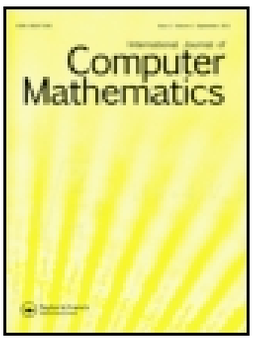

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: https://www.tandfonline.com/loi/gcom20

Numerical studies of a class of reaction–diffusion equations with Stefan conditions

Shuang Liu, Yihong Du & Xinfeng Liu

To cite this article: Shuang Liu, Yihong Du & Xinfeng Liu (2019): Numerical studies of a class of reaction–diffusion equations with Stefan conditions, International Journal of Computer Mathematics, DOI: 10.1080/00207160.2019.1599868

To link to this article: https://doi.org/10.1080/00207160.2019.1599868

Accepted author version posted online: 25 Mar 2019.
Published online: 04 Apr 2019.

Submit your article to this journal

Article views: 95

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=gcom20

INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS
https://doi.org/10.1080/00207160.2019.1599868

Taylor & Francis
Taylor & Francis Group

ARTICLE

Check for updates

Numerical studies of a class of reaction–diffusion equations with Stefan conditions

Shuang Liu[a], Yihong Du[b] and Xinfeng Liu[a]

[a]Department of Mathematics, University of South Carolina, Columbia, SC, USA; [b]School of Science and Technology, University of New England, Armidale, Australia

ABSTRACT

It is always very difficult to efficiently and accurately solve a system of differential equations coupled with moving free boundaries, while such a system has been widely applied to describe many physical/biological phenomena such as the dynamics of spreading population. The main purpose of this paper is to introduce efficient numerical methods within a general framework for solving such systems with moving free boundaries. The major numerical challenge is to track the moving free boundaries, especially for high spatial dimensions. To overcome this, a front tracking framework coupled with implicit solver is first introduced for the 2D model with radial symmetry. For the general 2D model, a level set approach is employed to more efficiently treat complicated topological changes. The accuracy and order of convergence for the proposed methods are discussed, and the numerical simulations agree well with theoretical results.

ARTICLE HISTORY
Received 9 May 2018
Revised 14 September 2018
Accepted 30 December 2018

KEYWORDS
Reaction–diffusion equations; free boundaries; level set method; front tracking method; diffusive logistic models

2010 MATHEMATICS SUBJECT CLASSIFICATION
65M06

1. Introduction

In this paper, we consider reaction–diffusion equations over a changing domain of the form

$$\frac{\partial U}{\partial t} - D\Delta U = f(U) \quad \text{for } \mathbf{x} \in \Omega(t),\ t > 0; \qquad U = 0 \quad \text{for } \mathbf{x} \in \partial\,\Omega(t),\ t > 0. \tag{1}$$

The nonlinear function $f(U)$ is assumed to be a C^1 function satisfying $f(0) = 0$, and in the literature, it is often taken to be the logistic function $f(U) = U(a - bU)$ with positive constants a and b. In the rest of this paper, we will take this logistic function as an example to demonstrate the numerical methods.

The evolution of the moving domain $\Omega(t) \subset \mathbb{R}^N$ or rather its boundary $\partial\Omega(t)$ is determined by the one phase Stefan condition which, in the case $\partial\Omega(t)$ is a C^1 manifold in $\mathbb{R}^N$, can be described as follows:

Any point $\mathbf{x} \in \partial\Omega(t)$ moves with velocity $\mu|\nabla_{\mathbf{x}} U(t, \mathbf{x})|\mathbf{n}(\mathbf{x})$, where $\mathbf{n}(\mathbf{x})$ is the unit outward normal of $\Omega(t)$ at $\mathbf{x}$, and μ is a given positive constant.

The moving boundary $\partial\Omega(t)$ is generally called the 'free boundary', and it is well known that, in general, its smoothness is not guaranteed, even if the initial function $u(0, \mathbf{x})$ and initial domain $\Omega(0)$ are both smooth; see, for example [10], where a weak solution setting is introduced for the general

CONTACT Xinfeng Liu xfliu@math.sc.edu Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

© 2019 Informa UK Limited, trading as Taylor & Francis Group

situation of this free boundary problem. Such a free boundary problem was first introduced in [11] in one space dimension, as a model for the spreading of an invading or new species with population density $U(t,\mathbf{x})$ whose spreading front is represented by the free boundary.

In [12], the regularity and long-time behaviour of $\partial\Omega(t)$ and $u(t,\mathbf{x})$ are studied, and it is shown that a spreading-vanishing dichotomy holds: either $\Omega(t)$ stays bounded (i.e. is contained in some fixed ball in $\mathbb{R}^N$) for all $t>0$, and in such a case $U(t,\mathbf{x})\to 0$ as $t\to\infty$ uniformly in $x\in\Omega(t)$, or $\Omega(t)$ converges to $\mathbb{R}^N$, with $\partial\Omega(t)$ approximating a moving sphere enlarging to infinity as $t\to\infty$. Moreover, in the latter case, for all large t, the free boundary $\partial\Omega(t)$ is a smooth closed manifold without boundary.

If $f(U)\equiv 0$, then this problem reduces to the classical Stefan problem, which has been extensively studied theoretically (see, e.g. [3] and the references therein). Other theoretical studies of related free boundary problems can be found in [2] and the references therein.

In contrast, very few numerical methods have been developed to solve such free boundary problems. In general, it is always difficult to efficiently and accurately handle the moving boundaries. To efficiently handle the moving boundaries, level set methods [13,25,29,30,34,35] and front tracking methods [19,27,33,36] are two popular numerical approaches. One distinct feature of front tracking [8,15–17,20,32] is using a pure Lagrangian approach to explicitly track locations of interfaces, but it is difficult to handle topological bifurcations in high dimensions, while the level set method can efficiently overcome such difficulties. The level set method has been successfully applied to solve the classical Stefan problem [4,6,7,13,14,24,26] and the references therein. In this paper, we will introduce a front-tracking framework and a front-fixing framework to solve system (3)–(6) for a 2D model with radial symmetry, and a level set approach is employed for the general 2D model.

It is not easy to check the accuracy of the level set method. In this paper, we do it by applying the level set method to a 2D problem with radial symmetry, for which it is also possible to use the front-tracking method. Our numerical test shows that the numerical results obtained by the two methods agree well. The accuracy of the front-tracking method is checked and compared with the front-fixing method for 2D radially symmetric models, which indicates that they are reliably accurate numerical schemes. In addition, our numerical simulations correlate nicely with theoretical results.

The rest of the paper is organized in the following way. In Section 2, the front-tracking approach and front-fixing approach are introduced separately for a two-dimensional case with radial symmetry (3)–(6), and the two methods are also compared with each other. In Section 3, a level set method is discussed for a more general two-dimensional case. In Section 4, numerical examples are performed to show the efficiency, accuracy and consistency for these different approaches. Finally, a brief conclusion is drawn in Section 5.

2. Numerical methods for a 2D model with radial symmetry

To solve the 2D diffusive logistic model in polar coordinates, the system can be written as

$$\frac{\partial U}{\partial t}-D\left(\frac{\partial^2 U}{\partial r^2}+\frac{1}{r}\frac{\partial U}{\partial r}+\frac{1}{r^2}\frac{\partial^2 U}{\partial^2\theta}\right)=U(a-bU),\quad t>0,\ \ 0\le\theta\le 2\pi,\ \ r>0, \tag{2}$$

where $(r\cos\theta, r\sin\theta)\in\Omega(t)$.

We assume that the environment and the solution are radially symmetric, i.e. we set the initial domain Ω_0 as a disk, the initial function $U_0(x)$ as radially symmetric, the moving boundary $\partial\Omega(t)$ as thus a circle whose radius we denote by $H(t)$ and the solution $U(t,r,\theta)=U(t,r)$, the 2D diffusive logistic model with radial symmetry can be written as a 1D diffusive logistic model

$$\frac{\partial U}{\partial t}-D\left(\frac{\partial^2 U}{\partial r^2}+\frac{1}{r}\frac{\partial U}{\partial r}\right)=U(a-bU),\quad t>0,\ \ 0<r<H(t). \tag{3}$$

together with the boundary conditions

$$\frac{\partial U}{\partial r}(t,0)=0, \quad U(t,H(t))=0, \ \ t>0, \tag{4}$$

the Stefan condition

$$H'(t)=-\mu\frac{\partial U}{\partial r}(t,H(t)), \quad t>0, \tag{5}$$

and the initial conditions

$$H(0)=H_0, \quad U(0,r)=U_0(r), \quad 0\le r\le H(0). \tag{6}$$

2.1. Front-fixing method for the 2D diffusive logistic model with radial symmetry

Here we extend the ideas of Piqueras *et al.* [28] for 1D case to the 2D system with radial symmetry. Let us transform the 1D diffusive logistic model (3)–(6) into a problem with a fixed domain $[0,1]$. Under the Landau transformation [9,18]

$$z(t,r)=\frac{r}{H(t)}, \quad W(t,z)=U(t,r), \tag{7}$$

moving front problem (3)–(6) reduces to

$$G(t)\frac{\partial W}{\partial t}-D\frac{\partial^2 W}{\partial z^2}-\left(\frac{D}{z}+\frac{zG'(t)}{2}\right)\frac{\partial W}{\partial z}=G(t)W(a-bW), \quad t>0, \ \ 0<z<1, \tag{8}$$

where

$$G(t)=H^2(t), \quad t\ge 0. \tag{9}$$

Boundary conditions (4) and Stefan condition (5) take the form:

$$\frac{\partial W}{\partial z}(t,0)=0, \quad W(t,1)=0, \quad t>0, \tag{10}$$

and

$$G'(t)=-2\mu\frac{\partial W}{\partial z}(t,1), \quad t>0, \tag{11}$$

respectively, while initial conditions (6) become

$$G(0)=H_0^2, \quad W(0,z)=W_0(z)=U_0(zH_0), \quad 0\le z\le 1. \tag{12}$$

Conditions (6) for the initial function $U_0(r)$ are translated to $W_0(z)$ as follows:

$$W_0(z)\in C^2([0,1]), \quad W_0'(0)=W_0(1)=0, \quad W_0(z)>0, \ \ 0\le z<1. \tag{13}$$

After the transformation, the new problem is to solve nonlinear parabolic partial differential equations (8) in the unbounded fixed domain $(0,\infty)\times[0,1]$ for the variables (t,z). Let us consider the step size discretization $k=\triangle t$, $h=\triangle z=1/M$, and the mesh points (t^n,z_j), with $t^n=kn, n\ge$

$0, z_j = jh, 0 \le j \le M$ and M is the total number of the subintervals of $[0,1]$. Let us denote the approximate value of $W(t^n, z_j)$ at the mesh point (t^n, z_j),

$$w_j^n \approx W(t^n, z_j), \tag{14}$$

and let g^n be the approximation of $G(t^n)$. Let us consider the forward approximation of the time derivatives,

$$\frac{w_j^{n+1} - w_j^n}{k} \approx \frac{\partial W}{\partial t}(t^n, z_j), \quad \frac{g^{n+1} - g^n}{k} \approx G'(t^n), \tag{15}$$

and the central approximation of the spatial derivatives,

$$\frac{w_{j+1}^n - w_{j-1}^n}{2h} \approx \frac{\partial W}{\partial z}(t^n, z_j), \quad \frac{w_{j-1}^n - 2w_j^n + w_{j+1}^n}{h^2} \approx \frac{\partial^2 W}{\partial z^2}(t^n, z_j). \tag{16}$$

From (15) and (16), Equation (8) is approximated by

$$\begin{aligned} g^n \frac{w_j^{n+1} - w_j^n}{k} - D\frac{w_{j-1}^n - 2w_j^n + w_{j+1}^n}{h^2} - \left(\frac{D}{z_j} + \frac{z_j}{2}\frac{g^{n+1} - g^n}{k}\right)\frac{w_{j+1}^n - w_{j-1}^n}{2h} \\ = g^n w_j^n (a - b w_j^n), \quad n \ge 0, \ \ 0 < j \le M-1. \end{aligned} \tag{17}$$

For the point at $j=0$, the value w_{-1}^n is eliminated from the second-order discretization of boundary condition (10) and (13),

$$\frac{w_1^n - w_{-1}^n}{2h} = 0, \quad w_M^n = 0, \ \ n \ge 0. \tag{18}$$

Transformed Stefan condition (11) is discretized using first-order forward approximation for $G'(t)$ and three points backward spatial approximation of $\frac{\partial W}{\partial z}(t, 1)$:

$$\frac{g^{n+1} - g^n}{k} = -\frac{\mu}{h}(3w_M^n - 4w_{M-1}^n + w_{M-2}^n), \quad n \ge 0. \tag{19}$$

to preserve accuracy of order $O(k) + O(h^2)$.

Finally, we have

$$\begin{aligned} w_0^{n+1} &= \left(1 - \frac{2Dk}{g^n h^2} + k(a - b w_0^n)\right) w_0^n + 2\frac{Dk}{g^n h^2} w_1^n, \\ w_j^{n+1} &= a_j^n w_{j-1}^n + b_j^n w_j^n + c_j^n w_{j+1}^n, \quad n \ge 0, \ \ 0 < j \le M-1, \\ w_M^{n+1} &= 0. \end{aligned} \tag{20}$$

where the coefficients are given by

$$\begin{aligned} a_j^n &= \frac{Dk}{g^n h^2} - \frac{Dk}{2hz_j g^n} - \frac{z_j \mu k(4w_{M-1}^n - w_{M-2}^n)}{4h^2 g^n}, \\ b_j^n &= 1 - \frac{2Dk}{g^n h^2} + k(a - b w_j^n), \\ c_j^n &= \frac{Dk}{g^n h^2} + \frac{Dk}{2hz_j g^n} + \frac{z_j \mu k(4w_{M-1}^n - w_{M-2}^n)}{4h^2 g^n}. \end{aligned} \tag{21}$$

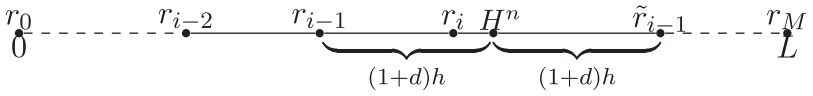

Figure 1. Case 1. $r_i \le H^n < r_{i+1}$.

2.2. Front-tracking method for the 2D model with radial symmetry

Let us consider moving front problem (3)–(6) in a fixed domain $[0, T] \times [0, L]$, i.e. we want to find the distribution of the population in the region $[0, L]$ up to time T. Set the step size discretization $h = \triangle r = L/M$, $k = \triangle t = 0.1(\triangle r)^2/D$, which satisfies the CFL condition, and the mesh points (t^n, r_i), with $t^n = kn, n \ge 0, r_i = ih, 0 \le i \le M$ and M is the total number of subintervals of $[0, L]$. Let us denote the approximate value of $U(t^n, r_i)$ at the mesh point (t^n, r_i) by

$$u_i^n \approx U(t^n, r_i), \tag{22}$$

and let H^n be the approximation of $H(t^n)$.

Step 1: Track the position of the moving front.

According to the Stefan condition (5), we use the central approximation of the spatial derivatives to approximate $\frac{\partial U}{\partial r}(t, H(t))$:

1. When $r_i \le H^n < r_{i+1}, i = 2, 3 \ldots M-1$ as shown in Figure 1, denoting $d = \frac{H^n - r_i}{h}$, we first consider the symmetric point of r_{i-1} with respect to the position H^n, which is denoted by $\tilde{r}_{i-1}$. Specifically when $H^n = r_i, \tilde{r}_{i-1} = r_{i+1}$. We use the Lagrange extrapolation method to construct a polynomial P^L from the value of $d, h, u_{i-2}^n, u_{i-1}^n$ and H^n [14]. At $\tilde{r}_{i-1}$, we use the value of P^L at $\tilde{r}_{i-1}$ instead of $u(\tilde{r}_{i-1})$,

$$\frac{\partial U}{\partial r}(t^n, H^n) \approx \frac{P^L(\tilde{r}_{i-1}) - u_{i-1}^n}{2(1+d)h}, \quad i = 2, 3, \ldots, M-1. \tag{23}$$

Remark: The most challenging part of the front tracking method is the evaluation of $\frac{\partial U}{\partial r}(t, H(t))$, where $H(t)$ is the moving boundary. It is difficult to find a uniform finite difference approximation of $\frac{\partial U}{\partial r}(t, H(t))$ with high order accuracy because the distance of the moving point H^n to the set of grid points $\{r_i\}$ does not have a uniform positive lower bound. The numerical methods of evaluating $\frac{\partial U}{\partial r}(t, H(t))$ in such a case need to be carefully designed to avoid singularity caused by H^n being very close to some grid point r_i. Evaluating $\frac{\partial U}{\partial r}(t, H(t))$ by combining the evaluation of $\frac{\partial U}{\partial r}(t, r_i)$ and $\frac{\partial U}{\partial r}(t, r_{i+1})$ can avoid such singularity; however, it destroys the accuracy when combined with the process of updating $U(t, r)$. In (23), we combine the classical central approximation of the spatial derivatives and the Lagrange extrapolation method to evaluate $\frac{\partial U}{\partial r}(t, H(t))$ to ensure second-order accuracy in space. For example, when $r_i < H^n < r_{i+1}$, we evaluate $\frac{\partial U}{\partial r}(t, H(t))$ on r_{i-1} and $\tilde{r}_{i-1}$ instead of r_i and $\tilde{r}_i$ to avoid singularity when H^n is very close to r_i, and when $H^n = r_i$, (23) becomes

$$\frac{\partial U}{\partial r}(t^n, H^n) \approx \frac{P^L(r_{i+1}) - u_{i-1}^n}{2h}, \quad i = 2, 3, \ldots, M-1.$$

2. When $r_0 < H^n \le r_1$, the central approximation of the spatial derivatives to approximate $\frac{\partial U}{\partial r}(t, H(t))$ involves the fictitious value u_{-1}^n at the point $(t^n, -h)$. The value u_{-1}^n is eliminated from the discretization of boundary condition (4),

$$\frac{u_1^n - u_{-1}^n}{2h} = 0$$

which means that $u_{-1}^n = u_1^n = 0$, we can see that all the values of u_i^n on the grid points are equal to 0 except u_0^n. The simulation should stop here indicating that a more refined mesh is needed.

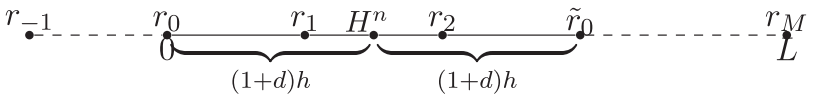

Figure 2. Case 3: $r_1 < H^n < r_2$.

3. When $r_1 < H^n < r_2$ as depicted in Figure 2, denoting $d = \frac{H^n - r_1}{h}$. Let us first consider the symmetric point of r_0 with respect to the position H^n, which is denoted by $\tilde{r}_0$. Then we consider the value of $u^n_{-1} = u^n_1$ and use the Lagrange extrapolation method to construct polynomial P^L from the value of h, d, u^n_{-1}, u^n_0 and H^n. At $\tilde{r}_0$, we use the value of P^L at $\tilde{r}_0$ instead of $u(\tilde{r}_0)$,

4. When $H^n = r_M$, it means that the spreading of the populations goes out of the computational domain $[0, L]$, and the simulation should stop here indicating for a larger computational domain.

Remark: In system (3)–(6), cases 2 and 3 will not happen, since the front $H(t)$ is increasing in time t. However, the front tracking method possesses preferable adaptability to track various moving front conditions, such as those used in [1], where the front need not be increasing in time, and therefore cases 2 and 3 indeed occur.

Step 2: Update the value of $U(t^{n+1}, r_i)$.

1. When $r_i = H^{n+1}$, we know that $U(t^{n+1}, r_i) = 0$. Let $u^{n+1}_j = 0$, for $j = i, i+1, \ldots M$. We consider the central approximation of the spatial derivatives at x_j, for $j = 0, 1, 2, \ldots, i-1$, where U is updated by the backward Euler method in time

$$\begin{aligned}
&\frac{u^{n+1}_0 - u^n_0}{k} = D\frac{2u^{n+1}_1 - 2u^{n+1}_0}{h^2} + au^{n+1}_0 - b(u^{n+1}_0)^2,\\
&\frac{u^{n+1}_j - u^n_j}{k} = D\left(\frac{u^{n+1}_{j-1} - 2u^{n+1}_j + u^{n+1}_{j+1}}{h^2} + \frac{u^{n+1}_{j+1} - u^{n+1}_{j-1}}{2jh^2}\right) + au^{n+1}_j - b(u^{n+1}_j)^2, \quad j = 1, \ldots i-1.
\end{aligned} \tag{24}$$

Then we use the *Picard* Iteration (or Newton Iteration) to solve nonlinear system (24).

2. When $r_i < H^{n+1} < r_{i+1}$, denoting $R = \frac{H^{n+1} - r_i}{h}$. Let $u^{n+1}_j = 0$, for $j = i+1, \ldots M$. We consider the central approximation of the spatial derivatives at x_j, for $j = 0, 1, 2, \ldots, i$. For updating u^{n+1}_i, we use the Lagrange extrapolation to construct polynomial P^L from the value of h, R, u^{n+1}_{i-2}, u^{n+1}_{i-1} and H^{n+1}. At r_{i+1}, we use the value of P^L at r_{i+1} instead of u^{n+1}_{i+1}. U is updated by the backward Euler method in time

$$\begin{aligned}
&\frac{u^{n+1}_0 - u^n_0}{k} = D\frac{2u^{n+1}_1 - 2u^{n+1}_0}{h^2} + au^{n+1}_0 - b(u^{n+1}_0)^2,\\
&\frac{u^{n+1}_j - u^n_j}{k} = D\left(\frac{u^{n+1}_{j-1} - 2u^{n+1}_j + u^{n+1}_{j+1}}{h^2} + \frac{u^{n+1}_{j+1} - u^{n+1}_{j-1}}{2jh^2}\right) + au^{n+1}_j - b(u^{n+1}_j)^2, \quad j = 1, \ldots i-1,\\
&\frac{u^{n+1}_i - u^n_i}{k} = D\left(\frac{u^{n+1}_{i-1} - 2u^{n+1}_i + P^L(x_{i+1})}{h^2} + \frac{P^L(x_{i+1}) - u^{n+1}_{i-1}}{2ih^2}\right) + au^{n+1}_i - b(u^{n+1}_i)^2.
\end{aligned} \tag{25}$$

Picard Iteration (or Newton Iteration) will be applied to solve nonlinear system (25).

3. Level set method for the general 2D diffusive logistic model

The general 2D diffusive logistic model for the density of population of the invasive species $U(t,x,y)$ depending on time t and spatial variables (x,y) has the form

$$\begin{aligned}
&\frac{\partial U}{\partial t} - D\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) = U(a - bU), \quad t > 0, \ \ (x,y) \in \Omega(t),\\
&U(t, \partial\Omega(t)) = 0, \quad t > 0,\\
&\mathbf{v}(t,x,y) = \mu|\nabla U(t,x,y)|\,\mathbf{n}(t,x,y) = -\mu\nabla U(t,x,y), \quad t > 0, \ \ (x,y) \in \partial\Omega(t),\\
&\Omega(0) = \Omega_0, \quad U(0,x,y) = U_0(x,y), \ \ (x,y) \in \Omega_0.
\end{aligned} \tag{26}$$

where $\mathbf{v}(t,x,y)$ and $\mathbf{n}(t,x,y)$ are, respectively, the velocity vector of the boundary point $(x,y) \in \partial\Omega(t)$, and the unit outward normal of $\Omega(t)$ at $(x,y) \in \partial\Omega(t)$. The initial function $U_0(x,y)$ is assumed to have the following properties:

$$U_0 \in C^2(\Omega_0), \quad U_0 > 0 \text{ in } \Omega_0, \quad U_0 = 0 \text{ on } \partial\Omega_0. \tag{27}$$

In what follows, to simplify notations, we will use $\tau(t)$ to denote the unknown moving boundary $\partial\Omega(t)$. The density of population is distributed in the domain $\Omega(t), D > 0$ is the dispersal rate and the positive parameters a and b are the intrinsic growth rate and the intra-specific competition rate, respectively. The parameter $\mu > 0$ is the proportionality constant between the population gradient at the front and the speed of the moving boundary.

Following the ideas of Chen *et al.* [6] and Fedkiw and Osher [13], we construct a level set function ϕ, then move ϕ with the correct speed $\mathbf{v}$ at the front and followed by updating $u(t,x,y)$. The new position of the front is stored implicitly in ϕ. We extend the level set approach to effectively capture the front at each new time step and a finite difference discretization of five-point stencils coupled with a forward Euler scheme to solve the system everywhere away from the front. The inter-extrapolation strategy and boundary conditions will be employed for the points near fronts. With this approach, we avoid the difficulties that arise from explicitly tracking the front and thus increase the efficiency to deal with complex interfacial geometries.

Step 1: Construct level set equation $\phi(t,x,y)$ and velocity function $V(t,x,y)$.

We introduce a level set function ϕ. Initially, ϕ is set to equal to the signed distance function from the front as follows:

$$\phi(0,x,y) = \begin{cases} +d, & x \in \Omega_0^c,\\ 0, & x \in \tau_0,\\ -d & x \in \Omega_0, \end{cases} \tag{28}$$

where d is the distance from the front.

We want to construct a speed function $V(t,x,y)$ over the whole computational domain, which governs the motion of ϕ by

$$\phi_t + V|\nabla\phi| = 0. \tag{29}$$

The basic idea behind introducing the level set function $\phi(t,x,y)$ is that the front is equal to the zero level set of ϕ at any time, i.e.

$$\tau(t) = \{(x,y) \in \Omega(t) : \quad \phi(t,x,y) = 0\}.$$

As the front moves at the velocity field $\mathbf{v}$, we construct the speed function $V(t,x,y)$ over the whole computational domain in the following way:

The set $\tau_1(t) := \{(x,y) : \phi(t,x,y) = 0\}$ coincides with $\tau(t)$ for all $t > 0$, or equivalently (29) yields the same equation for the velocity vector $\mathbf{v_1}(t,x,y)$ at $(x,y) \in \tau_1(t)$ whenever $\tau_1(t)$ coincides with $\tau(t)$ at some $t \geq 0$ (note that they coincide at time $t = 0$ by assumption).

Indeed, if $\tau_1(t) = \tau(t)$, from $\phi(t, \tau_1(t)) = 0$ and $\phi(t, x, y) < 0$ for (x, y) lying inside $\tau_1(t)$, we deduce

$$\phi_t + \nabla\phi \cdot \mathbf{v_1} = 0, \quad \mathbf{n} = \nabla\phi/|\nabla\phi| \quad \text{for } (x, y) \in \tau_1(t) = \tau(t), \tag{30}$$

and $\mathbf{v_1}$ has the same direction as $\mathbf{n}$, the unit outward normal of $\tau_1(t) = \tau(t)$, i.e.

$$\mathbf{v_1} = V_1\,\mathbf{n} \text{ for some } V_1 = V_1(t, x, y) > 0, \quad (x, y) \in \tau_1(t) = \tau(t).$$

These relations yield

$$\phi_t + V_1\,|\nabla\phi| = 0 \text{ on } \tau(t).$$

Combining this with (29) we obtain

$$V_1 = V \text{ for } (x, y) \in \tau(t).$$

Therefore

$$\mathbf{v_1} = V\,\mathbf{n} \text{ for } (x, y) \in \tau(t).$$

By the Stefan condition, we have

$$\mathbf{v} = V\,\mathbf{n} \text{ for } (x, y) \in \tau(t),$$

and thus, we have proved

$$\mathbf{v_1} = \mathbf{v} \text{ for } (x, y) \in \tau(t)$$

as wanted.

Therefore, we get the velocity function over the computational domain

$$V(t, x, y) = \mu|\nabla U(t, x, y)|, \tag{31}$$

which of course moves ϕ with the correct speed at the front, so that $\tau(t)$ will always coincide with the zero level set of ϕ at time t.

Step 2: Update $\phi(t, x, y)$.

According to (29)–(31), the equation governing the level set function turns into

$$\phi_t = \mu\nabla U(t, x, y) \cdot \nabla\phi. \tag{32}$$

The approximation to ∇U at $\tau(t)$ is based upon approximations to the derivatives of U in four coordinate directions to cut down on grid orientation effects; here we use the standard x,y Cartesian coordinates and the 45^o-rotated coordinates η and ζ. We extend each approximation to a derivative of U away from the front by the following four advection equations:

$$u^1_t + S(\phi\phi_x)u^1_x = 0, \tag{33}$$

$$u^2_t + S(\phi\phi_y)u^2_y = 0, \tag{34}$$

$$u^3_t + S(\phi\phi_\eta)u^3_\eta = 0, \tag{35}$$

$$u^4_t + S(\phi\phi_\zeta)u^4_\zeta = 0, \tag{36}$$

where $u^1 = \partial U/\partial x$, $u^2 = \partial U/\partial y$, $u^3 = \partial U/\partial\eta$ and $u^4 = \partial U/\partial\zeta$ on $\tau(t)$. Here S is equal to the sign function.

Equation (33) through Equation (36) continuously extend u^1, u^2, u^3, u^4 away from the front by advecting these fields in the proper upwind direction, and they are used to define V away from $\tau(t)$. Since ϕ is zero on $\tau(t)$, these equations will not degrade the value of V on the front.

Choosing the computational domain as a square box $[-\frac{L}{2}, \frac{L}{2}] \times [-\frac{L}{2}, \frac{L}{2}]$, we discretize the domain by setting $\triangle x = \triangle y = h$. The time step taken in the following sections is $\triangle t$, which satisfy the CFL condition. Let $u_{i,j}^n \approx U(n\triangle t, -\frac{L}{2} + (i-1)h, -\frac{L}{2} + (j-1)h), \phi_{i,j}^n \approx \phi(n\triangle t, -\frac{L}{2} + (i-1)h, -\frac{L}{2} + (j-1)h)$. We use a first-order upwind scheme to discretize (33)–(36). For example, the discretization of (33) is as follows:

$$\text{if } S_{i,j}(\phi\phi_x) > 0, \text{ then } u_{i,j}^{1(new)} = u_{i,j}^{1(old)} - cfl * (u_{i,j}^{1(old)} - u_{i-1,j}^{1(old)}),$$
$$\text{if } S_{i,j}(\phi\phi_x) < 0, \text{ then } u_{i,j}^{1(new)} = u_{i,j}^{1(old)} + cfl * (u_{i+1,j}^{1(old)} - u_{i,j}^{1(old)}),$$

with $cfl = 0.5$.

Here, the time step of this discretization satisfies $\triangle t_{advect}/h \leq 1$. According to [31], the time step of the advection function $\triangle t_{extend}$ is not necessarily related to the main time step $\triangle t$.

From (32), we end up solving for the right-hand side of the equation

$$\phi_t = \frac{\mu}{2}(u_{i,j}^1(\phi_x)_{i,j} + u_{i,j}^2(\phi_y)_{i,j} + u_{i,j}^3(\phi_\eta)_{i,j} + u_{i,j}^4(\phi_\zeta)_{i,j}), \tag{37}$$

where spatial first derivatives of ϕ are approximated by a second-order ENO scheme. We update ϕ by solving (37) with a third-order Runge-Kutta scheme [6].

Step 3: Reinitialize ϕ to be a signed distance function for one time step.

The level set function will cease to be an exact distance function even after one time step. In order to keep the accuracy of $\mathbf{n}$ and V , we need to avoid having steep or flat gradients developed in ϕ. To avoid these numerical difficulties, a good choice is to re-initialize the level set function to be an exact distance function from the evolving front $\tau(t)$ at each time step.

We use the reinitialization scheme of Sussman *et al.* [31] to reinitialize ϕ by

$$\phi_t = S(\phi_0)(1 - |\nabla\phi|), \tag{38}$$

where $\phi(0, x, y) = \phi_0(x, y)$ and S again denotes the sign function. The sign function S is smoothed by the equation

$$S_\varepsilon(\phi_0) = \frac{\phi_0}{\sqrt{{\phi_0}^2 + \varepsilon^2}} \tag{39}$$

to avoid numerical difficulties while implemented [31].

By iterating Equation (38) to a steady state, the original position of the front will not change, but at points away from $\tau(t)$, ϕ will be evolved into a distance function.

Step 4: Update $U(t, x, y)$.

After reinitializing ϕ to be very close to an exact signed distance function from $\tau(t)$ in *Step 3*, next we update $U(t, x, y)$ in the following three cases:

- For points away from the front, which means the nearby four grid points are all inside the domain $\Omega(t)$, we solve the Reaction–diffusion equation by combining with the forward Euler method and five-point stencil scheme.For example, suppose we update $U(t, x, y)$ at the grid point (i, j), where $\phi_{i,j} < 0, \phi_{i+1,j} < 0, \phi_{i-1,j} < 0, \phi_{i,j+1} < 0$ and $\phi_{i,j-1} < 0$, we update $U(t, x, y)$ at the grid point (i, j) as
$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\triangle t} - D\frac{u_{i+1,j}^n + u_{i-1,j}^n - 4u_{i,j}^n + u_{i,j-1}^n + u_{i,j+1}^n}{h^2} = u_{i,j}^n(a - bu_{i,j}^n). \tag{40}$$
- For points near the front $\tau(t)$, some special care should be taken. We employ an interpolation scheme to approximate the spatial double derivative of U. Since ϕ is an exact distance function after reinitialization, we can effectively capture the front by using the level set function ϕ. For

example, we use one-sided different sign of ϕ to incorporate the distances between a point on the front and grid points neighbouring it in either the vertical or horizontal direction.Suppose $X_f = (x_f, -\frac{L}{2} + (j-1)h) \in \tau(t)$ for some integer j, we consider the two grid points (i, j) and $(i+1, j)$ which border X_f in x-direction, i.e. $x_i \le x_f \le x_{i+1}$. Assuming $\phi_{i,j} < 0$, $\phi_{i-1,j} < 0$, $\phi_{i,j-1} < 0$, $\phi_{i,j+1} < 0$ and $\phi_{i+1,j} > 0$, we introduce

$$x_f - x_i = rh = -\frac{\phi_{i,j}}{\phi_{i+1,j} - \phi_{i,j}} h$$

and use $u^n_{i,j}, u^n_{i-1,j}, u^n_{i-2,j}, r$ and $U(n\triangle t, x_f, -\frac{L}{2} + (j-1)h) = 0$ to construct interpolating polynomial P. When updating $u^{n+1}_{i,j}$, once again we use a standard five-point stencil combining with the forward Euler method, here we replace $u^n_{i+1,j}$ by $P(-\frac{L}{2} + ih)$, i.e.

$$\frac{u^{n+1}_{i,j} - u^n_{i,j}}{\triangle t} - D\frac{u^n_{i-1,j} + u^n_{i,j-1} - 4u^n_{i,j} + P(-\frac{L}{2} + ih) + u^n_{i,j+1}}{h^2} = u^n_{i,j}(a - bu^n_{i,j}). \tag{41}$$

For the case when the front interacts with y-axis, we use the same process in the y-direction. In special case, where we cannot find enough grid points inside the domain to construct interpolating polynomial P, we employ the nearby gird points and intersect points of the front and x- and y-axes to construct quadratic polynomial or straight line as the interpolating polynomial P to update U. For the extreme configuration, where there are only intersect points of the front and x- and y-axes near the grid point, we update $U = 0$ at the grid point.

- If a grid point lies on the front, we set the value $U = 0$ at that point according to the boundary condition.

Step 5: Repeat Step 2 through Step 5 to update ϕ and U for the next time step.

4. Numerical experiments

4.1. Numerical tests of a 2D problem with radial symmetry: front-fixing method and front-tracking method

4.1.1. Verification of convergence rates

For the convergence rates, we compare the numerical approximation to the exact solution. Let's take the convergence of u in space, for example. However, the numerical approximation depends on the choice of the grid size (h). For instance, we denote the numerical approximation by $\tilde{u}_h$. If the numerical method is of order p, it means that there is a number C independent of h such that

$$|\tilde{u}_h - u| \le Ch^p,$$

at least for sufficiently small h. Often the error $|\tilde{u}_h - u|$ depends smoothly on h. Then, we have

$$|\tilde{u}_h - u| = Ch^p + O(h^{p+1}).$$

To evaluate the convergence order p, we need to check the sequence

$$\log|\tilde{u}_h - u| = \log|C| + p\log(h) + O(h),$$

for $h_1, h_2, \ldots$, and fit it to a linear function of $\log(h)$. A standard way to calculate p is to divide h by half every time and look at the ratios of the errors $|u - \tilde{u}_h|$ and $|u - \tilde{u}_{h/2}|$, i.e.

$$\frac{|\tilde{u}_h - u|}{|\tilde{u}_{h/2} - u|} = \frac{Ch^p + O(h^{p+1})}{C(h/2)^p + O((h/2)^{p+1})} = 2^p + O(h).$$

Table 1. Convergence analysis for the front-fixing method for the 2D model with radial symmetry.

$N_x \times N_t$	L_2error	Order	L_∞error	Order
Accuracy test of U of the front-fixing method				
21×5e4	6.20e−03		8.70e−03	
41×5e4	1.50e−03	2.01	2.20e−03	2.00
81×5e4	4.00e−04	2.07	5.00e−04	2.07
161×5e4	1.01e−04	2.32	1.00e−04	2.32
321×5e4	Reference			
Accuracy test of H of the front-fixing method				
21×5e4	1.10e−03		1.90e−03	
41×5e4	3.00e−04	2.00	5e−05	1.98
81×5e4	1.00e−04	2.05	1e−05	2.05
161×5e4	2.01e−05	2.31	2.02e−06	2.31
321×5e4	Reference			

Table 2. Convergence analysis for the front-tracking method for the 2D model with radial symmetry.

$N_x \times N_t$	L_2error	Order	L_∞error	Order
Accuracy test of U of the front-tracking method				
71×2e04	6.50e−04		2.71e−03	
141×2e04	1.42e−04	2.19	5.96e−04	2.19
281×2e04	3.24e−05	2.14	1.35e−04	2.14
561×2e04	6.27e−06	2.37	2.61e−05	2.37
1121×2e04	Reference			
Accuracy test of H of the front-tracking method				
71×2e04	3.02e−02		5.01e−03	
141×2e04	6.75e−03	2.16	1.07e−03	2.23
281×2e04	1.54e−03	2.14	2.42e−04	2.14
561×2e04	3.01e−04	2.35	4.67e−05	2.37
1121×2e04	Reference			

Hence

$$\log_2 \left| \frac{\tilde{u}_h - u}{\tilde{u}_{h/2} - u} \right| = p + O(h).$$

4.1.2. Convergence test of front-fixing method

We test the front-fixing method for solving the 2D logistic diffusion model with radial symmetry (2)–(6) with parameters $(D, \mu, a, b, H_0) = (0.4, 1, 1, 1, 1)$ and $U_0 = \cos(\frac{\pi r}{2})$.

In Table 1, the error (both L_2 and L_∞) and the order of accuracy in space of the front-fixing method are examined, with final time $T = 0.5$. The error is computed by the difference of the numerical solution with the exact solution. For all the examples below when the exact solution is not given, the solution with a fine resolution will be considered as reference or 'exact' solution. As expected, a second-order convergence in space can be observed.

4.1.3. Convergence test of the front-tracking method

We consider the 2D logistic diffusion model with radial symmetry (2)–(6) with parameters $(D, \mu, a, b, H_0) = (0.4, 10, 1, 1, 0.5)$ and $U_0 = \cos(\frac{\pi r}{2})$. The system is used to test the front-tracking method.

In Table 2, the error (both L_2 and L_∞) and the order of convergence in space to the solution of the front-tracking method is examined, with final time $T = 0.1$. Again second-order convergence in space can be observed.

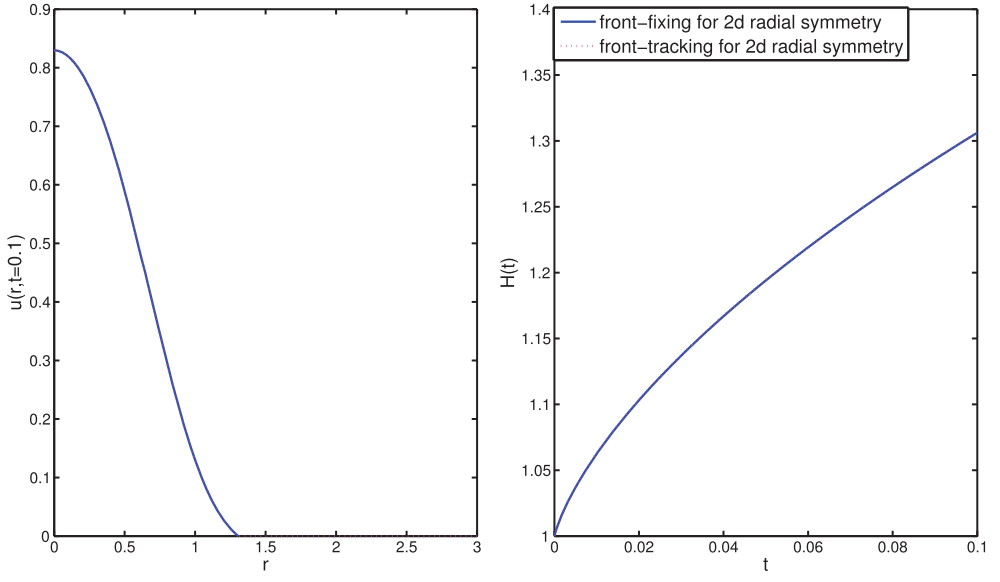

Figure 3. Front-tracking method vs. front-fixing method for the 2D model with radial symmetry.

4.1.4. The comparison of front-tracking with front-fixing for the 2D model with radial symmetry

In this section, we use the front-tracking method and front-fixing method to simulate the 2D logistic diffusion model with radial symmetry (2)–(6) with parameters $(D, \mu, a, b, H_0) = (0.4, 10, 1, 1, 1)$, $U_0 = \cos(\frac{\pi r}{2})$ and spatial size $h = 0.00625$. Figure 3 reveals that front-tracking method matches well with the front-fixing method for the 2D logistic diffusion model with radial symmetry (2)–(6). The numerical results here agree with the theoretical ones in [10], where it is shown that $H(t)$ is an increasing function and for large t, $H(t)$ behaves like a linear function c^*t for some positive constant c^*.

4.2. Numerical tests of the 2D model with the level set method

4.2.1. Convergence of the level set method for the 2D model with radial symmetry

Here we study 2D logistic diffusion model (26) by using the level set approach with parameter $(D, \mu, a, b) = (0.4, 10, 1, 1)$; τ_0 is a circle with radius 1 and $U_0 = 4\cos(\pi \frac{\sqrt{x^2+y^2}}{2})$.

For the boundaries of the species, we use the dotted curve to show the simulated boundary of the species, the solid circle is introduced to describe to what degree the boundary evolves like a circle. The radius of the solid circle is the average distance between the intersect points of $\tau(t)$ with x-axis and y-axis on the boundary and the origin, i.e.

$$r = \frac{\sum \sqrt{x^2 + y^2}}{\#of(x, y)}$$

where $(x, y) \in \tau(t)$ are all the intersect points of $\tau(t)$ with x-axis and y-axis.

According to [11], the solution of Equation (2)–(6) is unique and radially symmetric. Figure 4 shows the evolution of $U(t, x, y)$ and $\tau(t)$, where we can see that the solid circle matches exactly with the dotted curve, which means that the boundary $\tau(t)$ keeps the geometry. And it can be easily observed that $U(t, x, y)$ has radial symmetry as U_0.

We focus on the radius of the boundary $\tau(t)$, which we denote by $H(t)$. $U(t, r) = U(t, x, y)$ is used to learn about the order of accuracy in space of the level set method.

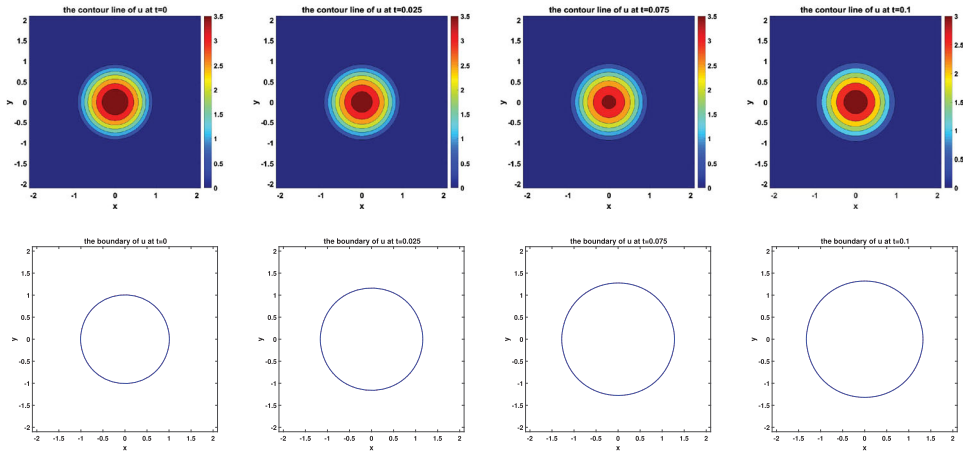

Figure 4. Disk.

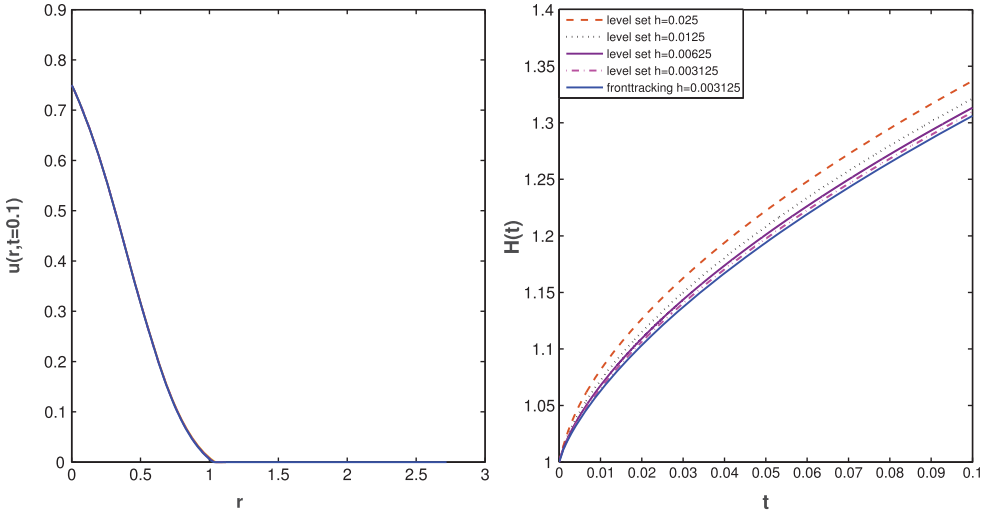

Figure 5. The convergence of the level set method for the 2D model with radial symmetry.

The convergence test for the solution of $u(r)$ at $T = 0.1$ and the front $H(t)$ can be observed from Figure 5 with different space sizes $h = 0.025$, $h = 0.0125$, $h = 0.00625$, $h = 0.003125$, and the results are compared to the results of front tracking method with the same initial setup and step size $h = 0.003125$. Figure 5 shows that the results of the level set method agree very nicely with the results of the front tracking method, which means the three methods are consistent with each other.

In Table 3, the error (both L_2 and L_∞) and the order of convergence to the solution of the level set method are examined, with final time $T = 0.1$. It reveals that the convergence orders for both the solution u and the front $H(t)$ are between 1 and 2.

4.3. Numerical tests of level set methods for the 2D model with different initial configuration

Example 4.1: In the 2D logistic diffusion model (26) with parameters $(D, \mu, a, b) = (4, 10, 1, 1)$, the initial boundary τ_0 is set to be an equilateral triangle which centres at the origin point $(0, 0)$ with

Table 3. Convergence analysis for the level set method.

$N_x \times N_y \times N_t$	L_2error	Order	L_∞error	Order
Accuracy test of U of the level set method				
29×29×160	5.58e−03		9.29e−03	
57×57×640	3.06e−03	0.86	5.01e−03	0.89
113×113×2560	1.40e−03	1.13	2.26e−03	1.15
225×225×10240	4.84e−04	1.54	7.79e−04	1.54
449×449×40960	Reference			
Accuracy test of H of the level set method				
29×29×160	4.19e−02		5.84e−02	
57×57×640	2.01e−02	1.06	2.76e−02	1.08
113×113×2560	8.70e−03	1.20	1.19e−02	1.22
225×225×10240	2.91e−03	1.57	3.92e−03	1.60
449×449×40960	Reference			

side-length 1. The initial value $u_0(x, y)$ and the initial level set function $\phi_0(x, y)$ are set as follows:

$$u_0(x,y) = \begin{cases} 400(\frac{\sqrt{3}}{2} - \frac{1}{\sqrt{3}} + y)(\sqrt{3}x - y + \frac{1}{\sqrt{3}})(-\sqrt{3}x - y + \frac{1}{\sqrt{3}}), & (x,y) \in \Omega_0, \\ 0 & (x,y) \in \Omega_0^c. \end{cases} \tag{42}$$

For $(x, y) \in \overline{\Omega}_0$, we set

$$\phi_0(x,y) = \begin{cases} -\min(\frac{\sqrt{3}}{2} - \frac{1}{\sqrt{3}} + y, (\sqrt{3}x - y + \frac{1}{\sqrt{3}})/2, (-\sqrt{3}x - y + \frac{1}{\sqrt{3}})/2), & (x,y) \in \Omega_0, \\ 0 & (x,y) \in \tau_0. \end{cases} \tag{43}$$

For $(x, y) \in \overline{\Omega}_0^c$, the magnitude of the signed distance $\phi_0(x, y)$ is the smallest distance of (x, y) to sides of the triangle, and the sign of $\phi_0(x, y)$ is positive.

For the boundaries of the species, we use the dotted curve to show the simulated boundary of the species and the triangle represents the initial boundary. Figure 6 shows the simulation of the evolvement of the species and moving boundaries along time with an equilateral triangle as the initial boundary.

From Figure 6, we can see that the dotted curve evolves into a circle, and then propagate as a circle, which also agrees with the theoretical results [12], where it is proved in Theorems 1.1 –1.3 that the moving boundary for large time is a smooth closed manifold close to an enlarging sphere as time increases.

Example 4.2: In the 2D logistic diffusion model (26) with parameters $(D, \mu, a, b) = (5, 10, 1, 1)$, the initial boundary of the species τ_0 is a rectangle with length= 1.2 and width= 1, centred at (0,0). And the initial function $u_0(x, y)$ and the initial level set function $\phi_0(x, y)$ are set as follows:

$$u_0(x,y) = \begin{cases} 200(0.5 - x)(0.5 + x)(0.6 - y)(0.6 + y), & (x,y) \in \Omega_0, \\ 0 & (x,y) \in \Omega_0^c. \end{cases} \tag{44}$$

For $(x, y) \in \overline{\Omega}_0$, we set

$$\phi_0(x,y) = \begin{cases} -\min(0.5 - |x|, 0.6 - |y|), & (x,y) \in \Omega_0, \\ 0 & (x,y) \in \tau_0. \end{cases} \tag{45}$$

For $(x, y) \in \overline{\Omega}_0^c$, we have

$$\phi_0(x,y) = \begin{cases} \sqrt{(|x| - 0.5)^2 + (|y| - 0.6)^2}, & |x| > 0.5 \text{ and } |y| > 0.6, \\ \min(|y - 0.6|, |y + 0.6|), & |x| \le 0.5, \\ \min(|x - 0.5|, |x + 0.5|), & |y| \le 0.6. \end{cases} \tag{46}$$

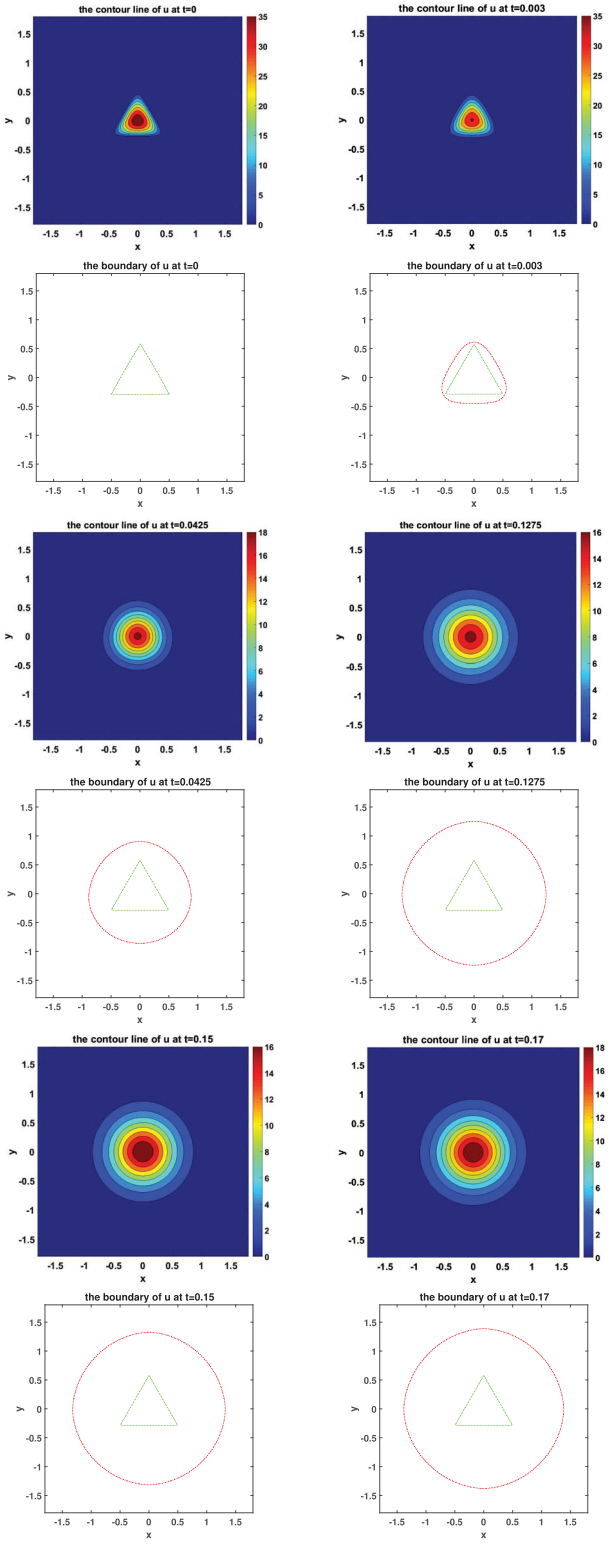

Figure 6. The simulated evolution of $u(x, y, t)$ and $\tau(t)$ with initial domain Ω_0 of an equilateral triangle in 2D. The snapshots are taken at the times $t =$ 0,0.003,0.0425,0.1275,0.15,0.17, respectively.

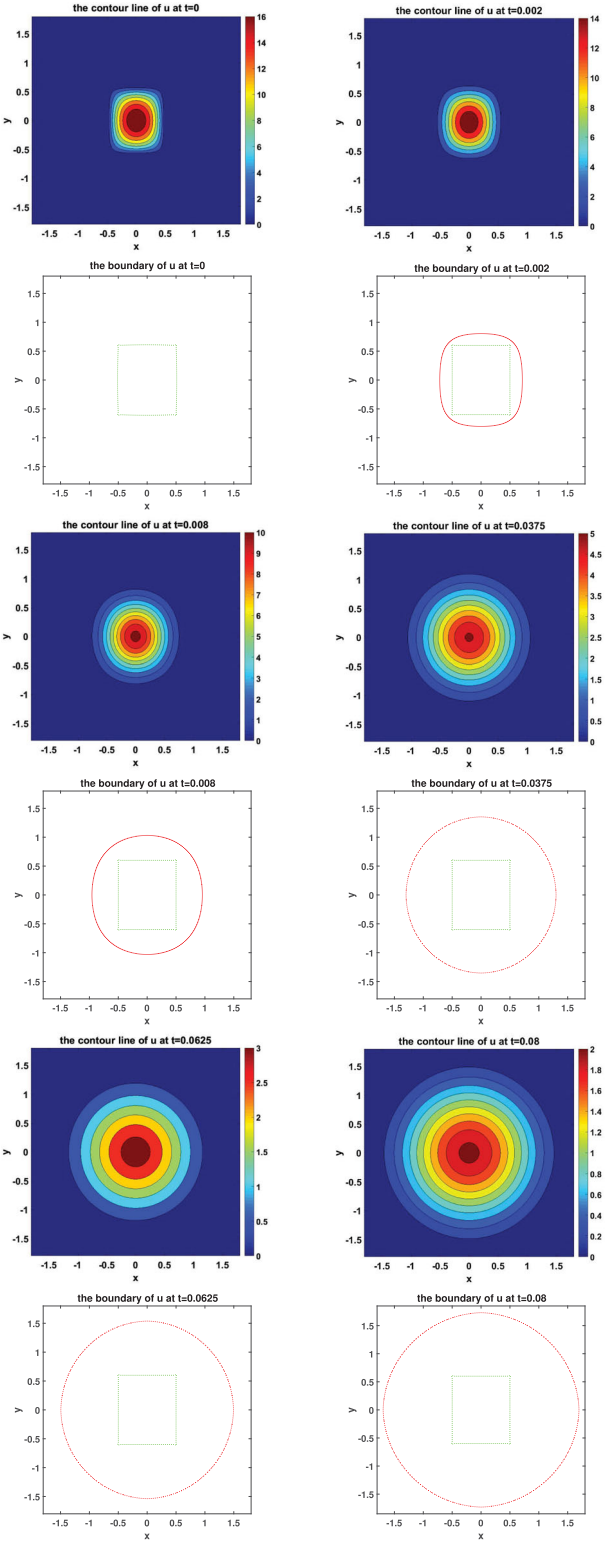

Figure 7. The simulated evolution of $u(x, y, t)$ and $\tau(t)$ for the initial domain of a rectangle in 2D. The snapshots are taken at the times $t = 0,0.002,0.008,0.0375,0.0625,0.08$, respectively.

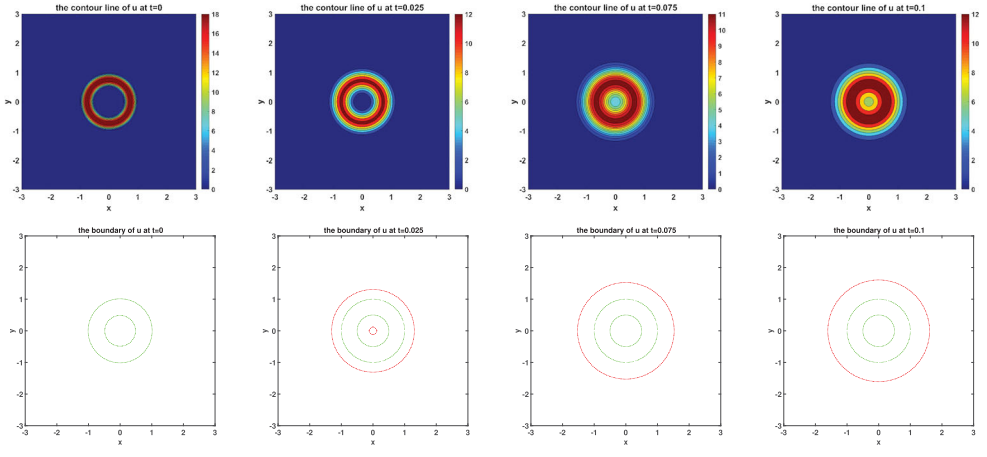

Figure 8. Annulus.

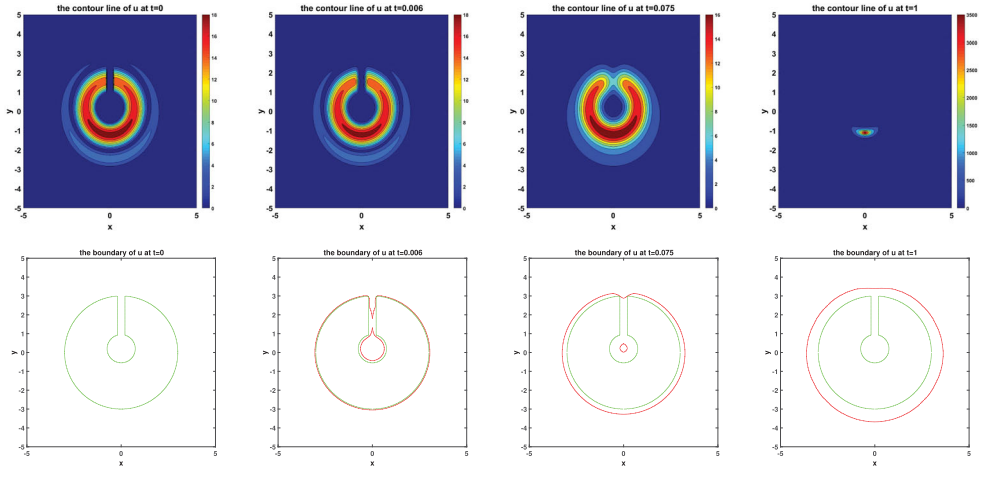

Figure 9. Annulus with a cut.

For the boundaries of the species, we use the dotted curve to show the simulated boundary of the species and the rectangle represents the initial boundary. Figure 7 shows the spreading of u and moving boundary along time with a rectangle as the initial boundary. It indicates that the boundary evolves into a circle, and then propagates like a circle, as predicted by the theoretical result in [12].

Example 4.3: Here we test the level set method for solving (26) with two other different initial domain setup: annulus (Figure 8) and Annulus with a cut (Figure 9). For the boundaries of the species, we use the outer dotted curve to show the simulated boundary of the species and the inner dotted curve represents the initial boundary. For all two different cases, the front will asymptotically evolve into circles that correlates exactly with theoretical results in [12], which predicts that as time increases, the bounded piece of the moving boundary will eventually disappear, and the outer moving boundary will evolve into a smooth closed manifold getting closer and closer to an enlarging sphere as time increases.

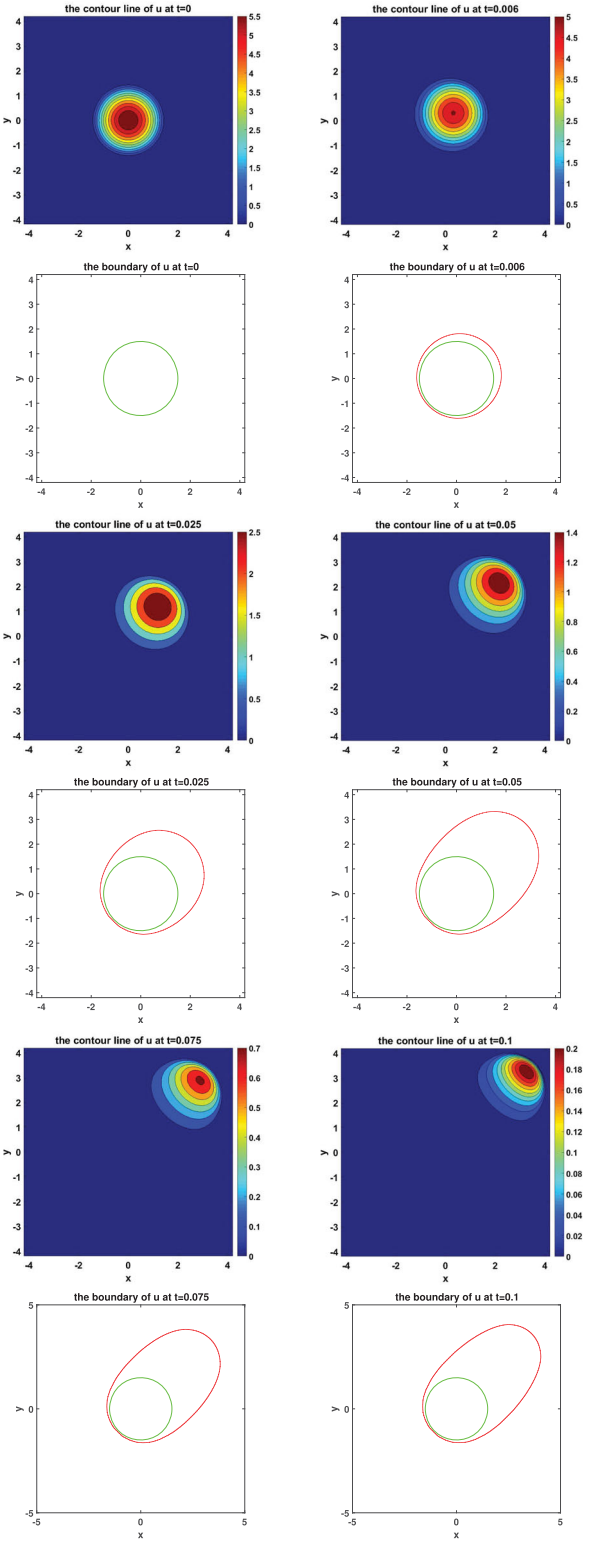

Figure 10. The simulated evolution of $u(x, y, t)$ and $\tau(t)$ for initial boundary a circle in 2D with an advection term. The snapshots are taken at the times $t = 0, 0.006, 0.025, 0.05, 0.075, 0.1$, respectively.

4.4. Numerical test of level set methods for 2D advection–reaction–diffusion model

We consider a 2D advection–reaction–diffusion (ARD) model with a free boundary of the form

$$\frac{\partial U}{\partial t} - D\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) + \beta\left(\frac{\partial U}{\partial x} + \frac{\partial U}{\partial y}\right) = U(a - bU), \quad t > 0,\ (x, y) \in \Omega(t), \tag{47}$$

together with the boundary conditions

$$U(t, \partial\Omega(t)) = 0, \quad t > 0, \tag{48}$$

the Stefan condition

$$\mathbf{v}(t, x, y) = \mu|\nabla U(t, x, y)|\mathbf{n}(t, x, y) = -\mu\nabla U(t, x, y), \quad t > 0,\ (x, y) \in \partial\Omega(t), \tag{49}$$

where $\mathbf{v}(t, x, y)$ and $\mathbf{n}(t, x, y)$ are, respectively, the velocity vector of the boundary point $(x, y) \in \partial\Omega(t)$ and the unit outward normal of $\Omega(t)$ at $(x, y) \in \partial\Omega(t)$. The initial conditions are

$$\Omega(0) = \Omega_0, \quad U(0, x, y) = U_0(x, y),\ (x, y) \in \Omega_0. \tag{50}$$

The initial function $U_0(x, y)$ is assumed to satisfy (27).

In (47), the advection term $\beta(\frac{\partial U}{\partial x} + \frac{\partial U}{\partial y})$ is in the north-east direction. We may think of (47) as describing the spreading of a flying insect species U affected by wind blowing to the north-east direction during the spreading process.

In the 2D ARD model (47)–(50) with parameters $(D, \mu, a, b, \beta) = (10, 10, 1, 1, 50)$, the initial boundary of the species τ_0 is a circle with radius equals 1.5, centred at (0,0). And the initial value $u_0(x, y)$ and the initial level set function $\phi_0(x, y)$ are set as follows:

$$u_0(x, y) = \begin{cases} 6\cos(\sqrt{x^2 + y^2}\pi), & (x, y) \in \Omega_0, \\ 0, & (x, y) \in \Omega_0^c, \end{cases} \tag{51}$$

$$\phi_0(x, y) = -(0.5 - \sqrt{x^2 + y^2}). \tag{52}$$

Figure 10 shows the spreading of U and the moving boundary of this ARD model as time increases, where the dotted curve represents the simulated boundary of the species. In order to clearly reveal the effect of the advection in the model, the initial boundary is indicated in the graph by the solid circle; the free boundary clearly expands faster in the north-east direction and slower in the south-west direction, due to the advection in the north-east direction.

5. Conclusion

In this paper, we have introduced a general numerical framework to efficiently solve a class of reaction–diffusion equations with moving free boundaries. A front tracking algorithm is first introduced for the 2D model with radial symmetry, which has been compared to a front-fixing method. The consistency for these two methods has been checked by several numerical examples. A level set framework is later applied for more general 2D models to overcome the difficulty of handling complicated topologically changes. All the proposed methods agree with each other through examination of an example of the 2D model with radial symmetry. The level set approach is also shown to be very robust to handle different complicated geometries.

Since the level set method is very robust to handle topological changes, in a separate work [21], we have extended level set approach to the systems of two competing species in which each species has its own moving boundary. The front will become more complicated and more challenging once

two moving fronts are tangled together. Moreover, extremely small time steps are usually required when the system is very stiff. To overcome this difficulty, currently we are incorporating the implicit integration factor (IIF) method [22] and its compact form (cIIF) [23] for such stiff systems. The major computation for IIF or cIIF arises from the computation of the exponential of discretized matrices. Due to the moving fronts, evaluation of the exponential of the discretized matrices is necessary for each time step. We also plan to combine Krylov subspace [5] to further improve the efficiency.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Science Foundation [DMS1853365] and the Australian Research Council.

References

[1] W. Bao, Y. Du, Z. Lin, and H. Zhu, *Free boundary models for mosquito range movement driven by climate warming*, J. Math. Biol. 76(4) (2018), pp. 841–875.
[2] G. Bunting, Y. Du, and K. Krakowski, *Spreading speed revisited: Analysis of a free boundary model*, NHM 7(4) (2012), pp. 583–603.
[3] L.A. Caffarelli and S. Salsa, *A geometric approach to free boundary problems*, Vol. 68. Providence, RI, American Mathematical Society, 2005.
[4] Y. Cao, A. Faghri, and W.S. Chang, *A numerical analysis of Stefan problems for generalized multi-dimensional phase-change structures using the enthalpy transforming model*, Int. J. Heat. Mass. Transf. 32(7) (1989), pp. 1289–1298.
[5] S.Q. Chen and Y.T. Zhang, *Krylov implicit integration factor methods for spatial discretization on high dimensional unstructured meshes: Application to discontinuous Galerkin methods*, J. Comput. Phys. 230(11) (2011), pp. 4336–4352.
[6] S. Chen, B. Merriman, S. Osher, and P. Smereka, *A simple level set method for solving Stefan problems*, J. Comput. Phys. 135(1) (1997), pp. 8–29.
[7] H. Chen, C. Min, and F. Gibou, *A numerical scheme for the Stefan problem on adaptive Cartesian grids with supralinear convergence rate*, J. Comput. Phys. 228(16) (2009), pp. 5803–5818.
[8] I.L. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv, *Front tracking for gas dynamics*, J. Comput. Phys. 62(1) (1986), pp. 83–110.
[9] J. Crank, *Free and moving boundary problems*, Clarendon Press, Oxford, 1984.
[10] Y. Du and Z. Guo, *The Stefan problem for the FisherKPP equation*, J. Differ. Equ. 253(3) (2012), pp. 996–1035.
[11] Y. Du and Z. Lin, *Spreading-vanishing dichotomy in the diffusive logistic model with a free boundary*, SIAM J. Math. Anal. 42(1) (2010), pp. 377–405.
[12] Y. Du, H. Matano, and K. Wang, *Regularity and asymptotic behavior of nonlinear Stefan problems*, Arch. Ration. Mech. Anal. 212(3) (2014), pp. 957–1010.
[13] S.O.R. Fedkiw and S. Osher, *Level set methods and dynamic implicit surfaces*, Surfaces 44 (2002), p. 77.
[14] F. Gibou and R. Fedkiw, *A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem*, J. Comput. Phys. 202(2) (2005), pp. 577–601.
[15] J. Glimm, X.L. Li, Y. Liu, and N. Zhao, *Conservative front tracking and level set algorithms*, Proc. Natl. Acad. Sci. 98(25) (2001), pp. 14198–14201.
[16] J. Hilditch and P. Colella, *A front tracking method for compressible flames in one dimension*, SIAM. J. Sci. Comput. 16(4) (1995), pp. 755–772.
[17] J. Hua, J.F. Stene, and P. Lin, *Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method*, J. Comput. Phys. 227(6) (2008), pp. 3358–3382.
[18] H.G. Landau, *Heat conduction in a melting solid*, Q. Appl. Math. 8(1) (1950), pp. 81–94.
[19] R.J. Leveque and Z. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM. J. Numer. Anal. 31(4) (1994), pp. 1019–1044.
[20] R.J. Leveque and K.M. Shyue, *Two-dimensional front tracking based on high resolution wave propagation methods*, J. Comput. Phys. 123(2) (1996), pp. 354–368.
[21] S. Liu and X.F. Liu, *Numerical methods for a two-species competition-diffusion model with free boundaries*, Mathematics 6(5) (2019), pp. 72.
[22] Q. Nie, Y.-T. Zhang, and R. Zhao, *Efficient semi-implicit schemes for stiff systems*, J. Comput. Phys. 214 (2006), pp. 521–537.
[23] Q. Nie, F. Wang, Y.-T. Zhang, and X.F. Liu, *Compact integration factor methods in high spatial dimensions*, J. Comput. Phys. 277 (2008), pp. 5238–5255.

[24] S. Osher and R.P. Fedkiw, *Level set methods: An overview and some recent results*, J. Comput. Phys. 169(2) (2001), pp. 463–502.
[25] S. Osher and J.A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys. 79(1) (1988), pp. 12–49.
[26] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A PDE-based fast local level set method*, J. Comput. Phys. 155(2) (1999), pp. 410–438.
[27] C.S. Peskin, *The immersed boundary method*, Acta Numer. 11 (2002), pp. 479–517.
[28] M.-A. Piqueras, R. Company, and L. Jodar, *A front-fixing numerical method for a free boundary nonlinear diffusion logistic population model*, J. Comput. Appl. Math. 309 (2017), pp. 473–481.
[29] J.A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. 93(4) (1996), pp. 1591–1595.
[30] J.A. Sethian, *Level set methods and fast marching methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, Vol. 3. Cambridge, MA, Cambridge University Press, 1999.
[31] M. Sussman, P. Smereka, and S. Osher, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys. 114(1) (1994), pp. 146–159.
[32] S.O. Unverdi and G. Tryggvason, *A front-tracking method for viscous, incompressible, multi-fluid flows*, J. Comput. Phys. 100(1) (1992), pp. 25–37.
[33] A. Wiegmann and K.P. Bube, *The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources*, SIAM. J. Numer. Anal. 35(1) (1998), pp. 177–200.
[34] J.J. Xu, Z. Li, J. Lowengrub, and H. Zhao, *A level-set method for interfacial flows with surfactant*, J. Comput. Phys. 212(2) (2006), pp. 590–616.
[35] H.K. Zhao, T. Chan, B. Merriman, and S. Osher, *A variational level set approach to multiphase motion*, J. Comput. Phys. 127(1) (1996), pp. 179–195.
[36] L. Zhu and C.S. Peskin, *Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method*, J. Comput. Phys. 179(2) (2002), pp. 452–468.