for Two-Phase Incompressible Flows, vol. 40. Springer, Cham (2011)
39. Grande, J., Reusken, A.: A higher order finite element method for partial differential equations on surfaces. SIAM J. Numer. Anal. **54**(1), 388–414 (2016)
40. Zhang, Z., Naga, A.: A new finite element gradient recovery method: superconvergence property. SIAM J. Sci. Comput.ng **26**(4), 1192–1213 (2005)
41. Lu, S., Xu, X.: Numerical investigations on trace finite element methods for the Laplace-Beltrami eigenvalue problem. J. Sci. Comput. **97**(1), 12 (2023)
42. Chrysafinos, K., Karatzas, E.N., Kostas, D.: Stability and error estimates of fully discrete schemes for the Brusselator system. SIAM J. Numer. Anal. **57**(2), 828–853 (2019)
43. Kovács, B., Li, B., Lubich, C., Power, G., Christian, A.: Convergence of finite elements on an evolving surface driven by diffusion on the surface. Numer. Math. **137**, 643–689 (2017)
44. Grande, J.R., Lehrenfeld, C., Reusken, A.: Analysis of a high-order trace finite element method for PDEs on level set surfaces. SIAM J. Numer. Anal. **56**(1), 228–255 (2018)
45. Larson, M.G., Zahedi, S.: Stabilization of high order cut finite element methods on surfaces. IMA J. Numer. Anal. **40**(3), 1702–1745 (2020)
46. Esedoglu, S., Guo, J.: A monotone, second order accurate scheme for curvature motion. SIAM J. Numer. Anal. **60**(5), 2435–2447 (2022)
47. Nishide, R., Ishihara, S.: Pattern propagation driven by surface curvature. Phys. Rev. Lett. **128**(22), 224101 (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.