

Computational Physics

An unconditionally stable adaptive finite difference scheme for the Allen–Cahn equation

Hyundong Kim^{a,b}, Seokjun Ham^c, Soobin Kwak^c, Junseok Kim^{c, ID, *}^a Department of Mathematics and Physics, Gangneung-Wonju National University, Gangneung 25457, Republic of Korea^b Institute for Smart Infrastructure, Gangneung-Wonju National University, Gangneung 25457, Republic of Korea^c Department of Mathematics, Korea University, Seoul, 02841, Republic of Korea

ARTICLE INFO

The review of this paper was arranged by Prof. David W. Walker

Keywords:

Adaptive finite difference method
Phase-field model
Maximum principle

ABSTRACT

We propose an unconditionally stable adaptive finite difference scheme for the Allen–Cahn (AC) equation. The AC equation is a reaction-diffusion equation used to model phase separation in multi-component alloy systems. It describes the temporal evolution of the order parameter, which denotes different phases, and incorporates both diffusion and nonlinear reaction terms to capture the interfacial dynamics between phases. A fundamental aspect of the dynamics of the AC equation is motion by mean curvature, which implies that an initial interface shrinks as time progresses. Therefore, it is highly efficient to reduce the computational domain as the interface shrinks. We use an operator splitting technique with a finite difference method and a closed-form solution. We conduct computational tests to validate the effectiveness of the proposed approach. The computational tests demonstrate that the proposed algorithm is effective, reliable, and robust across various test cases.

1. Introduction

We propose an unconditionally stable adaptive computational method for the Allen–Cahn (AC) equation [1]:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} + \Delta \phi(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (1)$$

where $\phi(\mathbf{x}, t)$ is the order parameter, ϵ is a parameter related to the interfacial thickness, $F(\phi) = 0.25(\phi^2 - 1)^2$ is the Helmholtz free energy density function, and $\Omega \subset \mathbb{R}^2$ is the two-dimensional (2D) domain. In this study, we use the zero Neumann boundary condition:

$$\mathbf{n} \cdot \nabla \phi(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \partial\Omega, \quad t \geq 0, \quad (2)$$

where \mathbf{n} denotes the normal vector on the boundary $\partial\Omega$ of the domain Ω .

A primary challenge associated with phase-field models is the necessity to accurately resolve the diffuse interface layer. This difficulty arises across various phase-field models, including the AC, Cahn–Hilliard (CH) [2], conservative AC [3], phase-field crystal [4], and diblock copolymer [5] equations.

Feng and Wu [6] developed an adaptive finite element method (FEM) to solve the AC equation, with a focus on error estimates. This method was validated through several numerical tests and demonstrated its robustness and effectiveness in simulating curvature-driven flows and phase transitions. The adaptivity of this method was highlighted as it enabled efficient computation by concentrating resources on critical areas of the computational domain. Chen et al. [7] investigated a time–space adaptive FEM for the AC equation. They incorporated a second-order accurate unconditionally energy stable method and an SCR-based a posteriori error estimator to guide mesh refinement and coarsening. Numerical results demonstrated the reliability and efficiency of the proposed estimator and adaptive algorithm, with further discussion on extending the approach to the CH equation. An efficient computational method for solving the AC equation was studied, using a diagonally implicit fractional-step θ -scheme and a conforming FEM. The authors explored the effectiveness of adaptive grids compared to uniform grids in handling the steady-state solution

* Corresponding author.

E-mail addresses: hdkim@gwnu.ac.kr (H. Kim), cfdkim@korea.ac.kr (J. Kim).

URL: <https://mathematicians.korea.ac.kr/cfdkim> (J. Kim).

of the equation, demonstrating computational efficiency improvements. This study validates the method through numerical examples that confirm its robustness and accuracy in solving nonlinear PDEs involving complex interfacial dynamics. Shah et al. [8] introduced an efficient and robust algorithm for solving the AC equation, in which they used a diagonally implicit fractional-step scheme for time discretization and a conforming FEM for spatial discretization. Through numerical examples, the authors demonstrated the advantages of adaptive grids over uniform grids for steady-state problems and compared the proposed scheme with other methods in terms of computational cost and temporal accuracy for unsteady problems. Ham et al. [9] introduced a fully explicit, adaptive numerical scheme designed to efficiently simulate a novel phase-field model for crystal growth in both 2D and 3D, which incorporates a curvature-eliminating term and a double-grid temperature approach to improve accuracy and computational performance.

A considerable number of studies have investigated the application of adaptive techniques to phase-field methods for the simulation of two-phase flows. These methods are designed to improve computational efficiency and accuracy by using dynamic mesh refinement based on interface evolution and a posteriori error estimates, which enables accurate resolution of complex interfacial structures while maintaining lower computational cost. Joshi and Jaiman [10] proposed an adaptive variational phase-field FEM on unstructured meshes for simulating two-phase flows, which incorporates a positivity-preserving conservative AC equation coupled with the incompressible Navier–Stokes equations. By using a residual-based error estimator and a coarsening strategy without hierarchical tree storage, the method significantly reduces computational cost and mass loss while maintaining accuracy, as demonstrated through benchmark tests including sloshing tank and dam-breaking problems. Khanwale et al. [11] proposed a fully-coupled, implicit-in-time framework for simulating thermodynamically consistent Cahn–Hilliard–Navier–Stokes systems using a Crank–Nicolson scheme with pressure stabilization and adaptive octree meshes. The method demonstrated second-order temporal accuracy, improved computational efficiency, and robust performance across benchmark problems, validated through comprehensive parallel numerical experiments. Building upon this, they later proposed a semi-implicit, projection-based FEM that decouples pressure via a projection step and incorporates a residual-based variational multiscale formulation, which enables efficient and scalable simulations with large time steps and accurate resolution of interfacial dynamics in two-phase flows [12]. In a related effort focusing on multiphysics coupling, Gao et al. [13] proposed a fully decoupled, linearized, and unconditionally energy-stable FEM to solve the Cahn–Hilliard–Navier–Stokes–Darcy system in coupled free fluid and porous media regions. They used auxiliary variables and interface treatment to ensure stability and efficiency. Numerical results confirmed the method’s accuracy, stability without time step restrictions, and effectiveness in simulating shape relaxation and buoyancy-driven flows with adaptive mesh refinement. To further handle implementation within existing multiphysics platforms, Mostafavi et al. [14] recently presented a fully-coupled, fully-implicit-in-time FEM based on the MOOSE platform for simulating incompressible laminar two-phase flows governed by the Cahn–Hilliard–Navier–Stokes equations. To resolve the lack of mass-conserving interpolation during adaptive mesh refinement, the authors introduced a time-dependent scalar Lagrange multiplier. They validated the method through benchmark tests and confirmed its accuracy and applicability to complex multi-dimensional two-phase flow problems. Meanwhile, with a focus on flows through porous boundaries, Luo et al. [15] presented a phase-field model incorporating the Cahn–Hilliard and Navier–Stokes equations with a generalized Navier slip and penetration boundary condition to simulate two-phase flows through permeable surfaces. The proposed finite element-based numerical algorithm effectively handled complex geometries and adaptive mesh refinement, and its accuracy and robustness were validated through numerical experiments involving bubble penetration and displacement in porous structures. In the context of modeling more intricate multiphysics phenomena, Chen et al. [16] developed two linear, decoupled, and fully discrete FEMs for a complex phase-field model of two-phase ferrofluids, which involves coupled Cahn–Hilliard, Navier–Stokes, magnetization, and magnetostatic equations. The proposed methods achieved energy stability, unique solvability, and efficient handling of nonlinear couplings, as confirmed by numerical experiments with adaptive mesh refinement.

The primary objective of this study is to develop and present an unconditionally stable, maximum principle preserving adaptive finite difference method (FDM) for the numerical solution of the AC equation. The proposed method aims to enhance stability and computational efficiency while guaranteeing that the numerical solution respects the maximum principle, and it preserves the physical bounds of the phase-field variable and accurately captures the interface dynamics inherent in the AC model.

The outline of this paper is as follows. Section 2 describes the proposed computational method. Section 3 presents the results of numerical experiments. We draw conclusions in Section 4.

2. Numerical method

Now, we describe the computational algorithm to find the numerical solution of the AC equation (1) in the 2D space. We use the operator splitting method (OSM) [17,18] to solve the AC equation (1) in the 2D domain $\Omega = (L_x, R_x) \times (L_y, R_y)$:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} + \phi_{xx}(\mathbf{x}, t) + \phi_{yy}(\mathbf{x}, t). \quad (3)$$

Let Δt be the discrete temporal step size. We define the linear operators $\mathcal{L}_x^{\Delta t}$ and $\mathcal{L}_y^{\Delta t}$ for x - and y -directions, respectively. In addition, we define the nonlinear operator $\mathcal{N}^{\Delta t}$. We update $\phi(\mathbf{x}, t)$ in the following order of operator compositions:

$$\phi(\mathbf{x}, t + \Delta t) = (\mathcal{L}_y^{\Delta t} \circ \mathcal{L}_x^{\Delta t} \circ \mathcal{N}^{\Delta t})\phi(\mathbf{x}, t). \quad (4)$$

It is important to note that using a relatively large size of temporal step affects the interface shape depending on the order of solving terms. Solving the diffusion term first followed by the nonlinear term results in a sharper interface. Conversely, solving the nonlinear term first and then the diffusion term leads to a more relaxed transition layer of the interface. In this work, we adopt the splitting order as shown in Eq. (4). In the numerical experiments section, we demonstrate that the numerical results corresponding to the two different operator orders are nearly identical when small time steps are used. Therefore, the numerical solution $\phi(\mathbf{x}, t + \Delta t)$ can be updated by splitting it into three subproblems and solving them sequentially: Firstly, considering the nonlinear operator $\mathcal{N}^{\Delta t}$, we can obtain a solution $\Phi^1(\mathbf{x})$ to Eq. (5), expressed as $\Phi^1(\mathbf{x}) = \mathcal{N}^{\Delta t}\phi(\mathbf{x}, t)$ with the initial condition $\psi(\mathbf{x}, 0) = \phi(\mathbf{x}, t)$:

$$\frac{\partial \psi(\mathbf{x}, t)}{\partial t} = -\frac{F'(\psi(\mathbf{x}, t))}{\epsilon^2}. \quad (5)$$

To solve the nonlinear equation (5), we apply the separation of variables [19]. Secondly, considering the x -direction linear operator $\mathcal{L}_x^{\Delta t}$, we can obtain a solution $\Phi^2(\mathbf{x})$ to the linear equation (6), expressed as $\Phi^2(\mathbf{x}) = \mathcal{L}_x^{\Delta t}\Phi^1(\mathbf{x})$:

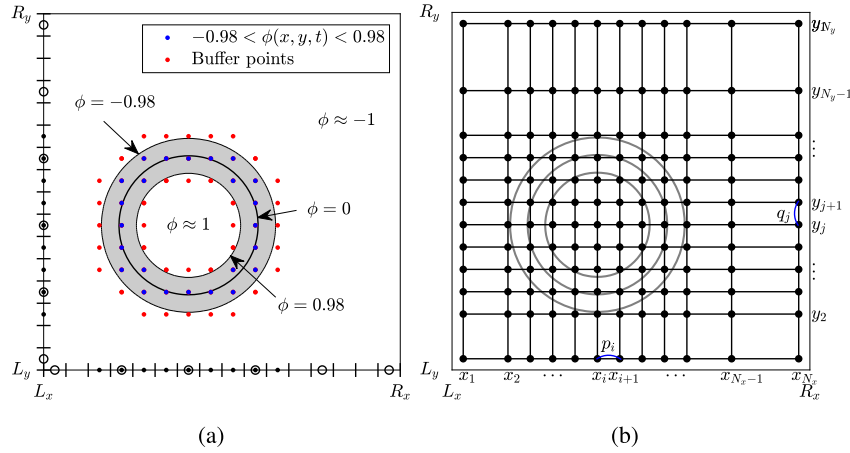


Fig. 1. Schematic illustration of the adaptive mesh generation algorithm. (a) Adaptive mesh generation process. The gray area indicates the interfacial region where $-0.98 < \phi(x, y, t) < 0.98$. Blue dots represent interfacial points, and red dots are buffer points. These are projected onto the axes, and the final nonuniform grid is formed by combining them with coarser grid points (black circles). (b) Final nonuniform grid constructed for solving the AC equation. The grid points are selectively refined near the interfacial region to enhance computational efficiency while preserving accuracy. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\frac{\partial \psi(\mathbf{x}, t)}{\partial t} = \psi_{xx}(\mathbf{x}, t), \quad (6)$$

with $\psi(\mathbf{x}, 0) = \Phi^1(\mathbf{x})$. Finally, using $\mathcal{L}_y^{\Delta t}$, we can obtain a solution $\phi(\mathbf{x}, t + \Delta t)$ to the linear equation (7), expressed as $\phi(\mathbf{x}, t + \Delta t) = \mathcal{L}_y^{\Delta t} \Phi^2(\mathbf{x})$:

$$\frac{\partial \psi(\mathbf{x}, t)}{\partial t} = \psi_{yy}(\mathbf{x}, t), \quad (7)$$

where the initial condition is considered as $\psi(\mathbf{x}, 0) = \Phi_2(\mathbf{x})$. Considering the homogeneous Neumann boundary condition, and the implicit Euler method on nonuniform mesh, we can solve the linear Eqs. (6) and (7).

We now describe a method for generating an adaptive mesh. The procedure for generating the adaptive mesh is outlined as follows: First, we define a reference uniform fine grid with spatial step size h . In Fig. 1(a), intervals of size h are shown along both the x - and y -axes, and the possible node points are positioned at the centers of these intervals. The gray area indicates the interfacial transition regions where the phase variable satisfies $-0.98 < \phi(x, y, t) < 0.98$, which is the criterion for introducing fine grid points. Computational resources are concentrated in these regions since the phase-field remains nearly constant elsewhere. The blue dots represent points located within the gray interfacial region, whereas the red dots indicate buffer points. These tagged points are subsequently projected onto the x - and y -axes. The final nonuniform grid is constructed by taking the union of these projected points and a set of coarser grid points (represented by black circles), which include the minimum and maximum coordinates of the reference points. Fig. 1(b) illustrates the final set of grid points used for the computational solution of the AC equation.

Let us consider a given nonuniform grid x_i for $1 \leq i \leq N_x$ and y_j for $1 \leq j \leq N_y$, where N_x and N_y are the total number of grid points along the x and y axes, respectively. This defines a nonuniform discrete computational domain $\Omega_h = \{(x_i, y_j) | i = 1, \dots, N_x, j = 1, \dots, N_y\}$, with the layout shown in Fig. 1(b). We define the spatial mesh intervals as $p_i = x_{i+1} - x_i$ and $q_j = y_{j+1} - y_j$, with the boundary mesh intervals given by $p_0 = p_1$, $p_{N_x} = p_{N_x-1}$, $q_0 = q_1$, and $q_{N_y} = q_{N_y-1}$. The variable $\phi(x_i, y_j, n\Delta t)$ is simply denoted as ϕ_{ij}^n . The homogeneous Neumann boundary conditions are applied as $\phi_{0j}^n = \phi_{1j}^n$, $\phi_{N_x+1,j}^n = \phi_{N_x,j}^n$, $\phi_{i0}^n = \phi_{i1}^n$, and $\phi_{i,N_y+1}^n = \phi_{i,N_y}^n$. The simulation runs up to a final time T with N_t time steps, and the time step size is $\Delta t = T/N_t$. For simplicity in notations, we use averaged mesh sizes $p_{i+\frac{1}{2}} = (p_{i+1} + p_i)/2$ and $q_{j+\frac{1}{2}} = (q_{j+1} + q_j)/2$. Now, the hybrid computational solution algorithm [20] is described for the AC equation (3) in the following three steps. To start, the method of separation of variables can be used to derive the analytical solution of the nonlinear equation presented in Eq. (5): For $1 \leq i \leq N_x$ and $1 \leq j \leq N_y$,

$$\Phi_{ij}^1 = \frac{\phi_{ij}^n}{\sqrt{e^{\frac{-2\Delta t}{\epsilon^2}} + (\phi_{ij}^n)^2 \left(1 - e^{\frac{-2\Delta t}{\epsilon^2}}\right)}}. \quad (8)$$

Let us define Δ_d on the 2D nonuniform mesh [21,22] to numerically solve Eqs. (6) and (7):

$$\begin{aligned} \Delta_d \phi_{ij} = (\mathcal{L}_d^x + \mathcal{L}_d^y) \phi_{ij} = & \frac{1}{p_{i-\frac{1}{2}}} \left(\frac{\phi_{i+1,j} - \phi_{ij}}{p_i} - \frac{\phi_{ij} - \phi_{i-1,j}}{p_{i-1}} \right) \\ & + \frac{1}{q_{j-\frac{1}{2}}} \left(\frac{\phi_{i,j+1} - \phi_{ij}}{q_j} - \frac{\phi_{ij} - \phi_{i,j-1}}{q_{j-1}} \right). \end{aligned}$$

Secondly and thirdly, we solve Eqs. (6) and (7) using the implicit Euler method. Here, first solve for the equation involving the operator \mathcal{L}_d^x using the initial condition Φ^1 to obtain Φ^2 for fixed j :

$$\frac{\Phi_{ij}^2 - \Phi_{ij}^1}{\Delta t} = \frac{1}{p_{i-\frac{1}{2}}} \left(\frac{\Phi_{i+1,j}^2 - \Phi_{ij}^2}{p_i} - \frac{\Phi_{ij}^2 - \Phi_{i-1,j}^2}{p_{i-1}} \right), \quad \text{for } i = 1, \dots, N_x. \quad (9)$$

Use Φ^2 as the initial condition to solve for ϕ^{n+1} via \mathcal{L}_d^y for fixed i :

$$\frac{\phi_{ij}^{n+1} - \Phi_{ij}^2}{\Delta t} = \frac{1}{q_{j-\frac{1}{2}}} \left(\frac{\phi_{i,j+1}^{n+1} - \phi_{ij}^{n+1}}{q_j} - \frac{\phi_{ij}^{n+1} - \phi_{i,j-1}^{n+1}}{q_{j-1}} \right), \text{ for } j = 1, \dots, N_y. \quad (10)$$

This operator splitting approach simplifies the multidimensional problem into a sequence of one-dimensional problems and thus enables more efficient numerical computation, especially on anisotropic or nonuniform grids. Equations (9) and (10) can be rewritten, respectively, as follows:

$$-\frac{1}{p_{i-1}p_{i-\frac{1}{2}}} \Phi_{i-1,j}^2 + \left(\frac{1}{\Delta t} + \frac{2}{p_i p_{i-1}} \right) \Phi_{ij}^2 - \frac{1}{p_i p_{i-\frac{1}{2}}} \Phi_{i+1,j}^2 = \frac{\Phi_{ij}^1}{\Delta t}, \quad (11)$$

$$-\frac{1}{q_{j-1}q_{j-\frac{1}{2}}} \phi_{i,j-1}^{n+1} + \left(\frac{1}{\Delta t} + \frac{2}{q_j q_{j-1}} \right) \phi_{ij}^{n+1} - \frac{1}{q_j q_{j-\frac{1}{2}}} \phi_{i,j+1}^{n+1} = \frac{\Phi_{ij}^2}{\Delta t}. \quad (12)$$

Finally, we can find a numerical solution ϕ_{ij}^{n+1} for the AC Eq. (3) by applying the three steps mentioned above.

We investigate the stability of the proposed numerical method. In particular, we prove that the numerical solution preserves the maximum bound principle. To begin, we state the following lemma, which is used in our analysis to verify the solvability of the linear systems arising in the proposed scheme.

Lemma 1. Let $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$ be a real square matrix. Suppose that \mathbf{A} is strictly diagonally dominant, i.e.,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad \text{for all } i = 1, \dots, n.$$

Then, \mathbf{A} is nonsingular. In other words, the matrix \mathbf{A} admits a unique inverse in $\mathbb{R}^{n \times n}$ [23].

The following theorem establishes the stability of the numerical solution in the maximum norm. Specifically, it guarantees that the solution remains bounded by 1, provided the initial value is bounded by 1.

Theorem 1. If $\|\phi^n\|_\infty \leq 1$, then the numerical solution satisfies $\|\phi^{n+1}\|_\infty \leq 1$.

Proof. We divide the proof into the following three steps for clarity.

Step 1: Nonlinear operator

The nonlinear term in the AC equation is solved analytically as

$$\Phi_{ij}^1 = \frac{\phi_{ij}^n}{\sqrt{e^{\frac{-2\Delta t}{\epsilon^2}} + (\phi_{ij}^n)^2 \left(1 - e^{\frac{-2\Delta t}{\epsilon^2}}\right)}} = \frac{\phi_{ij}^n}{\sqrt{e^{\frac{-2\Delta t}{\epsilon^2}} (1 - (\phi_{ij}^n)^2) + (\phi_{ij}^n)^2}}.$$

Since $\|\phi^n\|_\infty \leq 1$, we obtain the following inequality

$$|\Phi_{ij}^1| \leq \frac{|\phi_{ij}^n|}{\sqrt{(\phi_{ij}^n)^2}} = 1.$$

Therefore, $\|\Phi^1\|_\infty \leq 1$.

Step 2: x-direction linear operator

From Eq. (11), we can obtain the following matrix form:

$$\mathbf{A}_x \Phi_j^2 = \Phi_j^1, \text{ for } j = 1, \dots, N_y,$$

where

$$\mathbf{A}_x = \begin{pmatrix} \left(\frac{1}{\Delta t} + \frac{2}{p_1 p_0} \right) & -\frac{1}{p_1 p_{\frac{1}{2}}} & & & & 0 \\ -\frac{1}{p_1 p_{\frac{1}{2}}} & \left(\frac{1}{\Delta t} + \frac{2}{p_2 p_1} \right) & -\frac{1}{p_2 p_{\frac{3}{2}}} & & & \\ & -\frac{1}{p_2 p_{\frac{3}{2}}} & \left(\frac{1}{\Delta t} + \frac{2}{p_3 p_2} \right) & -\frac{1}{p_3 p_{\frac{5}{2}}} & & \\ & & \ddots & \ddots & \ddots & \\ 0 & & & -\frac{1}{p_{N_x-1} p_{N_x-\frac{3}{2}}} & \left(\frac{1}{\Delta t} + \frac{2}{p_{N_x} p_{N_x-1}} \right) \end{pmatrix}$$

is an $(N_x \times N_x)$ matrix, $\Phi_j^2 = (\Phi_{1,j}^2, \Phi_{2,j}^2, \dots, \Phi_{N_x,j}^2)^T$ and $\Phi_j^1 = (\Phi_{1,j}^1, \Phi_{2,j}^1, \dots, \Phi_{N_x,j}^1)^T$, for $j = 1, 2, \dots, N_y$. To examine the diagonal dominance of the matrix \mathbf{A}_x , we consider each row index i . For the rows corresponding to $i = 1$ and $i = N_x$, the diagonal dominance condition is satisfied as follows:

$$|a_{11}| - \sum_{\substack{j=1 \\ j \neq 1}}^{N_x} |a_{1j}| = \frac{1}{\Delta t} + \frac{1}{p_0 p_{\frac{1}{2}}} > 0,$$

$$|a_{N_x N_x}| - \sum_{\substack{j=1 \\ j \neq N_x}}^{N_x} |a_{N_x j}| = \frac{1}{\Delta t} + \frac{1}{p_{N_x} p_{N_x - \frac{3}{2}}} > 0.$$

For the rows corresponding to $i = 2, \dots, N_x - 1$, we have

$$|a_{ii}| - \sum_{\substack{j=1 \\ j \neq i}}^{N_x} |a_{ij}| = \frac{1}{\Delta t} > 0.$$

Hence, the matrix \mathbf{A}_x is strictly diagonally dominant, which guarantees the uniqueness and solvability of the numerical solution.

Equation (11) can be arranged as

$$\Phi_{ij}^2 + a_i(\Phi_{ij}^2 - \Phi_{i-1,j}^2) + b_i(\Phi_{ij}^2 - \Phi_{i+1,j}^2) = \Phi_{ij}^1, \quad (13)$$

where

$$a_i = \frac{\Delta t}{p_i p_{i-\frac{1}{2}}} > 0 \quad \text{and} \quad b_i = \frac{\Delta t}{p_i p_{i+\frac{1}{2}}} > 0.$$

Suppose that there exist indices M and m , with $1 \leq M, m \leq N_x$, such that for each fixed j ,

$$\Phi_{M,j}^2 = \max_{1 \leq i \leq N_x} \Phi_{ij}^2 > \max_{1 \leq i \leq N_x} \Phi_{ij}^1 \quad \text{and} \quad \Phi_{m,j}^2 = \min_{1 \leq i \leq N_x} \Phi_{ij}^2 < \min_{1 \leq i \leq N_x} \Phi_{ij}^1.$$

Then, from Eq. (13), for $i = M$ and m , we have the following:

$$\Phi_{M,j}^2 + a_M(\Phi_{M,j}^2 - \Phi_{M-1,j}^2) + b_M(\Phi_{M,j}^2 - \Phi_{M+1,j}^2) = \Phi_{M,j}^1, \quad (14)$$

$$\Phi_{m,j}^2 + a_m(\Phi_{m,j}^2 - \Phi_{m-1,j}^2) + b_m(\Phi_{m,j}^2 - \Phi_{m+1,j}^2) = \Phi_{m,j}^1. \quad (15)$$

Since $\Phi_{M,j}^2$ is the maximum, we have $\Phi_{M,j}^2 - \Phi_{M\pm 1,j}^2 \geq 0$, and hence the left-hand side of (14) is greater than or equal to $\Phi_{M,j}^2$. Similarly, since $\Phi_{m,j}^2$ is the minimum, the terms $\Phi_{m,j}^2 - \Phi_{m\pm 1,j}^2 \leq 0$, and so the left-hand side of Eq. (15) is less than or equal to $\Phi_{m,j}^2$. These contradict the assumption that

$$\Phi_{M,j}^2 > \max_{1 \leq i \leq N_x} \Phi_{ij}^1, \quad \text{and} \quad \Phi_{m,j}^2 < \min_{1 \leq i \leq N_x} \Phi_{ij}^1.$$

Therefore, we conclude

$$\max_{1 \leq i \leq N_x} \Phi_{ij}^2 \leq \max_{1 \leq i \leq N_x} \Phi_{ij}^1 \quad \text{and} \quad \min_{1 \leq i \leq N_x} \Phi_{ij}^2 \geq \min_{1 \leq i \leq N_x} \Phi_{ij}^1,$$

which implies

$$\|\Phi_j^2\|_\infty \leq \|\Phi_j^1\|_\infty.$$

By taking the maximum of both sides over all j , we obtain

$$\|\Phi^2\|_\infty \leq \|\Phi^1\|_\infty.$$

In particular, if the initial value satisfies $\|\Phi^1\|_\infty \leq 1$, then the updated solution satisfies

$$\|\Phi^2\|_\infty \leq 1.$$

Step 3: y-direction linear operator

Without loss of generality, we omit the full matrix form for the y -direction since the structure of the discretization and the corresponding matrix analysis is analogous to that of the x -direction. As in the x -direction, the matrix \mathbf{A}_y associated with the y -direction discretization is strictly diagonally dominant, which guarantees the uniqueness and solvability of the numerical solution of Eq. (12). Moreover, the maximum principle also holds:

$$\|\Phi^{n+1}\|_\infty \leq \|\Phi^n\|_\infty \leq 1, \quad (16)$$

under the assumption that the initial condition satisfies $\|\Phi^2\|_\infty \leq 1$. \square

By Theorem 1, if the initial solution satisfies $\|\Phi^n\|_\infty \leq 1$, the numerical solutions preserve the maximum bound principle for each time. Hence, the propose method is unconditionally stable.

The proposed algorithms have a flexible structure that allows a direct extension to 3D computational domains. This extension requires only minor modifications to the numerical implementation, and the underlying framework remains applicable to higher-dimensional simulations. The 3D nonuniform discrete computational domain is defined as $\Omega_h = \{(x_i, y_j, z_k) | i = 1, \dots, N_x, j = 1, \dots, N_y, k = 1, \dots, N_z\}$ for $z_k = L_z + (k - 0.5)r_k$,

Table 1
Maximum norm errors and convergence rates.

h	1/16	1/32	1/64	1/128	1/256
error	3.547895e-2	9.163102e-3	2.307796e-3	5.777571e-4	1.442050e-4
rate		1.95	1.99	2.00	2.00

where $r_k = z_{i+1} - z_i$. In the three-dimensional domain, the proposed hybrid scheme for the AC equation can be systematically extended based on the 2D numerical scheme through the following four steps.

$$\Phi_{ijk}^1 = \frac{\phi_{ijk}^n}{\sqrt{e^{\frac{-2\Delta t}{\epsilon^2}} + (\phi_{ijk}^n)^2 \left(1 - e^{\frac{-2\Delta t}{\epsilon^2}}\right)}}, \text{ for all } i, j, k.$$

In the following steps, we apply operator splitting in each spatial direction to advance the solution.

$$\frac{\Phi_{ijk}^2 - \Phi_{ijk}^1}{\Delta t} = \frac{1}{p_{i-\frac{1}{2}}} \left(\frac{\Phi_{i+1,jk}^2 - \Phi_{ijk}^2}{p_i} - \frac{\Phi_{ijk}^2 - \Phi_{i-1,jk}^2}{p_{i-1}} \right), \text{ for } i = 1, \dots, N_x.$$

This equation corresponds to the numerical update in the x -direction using a finite difference approximation, for fixed j and k . Next, we apply a similar procedure in the y -direction, for fixed i and k .

$$\frac{\Phi_{ijk}^3 - \Phi_{ijk}^2}{\Delta t} = \frac{1}{q_{j-\frac{1}{2}}} \left(\frac{\Phi_{ij+1,k}^3 - \Phi_{ijk}^3}{q_j} - \frac{\Phi_{ijk}^3 - \Phi_{ij-1,k}^3}{q_{j-1}} \right), \text{ for } j = 1, \dots, N_y.$$

Finally, the update in the z -direction is performed to obtain the solution at the next time step ϕ^{n+1} , for fixed i and j .

$$\frac{\phi_{ijk}^{n+1} - \Phi_{ijk}^3}{\Delta t} = \frac{1}{r_{k-\frac{1}{2}}} \left(\frac{\phi_{ij,k+1}^{n+1} - \Phi_{ijk}^{n+1}}{r_k} - \frac{\Phi_{ijk}^{n+1} - \phi_{ij,k-1}^{n+1}}{r_{k-1}} \right), \text{ for } k = 1, \dots, N_z.$$

3. Computational tests

The computational tests are performed to verify that the proposed numerical method satisfies certain properties of the AC equation. The interfacial thickness is controlled by setting $\epsilon = \epsilon_m = mh/[2\sqrt{2}\tanh^{-1}(0.9)]$ [24].

3.1. Convergence test

To validate the accuracy of the proposed computational method, we conduct a convergence test. The computational domain is $\Omega = (-0.5, 1.5) \times (0, 0.5)$, and the exact solution is as follows:

$$\phi_{\text{exact}}(x, y, t) = \frac{1}{2} \left(1 - \tanh \left(\frac{x - \frac{3t}{\sqrt{2}\epsilon}}{2\sqrt{2}\epsilon} \right) \right),$$

which represents a traveling interface profile centered at $x = 3t/\sqrt{2}\epsilon$, with interface thickness controlled by the parameter ϵ .

The convergence analysis is based on the maximum norm of the error, defined as $\|\phi - \phi_{\text{exact}}\|_{\infty}$, which captures the largest pointwise deviation between the numerical and exact solutions. The numerical tests are performed on a mesh with spatial step sizes h and $h/2$. The spatial grid size is halved from $h = 1/16$ to $1/256$, and the corresponding time step size is chosen as $\Delta t = 0.5h^2$ to ensure temporal accuracy consistent with the spatial discretization. The final time is fixed at $T = 0.0078125$, and the interface thickness parameter is set to $\epsilon = 0.25/(2\sqrt{2}\tanh^{-1}(0.9))$.

Table 1 summarizes the maximum norm errors and the observed convergence rates. The results indicate that the proposed method achieves first-order accuracy in time and second-order accuracy in space. Specifically, as the mesh is refined by a factor of two, the numerical error decreases approximately by a factor of four, which is consistent with second-order spatial convergence. The explicit Euler time-stepping discretization yields first-order temporal accuracy, which is preserved by adopting the time step size $\Delta t \sim h^2$.

3.2. Motion by mean curvature

We implement the numerical simulation using the proposed scheme to numerically solve the AC equation on the domain $\Omega_h = (-3, 3)^2$. The initial condition is

$$\phi(x, y, 0) = \tanh \left(\frac{1 - \sqrt{x^2 + y^2}}{\sqrt{2}\epsilon} \right)$$

with parameter values as $\epsilon = \epsilon_{15}$, $N_x = 200$, $N_y = 200$, $h = 6/200$, $\Delta t = 1.0h^2$.

Fig. 2 presents the computational results of solving the AC equation using the proposed adaptive FDM. Fig. 2(a) compares the analytic solution for the radius (denoted by the solid line) with the numerical solutions. Here, the numerical results are obtained using the proposed splitting order and reverse splitting order. In the reverse splitting order, the linear operators are solved before the nonlinear operator. The radius, represented by $R(t) = \sqrt{1 - 2t}$ [25], decreases over time, which indicates the shrinking of an interface due to the dynamics dictated by the AC equation. The

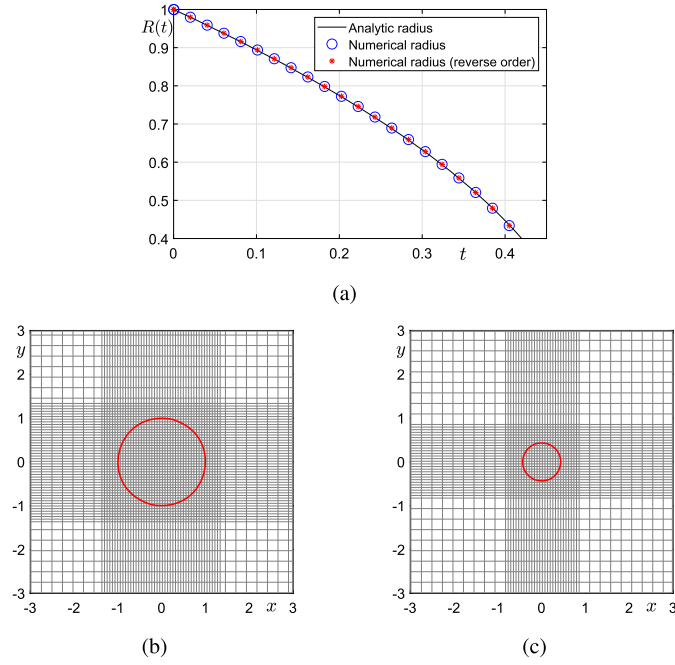


Fig. 2. (a) Temporal evolution of the numerical radius compared with the analytic radius over time. Blue circles and red dots denote the numerical radius computed in forward and reverse computational order, respectively, while the solid line represents the analytic solution. (b)–(c) Temporal evolution of the zero-level contour of numerical solutions with the adaptive refinement mesh at $t = 0$, and $t = 1000\Delta t$. The displayed mesh is shown at 50% resolution for clarity.

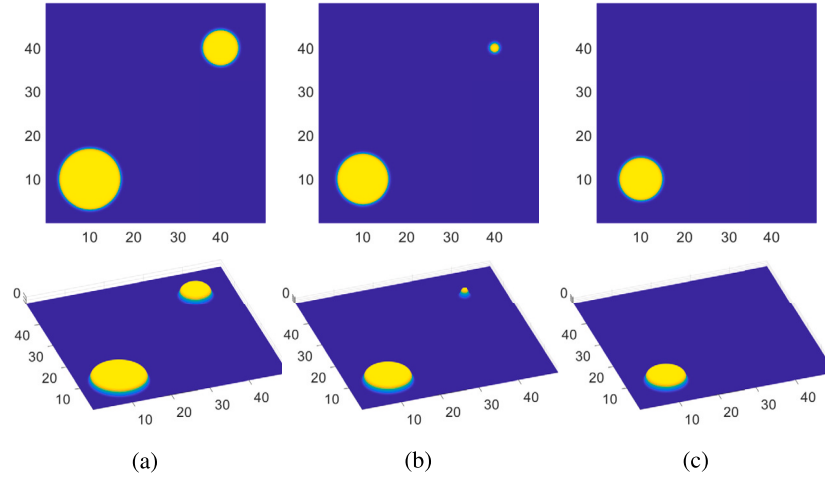


Fig. 3. Motion by mean curvature on a uniform mesh at times (a) $t = 0$, (b) $t = 3000\Delta t$, and (c) $t = 5000\Delta t$. First and second rows are top views and mesh plots, respectively.

close agreement between the numerical and analytic radii throughout the entire simulation demonstrates the accuracy of the adaptive scheme and indicates that the splitting order has little influence on the numerical solution. Figs. 2(b) and (c) display the spatial grids and the zero-level contours of the computational solution at two distinct times, $t = 0$ and $t = 1000\Delta t$, respectively. The initial state shows a well-defined circle that diminishes in size by $t = 1000\Delta t$. The adaptive mesh refinement around the interface is evident, with a denser grid near the circle to capture the sharp gradient accurately. This targeted refinement is crucial for maintaining the efficiency and accuracy of the numerical scheme without unnecessarily increasing computational costs across the entire domain.

3.3. Efficiency of adaptive mesh

We perform a computational experiment for solving the AC equation on the adaptive finite difference grids. The size of computational domain is set to $\Omega_h = (0, 50)^2$. To verify the efficiency of the proposed adaptive finite difference scheme, we consider two circles with different radii as initial condition (Fig. 3(a) and Fig. 4(a)):

$$\phi(x, y, 0) = 1 + \tanh\left(\frac{7 - \sqrt{(x-10)^2 + (y-10)^2}}{\sqrt{2}\epsilon}\right) + \tanh\left(\frac{4 - \sqrt{(x-40)^2 + (y-40)^2}}{\sqrt{2}\epsilon}\right).$$

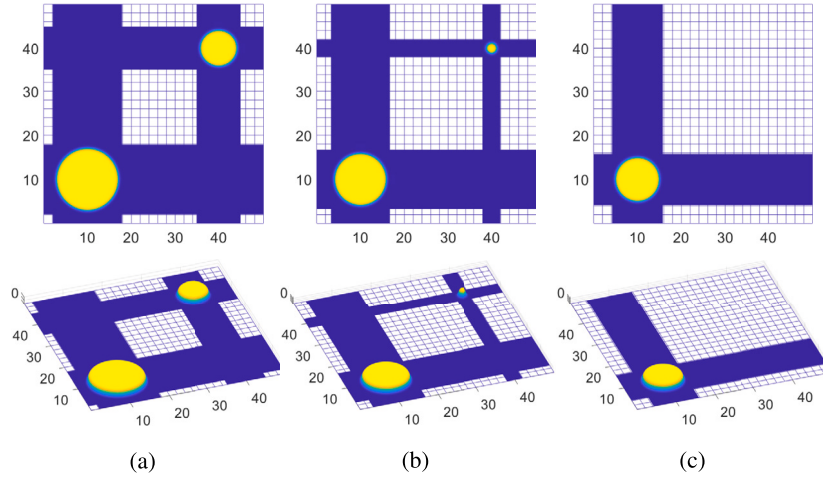


Fig. 4. Motion by mean curvature on an adaptive mesh at times (a) $t = 0$, (b) $t = 3000\Delta t$, and (c) $t = 5000\Delta t$. First and second rows are top views and mesh plots, respectively.

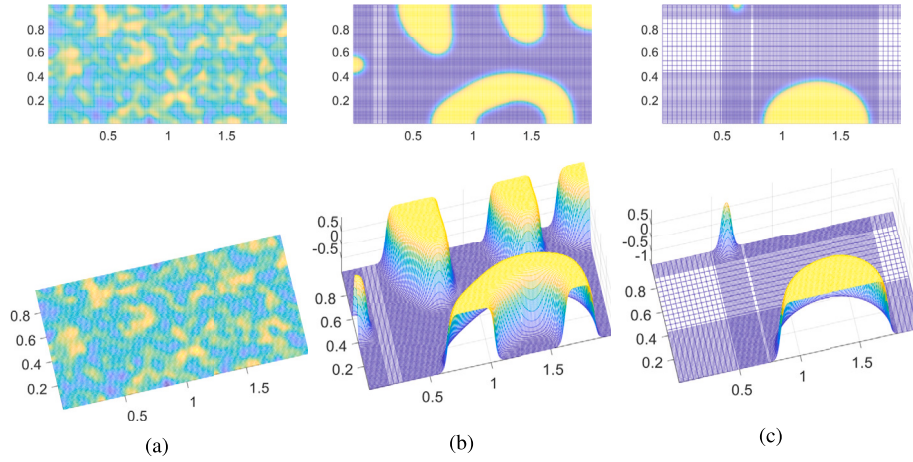


Fig. 5. Temporal evolution of AC equation on an adaptive mesh at times (a) $t = 10\Delta t$, (b) $t = 500\Delta t$, and (c) $t = 1600\Delta t$. First and second rows are top view and mesh plots, respectively.

Here, the values of parameters are used as $N_x = 1000$, $N_y = 1000$, $h = 50/1000$, $\Delta t = 1.0h^2$, and $\epsilon = \epsilon_{20}$. Numerical behaviors of the AC equation by solving the proposed scheme on the uniform and adaptive finite difference grids can be seen in Fig. 3 and Fig. 4. Over time, the two circles shrink due to mean curvature motion, with the circle of smaller radius disappearing first. As the interface of the two circles shrinks, the number of grids near the interface adaptively decreases according to the proposed method, see Figs. 4(b) and (c).

We recorded the CPU times required for the main iterations of the proposed method on both uniform and adaptive meshes, to assess the efficiency of the proposed adaptive mesh refinements. For 5000 iterations, the proposed method took 165.23 seconds on uniform grids and 40.41 seconds on adaptive meshes. These results demonstrate that the adaptive mesh approach was approximately 4 times faster than the uniform grid approach, confirming the efficiency of the proposed adaptive FDM.

3.4. Random initial perturbation

After setting a randomly perturbed initial condition, a numerical experiment is conducted in the computational region $\Omega_h = (0, 2) \times (0, 1)$. The initial condition is set to

$$\phi(x, y, 0) = 0.01(2\text{rand}(x, y) - 1),$$

where rand has a random value in $[0, 1]$. We used parameter values as $N_x = 200$, $N_y = 100$, $h = 2/200$, $\Delta t = 0.45h^2$ and $\epsilon = \epsilon_8$.

The initial random perturbation evolves over time due to the phase separation property of the AC equation as shown in Fig. 5(b). As time progresses, the perturbation coalesces into a single phase (see, Fig. 5(c)) and eventually shrinks, because the dynamics of the AC equation is governed by a non-conserved order parameter. In this test, we verify that the proposed method adaptively changes the finite difference grids as the interfaces of the phases evolve.

Now, we shall perform a computational test to numerically investigate the discrete maximum principle and energy dissipation properties. Let us consider the following discrete energy functional:

$$\mathcal{E}_h(\phi^n) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \frac{F(\phi_{ij}^n)}{\epsilon^2} p_{i-\frac{1}{2}} q_{j-\frac{1}{2}} + 0.5 \sum_{i=1}^{N_x-1} \sum_{j=1}^{N_y-1} \frac{q_j}{p_i} (\phi_{i+1,j}^n - \phi_{ij}^n)^2$$

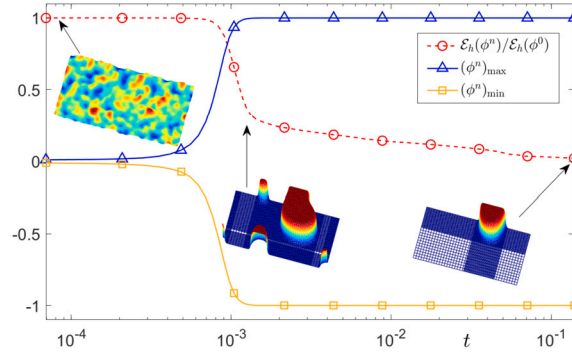


Fig. 6. Temporal evolution of the normalized discrete total free energy for the AC equation. Plots on a adaptive mesh at times $t = 10\Delta t$, $t = 700\Delta t$, and $t = 2000\Delta t$.

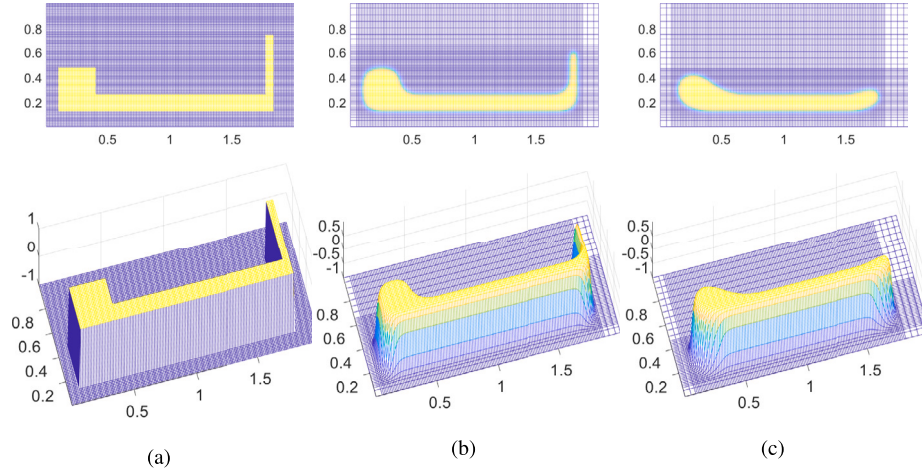


Fig. 7. Evolutionary dynamics of AC equation on a adaptive mesh at times (a) $t = 0$, (b) $t = 50\Delta t$, and (c) $t = 200\Delta t$. First and second rows are top view and mesh plots, respectively.

$$+0.5 \sum_{i=1}^{N_x-1} \sum_{j=1}^{N_y-1} \frac{p_i}{q_j} (\phi_{i,j+1}^n - \phi_{ij}^n)^2$$

The maximum value and minimum value are defined as: $(\phi^n)_{\max} = \max_{ij} \phi_{ij}^n$ and $(\phi^n)_{\min} = \min_{ij} \phi_{ij}^n$. Fig. 6 shows the temporal evolution of the normalized discrete total free energy for the AC equation. The snapshot plots three key metrics over a logarithmic time scale: the normalized total free energy $\mathcal{E}_h(\phi^n)/\mathcal{E}_h(\phi^0)$ (red color), and the maximum value (blue color) and minimum value (yellow color) of the field variable, ϕ^n . A curve of $\mathcal{E}_h(\phi^n)/\mathcal{E}_h(\phi^0)$ shows the relative change in the total free energy as a function of time, which indicates energy dissipation of the AC equation. The curve starts near unity and fluctuates slightly as time progresses, which suggests that while the governing equation loses energy due to the dynamics of phase separation, the proposed method effectively controls the rate of energy change, maintaining stability and accuracy. The maximum and minimum values of ϕ^n curves indicate the evolution of the extreme values of the field variable over time. Initially, the values span a wide range, which narrows as the AC equation evolves. This behavior is typical in phase separation processes modeled by the AC equation, where the extremes gradually approach the equilibrium state. The non-monotonic approach towards these equilibrium values reflects the complex interaction dynamics within the governing equation, possibly capturing transient growth in localized areas before stabilizing.

3.5. Evolution of a simple initial shape

We use the domain size and parameter values as $\Omega_h = (0, 2) \times (0, 1)$, $N_x = 160$, $N_y = 80$, $h = 2/160$, $\Delta t = 0.45h^2$, and $\epsilon = \epsilon_4$. The initial condition is considered key shape as shown in Fig. 7(a). From the numerical results shown in Figs. 7(b) and (c), we observe that over time, the interfaces corresponding to long and thin shape (tail) shrinks first, while the interfaces corresponding to short and thick shape shrink (head) later. We can see that the finite difference grids also change adaptively along the shape of interface.

3.6. Evolution of a complex initial shape

The domain size and parameter values are used as $\Omega_h = (0, 2) \times (0, 1)$, $N_x = 160$, $N_y = 80$, $h = 2/160$, $\Delta t = 0.45h^2$ and $\epsilon = \epsilon_4$. We set the initial condition as squared maze (Fig. 8(a)). The numerical behaviors of evolutionary dynamics can be seen in Fig. 8. The interface is reduced along the motion by mean curvature while maintaining its shape. As the shape of the interface evolves, we can see that the finite difference grids also adaptively change well along the shape of the interface.

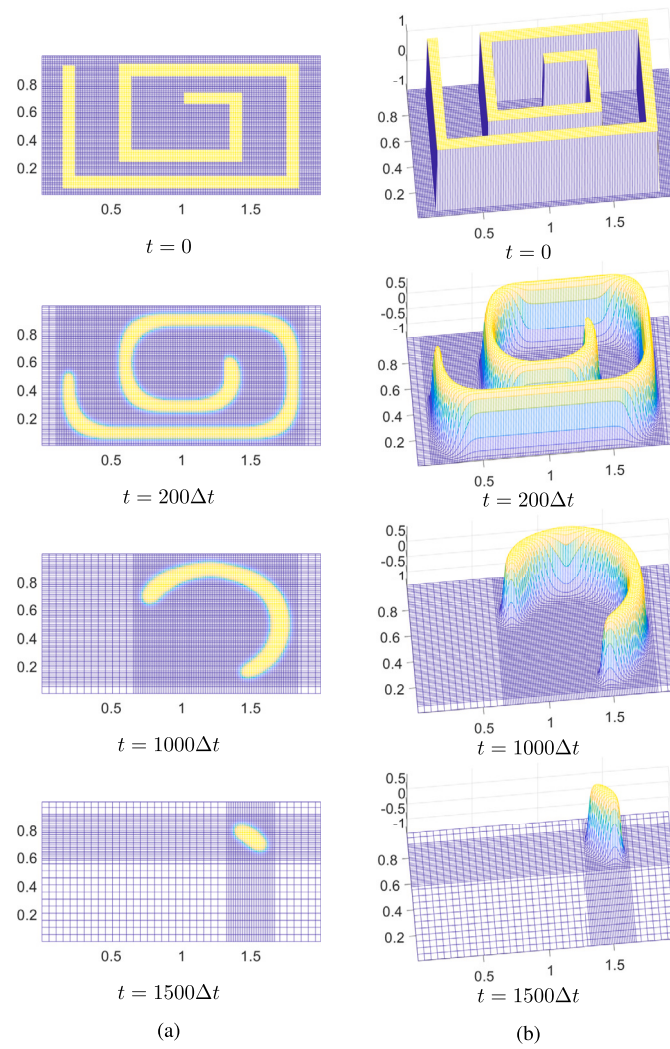


Fig. 8. Evolutionary dynamics for squared maze shape initial condition on a 2D adaptive mesh with $\epsilon = \epsilon_4$. (a) top views, (b) mesh plots.

4. Conclusions

In conclusion, we have introduced an unconditionally stable adaptive FEM for the AC equation, which plays an important role in modeling phase separation in multi-component alloy systems. The AC equation, characterized by its reaction-diffusion nature, effectively captures the complex dynamics of phase interfaces through the interplay of diffusion and nonlinear reaction terms. A key aspect of these dynamics, the motion by mean curvature, naturally leads to the shrinking of interfaces over time. Based on this principle, our proposed scheme efficiently reduces the computational domain as the interface contracts. Through a series of rigorous numerical experiments, we have demonstrated that our approach not only maintains unconditional stability but also significantly improves computational efficiency without compromising accuracy. The results consistently validate the robustness and reliability of the proposed algorithm across various scenarios and show its potential as a powerful tool for simulating phase separation phenomena in complex materials. This research provides the groundwork for future studies focused on advancing adaptive schemes for other reaction-diffusion equations and similar computational models in materials science.

CRediT authorship contribution statement

Hyundong Kim: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization, Supervision. **Seokjun Ham:** Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Methodology, Resources, Writing – review & editing. **Soobin Kwak:** Visualization, Validation, Resources, Writing – review & editing. **Junseok Kim:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the writing process

This article was not created using Artificial Intelligence (AI) tools.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The first author (Hyundong Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A1A03044326). The corresponding author (J.S. Kim) was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A2C1003844). We sincerely thank the reviewers for their careful reading and thoughtful comments, which helped us to improve the clarity and quality of the revised manuscript.

Appendix A

The following Listing 1 is the MATLAB source code to numerically implement for Fig. 5, which is also available in the public GitHub repository:

<https://github.com/hdkim4681/AdaptiveAC>

Listing 1: MATLAB source code for Fig. 5.

```

1 clear all; close all; clc;
2 Lx=0; Rx=2; Ly=0; Ry=1; % Domain definition
3 % Global grid size and step size
4 GNx=200; h=(Rx-Lx)/GNx; GNy=round((Ry-Ly)/h);
5 % Generate uniform global grid points in x and y
6 x=linspace(Lx+0.5*h, Rx-0.5*h, GNx);
7 y=linspace(Ly+0.5*h, Ry-0.5*h, GNy);
8 % Time step size and interfacial parameter
9 dt=0.45*h^2; m=8; epsilon = m*h/(2*sqrt(2)*atanh(0.9));
10 % Set random initial condition for phi in [-0.01, 0.01]
11 for i=1:GNx
12     for j=1:GNy
13         phi0(i,j)=0.01*(2*rand(1)-1);
14     end
15 end
16 % Identify regions near the interface and add buffer
17 xt=[]; yt=[]; bf=1; % buffer size
18 for i=1:GNx
19     for j=1:GNy
20         if abs(phi0(i,j))<0.98
21             ii=round((x(i)-Lx)/h+0.5);
22             jj=round((y(j)-Ly)/h+0.5);
23             xt=[xt max(ii-bf,1):min(ii+bf,GNx)];
24             yt=[yt max(jj-bf,1):min(jj+bf,GNy)];
25         end
26     end
27 end
28 % Background coarse grid sampling
29 s=4; bgx=[1:s:GNx GNx]; bgy=[1:s:GNy GNy];
30 % Merge tagged and background grids
31 xt=unique([xt bgx]); yt=unique([yt bgy]);
32 % Plot initial condition
33 newx=x(xt); newy=y(yt); phi1=phi0(xt,yt);
34 [xx,yy]=meshgrid(newx,newy);
35 figure(1); clf; mesh(xx,yy,phi1'); view(-13,77)
36 axis([Lx Rx Ly Ry -1 1]); axis image;
37 figure(2); clf; mesh(xx,yy,phi1'); view(0,90); axis image;
38 Nx=length(newx); Ny=length(newy);
39 % Main loop parameters
40 Nt=3000; % Total time steps
41 remesh=20; % Remeshing interval
42 ns=100; % Plotting interval
43 % Main iteration loop
44 for iter = 1:Nt
45     if mod(iter, remesh)==0 || iter==1 % Remeshing condition
46         if iter>1 % Recompute active region based on interface
47             xt=[]; yt=[];
48             for i=1:Nx
49                 for j=1:Ny
50                     if abs(phi1(i,j))<0.98

```

```

51         ii=round((newx(i)-Lx)/h+0.5);
52         jj=round((newy(j)-Ly)/h+0.5);
53         xt=[xt max(ii-bf,1):min(ii+bf,GNx)];
54         yt=[yt max(jj-bf,1):min(jj+bf,GNy)];
55     end
56 end
57 end
58 % Merge with background grid
59 xt=unique([xt bgx]); yt=unique([yt bgy]);
60 % Update mesh and interpolate phi
61 newx=x(xt); newy=y(yt); Nx=length(xt); Ny=length(yt);
62 [xx1,yy1]=meshgrid(newx,newy);
63 % Bilinear interpolation
64 phi1=interp2(xx,yy,phi1',xx1,yy1)'; xx=xx1; yy=yy1;
65 end
66 % Compute nonuniform grid spacing and harmonic mean
67 p=newx(2:end)-newx(1:end-1);
68 p=[p(1) p p(end)];
69 ph=(p(1:end-1)+p(2:end))/2;
70 q=newy(2:end)-newy(1:end-1);
71 q=[q(1) q q(end)];
72 qh=(q(1:end-1)+q(2:end))/2;
73 % Construct 1D coefficients in x-direction
74 ax=zeros(1,Nx); bx=ax; cx=ax;
75 for i=1:Nx
76     ax(i)=-1/(ph(i)*p(i));
77     bx(i)=1/dt+2/(p(i+1)*p(i));
78     cx(i)=-1/(ph(i)*p(i+1));
79 end
80 % Construct 1D coefficients in y-direction
81 ay=zeros(1,Ny); by=ay; cy=ay;
82 for j=1:Ny
83     ay(j)=-1/(qh(j)*q(j));
84     by(j)=1/dt+2/(q(j+1)*q(j));
85     cy(j)=-1/(qh(j)*q(j+1));
86 end
87 % Neumann boundary condition modification
88 bx(1)=bx(1)+ax(1); bx(Nx)=bx(Nx)+cx(Nx);
89 by(1)=by(1)+ay(1); by(Ny)=by(Ny)+cy(Ny);
90 end
91 % Nonlinear stabilization step
92 phi1=phi1./sqrt((1-phi1.^2)*exp(-2*dt/epsilon^2)+phi1.^2);
93 % Solve implicit x-direction using Thomas algorithm
94 for j=1:Ny
95     f=phi1(1:Nx,j)/dt; phi1(1:Nx,j)=thomas(ax, bx, cx, f);
96 end
97 % Solve implicit y-direction using Thomas algorithm
98 for i=1:Nx
99     g=phi1(i,1:Ny)/dt; phi1(i,1:Ny)=thomas(ay, by, cy, g)';
100 end
101 % Visualization
102 if mod(iter,ns)==0
103     figure(3); clf; mesh(xx,yy,phi1'); view(-13,77)
104     axis([Lx Rx Ly Ry -1 1]); axis image;
105     figure(4); clf; mesh(xx,yy,phi1'); view(0,90); axis image;
106 end
107 end
108 % Thomas algorithm for tridiagonal linear system
109 function x=thomas(alpha, beta, gamma, f)
110     n=length(f);
111     for i=2:n
112         mult=alpha(i)/beta(i-1);
113         beta(i)=beta(i)-mult*gamma(i-1);
114         f(i)=f(i)-mult*f(i-1);
115     end
116     x(n)=f(n)/beta(n);
117     for i=n-1:-1:1
118         x(i)=(f(i)-gamma(i)*x(i+1))/beta(i);
119     end
120 end

```

Data availability

All data used in this article can be shared upon appropriate request.

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 (6) (1979) 1085–1095.
- [2] S.J. Chen, H.Y. Yu, L-MAU: a multivariate time-series network for predicting the Cahn–Hilliard microstructure evolutions via low-dimensional approaches, *Comput. Phys. Commun.* 305 (2024) 109342.
- [3] Y. Wu, J. Yang, Z. Tan, A highly efficient variant of scalar auxiliary variable (SAV) approach for the phase-field fluid-surfactant model, *Comput. Phys. Commun.* 292 (2023) 108860.
- [4] Z. Tan, L. Chen, J. Yang, Generalized Allen–Cahn-type phase-field crystal model with FCC ordering structure and its conservative high-order accurate algorithm, *Comput. Phys. Commun.* 286 (2023) 108656.
- [5] Y. Wang, X. Xiao, H. Zhang, X. Qian, S. Song, Efficient diffusion domain modeling and fast numerical methods for diblock copolymer melt in complex domains, *Comput. Phys. Commun.* 305 (2024) 109343.
- [6] X. Feng, H.J. Wu, A posteriori error estimates and an adaptive finite element method for the Allen–Cahn equation and the mean curvature flow, *J. Sci. Comput.* 24 (2005) 121–146.
- [7] Y. Chen, Y. Huang, N. Yi, A SCR-based error estimation and adaptive finite element method for the Allen–Cahn equation, *Comput. Math. Appl.* 78 (1) (2019) 204–223.
- [8] A. Shah, M. Sabir, M. Qasim, P. Bastian, Efficient numerical scheme for solving the Allen–Cahn equation, *Numer. Methods Partial Differ. Equ.* 34 (5) (2018) 1820–1833.
- [9] S. Ham, Y. Li, S. Kwak, D. Jeong, J. Kim, An efficient and fast adaptive numerical method for a novel phase-field model of crystal growth, *Commun. Nonlinear Sci. Numer. Simul.* 131 (2024) 107822.
- [10] V. Joshi, R.K. Jaiman, An adaptive variational procedure for the conservative and positivity preserving Allen–Cahn phase-field model, *J. Comput. Phys.* 366 (2018) 478–504.
- [11] M.A. Khanwale, K. Saurabh, M. Fernando, V.M. Calo, H. Sundar, J.A. Rossmann, B. Ganapathysubramanian, A fully-coupled framework for solving Cahn–Hilliard Navier–Stokes equations: second-order, energy-stable numerical methods on adaptive octree based meshes, *Comput. Phys. Commun.* 280 (2022) 108501.
- [12] M.A. Khanwale, K. Saurabh, M. Ishii, H. Sundar, J.A. Rossmann, B. Ganapathysubramanian, A projection-based, semi-implicit time-stepping approach for the Cahn–Hilliard Navier–Stokes equations on adaptive octree meshes, *J. Comput. Phys.* 475 (2023) 111874.
- [13] Y. Gao, R. Li, X. He, Y. Lin, A fully decoupled numerical method for Cahn–Hilliard–Navier–Stokes–Darcy equations based on auxiliary variable approaches, *J. Comput. Appl. Math.* 436 (2024) 115363.
- [14] A. Mostafavi, M. Ranjbar, V. Yurkiv, A.L. Yarin, F. Mashayek, MOOSE-based finite element framework for mass-conserving two-phase flow simulations on adaptive grids using the diffuse interface approach and a Lagrange multiplier, *J. Comput. Phys.* (2025) 113755.
- [15] L. Luo, Q. Zhang, H. Liu, J. Zhang, X.P. Wang, A numerical study of two-phase flows in complex domain with a generalized Navier slip and penetration boundary condition on permeable boundaries, *J. Comput. Phys.* (2025) 113980.
- [16] X. Chen, R. Li, J. Li, The linear, decoupled and fully discrete finite element methods for simplified two-phase ferrohydrodynamics model, *Appl. Numer. Math.* 210 (2025) 123–146.
- [17] H.G. Lee, J. Shin, J.Y. Lee, First and second order operator splitting methods for the phase field crystal equation, *J. Comput. Phys.* 299 (2015) 82–91.
- [18] H.G. Lee, A semi-analytical Fourier spectral method for the Swift–Hohenberg equation, *Comput. Math. Appl.* 74 (8) (2017) 1885–1896.
- [19] A. Stuart, A.R. Humphries, *Dynamical Systems and Numerical Analysis*, Cambridge University Press, 1998.
- [20] H. Kim, G. Lee, S. Kang, S. Ham, Y. Hwang, J. Kim, Hybrid numerical method for the Allen–Cahn equation on nonuniform grids, *Comput. Math. Appl.* 158 (2024) 167–178.
- [21] H.D. Ceniceros, A.M. Roma, A nonstiff, adaptive mesh refinement-based method for the Cahn–Hilliard equation, *J. Comput. Phys.* 225 (2) (2007) 1849–1862.
- [22] G. Lee, S. Kwak, Y. Choi, S. Lee, S. Kang, S. Ham, J. Kim, An operator splitting method for the Cahn–Hilliard equation on nonuniform grids, *Comput. Math. Appl.* 167 (2024) 207–216.
- [23] R.S. Varga, *Iterative Analysis*, 1962, New Jersey.
- [24] J.J. Eggleston, G.B. McFadden, P.W. Voorhees, A phase-field model for highly anisotropic interfacial energy, *Physica D* 150 (1–2) (2001) 91–103.
- [25] C. Lee, S. Ham, Y. Hwang, S. Kwak, J. Kim, An explicit fourth-order accurate compact method for the Allen–Cahn equation, *AIMS Math.* 9 (1) (2024) 735–762.