Communications on Applied Mathematics and Computation
https://doi.org/10.1007/s42967-023-00261-1

ORIGINAL PAPER

Exponential Time Differencing Method for a Reaction-Diffusion System with Free Boundary

Shuang Liu[1] · **Xinfeng Liu**[2]

Dedicated to the memory of Professor Ching-Shan Chou.

Received: 31 July 2022 / Revised: 9 February 2023 / Accepted: 10 February 2023

Abstract
For reaction-diffusion equations in irregular domains with moving boundaries, the numerical stability constraints from the reaction and diffusion terms often require very restricted time step sizes, while complex geometries may lead to difficulties in the accuracy when discretizing the high-order derivatives on grid points near the boundary. It is very challenging to design numerical methods that can efficiently and accurately handle both difficulties. Applying an implicit scheme may be able to remove the stability constraints on the time step, however, it usually requires solving a large global system of nonlinear equations for each time step, and the computational cost could be significant. Integration factor (IF) or exponential time differencing (ETD) methods are one of the popular methods for temporal partial differential equations (PDEs) among many other methods. In our paper, we couple ETD methods with an embedded boundary method to solve a system of reaction-diffusion equations with complex geometries. In particular, we rewrite all ETD schemes into a linear combination of specific ϕ-functions and apply one state-of-the-art algorithm to compute the matrix-vector multiplications, which offers significant computational advantages with adaptive Krylov subspaces. In addition, we extend this method by incorporating the level set method to solve the free boundary problem. The accuracy, stability, and efficiency of the developed method are demonstrated by numerical examples.

Keywords Reaction diffusion equations · Free boundary · Integrating factor method · Level set method

This work is partially supported by the NSF DMS1853365.

✉ Xinfeng Liu
xfliu@math.sc.edu

Shuang Liu
shl083@ucsd.edu

[1] Department of Mathematics, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093-0112, USA

[2] Department of Mathematics, University of South Carolina, Columbia, South Carolina 29208, USA

Mathematics Subject Classification 65M12

1 Introduction

The systems of reaction-diffusion equations coupled with moving boundaries defined by the Stefan condition have been widely used to describe the dynamics of the spreading population. A moving boundary problem is characterized by the fact that the boundary of the domain is not known in advance but it has to be determined as a part of the solution. These problems are often called Stefan problems due to the Stefan condition that links the behavior of the boundary with the unknown solution [55, 56, 60].

The Stefan condition was first introduced to manage a moving boundary of a parabolic type PDE to describe the spreading of species population as introduced in [19], the reaction-diffusion system for the density of population of the invasive species $u(\mathbf{x}, t)$ depending on the time t and the spatial variable $\mathbf{x}$. In this paper, we consider solving the following system of a reaction-diffusion equation coupled with a free boundary:

$$\begin{cases} u_t = \nabla \cdot (\beta(\mathbf{x})\nabla u) + f(u) & \text{for } \mathbf{x} \in \Omega(t), \quad t > 0, \\ u(\mathbf{x}, t) = 0 & \text{for } \mathbf{x} \in \partial\Omega(t), \quad t > 0, \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega_0, \end{cases} \tag{1}$$

where $\beta(\mathbf{x})$ is a continuous function on $\Omega(t)$, relating to the diffusion rate of the species $u(\mathbf{x}, t)$, and $f(u)$ is assumed to be a C^1 function satisfying $f(0) = 0$. $u_0 \in C^2(\overline{\Omega_0})$, u_0 is positive in Ω_0, and $u_0(\mathbf{x}) = 0$ for $\mathbf{x} \in \partial\Omega_0$. $\partial\Omega(t)$ is the moving boundary of the evolution of the domain $\Omega(t)$, which represents the spreading front of the species $u(\mathbf{x}, t)$. Here 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 the velocity $\mathbf{v}(\mathbf{x}, t) = \mu|\nabla_{\mathbf{x}} u(\mathbf{x}, t)|\mathbf{n}(\mathbf{x})$, 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 is generally called the "free boundary", which has been extensively studied theoretically [10] and numerically [12–14, 23, 50–52] and the references therein. Other theoretical studies of related free boundary problems can be found in [7] and the references therein. When solving such a system numerically, difficulties arise from the stiffness along with moving boundaries. First of all, it is always extremely difficult to handle points near the boundary. To overcome this, various numerical techniques have been proposed with different choices for defining the ghost values to avoid the small cell stiffness, while those numerical treatments focus on introducing a small positive number as the threshold of the distance between the interior points and the boundary points [23, 24, 32, 42]. To some extent, these techniques can remove the large errors that could occur from dividing by small numbers to get second-order accurate solutions. However, remedies are required to keep the numerical accuracy of the gradients by not only proposing higher-order extrapolation for defining ghost points, but also combining higher-order interpolation for locating the interface [47].

To overcome this difficulty, we adopt an embedded boundary method to solve a variable coefficient Poisson equation in an irregular domain with Dirichlet boundary conditions.

Numerical solutions to the Poisson equation in irregular domains have been considered by many approaches, including finite difference [9, 24, 32, 38, 47, 53, 61], finite volume [31, 62, 49], and finite element [2, 6, 43, 58] using various meshing techniques. Among them, the embedded boundary method has a number of advantages, which includes simplifying the grid generation process for complicated geometries, enabling a fast computation approach in parallel, and shifting the complexity of dealing with complex geometries to the discretization approach. More importantly, the embedded boundary method is an excellent candidate with extension to the moving boundary problems, as it generates the mesh using a background regular mesh by taking special care of cut-cells where the geometry intersects the grid.

The placement of the *ghost point* is the subtle yet important distinction from a wide range of methods [35–37]. In contrast to the ghost fluid method introduced by [24], where *ghost points* are placed outside the computational domain, here we plan to use *interior ghost points* instead to ease the small cell stiffness when the interface is very close to grid points in the irregular domain. The proposed embedded boundary method results in a symmetric positive definite discretization matrix, thus we can use a wild number of fast linear solvers. For instance, the algebraic multigrid with both "V"-cycle and "W"-cycle can be applied as preconditioners to further speed up calculations.

On the other hand, extremely small time steps are required due to the stiffness of the nonlinear system. When explicit schemes are applied to solve such a system, due to the stability constraints, an extremely small time step should be used and it might take a long time to finish one single simulation. However, while applying an implicit scheme [8, 25, 44] may be able to remove the stability constraints on the time step Δt, it usually requires solving a large global system of nonlinear equations for each time step, and the computational cost could be significant.

To remove the stability constrains on the size of time steps, we employ exponential time differencing (ETD) methods in which the diffusion term is discretized by the embedded boundary method. As is known, the ETD schemes exhibit very nice stability properties, which allow for the relatively large time step size [5, 16–18, 28, 30, 33, 34, 39, 40]. In addition, by rewriting all ETD schemes as a linear combination of ϕ-functions, we combine a state-of-the-art algorithm: *phipm_simul_iom*2 [45] to evaluate the linear combination of matrix-vector multiplications, which offers significant computational advantages by adopting adaptive Krylov subspaces.

The rest of the paper is organized as follows. In Sect. 2, a second-order embedded boundary method is presented to discretize the diffusion term in irregular domains. In Sect. 3, we briefly describe the explicit ETD schemes along with the Runge-Kutta (RK) type of ETD schemes by rewriting all ETD schemes as a linear combination of ϕ-functions. Furthermore, we introduce a state-of-the-art algorithm for computing linear combinations of the matrix function $\phi_k(A)$ on vectors v_k. In Sect. 4, various numerical examples have been performed to demonstrate the accuracy, efficiency, and stability of the developed algorithm, and such schemes have been incorporated with a level set method to solve the free boundary problem as depicted in (1). Finally, in Sect. 5, we draw a brief conclusion and further discuss several possible extensions for future studies.

2 A Cartesian Grid Embedded Boundary Method

In this section, following [53], we briefly introduce a Cartesian grid embedded boundary method to develop a second-order symmetric positive definite discretization of a stationary Poisson equation with Dirichlet boundary conditions in irregular domains.

In order to illustrate the approach, we first consider a stationary Poisson equation with variable coefficients in a two-dimensional (2D) irregular domain Ω,

$$\nabla \cdot (\beta(x,y)\nabla u) = f(x,y), \quad (x,y) \in \Omega \tag{2}$$

with Dirichlet boundary conditions

$$u(x,y) = u_{\mathcal{D}}(x,y), \quad (x,y) \in \partial\Omega \tag{3}$$

on the interface. Without loss of generality, we assume that the irregular domain Ω is contained inside a rectangular domain $[x_{\mathrm{L}}, x_{\mathrm{R}}] \times [y_{\mathrm{L}}, y_{\mathrm{R}}]$, covered by a uniform grid with a grid function denoted by $u_{ij} = u(x_i, y_j)$, where

$$\mathbf{x}_{ij} = (x_i, y_j) = (x_{\mathrm{L}} + (i-1)h, y_{\mathrm{L}} + (j-1)h), \;\; i = 1, \cdots, n_x, \;\; j = 1, \cdots, n_y$$

with $x_{\mathrm{L}} + (n_x - 1)h = x_{\mathrm{R}}$, $y_{\mathrm{L}} + (n_y - 1)h = y_{\mathrm{R}}$. We denote grid points inside Ω on the fringe of the computational domain as *interior ghost points*, and all other interior grid points are denoted as *computational points*. An *interior ghost point* (x_i, y_j) satisfies the condition that (x_i, y_j) is inside Ω, but at least one of its four nearest neighbors is outside. A computational point (x_i, y_j) satisfies the condition that (x_i, y_j) and all its four nearest neighbors are all inside Ω. For example, in Fig. 1 (left), all colored grid points are *computational points* while all grid points with a circle are *interior ghost points*. Note that the Poisson system (2) will be solved only at *computational points* and not at *interior ghost points*.

There are three types of *computational points* to discretize the Laplacian operator as follows:

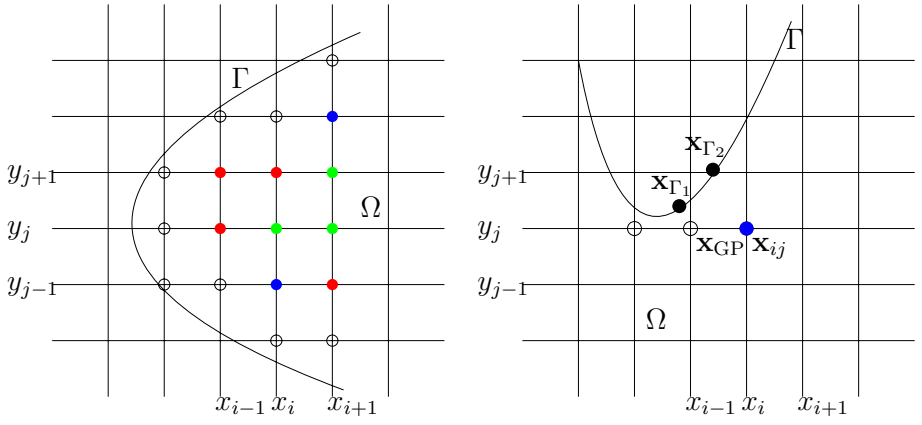

Fig. 1 Left: discretization in an irregular domain with circles representing *interior ghost points* and colored dots representing *computational points*. Right: illustration to construct an RBF interpolation to obtain u_{GP}

Case 1. If the *computational point* is in the absence of *interior ghost points* as its neighbors (green grid points in Fig. 1 (left)), the Laplacian operator is approximated by a standard central difference scheme. We take the point (x_i, y_j) in Fig. 1 (left) as an example,

$$\begin{aligned}\nabla \cdot (\beta(x_i, y_j)\nabla u(x_i, y_j)) \approx & \frac{\beta_{i+\frac{1}{2},j}u_{i+1,j} - \left(\beta_{i+\frac{1}{2},j} + \beta_{i-\frac{1}{2},j}\right)u_{i,j} + \beta_{i-\frac{1}{2},j}u_{i-1,j}}{h^2} \\ & + \frac{\beta_{i,j+\frac{1}{2}}u_{i,j+1} - \left(\beta_{i,j+\frac{1}{2}} + \beta_{i,j-\frac{1}{2}}\right)u_{i,j} + \beta_{i,j-\frac{1}{2}}u_{i,j-1}}{h^2}.\end{aligned} \tag{4}$$

Case 2. For one coordinate direction, if the *computational point* neighbors an *interior ghost point* while the *interior ghost point* borders the interface, for example, the red and blue grid points in Fig. 1 (left) for the y-coordinate direction, the Lagrange polynomial interpolation with a line by the line approach will be applied for this case.

For instance, we take (x_i, y_{j+1}) in the y-direction as an example for illustration (see Fig. 1 (left)). The intersection point of the grid line $x = x_i$ with the boundary Γ between (x_i, y_{j+2}) and (x_i, y_{j+3}) is denoted by (x_i, y_Γ), and the boundary value at (x_i, y_Γ) is given by u_Γ. Here y_Γ can be found by some root-finding algorithm, such as the secant method.

Next we introduce an interpolation polynomial such that the value of the *interior ghost point* $u_{\text{GP}} = u_{i,j+2}$ can be estimated as

$$u_{\text{GP}} = \mathcal{I}_P u(x_i, y_{j+2}) = u_\Gamma g_\Gamma(y_{j+2}) + u_{i,j+1}g_1(y_{j+2}) = u_\Gamma \frac{y_{j+2} - y_{j+1}}{y_\Gamma - y_{j+1}} - u_{i,j+1}\frac{y_{j+2} - y_\Gamma}{y_\Gamma - y_{j+1}}.$$

Substituting u_{GP} into the central difference approximation for the Laplacian operator in the y-direction at the point (x_i, y_{j+1}), we obtain

$$\frac{-\left(\beta_{i,j+\frac{3}{2}}(1 - g_1(y_{\text{GP}})) + \beta_{i,j+\frac{1}{2}}\right)u_{i,j+1} + \beta_{i,j+\frac{1}{2}}u_{i,j}}{h^2} + \frac{\beta_{i,j+\frac{3}{2}}u_\Gamma g_\Gamma(y_{\text{GP}})}{h^2}.$$

Noticing that

$$|\, g_\Gamma(y_{\text{GP}})\,| = \left|\frac{y_{\text{GP}} - y_{j+1}}{y_\Gamma - y_{j+1}}\right| = \left|\frac{y_{j+2} - y_{j+1}}{y_\Gamma - y_{j+1}}\right| \leqslant 1 \text{ and } |\, g_1(y_{\text{GP}})\,| = \left|\frac{y_{\text{GP}} - y_\Gamma}{y_\Gamma - y_{j+1}}\right| = \left|\frac{y_{j+2} - y_\Gamma}{y_\Gamma - y_{j+1}}\right| \leqslant 1. \tag{5}$$

The resulting linear system is still diagonal dominant with correct sign, and the symmetric structure is also preserved with only diagonal elements modified.

Case 3. For one coordinate direction, the *computational point* neighbors an *interior ghost point* while this *interior ghost point* does not border the interface, for example, the blue grid points in Fig. 1 (left) for the x-coordinate direction. For this case, a radial basis function (RBF) based interpolation [19, 24] will be employed since applying the Lagrange polynomial interpolation directly would cause loss of accuracy.

Without loss of generality, we consider the case in a non-convex geometry as presented in Fig. 1 (right) that will occur in later numerical testing examples. We take (x_i, y_j) in the x-coordinate direction as an example for illustration. Let $\mathbf{x}_{ij} = (x_i, y_j)$ be a *computational point* and $\mathbf{x}_{\text{GP}} = (x_{i-1}, y_j)$ be an *interior ghost point* neighboring $\mathbf{x}_{ij}$ in the x-direction.

We choose $\mathbf{x}_{\Gamma_1}$ and $\mathbf{x}_{\Gamma_2}$ to be the corresponding closet points on the boundary to $\mathbf{x}_{\text{GP}}$ and $\mathbf{x}_{ij}$. We use the following combination of RBF and a linear polynomial tail for interpolation at $\mathbf{x}_{\text{GP}}$:

$$\mathcal{I}_{\mathrm{R}}u(\mathbf{x}) = \lambda_{ij}\psi(\|\mathbf{x}-\mathbf{x}_{ij}\|) + \lambda_{\Gamma_1}\psi(\|\mathbf{x}-\mathbf{x}_{\Gamma_1}\|) + \lambda_{\Gamma_2}\psi(\|\mathbf{x}-\mathbf{x}_{\Gamma_2}\|) + \mu_1 + \mu_2 x + \mu_3 y,$$

where $\mathbf{x} = (x, y)$, $||\cdot||$ is the standard l_2 norm, and $\psi(\cdot)$ is an RBF. The linear polynomial tail is required to maintain the second-order accuracy [4]. The coefficients $\lambda = (\lambda_{ij}, \lambda_{\Gamma_1}, \lambda_{\Gamma_2})^{\mathrm{T}}$ and $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)^{\mathrm{T}}$ are determined by the linear system

$$B\begin{pmatrix}\lambda\\ \boldsymbol{\mu}\end{pmatrix} = \begin{pmatrix}A & \Pi^{\mathrm{T}}\\ \Pi & 0\end{pmatrix}\begin{pmatrix}\lambda\\ \boldsymbol{\mu}\end{pmatrix} = \begin{pmatrix}\mathbf{u}\\ \mathbf{0}\end{pmatrix}, \tag{6}$$

where $\mathbf{u} = (u_{ij}, u_{\Gamma_1}, u_{\Gamma_2})$,

$$A = \begin{pmatrix}\psi(0) & \psi(\|\mathbf{x}_{ij}-\mathbf{x}_{\Gamma_1}\|) & \psi(\|\mathbf{x}_{ij}-\mathbf{x}_{\Gamma_2}\|)\\ \psi(\|\mathbf{x}_{\Gamma_1}-\mathbf{x}_{ij}\|) & \psi(0) & \psi(\|\mathbf{x}_{\Gamma_1}-\mathbf{x}_{\Gamma_2}\|)\\ \psi(\|\mathbf{x}_{\Gamma_2}-\mathbf{x}_{ij}\|) & \psi(\|\mathbf{x}_{\Gamma_2}-\mathbf{x}_{\Gamma_1}\|) & \psi(0)\end{pmatrix}, \quad \text{and} \quad \Pi = \begin{pmatrix}1 & 1 & 1\\ x_{ij} & x_{\Gamma_1} & x_{\Gamma_2}\\ y_{ij} & y_{\Gamma_1} & y_{\Gamma_2}\end{pmatrix}. \tag{7}$$

The value at the *boundary point* $\mathbf{x}_{\mathrm{GP}}$ is assigned as

$$u_{\mathrm{GP}} = \mathcal{I}_{\mathrm{RBF}}u(\mathbf{x}_{\mathrm{GP}}) = \left(\boldsymbol{\psi}_{\mathrm{GP}}^{\mathrm{T}}, \mathbf{p}_{\mathrm{GP}}^{\mathrm{T}}\right)B^{-1}\begin{pmatrix}\mathbf{u}\\ \mathbf{0}\end{pmatrix} \tag{8}$$

with $\boldsymbol{\psi}_{\mathrm{GP}} = \left(\psi(\|\mathbf{x}_{\mathrm{GP}}-\mathbf{x}_{ij}\|), \psi(\|\mathbf{x}_{\mathrm{GP}}-\mathbf{x}_{\Gamma_1}\|), \psi(\|\mathbf{x}_{\mathrm{GP}}-\mathbf{x}_{\Gamma_2}\|)\right)^{\mathrm{T}}$ and $\mathbf{p}_{\mathrm{GP}} = (1, x_{\mathrm{GP}}, y_{\mathrm{GP}})^{\mathrm{T}}$. As the terms in the right-hand side in (8) appear only in the diagonal coefficient, the symmetry of the discrete matrix will not be broken. For this case, the Laplacian operator in the x-direction at the point (x_i, y_j) can be approximated by

$$\frac{\beta_{i+\frac{1}{2},j}u_{i+1,j} - \left(\beta_{i+\frac{1}{2},j} + \beta_{i-\frac{1}{2},j}\right)u_{i,j} + \beta_{i-\frac{1}{2},j}u_{\mathrm{GP}}}{h^2}.$$

3 ETD Schemes

In this section, we briefly discuss both the explicit ETD schemes and the RK type of ETD schemes with arbitrary order accuracy. For illustration, here we consider a reaction-diffusion system with certain boundary conditions,

$$\mathbf{u}_t = \nabla\cdot(\beta(x,y)\nabla\mathbf{u}) + \mathbf{f}(\mathbf{u}, t), \tag{9}$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^m$, $\beta(x, y)$ is the diffusion coefficient, and $\mathbf{f}(\mathbf{u}, t)$ represents nonlinear reaction terms.

Applying the spatial discretization with the embedded boundary method as previously illustrated in Sect. 2 for the Poisson equation, we reduce the equation (9) to a system of ODEs,

$$U_t = CU + F(U(t), t), \tag{10}$$

where $U = U(t)$ is the spatially discretized form of $\mathbf{u}$, and C is a constant matrix representing the finite difference approximation of the diffusion. After multiplying the equation (10) by the integrating factor e^{-Ct}, we integrate the equation over one time step from t_n to $t_{n+1} \equiv t_n + \Delta t$ to obtain

$$U(t_{n+1}) = U(t_n)\mathrm{e}^{C\Delta t} + \int_0^{\Delta t} \mathrm{e}^{(\Delta t-\tau)C} F(U(t_n+\tau), t_n+\tau)\mathrm{d}\tau. \tag{11}$$

While this formula is exact, the essence of the ETD methods is to derive numerical approximations to the integral in this expression.

3.1 Explicit ETD

For the derivation of ETD schemes, the integrand is approximated first through interpolation polynomials of the function $F(U(t_n+\tau), t_n+\tau)$ with $\mathrm{e}^{(\Delta t-\tau)C}$ unchanged. With the Lagrange interpolation being applied to approximate $F(U(t_n+\tau), t_n+\tau)$, a direct integration of the interpolation polynomial with the coefficient term $\mathrm{e}^{(\Delta t-\tau)C}$ yields the ETD method. If all interpolation points used for the integrand are with $\tau \leqslant 0$, the resulted temporal scheme is explicit. Otherwise, the scheme becomes implicit when the interpolation points contain the solution at t_{n+1}.

Assuming that $F(U(t), t)$ is constant such that $F(U(t), t) = F_n \equiv F(U_n, t_n)$ over the interval $t_n \leqslant t \leqslant t_{n+1}$, we introduce a scalar function

$$\phi_1(z) = \frac{\mathrm{e}^z - 1}{z},$$

and the first order ETD scheme (ETD1) is given by

$$\begin{aligned} U_{n+1} &= U_n\mathrm{e}^{C\Delta t} + \Delta t\phi_1(\Delta tC)F_n \\ &= U_n + \Delta t\phi_1(\Delta tC)(F_n + CU_n), \end{aligned}$$

where U_{n+1} is the numerical approximation to $U(t_{n+1})$ and $U(t_n)$ as U_n. Here

$$\phi_1(\Delta tC) = \frac{1}{\Delta t}\int_0^{\Delta t} \mathrm{e}^{(\Delta t-\tau)C}\mathrm{d}\tau = \int_0^1 \mathrm{e}^{(1-\lambda)\Delta tC}\mathrm{d}\lambda$$

is extended to the matrix form from the scalar function ϕ_1.

For the second-order approximation

$$F(U(t_n+\tau), t_n+\tau) \approx F_n + \tau\frac{F_n - F_{n-1}}{\Delta t},$$

the second-order ETD scheme (ETD2) can be achieved by

$$\begin{aligned} U_{n+1} &= U_n\mathrm{e}^{\Delta tC} + \Delta t\phi_1(\Delta tC)F_n + \Delta t\phi_2(\Delta tC)(F_n - F_{n-1}) \\ &= U_n + \Delta t\phi_1(\Delta tC)(F_n + CU_n) + \Delta t\phi_2(\Delta tC)(F_n - F_{n-1}). \end{aligned}$$

To derive even higher-order schemes, one can build up higher-order approximations of the integrand as shown in the equation (11), i.e., with a reminder term of $O(\Delta t^q)(q \geqslant 4)$. For example, one can approximate $F(U(t_n+\tau), t_n+\tau)$ by the high-order Taylor expansion and substitute into the integral term in the equation (11), leading to a family of ϕ functions (similar to ϕ_1),

$$\phi_k(\Delta tC) = \frac{1}{\Delta t^k}\int_0^{\Delta t} \mathrm{e}^{(\Delta t-\tau)C}\tau^{k-1}\mathrm{d}\tau = \int_0^1 \mathrm{e}^{(1-\lambda)\Delta tC}\lambda^{k-1}\mathrm{d}\lambda, \tag{12}$$

which are bounded satisfying the following recursion relation:

$$\phi_{k+1}(z) = \frac{k\phi_k(z)-1}{z},\ k \geqslant 1,\ \text{where } \phi_0(z) = \mathrm{e}^z. \tag{13}$$

As discussed in [15], the explicit multistep ETD schemes with arbitrary order have been derived based on a polynomial approximation of $F(U(t_n+\tau), t_n+\tau)$,

$$U_{n+1} = U_n\mathrm{e}^{\Delta tC} + \Delta t\sum_{m=0}^{s-1} g_m \sum_{k=0}^{m}(-1)^k\binom{m}{k}F_{n-k}, \tag{14}$$

where

$$g_m = (-1)^m\int_0^1 \mathrm{e}^{(1-\lambda)\Delta tC}\binom{-\lambda}{m}\mathrm{d}\lambda \quad \text{with} \quad \binom{-\lambda}{m} = \frac{(-\lambda)(-\lambda-1)\cdots(-\lambda-m+1)}{m!}.$$

3.2 ETD with RK Time Stepping

As mentioned in [15], since the multistep explicit ETD schemes require s previous evaluations of the nonlinear term F as depicted in (14), they are sometimes inconvenient to use. By adopting the RK type approach alternatively, this inconvenience can be avoided. In addition, RK type approaches typically have the advantages of smaller error constants and larger stability regions than the multistep explicit ETD methods. For instance, a brief summary of ETDRK schemes up to the fourth order is listed in the following:

- ETD2RK

$$\begin{aligned} a_n &= U_n + \Delta t\phi_1(\Delta tC)(F_n + CU_n),\\ U_{n+1} &= U_n + \Delta t\phi_1(\Delta tC)(F_n + CU_n) + \Delta t\phi_2(\Delta tC)(F(a_n, t_n+\Delta t) - F_n), \end{aligned}$$

- ETD3RK

$$\begin{aligned} a_n &= U_n + \frac{\Delta t}{2}\phi_1(\Delta tC/2)(F_n + CU_n),\\ b_n &= U_n + \Delta t\phi_1(\Delta tC)(CU_n + 2F(a_n, t_n+\Delta t/2) - F_n),\\ U_{n+1} &= U_n + \Delta t\phi_1(\Delta tC)(CU_n + F_n) + \Delta t\phi_2(\Delta tC)(-3F_n\\ &\quad + 4F(a_n, t_n+\Delta t/2) - F(b_n, t_n+\Delta t))\\ &\quad + \Delta t\phi_3(\Delta tC)(2F_n - 4F(a_n, t_n+\Delta t/2) + 2F(b_n, t_n+\Delta t)). \end{aligned}$$

- ETD4RK

$$
\begin{aligned}
a_n &= U_n + \frac{\Delta t}{2}\phi_1(\Delta tC/2)(F_n + CU_n),\\
b_n &= U_n + \frac{\Delta t}{2}\phi_1(\Delta tC/2)(F(a_n, t_n + \Delta t/2) + CU_n),\\
c_n &= a_n + \frac{\Delta t}{2}\phi_1(\Delta tC/2)(2F(b_n, t_n + \Delta t/2) - F_n + Ca_n),\\
U_{n+1} =& U_n + \Delta t\phi_1(\Delta tC)(CU_n + F_n)\\
&+ \Delta t\phi_2(\Delta tC)(-3F_n + 2(F(a_n, t_n + \Delta t/2) + F(b_n, t_n + \Delta t/2)) - F(c_n, t_n + \Delta t))\\
&+ \Delta t\phi_3(\Delta tC)(2F_n - 2(F(a_n, t_n + \Delta t/2) + F(b_n, t_n + \Delta t/2)) + 2F(c_n, t_n + \Delta t)).
\end{aligned}
$$

3.3 Computing a Linear Combination of ϕ-Functions

The implementation of ETD schemes only requires computing the action of the matrix function $\phi_k(A)$ on vectors v_k. For the evaluation of linear combinations of ϕ-functions acting on sets of vectors $v_0, v_1, \cdots, v_p$,

$$\phi_0(A)v_0 + \phi_1(A)v_1 + \cdots + \phi_p(A)v_p, \tag{15}$$

it is crucial within calculations of all ETD schemes described above.

A few state-of-the-art algorithms to efficiently evaluate linear combinations of the matrix function $\phi_k(A)$ on vectors v_k are presented in [1, 11, 22, 27, 45, 48]. In this paper, we employ the algorithm *phipm_simul_iom*2 in [45] to evaluate the linear combination as in the equation (15), which typically consists of $(p + 1)$ matrix-vector multiplications that can be carried out in a lower dimensional Krylov subspace, using the incomplete orthogonalization method (IOM) within the Arnoldi iteration. As a result, the computational cost can be reduced significantly.

The linear combination (15) is actually equivalent to the solution of the following ODE:

$$y'(t) = Ay(t) + v_1 + tv_2 + \cdots + \frac{t^{p-1}}{(p-1)!}v_p,\ y(0) = v_0,$$

i.e.,

$$
\begin{aligned}
y(1) &= \mathrm{e}^A v_0 + \int_0^1 \mathrm{e}^{(1-\lambda)A}\mathrm{d}\lambda v_1 + \int_0^1 \mathrm{e}^{(1-\lambda)A}\lambda\mathrm{d}\lambda v_2 + \cdots + \int_0^1 \mathrm{e}^{(1-\lambda)A}\lambda^{p-1}\mathrm{d}\lambda v_p,\\
&= \phi_0(A)v_0 + \phi_1(A)v_1 + \phi_2(A)v_2 + \cdots + \phi_p(A)v_p.
\end{aligned}
$$

4 Numerical Experiments

In this section, the accuracy, stability, and efficiency of the developed numerical methods as described above are investigated through various testing examples. First we start with the accuracy test of the embedded boundary method by solving a Poisson equation with a virus-shaped geometry. Next numerical experiments on the reaction-diffusion equation in irregular domains are presented to exhibit the accuracy, stability, and efficiency of ETD

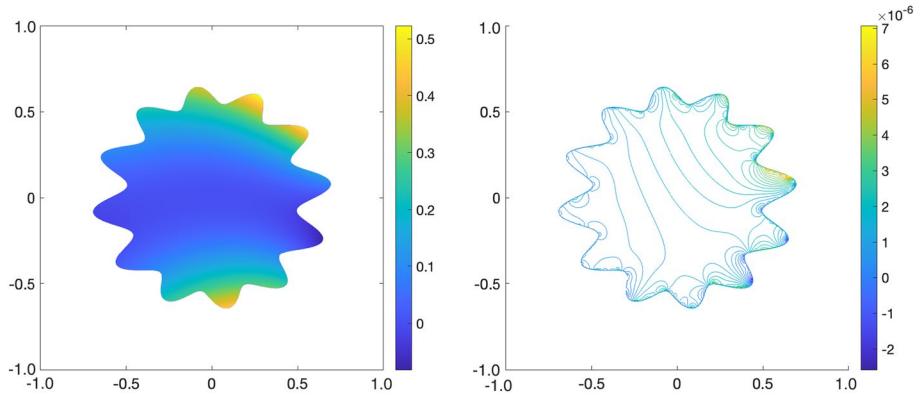

Fig. 2 Numerical solution and error of the Poisson equation on a virus-shape geometry with grid points 1 281 × 1 281. Left: numerical solution. Right: numerical error

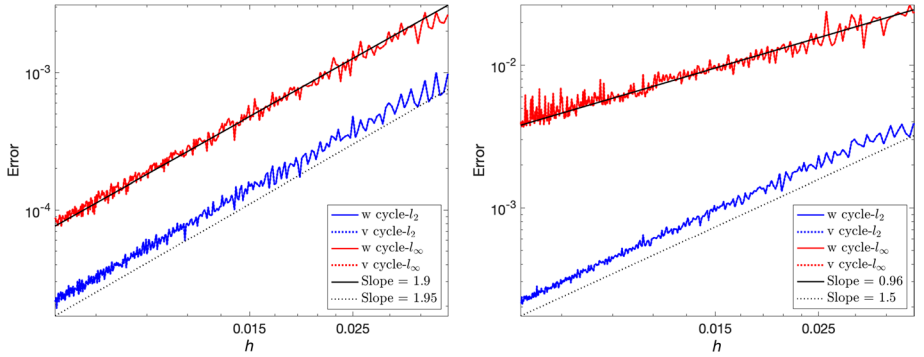

Fig. 3 Error analysis and convergence analysis of the solution with virus-shaped geometry using algebraic multigrid "W" shape and "V" shape. Left: the numerical error of the solution. Right: the numerical error of the gradient of the solution

schemes compared to other methods. Finally, we present the performance of the second-order ETD2 scheme combined with the level set method to solve a free boundary problem.

4.1 Convergence Study for the Embedded Boundary Method

In this example, we consider solving a Poisson equation $\nabla(\beta\nabla u) = f$ in an irregular domain Ω determined by the boundary interface which is parameterized by

$$\begin{cases} x(\theta) = (0.6 + 0.1\sin(12\theta))\cos(\theta), \\ y(\theta) = (0.6 + 0.05\sin(12\theta))\sin(\theta) \end{cases}$$

with $\theta \in [0, 2\pi)$. The exact solution on Ω for this case is $u = e^x(x^2\sin(y) + y^2)$ and $\beta = 2 + \sin(xy)$. An $n_x \times n_y$ uniform mesh partitioning $[-1, 1] \times [-1, 1]$ is used.

The numerical solution with $n_x = n_y = 1\,281$ is presented in Fig. 2. Sweeping from $n_x = n_y = 51$ to $n_x = n_y = 351$, the second-order accuracy for approximation of the solution

can be observed with the developed embedded boundary method both in the l_2 norm and l_∞ norm (see Fig. 3 (left)). Furthermore, we can also observe the $O(h^{1.5})$ accuracy in the l_2 norm and the $O(h^{0.96})$ accuracy in the l_∞ norm for approximation of the gradients of the solution (see Fig. 3 (right)).

4.2 Numerical Tests for ETD with Reaction-Diffusion Systems

In this section, we incorporate the embedded boundary method for the spatial discretization with ETD schemes to solve systems of reaction-diffusion equations in irregular domains. Numerical experiments are performed to demonstrate the accuracy, stability, and efficiency of ETD schemes. Without loss of generality and for the convenience of better comparison, the test example is selected with analytical solutions given. Specifically, we consider the following example with a peanut-shaped and non-convex geometry which is determined by the level-set function:

$$\rho(x, y) = 0.5 - \mathrm{e}^{-20\left(x^2+(y-0.25)^2\right)} - \mathrm{e}^{-20\left(x^2+(y+0.25)^2\right)}.$$

Here we solve the reaction-diffusion equation $u_t = \nabla(\beta\nabla u) + f$ defined inside the domain $\rho(x, y) < 0$, where $\beta(x, y) = 0.25 - x^2 - y^2$. The source term f is computed by assuming a exact solution $u(x, y, t) = \mathrm{e}^{-t}(x^2 + y^2 - 0.25)$. The computational domain is $[-1, 1] \times [-1, 1]$. In order to make a fair comparison between different methods, for this example, we will mainly focus on the following second-order numerical schemes: the second-order Crank-Nicolson, ETD2, and ETD2RK.

4.2.1 Accuracy Test

Numerical errors and corresponding convergence rates of the second-order Crank-Nicolson method, ETD2 and ETD2RK at the final time $t = 0.5$ with five different spatial and temporal resolutions are reported in Table 1, where the time step dt is taken as equal to the grid size h. As expected, we can clearly see a second-order accuracy for all three schemes. For illustration, the numerical solution and numerical error of the reaction-diffusion equation with ETD2 at the final time $t = 0.5$ with grid points $1\,281 \times 1\,281$ are presented in Fig. 4.

4.2.2 Stability Test

In this section, we test the stability properties of four numerical schemes: the standard explicit Runge-Kutta, ETD2, ETD2RK, and Crank-Nicolson for solving the example of the reaction-diffusion system with the peanut-shaped geometry as previously mentioned. We set the final time $t_{\text{end}} = 0.2$ and a uniform grid size $h = 0.004$ for all the simulations. The errors are measured in the l_2 norm between numerical solutions and the exact solutions by varying time steps.

From Fig. 5, it can be observed that the explicit RK scheme blows up with the time step size greater than 1×10^{-5} as expected. In contrast, all other three numerical schemes: Crank-Nicolson, ETD2, and ETD2RK exhibit very excellent stability conditions, which allow for very large time step size till d$t = 0.1$.

Table 1 Numerical errors in l_∞ norm and l_2 norm and corresponding convergence rates for Crank-Nicolson, ETD2, and ETD2RK schemes at $t = 0.5$

Convergence test for Crank-Nicolson				
$n_x \times n_y \times n_t$	l_∞-Error	Order	l_2-Error	Order
81×81×4	8.041×10^{-4}	–	2.296×10^{-4}	–
161×161×8	1.748×10^{-4}	2.201	4.436×10^{-5}	2.372
321×321×16	5.314×10^{-5}	1.718	1.059×10^{-5}	2.067
641×641×32	1.859×10^{-5}	1.515	2.751×10^{-6}	1.945
1 281×1 281×64	4.333×10^{-6}	2.101	6.254×10^{-7}	2.137
Convergence test for ETD2				
$n_x \times n_y \times n_t$	l_∞-Error	Order	l_2-Error	Order
81×81×4	8.234×10^{-4}	–	2.580×10^{-4}	–
161×161×8	1.887×10^{-4}	2.126	5.542×10^{-5}	2.219
321×321×16	5.759×10^{-5}	1.712	1.420×10^{-5}	1.964
641×641×32	1.956×10^{-5}	1.558	3.678×10^{-6}	1.949
1 281×1 281×64	4.583×10^{-6}	2.094	8.780×10^{-7}	2.067
Convergence test for ETD2RK				
$n_x \times n_y \times n_t$	l_∞-Error	Order	l_2-Error	Order
81×81×4	7.851×10^{-4}	–	2.276×10^{-4}	–
161×161×8	1.756×10^{-4}	2.161	4.465×10^{-5}	2.350
321×321×16	5.330×10^{-5}	1.720	1.071×10^{-5}	2.060
641×641×32	1.863×10^{-5}	1.517	2.784×10^{-6}	1.944
1 281×1 281×64	4.343×10^{-6}	2.101	6.346×10^{-7}	2.133

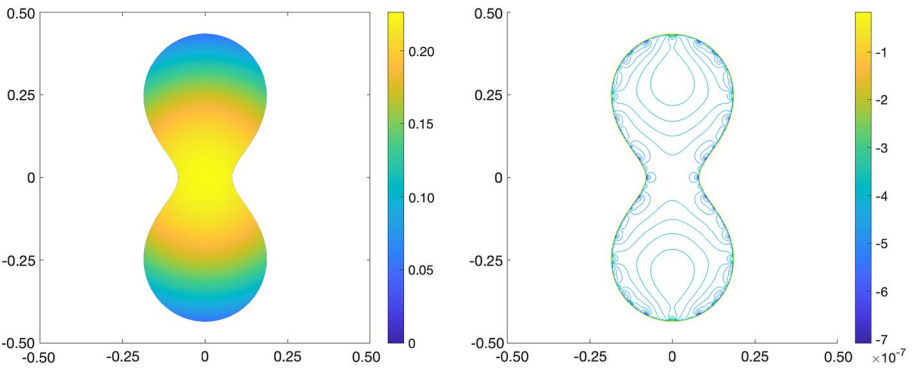

Fig. 4 Numerical solution and error of the reaction-diffusion equation at $t = 0.5$ with a peanut-shape geometry with grid points 1 281 × 1 281 using ETD2. Left: numerical solution. Right: numerical error

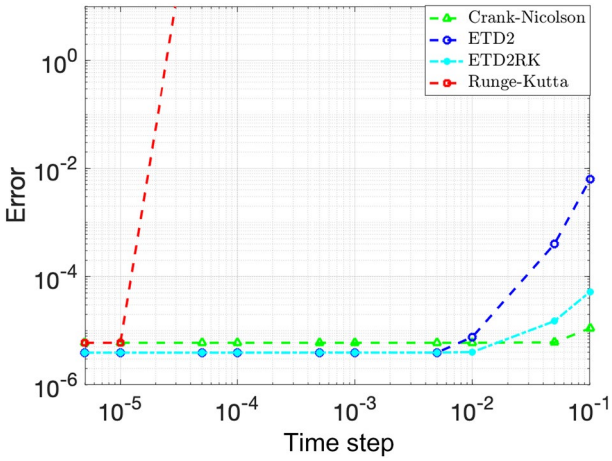

Fig. 5 Errors between numerical solutions and the exact solution using different numerical schemes by varying time step sizes with a fixed uniform grid size $h = 0.004$

Table 2 Efficiency test for a reaction-diffusion system by varying the number of grid points with $dt = h$ (Unit: seconds)

$n_x \times n_y$	$1\,001 \times 1\,001$	$2\,001 \times 2\,001$	$4\,001 \times 4\,001$
Crank-Nicolson	304.23	2 440.19	2 3847.71
ETD2	136.67	1 175.68	1 3470.78
ETD2RK	236.62	1 650.74	1 4137.79

4.2.3 Efficiency test

In Table 2, we compare the efficiency performance of three schemes with nice stability conditions: Crank-Nicolson, ETD2, and ETD2RK for a system with more refined grids. Here we choose the grid size $h = 0.002$, $h = 0.001$, and $h = 0.000\,5$, respectively. The time step size is $dt = h$ for all simulations. Note that when solving the linear system from the Crank-Nicolson method, we use the fast linear solver by the conjugate gradient method with an incomplete Cholesky preconditioner. However, solving a large size of the linear system in each time step is still very costly. By avoiding solving large linear systems, it can be clearly observed that ETD2 is 2–3 times faster than Crank-Nicolson, in which we adopt the adaptive Krylov space to compute the multiplication of matrix and vectors. Since ETD2RK is a two-stage numerical algorithm, its efficiency is slightly better or comparable to Crank-Nicolson for this example for the first two cases, while for the last case, ETD2RK shows an advantage for solving larger linear systems than Crank-Nicolson.

4.3 Numerical Tests of the Free Boundary Problem

For the free boundary problem as described in (1), it is very challenging and crucial to accurately and efficiently handle the reaction-diffusion equation with the changing domain for each time step. In this paper, we integrate the level set method to track the evolution of the moving boundary, and the ETD2 scheme with the discretization technique to solve the reaction-diffusion equation in each time step. As mentioned above, the ETD2 scheme

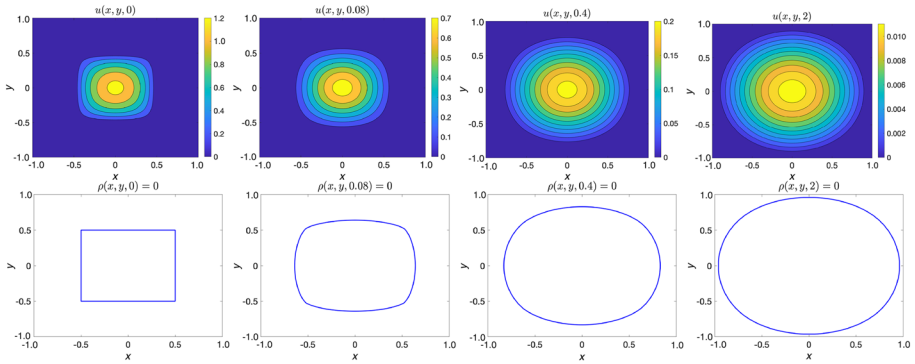

Fig. 6 Evolution of $u(x, y, t)$ and the moving boundary $\rho(x, y, t) = 0$ with the initial domain $\Omega(0)$ a square in 2D

exhibits very nice stability conditions by allowing for a large time step size, and it is also much faster than other schemes with the similar stability conditions like Crank-Nicolson and ETD2RK. Here we briefly introduce the numerical algorithm for solving a diffusive logistic model for the population of the invasive species $u(\mathbf{x}, t)$ with free boundaries as follows:

$$\begin{cases} u_t = D\Delta u + u(a - bu), & t > 0, \quad \mathbf{x} \in \Omega(t), \\ u(\mathbf{x}, t) = 0 \ \text{ and } \ \mathbf{v}(\mathbf{x}, t) = \mu |\nabla u(\mathbf{x}, t)| \mathbf{n}(\mathbf{x}), & t > 0, \quad \mathbf{x} \in \partial\Omega(t), \\ u(\mathbf{x}, 0) \in C^2(\overline{\Omega}(0)), \quad u(\mathbf{x}, 0) > 0 \text{ in } \Omega(0), & u(\mathbf{x}, 0) = 0 \text{ on } \partial\Omega(0). \end{cases}$$

We introduce a level set function $\rho(x, y, t)$, such that $\rho(x, y, t) = 0$ on the boundary, $\rho(x, y, t) < 0$ in $\Omega(t)$, and $\rho(x, y, t) > 0$ outside of $\Omega(t)$. $\rho(x, y, 0)$ is initialized as a signed distance function to the initial boundary $\partial\Omega(0)$.

Algorithm.

Step 0. Input all the parameters. Set the computational box $[-L, L]^2 \subset \mathbb{R}^2$ and cover it with a uniform finite-difference grid with grid sizes h. Discretize the time interval $[0, T]$ of interest with time step $\mathrm{d}t$. Initialize the level-set function ρ^0 and u^0. Set $m = 0$.

Step 1. Advance the interface.

Step 1.1 Extend the velocity $\mathbf{v}(\mathbf{x}, t_m)$ from the interface to the entire computational box. For $\mathbf{v} = v\mathbf{n}$, we extend v off $\partial\Omega(t)$ as a constant along the curve normal to $\partial\Omega(t)$ following the hyperbolic PDF,

$$v_t + S(\rho)\frac{\nabla\rho}{|\nabla\rho|} \cdot \nabla v = 0, \tag{16}$$

where $S(\rho)$ is the signature function of ρ defined as -1 if $\rho < 0$, or 0 if $\rho = 0$, or $+1$ if $\rho > 0$. We adopt fast marching method [57] and combine TVD-RK with WENO [29] to solve (16).

Step 1.2 Discretize the level set advection equation $\rho_t + \mathbf{v} \cdot \nabla\rho = 0$ with a TVD-RK with WENO [29]. Solve it to get the updated level-set function ρ^{m+1}.

Step 1.3 We use the reinitialization scheme of Sussman *et al.* [59] to reinitialize ρ by

$$\rho_t = S(\rho_0)(1 - |\nabla\rho|),$$

where ρ_0 is the initial given ρ to be reinitialized. The sign function S is smoothed by the equation

$$S_\varepsilon(\rho_0) = \frac{\rho_0}{\sqrt{\rho_0^2 + \varepsilon^2}}$$

to avoid numerical difficulties, here we take $\varepsilon = 10^{-10}$. The reinitialized ρ is still denoted by ρ^{m+1}.

Step 2. Extend u^m to new unknowns overlapping with Ω_{m+1} defined by ρ^{m+1} by a quadratic extrapolation in the normal direction by following [3]. Solve the reaction diffusion equation with ETD schemes in the irregular domain Ω_{m+1} to obtain u^{m+1}.

Step 3. Set $m := m + 1$. Repeat Steps 1–2 until the final simulation time is reached.

To show the potential application of the developed ETD2 for the Stefan-type free boundary problems, we consider one example from [41] with the following initial setups: $(D, \mu, a, b) = (1.5, 1, 1, 1)$, with an initial square domain of length 0.5 centered at $(0, 0)$. The initial level set function is

$$\rho(x, y, 0) = -\min(0.5 - |x|).$$

and the initial function is

$$u(x, y, 0) = \begin{cases} 20(0.5 - x)^2(0.5 + x)^2, & \text{if } (x, y) \in \Omega(0), \\ 0, & \text{otherwise.} \end{cases}$$

Figure 6 shows the evolution of the spreading of species $u(x, y, t)$ along with the moving boundary. For this example, the moving boundary will asymptotically evolve into circles, which correlates exactly with the theoretical asymptotic behavior described in [20].

5 Conclusion

In this paper, we have incorporated the embedded boundary method, ETD schemes with the level set method to systematically study reaction-diffusion systems in irregular domains with free boundaries. To our best knowledge, it is the first work to integrate ETD schemes with the embedded boundary method for time-dependent PDEs as well as to combine ETD with the level set method for solving free boundary problems in two dimensions. Through numerical experiments, we first show the accuracy of the embedded boundary method for a Poisson equation with a virus-shaped geometry. Next we test the accuracy, stability, and efficiency of the ETD schemes along with other methods. In order to significantly reduce the computational cost, we have adopted the state-of-the-art algorithm: *phipm_simul_iom*2 [45] to evaluate the linear combination of matrix-vector multiplications in ETD schemes using a lower dimensional adaptive Krylov subspace. In summary, the ETD2 scheme is superior to the other three selected schemes (RK, Crank-Nicolson, and ETD2RK) in terms of a combination of the accuracy, stability, and efficiency, especially in efficiency. More importantly, the ETD2 scheme has been successfully employed to a reaction-diffusion system with free boundaries, which produces very promising results for free boundary problems.

Our immediate next step is to further develop a fast and efficient numerical algorithm for other more complicated systems with free boundaries. Possible extensions include (but not limited to) combining the fast local level set method and the ETD schemes for free boundary problems, and the generalization of the developed methods to the Navier-Stokes equations and the Grad-Shafranov equations.

Compliance with Ethical Standards

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Al-Mohy, A.H., Higham, N.J.: Computing the action of the matrix exponential with an application to exponential integrators. SIAM J. Sci. Comput. **33**(2), 488–511 (2011)
2. Apel, T., Sändig, A.M., Whiteman, J. R.: Graded mesh refinement and error estimates for finite element solutions of elliptic boundary value problems in non-smooth domains. Math. Methods Appl. Sci. **19**(1), 63–85 (1996)
3. Aslam, T.D.: A partial differential equation approach to multidimensional extrapolation. J. Comput. Phys. **193**(1), 349–355 (2004)
4. Barnett, G.A.: A robust RBF-FD formulation based on polyharmonic splines and polynomials. PhD Thesis. Citeseer (2015)
5. Beylkin, G., Keiser, J.M., Vozovoi, L.: A new class of time discretization schemes for the solution of nonlinear PDEs. J. Comput. Phys. **147**, 362–387 (1998)
6. Braess, D.: The contraction number of a multigrid method for solving the Poisson equation. Numer. Math. **37**(3), 387–404 (1981)
7. Bunting, G., Du, Y., Krakowski, K.: Spreading speed revisited: analysis of a free boundary model. Net. Heterog. Med. **7**(4), 583 (2012)
8. Burrage, K., Butcher, J.C.: Stability criteria for implicit Runge-Kutta methods. SIAM J. Numer. Anal. **16**(1), 46–57 (1979)
9. Buzbee, B.L., Dorr, F.W., George, J.A., Golub, G.H.: The direct solution of the discrete Poisson equation on irregular regions. SIAM J. Numer. Anal. **8**(4), 722–736 (1971)
10. Caffarelli, L.A., Salsa, S.: A Geometric Approach to Free Boundary Problems, volume 68. American Mathematical Society, Providence, RI (2005)

11. Caliari, M., Kandolf, P., Ostermann, A., Rainer, S.: The Leja method revisited: backward error analysis for the matrix exponential. SIAM J. Sci. Comput. **38**(3), A1639–A1661 (2016)
12. Cao, Y., Faghri, A., Chang, W.S.: A numerical analysis of Stefan problems for generalized multi-dimensional phase-change structures using the enthalpy transforming model. Intern. J. Heat. Mass. Trans. **32**(7), 1289–1298 (1989)
13. Chen, H., Min, C., Gibou, F.: A numerical scheme for the Stefan problem on adaptive Cartesian grids with supralinear convergence rate. J. Comput. Phys. **228**(16), 5803–5818 (2009)
14. Chen, S., Merriman, B., Osher, S., Smereka, P.: A simple level set method for solving Stefan problems. J. Comput. Phys. **135**(1), 8–29 (1997)
15. Cox, S.M., Matthews, P.C.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**(2), 430–455 (2002)
16. Du, Q., Ju, L., Li, X., Qiao, Z.: Maximum bound principles for a class of semilinear parabolic equations and exponential time-differencing schemes. SIAM Rev. **63**(2), 317–359 (2021)
17. Du, Q., Zhu, W.: Stability analysis and applications of the exponential time differencing schemes. J. Comput. Math. **22**, 200 (2004)
18. Du, Q., Zhu, W.: Modified exponential time differencing schemes: analysis and applications. BIT. Numer. Math. **45**, 307–328 (2005)
19. Du, Y., Lin, Z.: Spreading-vanishing dichotomy in the diffusive logistic model with a free boundary. SIAM J. Math. Anal. **42**(1), 377–405 (2010)
20. Du, Y., Matano, H., Wang, K.: Regularity and asymptotic behavior of nonlinear Stefan problems. Archive Rat. Mech. Anal. **212**(3), 957–1010 (2014)
21. Franke, R.: Scattered data interpolation: tests of some methods. Math. Comput. **38**(157), 181–200 (1982)
22. Gaudreault, S., Pudykiewicz, J.A.: An efficient exponential time integration method for the numerical solution of the shallow water equations on the sphere. J. Comput. Phys. **322**, 827–848 (2016)
23. Gibou, F., Fedkiw, R.: 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), 577–601 (2005)
24. Gibou, F., Fedkiw, R.P., Cheng, L.T., Kang, M.: A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. J. Comput. Phys. **176**(1), 205–227 (2002)
25. Hairer, E., Wanner, G.: Stiff differential equations solved by Radau methods. J. Comput. Appl. Math. **111**(1/2), 93–111 (1999)
26. Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. J. Geophys. Res. **76**(8), 1905–1915 (1971)
27. Hochbruck, M., Lubich, C.: On Krylov subspace approximations to the matrix exponential operator. SIAM J. Numer. Anal. **34**(5), 1911–1925 (1997)
28. Hou, T.Y., Lowengrub, J.S., Shelley, M.J.: Removing the stiffness from interfacial flows with surface tension. J. Comput. Phys. **114**, 312–338 (1994)
29. Jiang, G.-S., Peng, D.: Weighted ENO schemes for Hamilton-Jacobi equations. SIAM J. Sci. Comput. **21**(6), 2126–2143 (2000)
30. Jiang, K., Ju, L., Li, J., Li, X.: Unconditionally stable exponential time differencing schemes for the mass-conserving Allen-Cahn equation with nonlocal and local effects. Numer. Methods Partial Differential Equations **38**(6), 1636–1657 (2021)
31. Johansen, H., Colella, P: A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. J. Comput. Phys. **147**(1), 60–85 (1998)
32. Jomaa, Z., Macaskill, C.: The embedded finite difference method for the Poisson equation in a domain with an irregular boundary and Dirichlet boundary conditions. J. Comput. Phys. **202**(2), 488–506 (2005)
33. Jou, H.J., Leo, P.H., Lowengrub, J.S.: Microstructual evolution in inhomogeneous elastic media. J. Comput. Phys. **131**, 109 (1997)
34. Kassam, A.-K., Trefethen, L.N.: Fourth-order time stepping for stiff PDEs. SIAM J. Sci. Comput. **26**, 1214–1233 (2005)
35. Kreiss, H.-O., Petersson, N.A.: A second order accurate embedded boundary method for the wave equation with Dirichlet data. SIAM J. Sci. Comput. **27**, 1141–1167 (2006)
36. Kreiss, H.-O., Petersson, N.A., Ystrom, J.: Difference approximations for the second order wave equation. SIAM J. Numer. Anal. **40**(5), 1940–1967 (2002)
37. Kreiss, H.-O., Petersson, N.A., Ystrom, J.: Difference approximations of the Neumann problem for the second order wave equation. SIAM J. Numer. Anal. **42**(3), 1292–1323 (2004)
38. Lai, M.-C.: A note on finite difference discretizations for Poisson equation on a disk. Numer. Methods Partial Differential Equations **17**(3), 199–203 (2001)
39. Leo, P.H., Lowengrub, J.S., Nie, Q.: Microstructural evolution in orthotropic elastic media. J. Comput. Phys. **157**, 44–88 (2000)

40. Li, J., Ju, L., Cai, Y., Feng, X.: Unconditionally maximum bound principle preserving linear schemes for the conservative Allen-Cahn equation with nonlocal constraint. J. Sci. Comput. **87**(3), 1–32 (2021)
41. Liu, S.: Numerical methods for a class of reaction-diffusion equations with free boundaries. PhD Thesis. University of South Carolina (2019)
42. Liu, X.D., Fedkiw, R.P., Kang, M.: A boundary condition capturing method for Poisson's equation on irregular domains. J. Comput. Phys. **160**(1), 151–178 (2000)
43. Louis, A.: Acceleration of convergence for finite element solutions of the Poisson equation. Numer. Math. **33**(1), 43–53 (1979)
44. Lowrie, R.B.: A comparison of implicit time integration methods for nonlinear relaxation and diffusion. J. Comput. Phys. **196**(2), 566–590 (2004)
45. Luan, V.T., Pudykiewicz, J.A., Reynolds, D.R.: Further development of efficient and accurate time integration schemes for meteorological models. J. Comput. Phys. **376**, 817–837 (2019)
46. Min, C., Gibou, F., Ceniceros, H.D.: A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids. J. Comput. Phys. **218**(1), 123–140 (2006)
47. Ng, Y.T., Chen, H., Min, C., Gibou, F.: Guidelines for Poisson solvers on irregular domains with Dirichlet boundary conditions using the ghost fluid method. J. Sci. Comput. **41**(2), 300–320 (2009)
48. Niesen, J., Wright, W.M.: Algorithm 919: a Krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators. ACM Trans. Math. Soft. (TOMS) **38**(3), 1–19 (2012)
49. Oevermann, M., Klein, R.: A Cartesian grid finite volume method for elliptic equations with variable coefficients and embedded interfaces. J. Comput. Phys. **219**(2), 749–769 (2006)
50. Osher, S., Fedkiw, R.P.: Level set methods: an overview and some recent results. J. Comput. Phys. **169**(2), 463–502 (2001)
51. Osher, S., Fedkiw, R.P.: Level Set Methods and Dynamic Implicit Surfaces. Springer Verlag, New York (2002)
52. Peng, D., Merriman, B., Osher, S., Zhao, H., Kang, M.: A PDE-based fast local level set method. J. Comput. Phys. **155**(2), 410–438 (1999)
53. Peng, Z., Appelö, D., Liu, S.: Universal AMG accelerated embedded boundary method without small cell stiffness. arXiv:2204.06083 (2022)
54. Perrone, N., Kao, R.: A general finite difference method for arbitrary meshes. Comput. Struct. **5**(1), 45–57 (1975)
55. Piqueras, M.-A., Company, R., Jódar, L.: A front-fixing numerical method for a free boundary nonlinear diffusion logistic population model. J. Comput. Appl. Math. **309**, 473–481 (2017)
56. Rubinshteĭ, L.I.: The Stefan Problem. American Mathematical Society (1971)
57. Sethian, J.A.: Fast marching methods. SIAM Rev. **41**(2), 199–235 (1999)
58. Sudicky, E.A.: The Laplace transform Galerkin technique: a time-continuous finite element theory and application to mass transport in groundwater. Wat. Res. Res. **25**(8), 1833–1846 (1989)
59. Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. J. Comput. Phys. **114**(1), 146–159 (1994)
60. Tayler, A.B.: Free and moving boundary problems. By J. Crank. Clarendon, Oxford, 1984. 425 pp. £ 45.00. J. Fluid Mech. **158**, 532–533 (1985)
61. Yoon, G., Min, C.: Analyses on the finite difference method by Gibou, et al. for Poisson equation. J. Comput. Phys. **280**, 184–194 (2015)
62. Zapata, M.A.U., Hernández-López, F.J.: A GPU parallel finite volume method for a 3D Poisson equation on arbitrary geometries. Intern. J. Comb. Optimizat. Prob. Inform. **9**(1), 3 (2018)