Electronic
Research Archive

https://www.aimspress.com/journal/era

ERA, 33(12): 7331–7359.
DOI: 10.3934/era.2025324
Received: 07 October 2025
Revised: 18 November 2025
Accepted: 27 November 2025
Published: 04 December 2025

Research article

A Cartesian grid-based kernel-free boundary integral method for Allen-Cahn equation

Min Zeng and Yaning Xie*

School of Mathematical Sciences, Zhejiang University of Technology, Hangzhou 310023, China

* **Correspondence:** Email: xieyaning@zjut.edu.cn.

Abstract: This paper investigates the numerical solution for Allen-Cahn equations with perturbation parameters and strong nonlinear terms in general computational domains. To get a global second-order accuracy, we use the second-order semi-implicit Runge-Kutta (SIRK) method and some implicit-explicit (IMEX) methods for time discretization. Then the original problem is transformed into boundary value problems (BVPs) of a modified Helmholtz equation at each time step, which can be solved by a Cartesian grid-based kernel-free boundary integral (KFBI) method. In the KFBI method, the BVPs are reformulated into a corresponding boundary integral equation and then solved iteratively by a class of subspace methods such as the matrix-free generalized minimal residual(GMRES) method, while integrals involved are regarded as solutions to their equivalent interface problems. Unlike traditional boundary integral methods, this method avoids numerical integration of the singular or nearly singular integrals. Instead, it utilizes grid-based operations as an alternative to direct evaluation. Therefore, integral evaluation only requires solving equivalent but much simpler interface problems in a bounding box so that fast elliptic solvers such as fast fourier transforms(FFTs) and geometric multigrid methods are applicable. This makes the KFBI method accurate and efficient when solving constant coefficient elliptic problems in general irregular domains. It can be seen that the accuracy of the present method is verified by numerical examples.

Keywords: Allen-Cahn equation; kernel-free boundary integral method; implicit-explicit method; boundary value problem; interface problem

---

1. Introduction

This paper investigates numerical methods for approximating solutions to the Allen-Cahn equation

$$\frac{\partial u}{\partial t} = \epsilon \Delta u - f(u) \quad \text{in } \Omega, \tag{1.1}$$

with the initial condition

$$u(\mathbf{p},0) = u_0(\mathbf{p}) \quad \text{for } t = 0 \text{ and } \mathbf{p} \in \Omega, \tag{1.2}$$

and the homogeneous Dirichlet boundary condition

$$u(\mathbf{p},t) = 0 \quad \text{for } t > 0 \text{ and } \mathbf{p} \in \Gamma. \tag{1.3}$$

Here, Ω is a bounded two-dimensional general irregular domain, Γ is at least twice continuously differentiable, $u = u(\mathbf{p},t)$ is an unknown phase field variable to be solved with $\mathbf{p} \in \mathcal{R}^2$ being a space point, and u_0 is a given smooth initial data. The positive parameter ϵ represents the interfacial width, which is small compared to the characteristic length of the laboratory scale, $f(u) = F'(u)$ with $F(u)$ being a given energy potential. The homogeneous Dirichlet boundary condition physically prohibits material flux across domain boundaries.

The Allen-Cahn equation was introduced by Allen and Cahn to describe the motion of antiphase boundaries in crystalline solids [1]. It is a type of nonlinear reaction diffusion equation and is widely used in fields such as phase field modeling [2], grain growth [3], materials science [4, 5], hydrodynamics [6, 7], biomathematics [8, 9], and mean curvature motion [10]. Hence, developing computationally efficient numerical methods is of vital importance.

Various numerical methods have been developed to improve the numerical stability and accuracy of the Allen-Cahn equation over the past two decades [11–14]. In [15], a temporally stable semi-implicit Fourier spectral scheme was employed for time discretization, which rigorously demonstrated the unconditional energy stability of a modified energy functional. For the Allen-Cahn model describing anti-phase domain coarsening in binary mixtures, reference [16] concluded that through an operator splitting approach, the Allen-Cahn equation can be decomposed into linear and nonlinear sub-equations. The linear sub-equation can be solved by a Fourier spectral method, which benefits from an exact analytical solution and consequently imposes no stability restrictions on the time step. Due to the intrinsic singularity of the logarithmic free energy, developing unconditionally stable numerical schemes for the Allen-Cahn equation with logarithmic potential presents significant challenges. These difficulties were successfully addressed by either introducing a stabilization term to the logarithmic energy or employing a regularized potential formulation. In this regard, reference [17] proposed an unconditionally stable splitting method with a special formulation that is physically more meaningful than conventional polynomial potentials. Reference [18] developed a novel second-order temporal approximation for the potential term. Through careful parameter design, this method guarantees both energy stability and unique solvability, demonstrating superior performance compared to other time discretization schemes.

Due to the fundamental property of the Allen-Cahn equation [19], the solution to the Allen-Cahn equation exhibits a monotonic dissipation of the total free energy over the whole time. The design of numerical schemes for phase-field models that satisfy the energy-decay property has been extensively studied in the past years. One widely adopted approach to achieve numerical energy property is the convex splitting method, which was first introduced by Elliott and Stuart in 1993 [20] and further developed by Eyre [21]. Specifically, the semi-implicit convex splitting method by Eyre guarantees that the numerical solution satisfies energy dissipation even in the discrete setting. It implicitly treats the convex part of the chemical potential while explicitly handling the concave part, ensuring the unique solvability and energy stability. Recently, convex splitting has been widely applied to various gradient flows, leading to the development of both first-order and second-order accurate algorithms in time [22–24]. Moreover, the breakthrough work by Eyre has significantly shaped the field of phase-field simulations, serving as a catalyst for the development of many modern time integration methods [25–27].

It is observed that the Allen-Cahn equation contains a small positive parameter in the diffusion term, which often leads to a stiff system after spatial discretization. In such cases, the IMEX technique introduced for time-dependent partial differential equations (PDEs) plays an important role [28]. Such schemes were proposed and analyzed as early as the late 1970s. For the latest developments, we refer to [29, 30]. The IMEX approach typically demonstrates superior stability properties over wide parameter regimes when contrasted with competing schemes. Recent applications span a range of incompressible Navier-Stokes simulations [31] and asymptotic-preserving discretizations for kinetic equations [32, 33], where implicit diffusion or explicit convection treatment remains canonical.

For spatial discretization, early numerical methods were represented by the finite difference method (FDM) [34]. The FDM is intuitive and efficient, but it heavily relies on regular, structured grids, making it difficult to handle complex geometries. In contrast, the finite element method (FEM), due to its exceptional adaptability to complex geometries, has become an indispensable choice in fields such as viscous and compressible flow [35]. Typically, the enriched or extended FEM formulations can capture sharp interfaces or localized phase transitions effectively and accurately, but without excessive mesh refinement [36, 37]. For problems defined in irregular domains, the immersed boundary method and overset grid techniques have been developed over the decades, which allow the mesh grid to be nonconforming with solid boundaries, significantly simplifying mesh generation [38, 39].

Besides these grid-based methods, the boundary integral methods (BIMs) transform differential problems into the boundary integral equations (BIEs) on the domain boundary, thus reducing the dimensionality of the model. The BIE for elliptic BVPs can be transformed into a Fredholm integral equation of the second kind, which can be subsequently solved by the Krylov subspace iterative methods. However, traditional BIEs often involve singular kernel functions [40]. To circumvent the direct computation of these kernels, Ying et al. proposed the KFBI method [41], which essentially eliminates the need to directly compute integrals with singular kernels. In the solution to the equivalent interface problems of the integrals, the coefficient matrix of the discrete system remains unchanged, it can be further solved by a FFT-based elliptic solver, whose computational work is roughly proportional to the number of grid nodes in the Cartesian grid used. As the discrete boundary integral equations are well-conditioned, the iteration converges within an essentially fixed number of steps, which is independent of the mesh parameter. As a result, the KFBI method achieves efficient, accurate, and stable numerical solutions for elliptic problems in general domains [42, 43].

As a combination of the traditional BIM and the Cartesian grid-based FDM, the KFBI method benefits both of their advantages, such as easy grid generation, the reduced dimensionality of the model, and the ease of complex boundary capturing. These implie less computer memory requirements and the convenience in moving boundary description. More importantly, the BIM converts the potential problem into an integral equation, whose discretized form becomes a well-conditioned system, which is totally different from the case of classical FDM and FEM. For constant coefficient elliptic problems with irregular boundaries/interfaces, the discrete system only modifies the right-hand side while the left-hand side remains unchanged. Thus, a series of fast elliptic solvers can be applied. Besides, the KFBI method works on the Cartesian grid with no need of complex mesh generation. All of these reduce the computational complexity of the KFBI method. In the consideration of algorithm accuracy, the accuracy of the KFBI method depends on the accuracy of the FDM used for integral evaluation. In recent years, the KFBI method has developed to be a general purpose fourth-order method for elliptic PDEs with various applications [44–46]. The present work follows the main idea of the KFBI method as the spatial discretization method, providing an alternative perspective in solving the phase field problem, especially for problems in complex domains.

An outline of this paper is presented as follows. In Section 2, we introduce several parameterized IMEX time discretization schemes. Section 3 provides a brief description of BVPs for elliptic PDEs, the equivalent interface problems of the integrals, and the reformulation of BIEs. Section 4 gives a concise introduction of the KFBI method, including the computational modules of discretization, correction, solution, and interpolation. In Section 5, we present a series of numerical experiments. The conclusions of this study and potential extensions are discussed in the final section.

2. Time discretization method

Time discretization of the Allen-Cahn equation (1.1) can be attributed to an extensive system of reaction-diffusion equations, which is often typically in the pattern

$$\frac{du}{dt} = \epsilon\Delta u + G(u), \tag{2.1}$$

where G is a nonlinear function of u, and ϵ is a small positive parameter. To discretize the Eq (2.1) in time, we mainly employ two methods: a second-order SIRK method and a class of two-step IMEX methods. A detailed explanation of these two methods will be provided next.

2.1. Second-order SIRK method

In the SIRK framework, the linear/stiff term Δu is treated implicitly, while the nonlinear/stiff term $G(u)$ is handled explicitly. A typical second-order SIRK scheme reads

$$\begin{aligned} K_0 &= \epsilon\Delta(u^0 + \frac{1}{2}K_0\Delta t) + G(u^0),\\ K_1 &= \epsilon\Delta(u^0 + \frac{1}{2}K_1\Delta t) + G(u^0 + K_0\Delta t),\\ u^{n+1} &= u^n + \frac{1}{2}\Delta t(K_0 + K_1). \end{aligned} \tag{2.2}$$

Here, Δt is the time step, $t^n = n\Delta t$, and $u^n \approx u(\mathbf{p}, t^n)$ represents the approximate solution at the time level t^n. By introducing $v_0 = u^n$, $v_1 = u^n + \frac{1}{2}K_0\Delta t \approx u^{n+\frac{1}{2}}$, and $v_2 = u^n + \frac{1}{2}K_1\Delta t \approx u^{n+\frac{1}{2}}$, then (2.2) can be simplified as follows:

$$\begin{aligned} \frac{v_1 - v_0}{\frac{1}{2}\Delta t} &= K_0 = \epsilon\Delta v_1 + G(v_0),\\ \frac{v_2 - v_1}{\frac{1}{2}\Delta t} &= K_1 = \epsilon\Delta v_2 + G(2v_1 - v_0),\\ u^{n+1} &= v_1 + v_2 - v_0. \end{aligned} \tag{2.3}$$

By rearranging the terms in each formula, we obtain the following form:

$$\begin{aligned}
&-\Delta v_1 + \frac{2}{\epsilon\Delta t} v_1 = \frac{2}{\epsilon\Delta t} v_0 + \frac{1}{\epsilon} G(v_0),\\
&-\Delta v_2 + \frac{2}{\epsilon\Delta t} v_2 = \frac{2}{\epsilon\Delta t} v_0 + \frac{1}{\epsilon} G(2v_1 - v_0),\\
&u^{n+1} = v_1 + v_2 - v_0.
\end{aligned} \tag{2.4}$$

The first two equations are the modified Helmholtz equations in the spatial domain, whose boundary conditions at certain time steps are given by (1.3). Therefore, we can transform (2.4) into two corresponding BVPs

$$\begin{aligned}
&-\Delta v_1 + \frac{2}{\epsilon\Delta t} v_1 = \frac{2}{\epsilon\Delta t} v_0 + \frac{1}{\epsilon} G(v_0) && \text{in } \Omega,\\
&v_1(\mathbf{p}) = g^D(\mathbf{p}, t^{n+\frac{1}{2}}) && \text{on } \Gamma,
\end{aligned} \tag{2.5}$$

and

$$\begin{aligned}
&-\Delta v_2 + \frac{2}{\epsilon\Delta t} v_2 = \frac{2}{\epsilon\Delta t} v_0 + \frac{1}{\epsilon} G(2v_1 - v_0) && \text{in } \Omega,\\
&v_2(\mathbf{p}) = g^D(\mathbf{p}, t^{n+\frac{1}{2}}) && \text{on } \Gamma.
\end{aligned} \tag{2.6}$$

The values of v_1 and v_2 can be obtained by solving the BVPs (2.5) and (2.6), and then substituting them into the formula of u^{n+1} to update its value.

2.2. *Two-step IMEX method*

We still focus on discretizing the Eq (2.1). A k-step IMEX scheme for Eq (2.1) takes the form

$$\frac{1}{\Delta t}\left(u^{n+1} + \sum_{j=0}^{k-1} \alpha_j u^{n-j}\right) = \epsilon \sum_{j=-1}^{k-1} \beta_j \Delta u^{n-j} + \sum_{j=0}^{k-1} \gamma_j G(u^{n-j}), \tag{2.7}$$

where $\beta_{-1} \neq 0$ [47]. Assume that the function $u(t)$ is sufficiently smooth. Applying the Taylor expansion at t^n, we obtain the following truncation error:

$$\begin{aligned}
&\frac{1}{\Delta t}\left(1 + \sum_{j=0}^{k-1} \alpha_j\right) u(t^n) + \left(1 - \sum_{j=1}^{k-1} j\alpha_j\right) u^{(1)}(t^n) + \cdots + \frac{\Delta t^{p-1}}{p!}\left(1 + \sum_{j=1}^{k-1} (-j)^p \alpha_j\right) u^{(p)}(t^n)\\
&-\epsilon \sum_{j=-1}^{k-1} \beta_j \Delta u(t^n) - \epsilon\Delta t\left(\beta_{-1} - \sum_{j=-1}^{s-1} \beta_j\right) \Delta u^{(1)}(t^n) - \cdots - \frac{\epsilon\Delta t^{p-1}}{(p-1)!}\left(\beta_{-1} + \sum_{j=0}^{k-1} \beta_j (-j)^{p-1}\right) \Delta u^{(p-1)}(t^n)\\
&-\sum_{j=0}^{k-1} \gamma_j G(u(t^n)) + \Delta t \sum_{j=1}^{k-1} j\gamma_j G^{(1)} - \cdots - \frac{\Delta t^{p-1}}{(p-1)!} \sum_{j=1}^{k-1} (-j)^{p-1} \gamma_j G^{(p-1)} = o(\Delta t^p).
\end{aligned}$$

Here, the parameter p denotes the order of accuracy of the numerical method, $u^{(p)}(t^n)$ signifies the p-th order derivative of u at time t^n, and $G^{(p)}$ denotes the p-th order derivative of G with respect to $u(t^n)(p \geq 1)$. To achieve p-th order accuracy, it is necessary to ensure that all terms of order lower than Δt^p in the Taylor expansion vanish. These conditions are fulfilled by appropriately choosing the coefficients α_j,β_j, and γ_j, thus preserving high-order convergence during the discretization process.

By analyzing the local truncation error through the original equation, we derive a p-order scheme under the following requirements:

$$
\begin{aligned}
&1+\sum_{j=0}^{k-1}\alpha_j=0,\\
&1-\sum_{j=1}^{k-1}j\alpha_j=\sum_{j=-1}^{k-1}\beta_j=\sum_{j=0}^{k-1}\gamma_j,\\
&\vdots\\
&\frac{1}{p!}\left(1+\sum_{j=1}^{k-1}(-j)^p\alpha_j\right)=\frac{1}{(p-1)!}\left(\beta_{-1}+\sum_{j=1}^{k-1}\beta_j(-j)^{p-1}\right)=\frac{1}{(p-1)!}\sum_{j=1}^{k-1}(-j)^{p-1}.
\end{aligned}
\tag{2.8}
$$

The general k-step IMEX scheme on the order of k depends on the parameter p with ($p \leq k$) [28,48]. We proceed to solve the underdetermined equations (2.8) to derive concrete examples of the second-order IMEX scheme.

2.2.1. Second-order IMEX scheme with parameter

Following the approach in [47], we use an IMEX time integration method, a widely adopted technique for efficiently solving time-dependent problems involving stiffness. According to Eqs (2.2) and (2.7) above, the numerical scheme achieves second-order accuracy when $p = k = 2$, so if we center our schemes in time at $t^{n-\alpha/(4+2\alpha)}$, the second-order IMEX schemes can be written as

$$
\begin{aligned}
&\frac{u^{n+1}+\alpha u^n-(1+\alpha)u^{n-1}}{\Delta t}\\
&=\epsilon\Delta\left((\beta-\frac{\alpha}{2})u^{n+1}+(2+\frac{3}{2}\alpha-2\beta)u^n+\beta u^{n-1}\right)+\left((2+\frac{\alpha}{2})G(u^n)+\frac{\alpha}{2}G(u^{n-1})\right),
\end{aligned}
\tag{2.9}
$$

where α and β are two free parameters. The present scheme is a three-step method so that u^0 and u^1 are indispensable to set up the time evolution. Here, we use the second-order SIRK method to compute u^1. By uniting the like terms and rearranging the sequence, Eq (2.9) can be simplified to a modified Helmholtz equation that

$$
\begin{aligned}
&-\Delta u^{n+1}+\frac{1}{\epsilon\Delta t(\beta-\frac{\alpha}{2})}u^{n+1}\\
&=-\frac{\alpha}{\epsilon\Delta t(\beta-\frac{\alpha}{2})}u^n+\frac{(2+\frac{3}{2}-2\beta)}{\beta-\frac{\alpha}{2}}\Delta u^n+\frac{\beta}{\beta-\frac{\alpha}{2}}\Delta u^{n-1}+\frac{1+\alpha}{\epsilon\Delta t(\beta-\frac{\alpha}{2})}u^{n-1}\\
&+\frac{1}{\epsilon(\beta-\frac{\alpha}{2})}((2+\frac{\alpha}{2})G(u^n)+\frac{\alpha}{2}G(u^{n+1}))\quad(n\geq 2).
\end{aligned}
\tag{2.10}
$$

Together with the Dirichlet boundary condition (1.3) at the time step $t^{n-\alpha/(4+2\alpha)}$, this results in a BVP of the modified Helmholtz equation. The BVP can be solved by the KFBI method, which serves as the spatial discretization method in this work.

2.2.2. Semi-implicit Leap-Frog scheme

Considering the second-order IMEX scheme (2.9), if we set the parameters as $(\alpha,\beta) = (0, 1)$, then we obtain the semi-implicit leapfrog(SILF) scheme:

$$\frac{u^{n+1} - u^{n-1}}{2\Delta t} = \epsilon\Delta\left(\frac{u^{n+1} + u^{n-1}}{2}\right) + G(u^n). \tag{2.11}$$

By uniting the like terms and rearranging the sequence, the above scheme implies the form of the modified Helmholtz equation that

$$-\Delta u^{n+1} + \frac{1}{\epsilon\Delta t}u^{n+1} = \Delta u^{n-1} + \frac{1}{\epsilon\Delta t}u^{n-1} + \frac{2}{\epsilon}G(u^n). \tag{2.12}$$

Together with the Dirichlet boundary condition (1.3) at the corresponding time step $t^{\frac{n}{4}}$, the equation also results in a BVP of the modified Helmholtz equation.

This scheme combines an explicit leapfrog time advancement with an implicit treatment technique. The explicit part adopts explicit second-order center difference for non-rigid or fluctuating terms to maintain energy conservation characteristics, while the implicit part adopts implicit processing for rigid terms (such as diffusion and damping terms) to improve stability. With second-order temporal accuracy, the scheme outperforms the purely explicit Euler method. Moreover, the implicit processing of the rigid term allows for a larger time step, which is the advantage of the SILF scheme.

2.2.3. Crank-Nicolson/Adams-Bashforth schemes

We also investigate an explicit multistep approach, the Crank-Nicolson/Adams-Bashforth (CNAB) method, which is a hybrid time integration scheme that combines the implicit Crank-Nicolson (CN) method with the explicit Adams-Bashforth (AB) method. The second-order CNAB scheme can be simply derived by setting the parameters in (2.9) as $(\alpha,\beta) = (-1, 0)$. For time integration of Eq (2.1), the nonlinear term $G(u)$ is treated explicitly using the AB method, and the diffusion term is handled implicitly through CN. Then we obtain the second-order CNAB scheme:

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{\epsilon}{2}\left(\Delta u^{n+1} + \Delta u^n\right) + \left(\frac{3}{2}G(u^n) - \frac{1}{2}G(u^{n-1})\right). \tag{2.13}$$

By simply rearranging the sequence, this yields a modified Helmholtz equation of the following form

$$-\Delta u^{n+1} + \frac{2}{\epsilon\Delta t}u^{n+1} = \Delta u^n + \frac{2}{\epsilon\Delta t}u^n + \frac{3}{\epsilon}G(u^n) - \frac{1}{\epsilon}G(u^{n-1}), \tag{2.14}$$

which can also be solved by the KFBI method to update u^{n+1}.

2.2.4. Modified CNAB scheme

Considering the numerical scheme with parameters again, if α and β are taken as -1 and $1/16$, respectively, a mixed numerical method can be obtained, namely the modified CNAB scheme. We take the modified CNAB scheme to the Eq (2.1), then it holds

$$\frac{u^{n+1} - u^n}{\Delta t} = \epsilon\Delta\left(\frac{9}{16}u^{n+1} + \frac{3}{8}u^n + \frac{1}{16}u^{n-1}\right) + \left(\frac{3}{2}G(u^n) - \frac{1}{2}G(u^{n-1})\right). \tag{2.15}$$

Similarly, by simply uniting the like terms and rearranging the sequence, the modified Helmholtz equation can be obtained

$$-\Delta u^{n+1} + \frac{16}{9\epsilon\Delta t}u^{n+1} = \frac{2}{3}\Delta u^n + \frac{16}{9\epsilon\Delta t}u^n + \frac{1}{9}\Delta u^{n-1} + \left(\frac{8}{3\epsilon}G(u^n) - \frac{8}{9\epsilon}G(u^{n-1})\right), \tag{2.16}$$

which is then solved by the KFBI method to update u^{n+1}.

3. Spatial discretization method

As can be seen above, all the time discretization schemes (2.9), (2.11), (2.13), and (2.15) mentioned in the previous chapter result in a spatial elliptic equation in the form of the modified Helmholtz equation at each time step. In this section, the modified Helmholtz equation, together with the proper boundary condition, is solved by a Cartesian grid-based boundary integral method.

3.1. Boundary value problem

Let $\Omega \subset \mathcal{R}^2$ be a bounded irregular domain with smooth boundary $\Gamma = \partial\Omega$. Let $\mathbf{p} \in \mathcal{R}^2$ the spatial coordinate. We consider numerical solutions to the Dirichlet BVP

$$\mathcal{L}u \equiv -\Delta u + \kappa u = f \qquad \text{in } \Omega, \tag{3.1}$$

$$u = g^D \qquad \text{on } \Gamma. \tag{3.2}$$

Here, $\mathcal{L}$ denotes the differential operator of the elliptic PDE, g^D represents the Dirichlet boundary data, κ is a constant, $f = f(\mathbf{p})$ is a smooth function defined on Ω, and $u = u(\mathbf{p})$ is the unknown function to be determined. To use the KFBI method, we introduce a rectangular domain $\mathcal{B}$. Then Γ becomes a smooth interface within $\mathcal{B}$ that divides the rectangle into two subdomains Ω_i and Ω_e, so that $\Omega_i \cup \Gamma \cup \Omega_e = \mathcal{B}$. Let G denote the Green's function for the differential operator $\mathcal{L}$ on $\mathcal{B}$, which satisfies the following constraints:

$$\begin{aligned} &\mathcal{L}G = -\Delta_{\mathbf{p}}G(\mathbf{p},\mathbf{q}) + \kappa G(\mathbf{p},\mathbf{q}) = \delta(\mathbf{p}-\mathbf{q}) && \mathbf{p} \in \mathcal{B}, \\ &G(\mathbf{p},\mathbf{q}) = 0 && \mathbf{p} \in \partial\mathcal{B}, \end{aligned} \tag{3.3}$$

for any $\mathbf{q} \in \mathcal{B}$. Here, $\delta(\mathbf{p}-\mathbf{q})$ denotes the Dirac function. According to potential theory [49], the Dirichlet BVP (3.1) and (3.2) can be restated as a Fredholm BIE of the second kind

$$\frac{1}{2}\varphi(\mathbf{p}) + \mathcal{M}\varphi(\mathbf{p}) = g^D(\mathbf{p}) - \mathcal{G}f(\mathbf{p}) \qquad \mathbf{p} \in \Gamma. \tag{3.4}$$

The solution to Dirichlet BVP reads

$$u(\mathbf{p}) = \mathcal{M}\varphi(\mathbf{p}) + \mathcal{G}f(\mathbf{p}) \qquad \mathbf{p} \in \Omega, \tag{3.5}$$

with

$$\mathcal{M}\varphi(\mathbf{p}) = \int_\Gamma \frac{\partial G(\mathbf{q},\mathbf{p})}{\partial \mathbf{n}_\mathbf{q}}\varphi(\mathbf{q})\,ds_\mathbf{q},$$

$$\mathcal{G}f(\mathbf{p}) = \int_\Omega G(\mathbf{q},\mathbf{p})f(\mathbf{q})\,d\mathbf{q}.$$

Here, $\varphi(\mathbf{p})$ is a density function defined on Γ, and $\mathbf{n_q}$ denotes the outward unit normal vector at the boundary point $\mathbf{q} \in \Gamma$. Equation (3.4) can be solved using Krylov subspace methods (such as the GMRES method [50]) or a simple Richardson iteration. As the iteration converges, $u(\mathbf{p})$ can be calculated by the formula (3.5).

3.2. Redefinition of boundary integral equations

All the boundary and volume integrals can be reinterpreted as the solution to their equivalent interface problems. For the piecewise smooth dependent variable $v(\mathbf{p})$ defined on a larger rectangular domain $\mathcal{B}$, which is discontinuous only across the interface Γ, let v^+ and v^- be the restrictions of v in the subdomains Ω_i and Ω_e, respectively. For any interface point $\mathbf{p} \in \Gamma$, $v^+(\mathbf{p})$, and $v^-(\mathbf{p})$ are interpreted as the limiting values of $v(\mathbf{p})$ from the corresponding side of the interface. Hence, the jump of v across the interface Γ can be naturally defined by

$$[v(\mathbf{p})] = v^+(\mathbf{p}) - v^-(\mathbf{p}) \qquad \mathbf{p} \in \Gamma.$$

According to potential theory [49], the double layer potential $v = \mathcal{M}\varphi$ satisfies the following interface problem:

$$\begin{aligned} &\mathcal{L}v = -\Delta v + \kappa v = 0 && \text{in } \mathcal{B} \setminus \Gamma, \\ &[v] = \varphi && \text{on } \Gamma, \\ &[\partial_{\mathbf{n}} v] = 0 && \text{on } \Gamma, \\ &v = 0 && \text{on } \partial\mathcal{B}. \end{aligned} \tag{3.6}$$

This means the double layer potential is discontinuous across the interface. Specifically, the discontinuity of the double layer potential is given by

$$\begin{aligned} v^+(\mathbf{p}) &= \frac{1}{2}\varphi(\mathbf{p}) + \mathcal{M}\varphi(\mathbf{p}) \qquad \mathbf{p} \in \Gamma, \\ v^-(\mathbf{p}) &= -\frac{1}{2}\varphi(\mathbf{p}) + \mathcal{M}\varphi(\mathbf{p}) \quad\ \mathbf{p} \in \Gamma. \end{aligned}$$

The volume integral $v = \mathcal{G}f$ satisfies the simple interface problem

$$\begin{aligned} &\mathcal{L}v = -\Delta v + \kappa v = \begin{cases} f & \text{in } \Omega, \\ 0 & \text{in } \Omega^c, \end{cases} \\ &[v] = 0 && \text{on } \Gamma, \\ &[\partial_{\mathbf{n}} v] = 0 && \text{on } \Gamma, \\ &v = 0 && \text{on } \partial\mathcal{B}. \end{aligned} \tag{3.7}$$

This means the volume potential and its normal derivative are continuous across the interface. Due to the continuity or discontinuity of the boundary and volume potentials, we can rewrite the BIE (3.4) in the following simpler form:

$$(\mathcal{M}\varphi)^+(\mathbf{p}) = g^D(\mathbf{p}) - (\mathcal{G}f)^+(\mathbf{p}) \qquad \mathbf{p} \in \Gamma, \tag{3.8}$$

where the superscript '+' represents the one-sided limit from the domain Ω_i.

The homogeneous Dirichlet boundary conditions in (3.6) and (3.7) hold since the involved Green's function satisfies the same Dirichlet zero condition as supposed in its definition (3.3). Based on the description in Section 3.1, equivalent simple interface problems (3.6) and (3.7) can be expressed in a unified form

$$\begin{aligned} &\mathcal{L}v = -\Delta v + \kappa v = \mathrm{F} && \text{in } \mathcal{B} \setminus \Gamma, \\ &[v] = \Phi && \text{on } \Gamma, \\ &[\partial_{\mathbf{n}} v] = \Psi && \text{on } \Gamma, \\ &v = 0 && \text{on } \partial\mathcal{B}. \end{aligned} \tag{3.9}$$

On the right-hand side F, Φ, and Ψ are determined according to the potentials of different types. In the context of the KFBI method, we first iteratively solve the reformulated BIE (3.8) with the Krylov subspace method, and then substitute the approximate density function φ into the solution representation formula (3.5) to get the approximation of u. Through the whole calculation process, all the integrals are evaluated by solving the equivalent and simple interface problem (3.9) with a Cartesian grid-based boundary integral method, which will be explained in detail in the next section.

4. Kernel-free boundary integral method

The KFBI method is an extension of the classical boundary integral method, which enables the solution of elliptic PDEs within the framework of BIEs [41]. Since the Green's function or the kernel involved in boundary integrals for elliptic PDEs are generally not available in a closed form [51], directly computing these boundary integrals is typically challenging or even infeasible. To address this, the KFBI substitutes the direct evaluation of boundary or volume integrals with interpolation from a grid-based solution that satisfies a discretized interface problem on a larger, regular grid. This section provides a detailed explanation of the method for two-dimensional space.

4.1. Discretization with corrected finite difference scheme

In the two-dimensional case, suppose $\mathcal{B} = (a, b) \times (c, d)$ is a square domain introduced in (3.9). We partition this square uniformly into an M × M grid along both the horizontal and vertical directions, where M denotes the number of grid subdivisions. After partitioning, the size of the grid is given by $h = (b - a)/\mathrm{M} = (d - c)/\mathrm{M}$. The grid nodes/points are denoted by (x_i, y_j), where $x_i = a + ih$ and $y_j = c + jh$, with $i, j = 0, 1, 2, ..., \mathrm{M}$. For a general function v defined in $\mathcal{B}$, we denote $v_{i,j} \approx v(x_i, y_j)$ as its grid approximation. For a given function f, we write $f_{i,j} = f(x_i, y_j)$ as its value at the nodes of the grid. Using traditional second-order partial differential approximations [52], we derive the following five-point finite difference scheme for the modified Helmholtz equation:

$$\mathcal{L}_h v_{i,j} \equiv \frac{4v_{i,j} - v_{i-1,j} - v_{i+1,j} - v_{i,j-1} - v_{i,j+1}}{h^2} + \kappa v_{i,j} = f_{i,j}. \tag{4.1}$$

Here, $\mathcal{L}_h$ denotes the second-order finite difference operator for the control equation of (3.9). On the Cartesian grid, all the grid points are classified into two categories: regular points and irregular points. A grid point is designated as irregular if the five-point compact difference stencil intersects with the interface Γ (as illustrated in Figure 1); otherwise it is classified as regular. The interface Γ divides the rectangular domain into two distinct regions, resulting in a discontinuous solution throughout the domain and significant local truncation errors near the interface. Since irregular points exhibit excessively large local truncation errors, we should identify and correct all such points to ensure the global accuracy of the finite difference scheme. To formally achieve second-order accuracy of the approximate solution, the finite difference scheme (4.1) is corrected accordingly into the form

$$\mathcal{L}_h v_{i,j} = \begin{cases} f_{i,j} & \text{if } (x_i, y_j) \text{ is regular,} \\ f_{i,j} + \frac{1}{h^2} \sum_{(x_{i+r}, y_{j+s}) \in T_{i,j}^{(5)} \cap \Omega_e} C_{i+r,j+s} & \text{if } (x_i, y_j) \text{ is irregular.} \end{cases} \tag{4.2}$$

In the second case, (x_i, y_j) is typically supposed to be an irregular grid point inside Ω_i, and $T_{i,j}^{(5)}$ denotes the corresponding five-point finite difference template. The summation is known as the correction of the modified scheme with $C_{i+r,j+s}$ being the correction term at the neighboring grid point involved in the template but belonging to the opposite side Ω_e. As the primary cause of the large local truncation error, the correction term $C_{i+r,j+s}$ is defined by

$$C_{i+r,j+s} := v^+(x_{i+r}, y_{j+s}) - v^-(x_{i+r}, y_{j+s}). \tag{4.3}$$

The superscript "+" means the corresponding grid point is located in the domain Ω_i (referred to as the interior point), while "−" suggests that the grid point is inside the domain Ω_e (referred to as the exterior point). Therefore, v^+ and v^- are smooth extension functions of v from two different sides Ω_i and Ω_e, respectively.

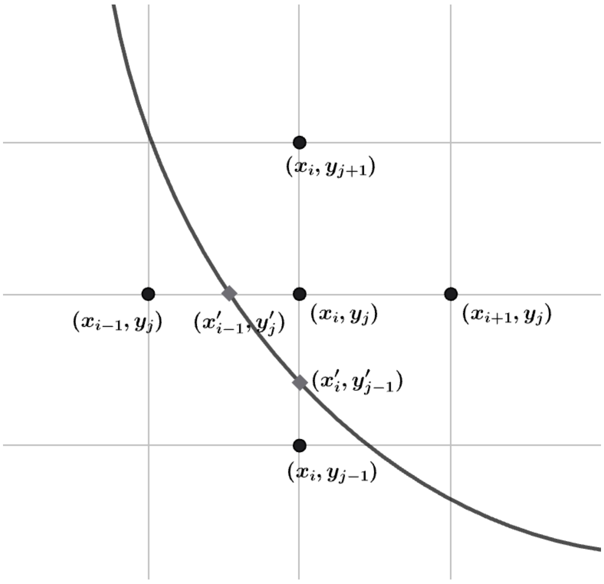


Figure 1. A five-point compact difference template at an irregular point (x_i, y_j) (black circle), intersections of the grid lines, and the interface (gray diamond).

The underlying reason for this definition is plotted in Figure 2, which gives an illustration of a degraded one-dimensional case. For a piecewise smooth function near an interface point z_i, the standard centered difference scheme at x_i has large local truncation error. To eliminate the leading order of the truncation error, at the grid point x_{i+1}, which is on the other side of the interface compared to x_i, the expected ghost function value v_{i+1}^+ can be calculated by adding the real function value v_{i+1}^-

and the difference between them, which is indeed the correction term defined there. In this way, the discontinuities of the piecewise function can be approximately removed in a local sense.

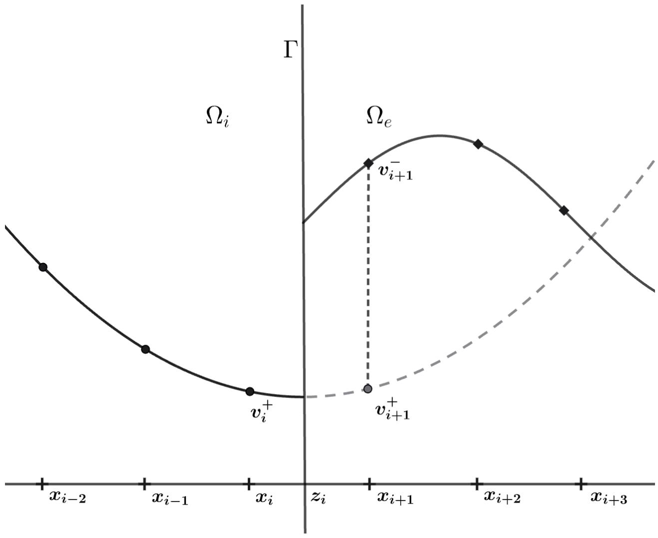


Figure 2. A piecewise smooth function along a straight line with an interface, where z_i is the interface point. The real function v^- in Ω_e (full line) is replaced by the corresponding ghost function v^+ (dashed line).

In this work, two kinds of correction techniques are applied to this end. The first one is the Taylor expansion, so that $C_{i+r,j+s}$ can be estimated by

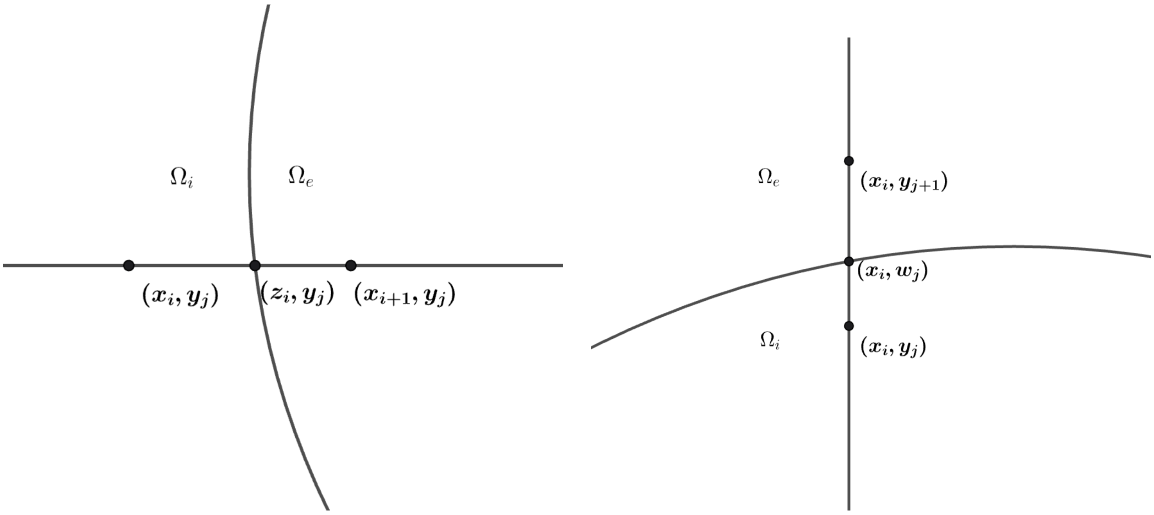
$$\begin{aligned} & v^+(x_{i+r}, y_{j+s}) - v^-(x_{i+r}, y_{j+s}) \\ &= \sum_{n=0}^{2} \frac{1}{n!} \left(\xi_{i+r} \frac{\partial}{\partial x} + \eta_{j+s} \frac{\partial}{\partial y} \right)^n \left\{ v^+(x'_{i+r}, y'_{j+s}) - v^-(x'_{i+r}, y'_{j+s}) \right\} + O(h^3) \\ &= \sum_{n=0}^{2} \frac{1}{n!} \left(\xi_{i+r} \frac{\partial}{\partial x} + \eta_{j+s} \frac{\partial}{\partial y} \right)^n [v](x'_{i+r}, y'_{j+s}) + O(h^3). \end{aligned} \tag{4.4}$$

Here, (x'_{i+r}, y'_{j+s}) is the intersection point between the centering grid point $(x_i, y_j) \in \Omega_i$ and the neighboring grid point $(x_{i+r}, y_{j+s}) \in \Omega_e$ (see Figure 1 for illustration), and $\xi_{i+r} = x_{i+r} - x'_{i+r}$ and $\eta_{j+s} = y_{j+s} - y'_{j+s}$ are displacements in the horizontal and vertical directions between them. $v^+(x'_{i+r}, y'_{j+s})$ and $v^-(x'_{i+r}, y'_{j+s})$ are the one-sided limit of v at the interface point (x'_{i+r}, y'_{j+s}) from the domain Ω_i or Ω_e, respectively. The abbreviation of derivative operation means

$$\begin{aligned} \left(\xi_{i+r} \frac{\partial}{\partial x} + \eta_{j+s} \frac{\partial}{\partial y} \right)^n \cdot &= \sum_{m=0}^{n} \frac{n!}{m!(n-m)!} \xi_{i+r}^m \eta_{j+s}^{n-m} \frac{\partial^n \cdot}{\partial x^m \partial y^{n-m}}, \\ \left(\xi_{i+r} \frac{\partial}{\partial x} + \eta_{j+s} \frac{\partial}{\partial y} \right)^n [\,\cdot\,] &= \sum_{m=0}^{n} \frac{n!}{m!(n-m)!} \xi_{i+r}^m \eta_{j+s}^{n-m} \left[\frac{\partial^n \cdot}{\partial x^m \partial y^{n-m}} \right], \end{aligned}$$

where $[\,\cdot\,]$ denotes the jump of the involved function. Hence, the difference is finally represented in terms of a Taylor expansion-based linear combination of the jumps of v.

The above formulas are given in a general form of two-dimensional Taylor expansion. However, we remark that since the five points in the finite difference template are all along the grid lines compared to the centering point (x_i, y_j), the Taylor expansion is actually the one-dimensional Taylor expansion (see Figure 3 for illustration), and the involved jumps are exactly $[v]$, $[v_x]$, $[v_{xx}]$, $[v_y]$, and $[v_{yy}]$ only.



(a) Correction in horizontal direction. Intersection point (z_i, y_j) of the interface and line segment between (x_i, y_j) and (x_{i+1}, y_j) in the five-point template.

(b) Correction in vertical direction. Intersection point (x_i, w_j) of the interface and line segment between (x_i, y_j) and (x_i, y_{j+1}) in the five-point template.

Figure 3. Schematic diagram of the intersections of the interface and grid lines in the horizontal directions and vertical directions. Taylor expansions is performed at the horizontal intersection point (z_i, y_j) or the vertical intersection point (x_i, w_j).

Since the jumps of v are all computable quantities, the correction term can be numerically computed. The jumps in the partial derivatives of the function v can be determined from the jumps of both the function itself and the two interface conditions of (3.9) [41, 53]. After the correction procedure, the local truncation error of the finite difference scheme is reduced to first-order at irregular grid points. Even so, the global second-order convergence can be guaranteed in the numerical solution. The modified finite difference scheme maintains the same coefficient matrix as the original standard five-point compact finite difference scheme. Consequently, the modified discrete system can also be solved by an FFT-based fast elliptic solver.

The second approach is based on a correction function method, where the correction term in (4.3) is regarded as a smooth function and defined as a solution of a local Cauchy problem [42]:

$$\begin{aligned} & -\Delta C(\mathbf{x}) + \kappa C(\mathbf{x}) = \bar{\mathrm{F}}(\mathbf{x}) && \mathbf{x} \in \Omega_\Gamma, \\ & C(\mathbf{x}) = \Phi(\mathbf{x}) && \mathbf{x} \in \Gamma \cap \Omega_\Gamma, \\ & \partial_{\mathbf{n}} C(\mathbf{x}) = \Psi(\mathbf{x}) && \mathbf{x} \in \Gamma \cap \Omega_\Gamma. \end{aligned} \tag{4.5}$$

Here, $\bar{\mathrm{F}} = [\mathrm{F}]$, Ω_Γ is a small partition of the narrow band in a neighborhood of a given interface point $\mathbf{x} \in \Gamma$. The local Cauchy problem can be solved by a mesh-free collocation method. To get the second-order accuracy, we use $\phi_0 = 1$, $\phi_1 = x$, $\phi_2 = y$, $\phi_3 = x^2$, $\phi_4 = y^2$, and $\phi_5 = xy$, the basis of the Taylor polynomial of degree no more than two, as the collection functions. Collocation points consist of three parts for each identity in (4.5) satisfied, namely grid collocation points for control equation satisfaction, interface collocation points for Dirichlet condition satisfaction, and also interface collocation points for Neumann condition satisfaction. The number of three kinds of collocation points are one, three and two, respectively. Distribution of those points are plotted in Figure 4.

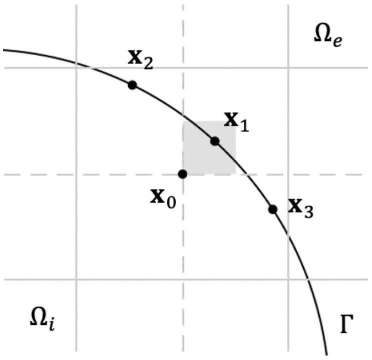


Figure 4. Collocation points stencil $\mathbf{x}_i$ ($i = 0, 1, 2, 3$) used for correction evaluation at a given interface point $\mathbf{x}$ in the shaded region. The stencil is determined by the quadrant of interface point $\mathbf{x}$ in a grid cell compared to the closest grid point.

For a given interface point $\mathbf{x}$ in the shaded region, the corresponding correction term $C(\mathbf{x})$ is computed in the following steps: 1) Find the closest grid point $\mathbf{x}_0$, the closest discrete interface point $\mathbf{x}_1$, and its two neighboring points $\mathbf{x}_2$ and $\mathbf{x}_3$. Choose $\mathbf{x}_0$ for control equation satisfaction, choose the three consecutive interface points $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ for the Dirichlet condition, and the two adjacent interface points $\mathbf{x}_2$, and $\mathbf{x}_3$ for the Neumann condition. 2) Express the approximate correction function $C(\mathbf{x})$ as a linear combination of the basis

$$C(\mathbf{x}) = \sum_{i=0}^{5} d_i \phi_i(\mathbf{x}) \qquad \mathbf{x} \in \Omega_\Gamma,$$

where d_i is the linear combination coefficients to be determined and $\mathbf{y}_i$ is the collection points ordered in sequence, $\mathbf{y}_0 = \mathbf{x}_0$, $\mathbf{y}_1 = \mathbf{x}_1$, $\mathbf{y}_2 = \mathbf{x}_2$, $\mathbf{y}_3 = \mathbf{x}_3$, $\mathbf{y}_4 = \mathbf{x}_2$, and $\mathbf{y}_5 = \mathbf{x}_3$. 3) Set up the finite-dimensional interpolate problem

$$\begin{aligned}
&\sum_{i=0}^{5} (-\Delta + \kappa) d_i \phi_i(\mathbf{y}_j) = \bar{\mathrm{F}}(\mathbf{y}_j) && j = 0, \\
&\sum_{i=0}^{5} d_i \phi_i(\mathbf{y}_j) = \Phi(\mathbf{y}_j) && j = 1, 2, 3, \\
&\sum_{i=0}^{5} d_i \mathbf{n}(\mathbf{y}_j) \cdot \nabla \phi_i(\mathbf{y}_j) = \Psi(\mathbf{y}_j) && j = 2, 3.
\end{aligned} \tag{4.6}$$

After solving the six by six system, one can get the combination coefficients d_i to further determine the correction function $C(\mathbf{x})$. The correction function method provides an accurate and stable technique but doesn't need to know the jumps as in the Taylor expansion-based method. So, it is more efficient to a certain extent.

4.2. Interpolation for integral values on the boundary

After solving the linear system of the corrected finite difference equation (4.2), one can obtain the second-order grid-based numerical solution. However, Dirichlet one-sided boundary data is repeatedly needed in the KFBI method as shown in the BIE (3.8). In order to extract boundary data of the grid-based numerical solution, a quadratic polynomial interpolation is used here.

To set up a two-dimensional quadratic interpolation, six distinct grid points should be selected where Taylor expansions are performed to construct a linear system. For any discrete point $\mathbf{q}$ on the interface Γ, we first select the closest grid point and denote it $\mathbf{p}_1$. The four adjacent grid points surrounding $\mathbf{p}_1$ (up, down, left, and right) are chosen and labeled with $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$, respectively. The sixth point $\mathbf{p}_6$ is chosen as the remaining point of the square where $\mathbf{q}$ is located (see Figure 5 for illustration).

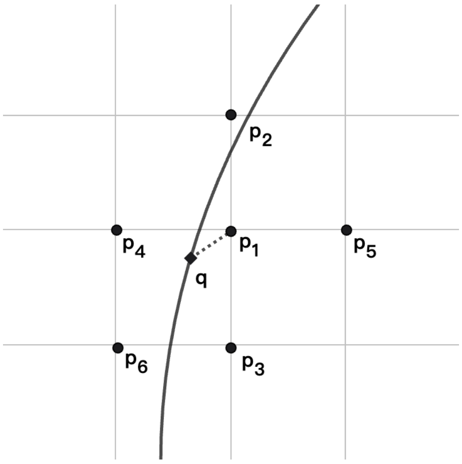


Figure 5. Six-point template for quadratic interpolation.

We assume that the grid-based piecewise smooth solution obtained from the corrected finite difference equation (4.2) also satisfies the jump interface conditions in (3.9). Then the Taylor expansion of v at any grid point $\mathbf{p} \in \mathcal{B}$ takes the form

$$\begin{aligned} v(\mathbf{p}) = v^+(\mathbf{q}) &+ \frac{\partial v^+(\mathbf{q})}{\partial x}\zeta + \frac{\partial v^+(\mathbf{q})}{\partial y}\eta + \frac{1}{2}\frac{\partial^2 v^+(\mathbf{q})}{\partial x^2}\zeta^2 \\ &+ \frac{\partial^2 v^+(\mathbf{q})}{\partial x \partial y}\zeta\eta + \frac{1}{2}\frac{\partial^2 v^+(\mathbf{q})}{\partial y^2}\eta^2 + O(|\mathbf{p}-\mathbf{q}|^3), \qquad \text{if } \mathbf{p} \in \Omega_i, \end{aligned}$$

$$\begin{aligned} v(\mathbf{p}) = v^-(\mathbf{q}) &+ \frac{\partial v^-(\mathbf{q})}{\partial x}\zeta + \frac{\partial v^-(\mathbf{q})}{\partial y}\eta + \frac{1}{2}\frac{\partial^2 v^-(\mathbf{q})}{\partial x^2}\zeta^2 \\ &+ \frac{\partial^2 v^-(\mathbf{q})}{\partial x \partial y}\zeta\eta + \frac{1}{2}\frac{\partial^2 v^-(\mathbf{q})}{\partial y^2}\eta^2 + O(|\mathbf{p}-\mathbf{q}|^3), \qquad \text{if } \mathbf{p} \in \Omega_e. \end{aligned}$$

Here, $(\zeta, \eta)^{\mathrm{T}} = \mathbf{p} - \mathbf{q}$. For simplicity, the following expression is used to represent the approximate solution $v^{\pm}$ and its partial derivatives:

$$
\begin{aligned}
w^{\pm} &= v^{\pm}(\mathbf{q}), & w_x^{\pm} &= \frac{\partial v^{\pm}(\mathbf{q})}{\partial x}, & w_y^{\pm} &= \frac{\partial v^{\pm}(\mathbf{q})}{\partial y}, \\
w_{xx}^{\pm} &= \frac{\partial^2 v^{\pm}(\mathbf{q})}{\partial x^2}, & w_{xy}^{\pm} &= \frac{\partial^2 v^{\pm}(\mathbf{q})}{\partial x \partial y}, & w_{yy}^{\pm} &= \frac{\partial^2 v^{\pm}(\mathbf{q})}{\partial y^2}.
\end{aligned}
$$

Then the Taylor series at each interpolation point $\mathbf{p}_i$ $(i = 1,\ 2, \cdots,\ 6)$ can be simply expressed as

$$
w^{+} + w_x^{+}\zeta_j + w_y^{+}\eta_j + \frac{1}{2}\zeta_j^2 w_{xx}^{+} + \zeta_j\eta_j w_{xy}^{+} + \frac{1}{2}\eta_j^2 w_{yy}^{+} = w_j \qquad \text{if } \mathbf{p}_j \in \Omega_i, \tag{4.7}
$$

and

$$
w^{-} + w_x^{-}\zeta_j + w_y^{-}\eta_j + \frac{1}{2}\zeta_j^2 w_{xx}^{-} + \zeta_j\eta_j w_{xy}^{-} + \frac{1}{2}\eta_j^2 w_{yy}^{-} = w_j \qquad \text{if } \mathbf{p}_j \in \Omega_e, \tag{4.8}
$$

with $w_j = v(\mathbf{p}_j)$ and $(\zeta_j, \eta_j)^{\mathrm{T}} = \mathbf{p}_j - \mathbf{q}$. However, only the boundary datas labeled with "+", which means the one-sided limit from the subdomain Ω_i, are needed in the KFBI method. Hence, the corresponding terms labeled with "−" are expected to be replaced by their counterpart labeled with "+". For this end, we introduce the following corrected quadratic interpolation scheme for $\mathbf{p}_j \in \Omega_e$:

$$
w^{+} + w^{+}\zeta_j + w_j^{+}\eta_j + \frac{1}{2}\zeta_j^2 w_{xx}^{+} + \zeta_j\eta_j w_{xy}^{+} + \frac{1}{2}\eta_j^2 w_{yy}^{+} = w_j + C(\mathbf{p}_j) \qquad \text{if } \mathbf{p}_j \in \Omega_e. \tag{4.9}
$$

Here, $C(\mathbf{p}_j)$ is the correction term as similarly defined in (4.3) to eliminate the discontinuities of v near the interface. Thus, $C(\mathbf{p}_j)$ can be computed either by the correction function method or linear combination in truncated Taylor series that

$$
C(\mathbf{p}_j) = [w] + [w_x]\zeta_j + [w_y]\eta_j + \frac{1}{2}[w_{xx}]\zeta_j^2 + [w_{xy}]\zeta_j\eta_j + \frac{1}{2}[w_{yy}]\eta_j^2. \tag{4.10}
$$

From Eqs (4.7) and (4.9), we obtain a well-posed linear system involving six unknowns, which is then solved using any direct or simple iterative methods, such as LU decomposition and CG iteration. The derived first unknown w^{+} exactly serves as the Dirichlet one-sided boundary data, which is then inserted into the BIE (3.8). As can be seen in the BIE (3.8), the volume integral $(\mathcal{G}f)^{+}$ only exists on the right-hand side of the equation, thus interpolation for computing $(\mathcal{G}f)^{+}$ is performed before the iteration is initiated. The boundary integral $(\mathcal{M}\varphi)^{+}$ actually represents an operator on the unknown density φ, so interpolation for this term is applied at every step of the loop until the iteration converges.

We note that the involved jumps in either correction or interpolation modules (Eqs (4.4) and (4.10)) can be computed in advance due to the control equation and the two jump conditions of the interface problem (3.9). Specific computation details refer to [54].

5. Numerical results

In this section, numerical experiments for initial boundary value problems of the Allen-Cahn equation on irregular domains with the proposed KFBI are presented to demonstrate the numerical accuracy, efficiency, and stability. For convenience, in the following examples, the bounding box $\mathcal{B}$ is set to be $[-1.2, 1.2] \times [-1.2, 1.2]$, and the computational domain is prescribed as elliptic domain, tree-shaped domain, and shell-shaped domain, respectively. The analytical formula for the boundary curve of the shell-shaped domain is given by

$$\begin{cases} x = [(1-c) + c\cos(2\cdot\theta)]\cos\theta \\ y = [(1-c) + c\cos(\theta)]\sin\theta - 0.2 \end{cases} \quad \text{for } \theta \in [0,\ 2\pi). \tag{5.1}$$

The analytical formulas for the boundary curves of the tree-shaped domain are given by

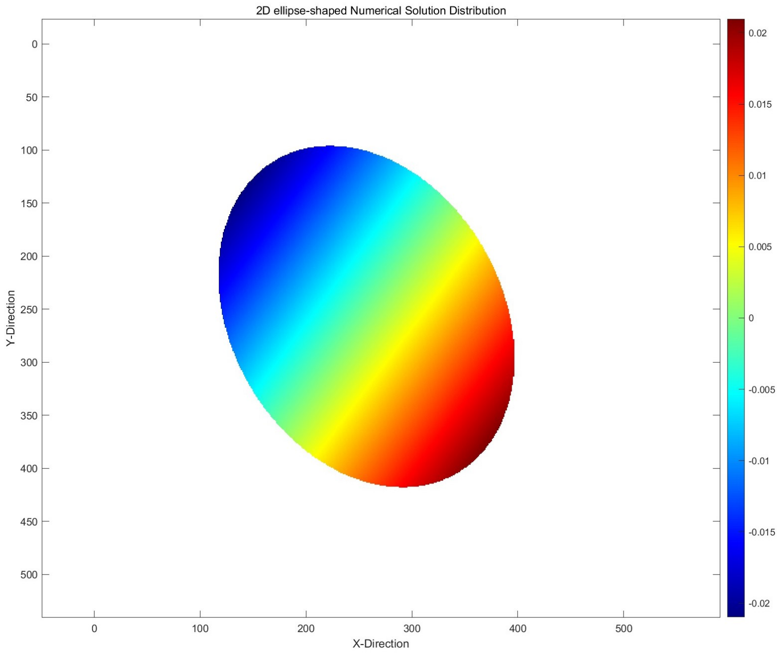
$$\begin{cases} x = [(1-c) - c\cos(4\cdot\theta)]\cos\theta \\ y = [(1-c) - c\cos(3\cdot\theta)]\sin\theta - 0.1 \end{cases} \quad \text{for } \theta \in [0, 2\pi). \tag{5.2}$$

In the following examples, the boundary condition is supposed as a Dirichlet zero condition (1.3), and the initial condition is given by

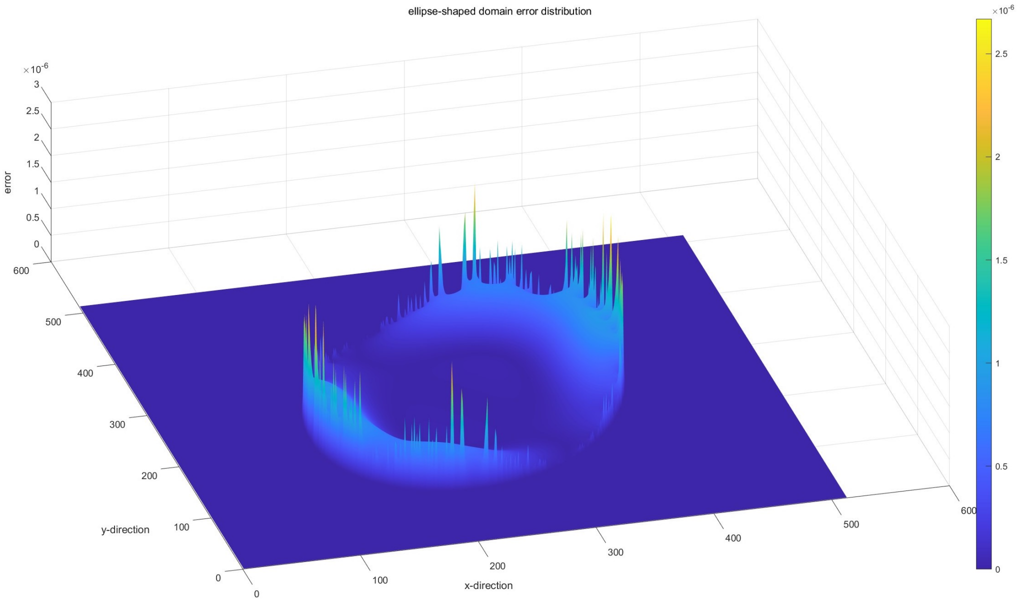
$$u_0(x, y) = \sin(\epsilon(0.6x + 0.8y)) \qquad (x, y) \in \Gamma. \tag{5.3}$$

Therefore, the examples considered in this work have no exact solution with analytical expression. To test the numerical accuracy, we adopt the numerical scheme described in Section 2 and use the numerical solution obtained with a grid size of 1024×1024 and a time step $\Delta t = 1/512$ as the exact/reference solution. Numerical accuracy is verified at terminal time $T = 1.0$ with the error measured in the maximum norm and denoted by $\|u_h - u_{ref}\|_\infty$ hereafter. We denote M as the grid number along x or y direction. Then the number of the quasi-uniformly spaced points on the boundary curve is fixed to be $2M$ throughout. The perturbation parameter ϵ is set to be 0.1, 0.01, and 0.001, respectively. The tolerance for the GMRES method in solving the BIE is fixed to be 10^{-9}, and the average iteration number of the GMRES method over the time evolution is denoted by $\#\text{GMRES}_{ave}$. The GMRES iteration always starts with the zero initial guess. The maximum number of iterations is 1000. The upper bound of the restart parameter is set to be 400. When the mesh parameter M is less than the upper bound, we set the restart parameter to be equal to M. In the KFBI method, the correction and interpolation are implemented based on Taylor expansion or correction function. As shown in the following results, we can observe that the maximum norm of the calculation error roughly exhibits second-order accuracy.

Example 1. This example solves the Allen-Cahn equation on an ellipse domain whose radii are set to be 0.8 and 0.6, respectively. The nonlinear term f is supposed as $f(u) = u^3 - u$ [1]. In this example, the correction and interpolation procedure in the KFBI method are both implemented with the Taylor expansion based scheme. For time integration, the IMEX scheme (2.9) with $(\alpha,\beta) = (-0.75, 0.5)$ and $(\alpha,\beta) = (-1, 0.5)$, the SILF scheme (2.11), the CNAB scheme (2.13), as well as the modified CNAB scheme (2.15) are used to conduct an effect comparison. It can be seen from the tables that the IMEX scheme with the parameters $(\alpha,\beta) = (-0.75, 0.5)$ relatively performs best from the aspects of accuracy and efficiency, especially for small perturbation parameter ϵ. Figure 6 shows the distribution of the numerical solution and discrete errors in the computational domain of this example.



(a) Solution distribution



(b) Error distribution

Figure 6. Figure (a) shows the 2D ellipse-shaped numerical solution distribution in Example 1. Figure (b) shows the 3D error distribution on 512×512 grid when T = 1.0 and $\epsilon = 0.01$.

Table 1. Numerical results of Example 1 by the second-order IMEX scheme with $(\alpha,\beta) = (-0.75, 0.5)$.

ϵ	grid size	64×64	128×128	256×256	512×512
$\epsilon = 10^{-1}$	time step	1/32	1/64	1/128	1/256
	$\|u_h - u_{ref}\|_\infty$	1.75E-4	4.54E-5	1.21E-5	2.94E-6
	#GMRES$_{ave}$	12	11	13	15
	CPU time	1.70E+01	2.15E+02	2.20E+02	2.61E+03
	MEM	1.87E+04	1.89E+04	1.96E+04	2.03E+04
$\epsilon = 10^{-2}$	time step	1/32	1/64	1/128	1/256
	$\|u_h - u_{ref}\|_\infty$	2.00E-4	4.44E-5	1.18E-5	2.66E-6
	#GMRES$_{ave}$	15	14	12	11
	CPU time	2.53E+03	2.88E+03	2.93E+03	3.34E+03
	MEM	1.88E+04	1.91E+04	2.01E+04	2.04E+04
$\epsilon = 10^{-3}$	time step	1/32	1/64	1/128	1/256
	$\|u_h - u_{ref}\|_\infty$	5.04E-4	1.32E-4	2.09E-5	5.78E-6
	#GMRES$_{ave}$	19	18	18	15
	CPU time	3.12E+03	3.55E+03	3.82E+03	4.21E+03
	MEM	1.88E+04	1.92E+04	2.02E+04	2.05E+04

Table 2. Numerical results of Example 1 by the second-order IMEX scheme with $(\alpha,\beta) = (-1, 0.5)$.

ϵ	grid size	64×64	128×128	256×256	512×512
$\epsilon = 10^{-1}$	time step	1/32	1/64	1/128	1/256
	$\|u_h - u_{ref}\|_\infty$	1.93E-3	4.72E-4	1.20E-4	2.62E-5
	#GMRES$_{ave}$	13	13	12	15
	CPU time	4.40E+03	5.18E+02	5.20E+02	5.61E+03
	MEM	1.87E+04	1.89E+04	1.96E+04	2.03E+04
$\epsilon = 10^{-2}$	time step	1/32	1/64	1/128	1/256
	$\|u_h - u_{ref}\|_\infty$	1.21E-3	3.14E-4	8.07E-5	1.97E-5
	#GMRES$_{ave}$	16	14	12	11
	CPU time	4.37E+03	5.17E+03	5.22E+03	5.60E+03
	MEM	1.88E+04	1.91E+04	2.02E+04	2.05E+04
$\epsilon = 10^{-3}$	time step	1/32	1/64	1/128	1/256
	$\|u_h - u_{ref}\|_\infty$	9.72E-4	1.81E-4	4.21E-5	1.01E-5
	#GMRES$_{ave}$	18	16	15	15
	CPU time	4.30E+03	5.02E+03	5.10E+03	5.41E+03
	MEM	1.89E+04	1.92E+04	2.05E+04	2.07E+04

Table 3. Numerical results of Example 1 with the second-order SILF scheme.

ϵ	grid size	64×64	128×128	256×256	512×512
$\epsilon = 10^{-1}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	1.90E-3	4.89E-4	1.15E-4	2.77E-5
	#GMRES$_{ave}$	14	13	11	14
	CPU time	4.25E+03	5.04E+03	5.07E+03	5.42E+03
	MEM	1.85E+04	1.88E+04	1.93E+04	2.00E+04
$\epsilon = 10^{-2}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	1.65E-3	4.33E-4	1.16E-4	2.83E-5
	#GMRES$_{ave}$	15	14	14	11
	CPU time	4.33E+03	5.07E+03	5.13E+03	5.50E+03
	MEM	1.87E+04	1.89E+04	2.00E+04	2.03E+04
$\epsilon = 10^{-3}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	1.34E-3	4.11E-4	1.01E-4	2.52E-5
	#GMRES$_{ave}$	16	15	15	13
	CPU time	4.50E+03	5.26E+03	5.34E+03	5.67E+03
	MEM	1.88E+04	1.90E+04	2.01E+04	2.05E+04

Table 4. Numerical results of Example 1 with the second-order CNAB scheme.

ϵ	grid size	64×64	128×128	256×256	512×512
$\epsilon = 10^{-1}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	2.33E-3	6.21E-4	1.66E-4	4.21E-5
	#GMRES$_{ave}$	12	11	12	14
	CPU time	4.36E+03	5.02E+03	5.07E+03	5.46E+03
	MEM	1.82E+04	1.97E+04	1.99E+04	2.05E+04
$\epsilon = 10^{-2}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	8.09E-3	9.77E-4	2.26E-4	5.49E-5
	#GMRES$_{ave}$	17	15	14	12
	CPU time	4.33E+03	4.87E+03	4.93E+03	5.32E+03
	MEM	1.78E+04	1.80E+04	1.92E+04	2.00E+04
$\epsilon = 10^{-3}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	9.24E-3	2.01E-3	4.78E-4	1.12E-4
	#GMRES$_{ave}$	19	18	16	14
	CPU time	4.31E+03	4.80E+03	4.88E+03	5.27E+03
	MEM	1.74E+04	1.76E+04	1.79E+04	1.84E+04

Table 5. Numerical results of Example 1 with the second-order modified CNAB scheme.

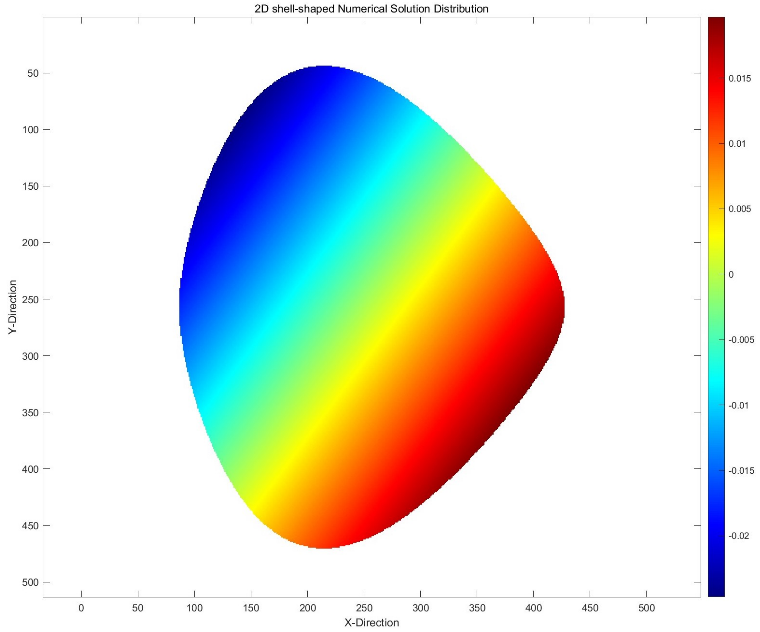
ϵ	grid size	64×64	128×128	256×256	512×512
$\epsilon = 10^{-1}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	1.94E-3	4.47E-4	1.19E-4	3.16E-5
	#GMRES$_{ave}$	16	15	12	12
	CPU time	4.37E+03	5.10E+03	5.15E+03	5.53E+03
	MEM	1.80E+04	1.92E+04	1.93E+04	2.01E+04
$\epsilon = 10^{-2}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	4.29E-3	7.79E-4	1.89E-4	4.61E-5
	#GMRES$_{ave}$	15	15	13	12
	CPU time	4.30E+03	4.83E+03	4.88E+03	5.27E+03
	MEM	1.72E+04	1.79E+04	1.90E+04	1.99E+04
$\epsilon = 10^{-3}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	6.31E-3	1.50E-3	3.58E-4	8.39E-5
	#GMRES$_{ave}$	14	13	13	10
	CPU time	4.25E+03	4.67E+03	4.73E+03	5.02E+03
	MEM	1.60E+04	1.67E+04	1.75E+04	1.86E+04

Table 6. Numerical results of Example 2 obtained by the second-order IMEX scheme with $(\alpha,\beta) = (-1.0, 0.5)$.

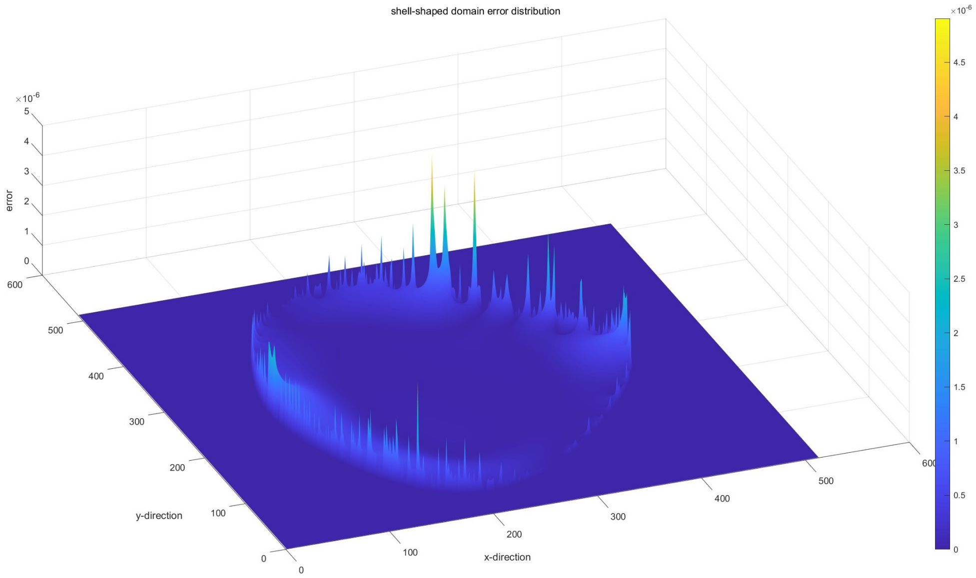
ϵ	grid size	64×64	128×128	256×256	512×512
$\epsilon = 10^{-1}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	1.63E-4	5.74E-5	1.62E-5	4.67E-6
	#GMRES$_{ave}$	10	10	12	13
	CPU time	7.68E+02	7.99E+02	8.11E+03	8.76E+03
	MEM	1.84E+04	1.86E+04	1.92E+04	2.02E+04
$\epsilon = 10^{-2}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	1.95E-4	5.34E-5	1.45E-5	3.90E-6
	#GMRES$_{ave}$	37	17	12	13
	CPU time	8.01E+03	8.03E+03	8.15E+03	9.13E+03
	MEM	1.85E+04	1.88E+04	1.93E+04	2.03E+04
$\epsilon = 10^{-3}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	3.98E-4	1.04E-4	2.66E-5	4.31E-6
	#GMRES$_{ave}$	40	21	21	20
	CPU time	8.04E+03	8.10E+03	8.57+03	9.31E+03
	MEM	1.86E+04	1.90E+04	1.97E+04	2.04E+04

Example 2. This example solves the Allen-Cahn equation on a shell-shaped domain whose curve is

given in (5.1) with the parameter $c = 0.2$. The nonlinear term f is supposed as $f(u) = u^5 - u^3 + u$, which is used to describe more complex phase transition behavior [55]. In this example, the correction and interpolation procedure in the KFBI method are both implemented with the Taylor expansion-based scheme. For time integration, we use the numerical scheme (2.9), where the parameters α and β are set to be -1 and 0.5, respectively. Numerical results are listed in Table 6. Figure 7 shows the distribution of solutions and errors in the computational domain of this example.



(a) Solution distribution



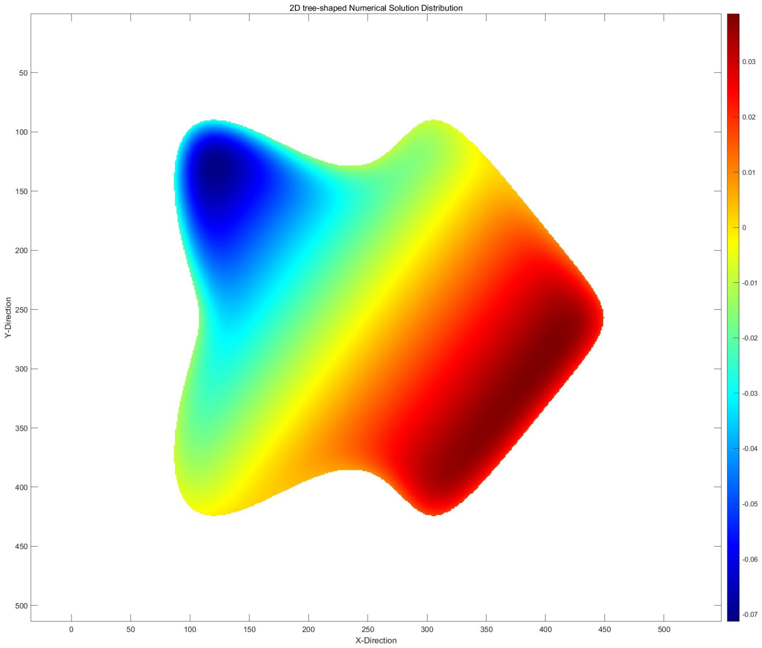
(b) Error distribution

Figure 7. Figure (a) shows the 2D shell-shaped numerical solution distribution in Example 2. Figure (b) shows the 3D error distribution on 512×512 grid when T = 1.0 and $\epsilon = 0.01$.

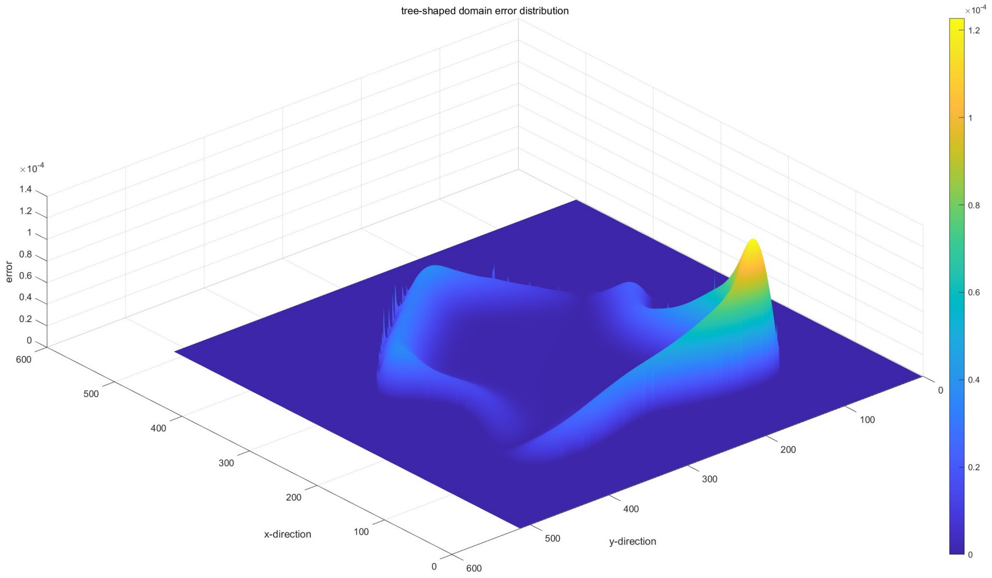
Example 3. This example solves the Allen-Cahn equation on a tree-shaped domain whose curve is given in (5.2) with the parameter $c = 0.2$. The non-linear term f is supposed as the asymmetric double well potential $f(u) = 2u(u-1)(2u-1)$ [56]. In this example, the correction and interpolation procedure in the KFBI method are both implemented with the correction function-based scheme. For time integration, we use the SILF scheme (2.11). Numerical results are listed in Table 7. Figure 8 shows the distribution of solutions and errors in the computational domain of this example.

Table 7. Numerical results of Example 3 obtained with the second-order SILF scheme.

ϵ	grid size	64×64	128×128	256×256	512×512
$\epsilon = 10^{-1}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	1.80E-4	5.11E-5	1.34E-5	3.84E-6
	#GMRES$_{ave}$	10	10	10	12
	CPU time	1.24E+04	1.40E+04	1.56E+04	1.62E+04
	MEM	1.84E+04	1.85E+04	1.93E+04	2.07E+04
$\epsilon = 10^{-2}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	3.91E-3	1.25E-3	3.82E-4	1.22E-4
	#GMRES$_{ave}$	36	14	13	16
	CPU time	1.08E+04	1.12E+04	1.13E+04	1.20E+04
	MEM	1.85E+04	1.87E+04	1.94E+04	2.08E+04
$\epsilon = 10^{-3}$	time step	1/32	1/64	1/128	1/256
	$\|\|u_h - u_{ref}\|\|_\infty$	9.01E-3	2.12E-3	5.41E-4	1.63E-4
	#GMRES$_{ave}$	52	37	36	20
	CPU time	3.89E+03	3.95E+03	3.96E+03	4.01E+04
	MEM	1.86E+04	1.89E+04	1.96E+04	2.11E+04



(a) Solution distribution



(b) Error distribution

Figure 8. Figure (a) shows the 2D tree-shaped numerical solution distribution in Example 3. Figure (b) shows the 3D error distribution on 512×512 grid when T = 1.0 and $\epsilon = 0.01$.

6. Conclusions

This study employs the KFBI method as the spatial discretization method to solve the Allen-Cahn equation, with a class of IMEX schemes as the temporal discretization method. The entire calculation process can be roughly summarized into the following steps: discretize in the time dimension, and then solve the corresponding spatial elliptic equations at each discrete time step. In the process of spatial discretization, we first select a rectangular domain that completely encompasses the interface, then establish a Cartesian grid and identify all irregular grid points on both sides of the interface for correction. Next, we use the FFT-based fast solver to get the numerical solutions at all grid points, followed by polynomial interpolation to calculate the function values on the interface. This process is repeatedly conducted in the iteration of solving the restated BIEs.

Numerical experiments demonstrate that the KFBI method can effectively solve problems with various forms of non-stiff terms and small perturbation parameters ϵ, achieving approximately second-order accuracy in solution convergence. The average number of iterations of the GMRES method will not necessarily increase during the grid refinement. This is because the discrete boundary integral equations are well-conditioned for relatively large ϵ. Thus, the iteration converges within an essentially fixed number of steps, independent of the mesh parameter. However, for even smaller values of ϵ, the procedure fails to converge as the discrete system becomes ill-conditioned. Instead, a class of fitted operator methods or fitted mesh methods on non-uniform mesh may generate a more accurate solution for singular perturbation problems [57]. The advantage of the proposed method lies in that, based on Green's third identity, the BVPs at each time step are reformulated into corresponding BIEs, where the integrals are treated as solutions to their equivalent interface problems, thereby circumventing the complexity of analytical expressions of the kernels. This is the true essence of the KFBI method termed with "kernel-free".

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. DMS-12101553), and the Natural Science Foundation of Zhejiang Province (Grant No. LQ22A010017).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. S. M. Allen, J. W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, *Acta Metall.*, **27** (1979), 1085–1095. https://doi.org/10.1016/0001-6160(79)90196-2
2. C. M. Elliott, B. Stinner, Computation of two-phase biomembranes with phase dependent material parameters using surface finite elements, *Commun. Comput. Phys.*, **13** (2013), 325–360. https://doi.org/10.4208/cicp.170611.130112a

3. L. Golubović, A. Levandovsky, D. Moldovan, Interface dynamics and far-from-equilibrium phase transitions in multilayer epitaxial growth and erosion on crystal surfaces: Continuum theory insights, *East Asian J. Appl. Math.*, **1** (2011), 297–371. https://doi.org/10.4208/eajam.040411.030611a
4. J. W. Cahn, J. E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, *J. Chem. Phys.*, **28** (1958), 258–267. https://doi.org/10.1063/1.1744102
5. R. Kobayashi, J. A. Warren, W. C. Carter, A continuum model of grain boundaries, *Physica D*, **140** (2000), 141–150. https://doi.org/10.1016/S0167-2789(00)00023-3
6. J. Kim, Phase-field models for multi-component fluid flows, *Commun. Comput. Phys.*, **12** (2012), 613–661. https://doi.org/10.4208/cicp.301110.040811a
7. D. Jacqmin, Contact-line dynamics of a diffuse fluid interface, *J. Fluid Mech.*, **402** (2000), 57–88. https://doi.org/10.1017/S0022112099006874
8. J. S. Lowengrub, H. B. Frieboes, F. Jin, Y. L. Chuang, X. Li, P. Macklin, et al., Nonlinear modelling of cancer: bridging the gap between cells and tumours, *Nonlinearity*, **23** (2009), R1–R91. https://doi.org/10.1088/0951-7715/23/1/R01
9. D. Shao, W. J. Rappel, H. Levine, Computational model for cell morphodynamics, *Phys. Rev. Lett.*, **105** (2010), 108104. https://doi.org/10.1103/PhysRevLett.105.108104
10. M. Beneš, K. Mikula, Simulation of anisotropic motion by mean curvature–comparison of phase-field and sharp-interface approaches, *Acta Math. Univ. Comenianae*, **67** (1998), 17–42.
11. S. Gu, X. Zhou, Convex splitting method for the calculation of transition states of energy functional, *J. Comput. Phys.*, **353** (2018), 417–434. https://doi.org/10.1016/j.jcp.2017.10.028
12. J. Shen, X. Yang, Numerical approximations of Allen-Cahn and Cahn-Hilliard equations, *Discrete Contin. Dyn. Syst.*, **28** (2010), 1669–1691. https://doi.org/10.3934/dcds.2010.28.1669
13. D. Jeong, S. Lee, D. Lee, J. Shin, J. Kim, Comparison study of numerical methods for solving the Allen–Cahn equation, *Comput. Mater. Sci.*, **111** (2016), 131–136. https://doi.org/10.1016/j.commatsci.2015.09.005
14. J. Long, C. Luo, Q. Yu, Y. Li, An unconditional stable compact fourth-order finite difference scheme for three dimensional Allen–Cahn equation, *Comput. Math. Appl.*, **77** (2019), 1042–1054. https://doi.org/10.1016/j.camwa.2018.10.028
15. D. Li, Z. Qiao, On second order semi-implicit Fourier spectral methods for 2D Cahn–Hilliard equations, *J. Sci. Comput.*, **70** (2017), 301–341. https://doi.org/10.1007/s10915-016-0251-4
16. Z. Weng, L. Tang, Analysis of the operator splitting scheme for the Allen–Cahn equation, *Numer. Heat Transfer, Part B*, **70** (2016), 472–483. https://doi.org/10.1080/10407790.2016.1215714
17. J. Park, C. Lee, Y. Choi, H. G. Lee, S. Kwak, Y. Hwang, et al., An unconditionally stable splitting method for the Allen–Cahn equation with logarithmic free energy, *J. Eng. Math.*, **132** (2022), 18. https://doi.org/10.1007/s10665-021-10203-6
18. F. Guillén-González, G. Tierra, Second order schemes and time-step adaptivity for Allen–Cahn and Cahn–Hilliard models, *Comput. Math. Appl.*, **68** (2014), 821–846. https://doi.org/10.1016/j.camwa.2014.07.014

19. P. C. Fife, *Dynamics of Internal Layers and Diffusive Interfaces*, 1988. https://doi.org/10.1137/1.9781611970180

20. C. M. Elliott, A. M. Stuart, The global dynamics of discrete semilinear parabolic equations, *SIAM J. Numer. Anal.*, **30** (1993), 1622–1663. https://doi.org/10.1137/0730084

21. D. J. Eyre, An unconditionally stable one-step scheme for gradient systems, 1997.

22. X. Wu, G. J. Van Zwieten, K. G. van der Zee, Stabilized second-order convex splitting schemes for Cahn–Hilliard models with application to diffuse-interface tumor-growth models, *Int. J. Numer. Methods Biomed. Eng.*, **30** (2014), 180–203. https://doi.org/10.1002/cnm.2597

23. S. Zhai, Z. Weng, X. Feng, Investigations on several numerical methods for the non-local Allen–Cahn equation, *Int. J. Heat Mass Transfer*, **87** (2015), 111–118. https://doi.org/10.1016/j.ijheatmasstransfer.2015.03.071

24. P. Vignal, L. Dalcin, D. L. Brown, N. Collier, V. M. Calo, An energy-stable convex splitting for the phase-field crystal equation, *Comput. Struct.*, **158** (2015), 355–368. https://doi.org/10.1016/j.compstruc.2015.05.029

25. X. Feng, T. Tang, J. Yang, Stabilized Crank-Nicolson/Adams-Bashforth schemes for phase field models, *East Asian J. Appl. Math.*, **3** (2013), 59–80. https://doi.org/10.4208/eajam.200113.220213a

26. F. Chen, J. Shen, Efficient energy stable schemes with spectral discretization in space for anisotropic Cahn-Hilliard systems, *Commun. Comput. Phys.*, **13** (2013), 1189–1208. https://doi.org/10.4208/cicp.101111.110512a

27. X. Yang, Error analysis of stabilized semi-implicit method of Allen-Cahn equation, *Discrete Contin. Dyn. Syst. - Ser. B*, **11** (2009), 1057–1070. https://doi.org/10.3934/dcdsb.2009.11.1057

28. U. M. Ascher, S. J. Ruuth, B. T. R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, *SIAM J. Numer. Anal.*, **32** (1995), 797–823. https://doi.org/10.1137/0732037

29. F. de la Hoz, F. Vadillo, A Sylvester-based IMEX method via differentiation matrices for solving nonlinear parabolic equations, *Commun. Comput. Phys.*, **14** (2013), 1001–1026. https://doi.org/10.4208/cicp.050612.180113a

30. S. Boscarino, L. Pareschi, G. Russo, Implicit-explicit Runge–Kutta schemes for hyperbolic systems and kinetic equations in the diffusion limit, *SIAM J. Sci. Comput.*, **35** (2013), A22–A51. https://doi.org/10.1137/110842855

31. B. E. Griffith, R. D. Hornung, D. M. McQueen, C. S. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, *J. Comput. Phys.*, **223** (2007), 10–49. https://doi.org/10.1016/j.jcp.2006.08.019

32. G. Dimarco, L. Pareschi, Numerical methods for kinetic equations, *Acta Numer.*, **23** (2014), 369–520. https://doi.org/10.1017/S0962492914000063

33. S. Jin, Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations, *SIAM J. Sci. Comput.*, **21** (1999), 441–454. https://doi.org/10.1137/S1064827598334599

34. R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, 2007. https://doi.org/10.1137/1.9780898717839

35. D. H. Norrie, G. De Vries, *The Finite Element Method: Fundamentals and Applications*, Academic Press, 2014.
36. F. Nudo, Two one-parameter families of nonconforming enrichments of the Crouzeix–Raviart finite element, *Appl. Numer. Math.*, **203** (2024), 160–172. https://doi.org/10.1016/j.apnum.2024.05.023
37. F. Dell'Accio, A. Guessab, F. Nudo, New quadratic and cubic polynomial enrichments of the Crouzeix-Raviart finite element, *Comput. Math. Appl.*, **170** (2024), 204–212. https://doi.org/10.1016/j.camwa.2024.06.019
38. C. S. Peskin, The immersed boundary method, *Acta Numer.*, **11** (2002), 479–517. https://doi.org/10.1017/S0962492902000077
39. R. J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.*, **31** (1994), 1019–1044. https://doi.org/10.1137/0731054
40. A. N. Marques, J. C. Nave, R. R. Rosales, High order solution of Poisson problems with piecewise constant coefficients and interface jumps, *J. Comput. Phys.*, **335** (2017), 497–515. https://doi.org/10.1016/j.jcp.2017.01.029
41. W. Ying, C. S. Henriquez, A kernel-free boundary integral method for elliptic boundary value problems, *J. Comput. Phys.*, **227** (2007), 1046–1074. https://doi.org/10.1016/j.jcp.2007.08.021
42. H. Zhou, W. Ying, A correction function-based kernel-free boundary integral method for elliptic PDEs with implicitly defined interfaces, *J. Comput. Phys.*, **496** (2024), 112545. https://doi.org/10.1016/j.jcp.2023.112545
43. Y. Xie, W. Ying, A fourth-order kernel-free boundary integral method for implicitly defined surfaces in three space dimensions, *J. Comput. Phys.*, **415** (2020), 109526. https://doi.org/10.1016/j.jcp.2020.109526
44. Y. Xie, S. Li, W. Ying, A fourth-order kernel-free boundary integral method for interface problems, *Commun. Comput. Phys.*, **33** (2023), 764–794. https://doi.org/10.4208/cicp.OA-2022-0236
45. Y. Xie, S. Li, W. Ying, A fourth-order Cartesian grid method for multiple acoustic scattering on closely packed obstacles, *J. Comput. Appl. Math.*, **406** (2022), 113885. https://doi.org/10.1016/j.cam.2021.113885
46. Y. Xie, W. Ying, A high-order kernel-free boundary integral method for incompressible flow equations in two space dimensions, *Numer. Math. Theory Methods Appl.*, **13** (2020), 595–619. https://doi.org/10.4208/nmtma.OA-2019-0175
47. X. Feng, H. Song, T. Tang, J. Yang, Nonlinear stability of the implicit-explicit methods for the Allen-Cahn equation, *Inverse Probl. Imaging*, **7** (2013), 679–695. https://doi.org/10.3934/ipi.2013.7.679
48. U. M. Ascher, S. J. Ruuth, R. J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, *Appl. Numer. Math.*, **25** (1997), 151–167. https://doi.org/10.1016/S0168-9274(97)00056-1
49. O. Steinbach, *Numerical Approximation Methods for Elliptic Boundary Value Problems: Finite and Boundary Elements*, Springer Science & Business Media, 2007.

50. A. Frangi, A. di Gioia, Multipole BEM for the evaluation of damping forces on MEMS, *Comput. Mech.*, **37** (2005), 24–31. https://doi.org/10.1007/s00466-005-0694-1
51. D. Gilbarg, N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer, 1977. https://doi.org/10.1007/978-3-642-96379-7
52. A. A. Samarskii, *The Theory of Difference Schemes*, CRC Press, 2001. https://doi.org/10.1201/9780203908518
53. J. T. Beale, A. T. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces, *Commun. Appl. Math. Comput. Sci.*, **1** (2006), 91–119. https://doi.org/10.2140/camcos.2006.1.91
54. W. Ying, W. C. Wang, A kernel-free boundary integral method for variable coefficients elliptic PDEs, *Commun. Comput. Phys.*, **15** (2014), 1108–1140. https://doi.org/10.4208/cicp.170313.071113s
55. P. H. Rabinowitz, Some global results for nonlinear eigenvalue problems, *J. Funct. Anal.*, **7** (1971), 487–513. https://doi.org/10.1016/0022-1236(71)90030-9
56. L. Q. Chen, Phase-field models for microstructure evolution, *Annu. Rev. Mater. Res.*, **32** (2002), 113–140. https://doi.org/10.1146/annurev.matsci.32.112001.132041
57. J. J. H. Miller On the convergence, uniformly in $varepsilon$, of difference schemes for a two point boundary singular perturbation problem, in *Numerical Analysis of Singular Perturbation Problems*, (1979), 467–474.

AIMS Press