Physica A 670 (2025) 130625

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Minireview

A review of the numerical methods for solving the binary Allen–Cahn equation

Hyun Geun Lee [a], Yibao Li [b], Junxiang Yang [c], Soobin Kwak [d], Youngjin Hwang [d], Seokjun Ham [d], Hyundong Kim [e,f], Jyoti [g], Yunjae Nam [h], Junseok Kim [d,*]

[a] Department of Mathematics, Dongguk University, Seoul 04620, Republic of Korea
[b] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China
[c] School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macao Special Administrative Region of China
[d] Department of Mathematics, Korea University, Seoul, 02841, Republic of Korea
[e] Department of Mathematics and Physics, Gangneung-Wonju National University, Gangneung 25457, Republic of Korea
[f] Smart infrastructure, Gangneung-Wonju National University, Gangneung 25457, Republic of Korea
[g] The Institute of Basic Science, Korea University, Seoul 02841, Republic of Korea
[h] Program in Actuarial Science and Financial Engineering, Korea University, Seoul 02841, Republic of Korea

ARTICLE INFO

Keywords:
Allen–Cahn equation
Numerical methods
Phase-field model

ABSTRACT

In this review, we present an overview of numerical methods to solve the binary Allen–Cahn (AC) equation, which is extensively used to model phase separation processes in materials science. It describes the time-dependent evolution of interfaces between two phases and accounts for both local reaction kinetics and diffusion effects. This equation plays a critical role in understanding the behavior of interfaces. The AC equation has various applications across fields such as materials science, physics, and biology, where it helps to analyze and predict phenomena such as phase transitions, grain boundary motion, and pattern formation in complex systems. Its importance lies in its ability to model the dynamics of interfaces and help the study of pattern formation and phase transitions in diverse environments. We discuss various computational methodologies developed for this important mathematical model and describe their strengths, limitations, and applications in diverse scientific domains.

Contents

* Corresponding author.
E-mail address: cfdkim@korea.ac.kr (J. Kim).
URL: https://mathematicians.korea.ac.kr/cfdkim (J. Kim).

https://doi.org/10.1016/j.physa.2025.130625
Received 23 December 2024; Received in revised form 3 March 2025
Available online 13 May 2025

1. Introduction

The Allen–Cahn (AC) equation (1), named after John W. Cahn and Sam Allen, models phase separation processes in binary alloy systems. It governs the dynamics of order–disorder transitions, where different phases within a material develop and separate over time. This equation models the evolution of the state of the components as a result of the combined effects of diffusion and local reactions. The AC equation is commonly used in the study of materials science, particularly in understanding how microstructures evolve during phase transitions, such as in alloys or polymers [1]. The Ginzburg–Landau free energy functional has been used to solve a broad spectrum of problems, including phase separations [2], fluid flows [3], energy minimizers, crystal growth [4], dendritic growth without artificial curvature effects [5], vector-valued AC equation, and fluid topology optimization [6,7], shell-infill structures [8], and optimization without curvature effects [9]. Furthermore, the many modified forms of the AC equation have been used to image analysis [10,11], motion by mean curvature [12–14], topology optimization [15], and shape transformation [16,17]. Shape transformation has been applied to remote sensing, medicine, computer graphics, and special effects creation. Shape transformation can be performed between objects of two- and three-dimensional shapes and it can be modeled by the AC equation with the difference between the source and target shapes [16,17].

In this review, we present a comprehensive overview of numerical methodologies used to solve the binary AC equation [1], which models the dynamics of interfaces between two distinct phases, *e.g.*, solid–liquid separation in alloys:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = -\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t), \tag{1}$$

where $\phi(\mathbf{x},t)$ is the difference between the concentrations of the two mixture components (the value of ϕ typically varies between -1 and 1, where $\phi=-1$ corresponds to one phase and $\phi=1$ corresponds to the other phase), ϵ is a parameter associated with the interface thickness between phases, and $F(\phi)=0.25(\phi^2-1)^2$, see Fig. 1(a). This quartic polynomial can be regarded as a Taylor series approximation of the following logarithmic potential [18]:

$$F(\phi)=\theta\left[\frac{1+\phi}{2}\ln\left(\frac{1+\phi}{2}\right)+\frac{1-\phi}{2}\ln\left(\frac{1-\phi}{2}\right)\right]+\frac{\theta_c}{2}(1-\phi^2), \tag{2}$$

where θ and θ_c are positive parameters, see Fig. 1(b). For the derivation of the Helmholtz free energy density $F(\phi)$ for a homogeneous system based on fundamental thermodynamic principles, refer to the review paper [19].

As illustrated in Fig. 2(a), when the value of ϕ ranges between -1 and 1, the region where $-0.9 \le \phi \le 0.9$ is approximately considered the interface thickness, although this definition is not absolute. If the desired interface thickness is mh, where h is the spatial step, then ϵ can be defined as $\epsilon_m = mh/(2\sqrt{2}\tanh^{-1}(0.9))$ [22]. We note that the concentration value is also restricted between 0 and 1. When we define $u=(\phi+1)/2$, the u value represents different phases when it is between 0 and 1. Thus, the free energy takes the shape shown in Fig. 1(c), and the interface thickness, as shown in Fig. 2(b), is determined by locating the region where $0.05 \le u \le 0.95$, and is represented as $\epsilon_m = mh/(4\sqrt{2}\tanh^{-1}(0.9))$.

Typically, the homogeneous Neumann boundary condition is used:

$$\mathbf{n}\cdot\nabla\phi = 0 \text{ on } \partial\Omega, \tag{3}$$

where $\mathbf{n}$ denotes the normal vector on $\partial\Omega$. Physically, this boundary condition imposes an orthogonal contact interface to the domain boundary. We can also use periodic boundary condition.

By applying the L^2-gradient flow to the total free energy functional the AC equation can be obtained as shown below:

$$\mathcal{E}(\phi)=\int_\Omega\left(\frac{F(\phi)}{\epsilon^2}+\frac{1}{2}|\nabla\phi|^2\right)d\mathbf{x}. \tag{4}$$

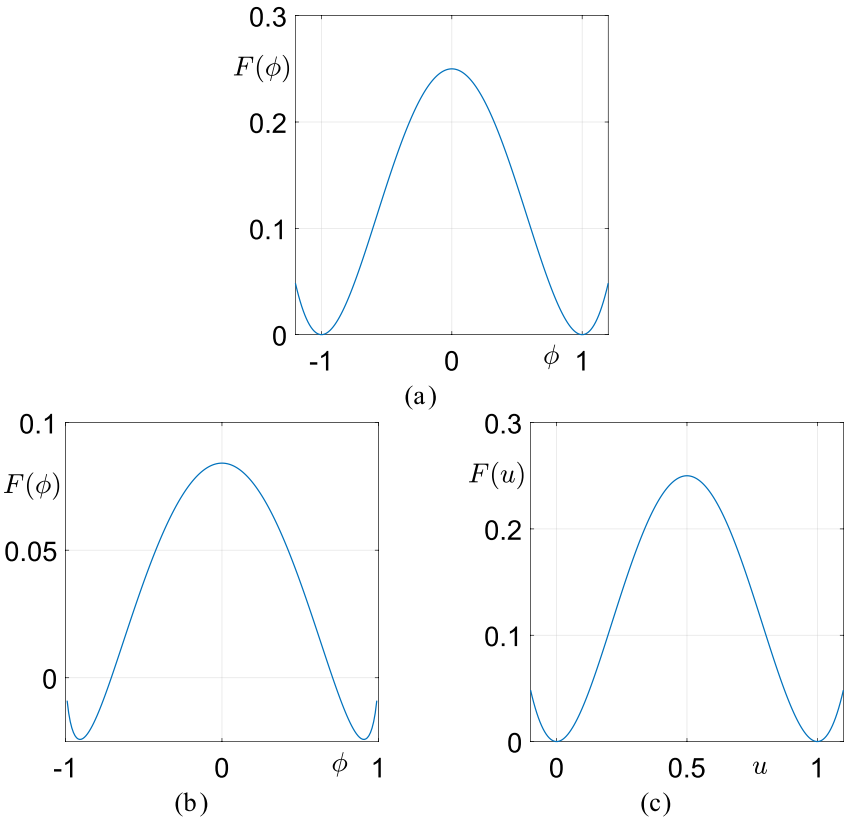

Fig. 1. (a) $F(\phi) = 0.25(\phi^2 - 1)^2$, (b) logarithmic potential, and (c) $F(u) = 4u^2(u-1)^2$ [20,21].

Differentiating the energy $\mathcal{E}(\phi)$ with respect to t gives

$$\begin{aligned}\frac{d}{dt}\mathcal{E}(\phi) &= \int_\Omega \left(\frac{F'(\phi)}{\epsilon^2}\phi_t + \nabla\phi\cdot\nabla\phi_t\right)d\mathbf{x} = \int_\Omega \left(\frac{F'(\phi)}{\epsilon^2} - \Delta\phi\right)\phi_t d\mathbf{x}\\ &= -\int_\Omega (\phi_t)^2 d\mathbf{x} \le 0.\end{aligned} \tag{5}$$

This follows from the application of integration by parts and the zero Neumann boundary condition (3), which guarantees a decrease in total energy over time.

Many numerical methods are used to solve the AC equation. Lee [23] developed a numerical scheme for the AC equation with a fixed boundary to consider the area-minimizing surface. In [24], Wei and Yang studied and proved the existence of phase transition layer solutions for anisotropic AC equations. Ntsokongo [25] investigated the asymptotic dynamics of the AC type equation with Dirichlet boundary conditions and temperature. Alsayed et al. [26] considered the optimal control problem for tumor growth models by coupling the AC equation and a reaction diffusion equation that describes tumor evolution in the presence of a nutrient supply. Nara [27] considered the initial value problem of the AC equation and investigates the large time dynamics for solutions with spreading fronts. Li et al. [28] proved the fractional energy dissipation laws for time-practical phase field equations including the AC and Cahn–Hilliard (CH) equations. Batangouna [29] investigated the relationship with AC equations by considering the time semi-discretization of the Ginzburg–Landau equation using the backward Euler method.

Furthermore, data-driven methods have gained considerable attention in recent years to solve the AC equation. Neural networks and deep learning-based methods are employed by [30] to solve both classical and conservative AC equations. To capture the fully discrete operators between consecutive time steps, two specialized convolutional neural network models are developed, one for each equation. These methods autonomously generate relevant samples throughout the time evolution, aiding in the training process. Xia et al. [31] proposed a simple algorithm for implementing the data assimilation method using the AC equation. A higher-dimensional AC equation with higher-order polynomial potential functions was solved by [32] using backward stochastic differential equations and deep neural networks. Mattey and Ghosh [33] proposed a backward compatible PINN (bc-PINN) scheme, which sequentially solves PDEs over successive time segments using a single neural network and retrains it while ensuring compatibility with previous solutions, and demonstrated superior accuracy and efficiency through applications on the CH and AC equations. An adaptive deep learning approach was developed to solve the AC equation, leveraging PINNs integrated with an adaptive collocation strategy [34]. This method improves the robustness of the model by using a multi-step discrete time modeling approach, which improves the

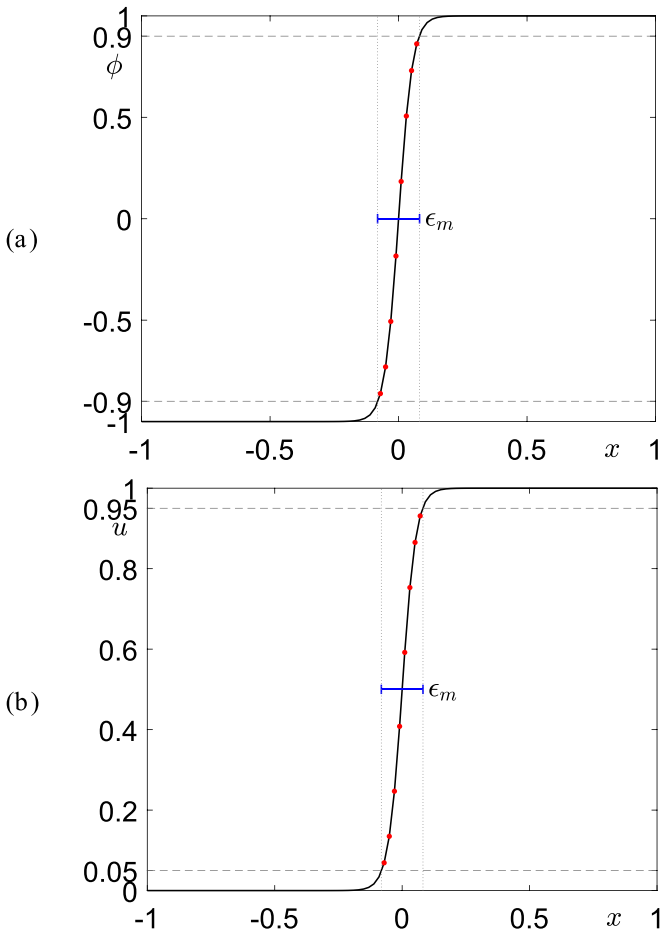

Fig. 2. Schematic illustration of interface thickness.

selection of initial training data for training. The authors demonstrate through computational experiments that this approach not only maintains the accuracy of short-time predictions but also shows promising results in the long-term behavior prediction of solutions, thereby solving one of the key challenges in using neural networks for dynamic systems modeled by partial differential equations. Wang et al. [35] introduced an innovative classification algorithm that combines a vector-valued AC equation with a deep neural network. Cheng et al. [36] considered the phase field sintering model, which consists of a CH type equation and multiple AC type equations. Kim et al. [37] proposed a model architecture using padding, convolution operations, and GPU acceleration to optimize the simulation speed of the AC equation. They demonstrated that this approach significantly improves computation speed while maintaining accuracy and validated its effectiveness through computational experiments in both two- and three-dimensional spaces.

In addition, the conservative AC equation has been extensively studied. Teng et al. [38] presented a numerical method that satisfies the maximum principle and mass conservation based on the Runge–Kutta-type integrators that are free of time delay for solving the conservative nonlocal AC equation. Tan et al. [39] developed a time-dependent auxiliary variable method for modeling ternary conservative AC fluids. Wang et al. [40] investigated the fractal properties of phase value time series in the conservative AC equation. Choi and Kim [41] developed a numerical method for the conservative AC equation that preserves the maximum principle and unconditional stability.

In this review, the content is organized as follows. Section 2 presents a range of numerical methods for solving the binary AC equation, while Section 3 provides concluding remarks.

2. Computational solution algorithms

We now consider various computational solution algorithms for the AC equation, such as the finite difference, finite element, finite volume, Fourier spectral, invariant energy quadratization, and scalar auxiliary variable methods, among others.

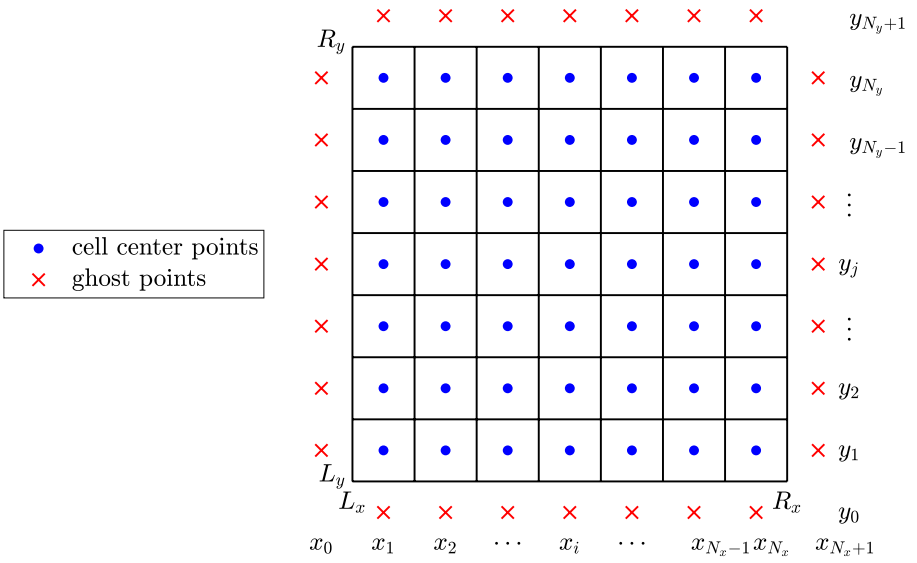

Fig. 3. Discrete computational domain.

2.1. Finite difference method

The finite difference method (FDM) is a computational methodology applied to approximate solutions to differential equations by partitioning the domain into a discrete grid and estimating derivatives through finite difference approximations. It is widely applied in various fields such as physics, engineering, and finance, for solving differential equations computationally. Zhang et al. [42] analyzed the maximum principle characteristic of the integrating factor Runge–Kutta (IFRK) scheme, which is fully explicit and temporally fourth order accurate. In [43], the authors presented the analysis of the stability of an explicit FDM for the AC equation. Yang et al. [44] developed a fast evolution computational algorithm for solving the AC equation. They developed a time rescaling scheme for the nonlinear portion of the AC equation to overcome the problem of rapid interface transition layers caused by the dominance of the nonlinear part in the operator splitting method with large time steps. Hwang et al. [45] developed an efficient and fast FDM for solving the AC equation on the cubic surface while taking into account the properties of the cubic surface.

To provide a concrete example of the FDM, we consider a computational method for the AC equation on a domain $\Omega = (L_x, R_x) \times (L_y, R_y)$ in two-dimensional space. Let $\Omega_h = \{(x_i, y_j) | x_i = L_x + (i - 0.5)h,\ y_j = L_y + (j - 0.5)h,\ 1 \le i \le N_x,\ 1 \le j \le N_y\}$ be the discrete domain, where N_x and N_y are integers and $h = (R_x - L_x)/N_x$ represents the spatial step size, and see Fig. 3.

Let $\phi_{ij}^n = \phi(x_i, y_j, n\Delta t)$ be defined on the discrete computational domain, where Δt is the time step. The AC equation (1) can then be discretized explicitly using the FDM as follows:

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} = -\frac{F'(\phi_{ij}^n)}{\epsilon^2} + \frac{\phi_{i+1,j}^n + \phi_{i-1,j}^n - 4\phi_{ij}^n + \phi_{i,j+1}^n + \phi_{i,j-1}^n}{h^2}, \tag{6}$$

which can be rewritten explicitly as

$$\phi_{ij}^{n+1} = \phi_{ij}^n + \Delta t \left(-\frac{F'(\phi_{ij}^n)}{\epsilon^2} + \frac{\phi_{i+1,j}^n + \phi_{i-1,j}^n - 4\phi_{ij}^n + \phi_{i,j+1}^n + \phi_{i,j-1}^n}{h^2} \right). \tag{7}$$

The boundary condition (3) at the ghost points (see Fig. 3) is defined as follows:

$$\phi_{i0}^n = \phi_{i1}^n \quad \text{and} \quad \phi_{i,N_y+1}^n = \phi_{iN_y}^n, \quad \text{for } i = 1,\ 2,\ \ldots,\ N_x, \tag{8}$$

$$\phi_{0j}^n = \phi_{1j}^n \quad \text{and} \quad \phi_{N_x+1,j}^n = \phi_{N_x j}^n, \quad \text{for } j = 1,\ 2,\ \ldots,\ N_y. \tag{9}$$

2.2. Finite element method

The finite element method (FEM) is a numerical methodology used for solving engineering and mathematical problems [46]. It discretizes a continuous domain into smaller elements to approximate complex systems and allows for efficient analysis of structures, heat transfer, fluid flow, and other phenomena in various fields of science and engineering. When sufficient elements and appropriate mesh density are used, FEM can produce accurate numerical results. Additionally, higher accuracy can be achieved when FEM is used in conjunction with high-order numerical schemes [47]. FEM can handle problems with complex geometry structures and irregular boundary conditions more effectively than FDM, as it allows for non-uniform mesh refinement and adaptive meshing [48]. FEM

typically provides higher accuracy than FDM, especially for problems with curved boundaries or strong spatial variations [49,50]. FEM is less sensitive to grid spacing compared to FDM, as it allows for local refinement in regions of interest. However, FEM implementation can be more complex than FDM, particularly for problems involving non-linearities or transient dynamics. In terms of computational cost, FEM solutions may require more computational resources compared to FDM, especially for problems with large numbers of elements or degrees of freedom. While both FDM and FEM are powerful numerical techniques for solving partial differential equations, the choice between them depends on the specific conditions and structures of the problem at hand. FDM may be more suitable for problems with simple geometries and regular boundaries, where computational efficiency is crucial. On the other hand, FEM provides greater flexibility and accuracy, which makes it preferable for problems with complex geometries and irregular boundaries, despite its higher implementation and computational costs.

To illustrate the finite element method with a concrete example, let us consider an unconditionally stable hybrid method to solve the AC equation using FEM [51]. We formally divide the AC equation (1) into the following two separate equations:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = \Delta\phi(\mathbf{x},t), \tag{10}$$

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = -\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2}. \tag{11}$$

Eqs. (10) and (11) are solved using a fully implicit FEM and a closed-form analytical solution, respectively. We divide the domain Ω into a collection $\mathcal{T}_h$, which is composed of triangular elements denoted by σ. We define the finite element space as $\boldsymbol{\Phi}_h = \{\psi \in C(\bar{\Omega}) \ : \ \psi|_\sigma$ is a linear function $\forall \sigma \in \mathcal{T}_h\}$. Let $\{\mathbf{x}_i\}_{i=1}^N$ represent the nodes of $\mathcal{T}_h$, and let $\{\eta_i\}_{i=1}^N$ denote the linear basis functions, where $\eta_i \in \boldsymbol{\Phi}_h$, $\eta_i(\mathbf{x}_j) = \delta_{ij}$, for $i, j = 1, \ldots, N$. The finite element solution ϕ_h^* to Eq. (10) is derived by solving the following equation:

$$\left(\frac{\phi_h^* - \phi_h^n}{\Delta t}, \psi_h\right) + (\nabla\phi_h^*, \nabla\psi_h) = 0, \quad \text{for all} \quad \psi_h \in \boldsymbol{\Phi}_h. \tag{12}$$

Next, Eq. (11) is solved analytically using the initial condition ϕ_h^*.

$$\phi_h^{n+1} = \frac{\phi_h^*}{\sqrt{e^{-\frac{2\Delta t}{\epsilon^2}} + \left(\phi_h^*\right)^2\left(1 - e^{-\frac{2\Delta t}{\epsilon^2}}\right)}}. \tag{13}$$

Fig. 4 displays a triangular mesh used in the finite element analysis.

The FEM is widely regarded as a highly effective approach for solving problems within complex domains, particularly those involving irregular geometries and boundary conditions. Its ability to discretize the domain into smaller, flexible elements allows for accurate approximations. We note that FEM can also be applied to the CH equation [52].

2.3. Finite volume method

The finite volume method (FVM) is a powerful numerical approach that approximates differential equations by transforming them into a conservative form. This method divides the computational domain into a finite number of control volumes CV and applies conservation laws to each volume. FVM preserves physical conservation properties and can be flexibly applied to problems with complex boundaries. To apply the FVM, the computational domain Ω is divided into a finite number of control volumes CV, and the AC equation is integrated over each control volume as follows. By dividing the numerical domain Ω into a finite number of control volumes CV and integrating the AC equation over each control volume CV, we obtain:

$$\int_{CV} \frac{\partial \phi}{\partial t} d\mathbf{x} = \int_{CV} \Delta\phi d\mathbf{x} - \int_{CV} \frac{F'(\phi)}{\epsilon^2} d\mathbf{x}. \tag{14}$$

Applying Gauss's divergence theorem to the Laplacian term $\Delta\phi$, we obtain:

$$\int_{CV} \Delta\phi \, d\mathbf{x} = \int_{\partial CV} \nabla\phi \cdot \mathbf{n} dS.$$

Using the above equation, the conservation form of the Eq. (14) is expressed as:

$$\int_{CV} \frac{\partial \phi}{\partial t} d\mathbf{x} - \sum_{\sigma \in \partial CV} \int_\sigma \nabla\phi \cdot \mathbf{n} dS = -\frac{1}{\epsilon^2} \int_{CV} F'(\phi) d\mathbf{x}. \tag{15}$$

Here, $\mathbf{n}$ represents the outward normal vector on the boundary ∂CV of control volume CV. Eq. (15) can be solved by computing the flux using various numerical methods. The flux term $\int_\sigma \nabla\phi \cdot \mathbf{n} dS$ is typically approximated using finite difference schemes. Strachota and Beneš [53] derived an error estimate for the numerical solution of the FVM applied to the isotropic AC equation and analyzed its numerical convergence by conducting 2D and 3D simulations.

2.4. Fourier spectral method

The Fourier spectral method is a numerical technique used in mathematics and engineering to approximate solutions to differential equations by representing functions as combinations of sinusoidal waves via Fourier series and enables efficient computation of complex phenomena with high accuracy. To improve the efficiency of the Fourier spectral method, numerous researchers have adopted stabilized semi-implicit schemes [54–57] for the temporal variable. These schemes treat the main elliptic

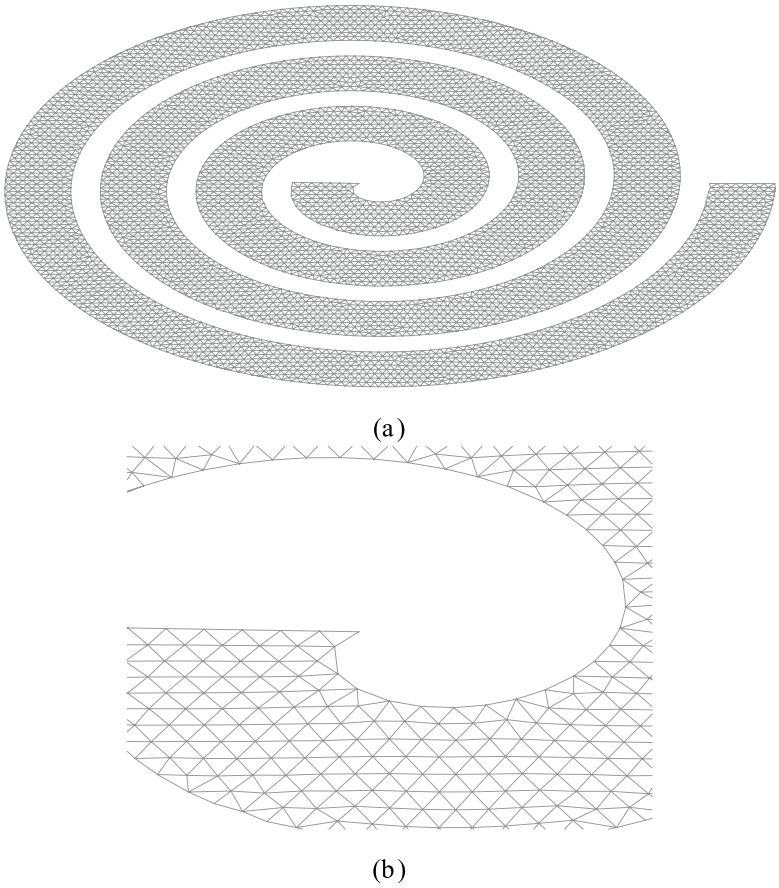

Fig. 4. (a) A triangular mesh and (b) a magnified view of the triangular mesh for FEM.

operator implicitly and the nonlinear term explicitly to avoid the costly task of solving nonlinear equations at every time step. In addition, a stabilizing term is incorporated to improve stability and maintain simplicity. Stabilized semi-implicit schemes reduce the time step restrictions inherent in explicit schemes, and allow for significantly larger time steps. However, it has been observed that when a large time step is employed, the effective time step becomes smaller than the one prescribed in the stabilized semi-implicit method [54,58]. As a result, the stabilized semi-implicit method suffers from inaccuracy, and produces incorrect morphologies in phase separation processes. Lee and Lee [58] proposed the semi-analytical Fourier spectral method for solving the AC equation, which allows the use of a sufficiently large time step without sacrificing accuracy or introducing technical challenges. The key idea of the proposed methodology involves decomposing the AC equation into linear and nonlinear equations, which possess closed-form solutions in the Fourier and physical spaces, respectively.

We briefly review the semi-analytical Fourier spectral method and consider the AC equation (1) in $\Omega = (L_x, R_x) \times (L_y, R_y)$. Let the "*linear operator*" $\mathcal{L}^{\Delta t}$ be defined as follows:

$$\mathcal{L}^{\Delta t}(\phi(t^n)) := \phi(t^n + \Delta t),$$

where $\phi(t^n + \Delta t)$ represents the solution of the following linear diffusion equation:

$$\frac{\partial \phi}{\partial t} = \Delta \phi,$$

where we use $\phi(t^n)$ as an initial condition. We use the discrete cosine transform to solve Eq. (1) with a zero Neumann boundary condition $\mathbf{n} \cdot \nabla \phi = 0$ on $\partial\Omega$: for $k_x = 0, 1, \dots, N_x - 1$ and $k_y = 0, 1, \dots, N_y - 1$,

$$\widehat{\phi}_{k_x k_y} = \alpha_{k_x} \beta_{k_y} \sum_{l_x=0}^{N_x-1} \sum_{l_y=0}^{N_y-1} \phi_{l_x l_y} \cos\left[\frac{\pi}{N_x}\left(L_x + l_x + \frac{1}{2}\right) k_x\right] \cos\left[\frac{\pi}{N_y}\left(L_y + l_y + \frac{1}{2}\right) k_y\right],$$

where $\alpha_0 = \sqrt{1/N_x}$, $\beta_0 = \sqrt{1/N_y}$, $\alpha_{k_x} = \sqrt{2/N_x}$, $\beta_{k_y} = \sqrt{2/N_y}$ for $k_x, k_y \geq 1$, and $\phi_{l_x l_y} = \phi\left(\frac{R_x - L_x}{N_x}\left(L_x + l_x + \frac{1}{2}\right), \frac{R_y - L_y}{N_y}\left(L_y + l_y + \frac{1}{2}\right)\right)$. Then, we have an analytical formula for $\mathcal{L}^{\Delta t}$ in the discrete cosine space

$$\mathcal{L}^{\Delta t}(\phi) = C^{-1}\left[e^{A_{k_x k_y}\Delta t} C[\phi]\right],$$

where $A_{k_x k_y} = \pi^2\left[\left(\frac{k_x}{R_x - L_x}\right)^2 + \left(\frac{k_y}{R_y - L_y}\right)^2\right]$ and C is the discrete cosine transform and C^{-1} is its inverse transform. We also define the "*nonlinear operator*" $\mathcal{N}^{\Delta t}$ as follows:

$$\mathcal{N}^{\Delta t}(\phi(t^n)) := \phi(t^n + \Delta t),$$

where $\phi(t^n + \Delta t)$ represents the solution of the following nonlinear equation

$$\frac{\partial \phi}{\partial t} = -\frac{F'(\phi)}{\epsilon^2},$$

where we use $\phi(t^n)$ as an initial condition. A closed-form solution for $\mathcal{N}^{\Delta t}$ is as follows:

$$\mathcal{N}^{\Delta t}(\phi) = \frac{\phi}{\sqrt{e^{-\frac{2\Delta t}{\epsilon^2}} + \phi^2(1 - e^{-\frac{2\Delta t}{\epsilon^2}})}}.$$

Then, the first- and second-order semi-analytical Fourier spectral methods for the AC equation can be expressed as follows:

$$\phi^{n+1} = \left(\mathcal{L}^{\Delta t} \circ \mathcal{N}^{\Delta t}\right)\phi^n$$

and

$$\phi^{n+1} = \left(\mathcal{L}^{\Delta t/2} \circ \mathcal{N}^{\Delta t} \circ \mathcal{L}^{\Delta t/2}\right)\phi^n,$$

respectively. Here, ϕ^n and ϕ^{n+1} are approximations of $\phi(t^n)$ and $\phi(t^n + \Delta t)$, respectively. In the Appendix, we provide MATLAB code for a Fourier spectral method.

2.5. Invariant energy quadratization method

The invariant energy quadratization (IEQ) scheme was originally proposed by Yang [59] to efficiently construct an energy-stable time-marching method for gradient flows. The basic idea of the IEQ method is to introduce an auxiliary variable

$$\psi(\mathbf{x}, t) = \sqrt{\frac{F(\phi)}{\epsilon^2} + C}, \tag{16}$$

where C is a constant such that $\frac{F(\phi)}{\epsilon^2} + C > 0$. Then, one can redefine the energy $\mathcal{E}(\phi)$ as

$$\mathcal{E}_{IEQ}(\phi, \psi) = \int_\Omega \left(\psi^2 + \frac{1}{2}|\nabla\phi|^2\right) d\mathbf{x} - C|\Omega| \tag{17}$$

and reformulate the AC equation as

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= -G(\phi)\psi + \Delta\phi, \\ \frac{\partial \psi}{\partial t} &= \frac{1}{2}G(\phi)\frac{\partial \phi}{\partial t}, \end{aligned} \tag{18}$$

where

$$G(\phi) = \frac{f(\phi)}{\epsilon^2\sqrt{\frac{F(\phi)}{\epsilon^2} + C}}.$$

The new system (18) still satisfies the energy dissipation property:

$$\frac{d\mathcal{E}_{IEQ}}{dt} = \int_\Omega \left(2\psi\frac{\partial \psi}{\partial t} - \Delta\phi\frac{\partial \phi}{\partial t}\right) d\mathbf{x} = -\int_\Omega \left(\frac{\partial \phi}{\partial t}\right)^2 d\mathbf{x} \leq 0.$$

Let $\phi^n(\mathbf{x}) = \phi(\mathbf{x}, t)$ at $t = n\Delta t$. Here, $\Delta t = T/N_T$ is the uniform temporal step, T is the total computational time, and N_T is the number of time iterations. The temporally first-order accurate method based on backward Euler scheme is as follows:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = -G(\phi^n)\psi^{n+1} + \Delta\phi^{n+1}, \tag{19}$$

$$\frac{\psi^{n+1} - \psi^n}{\Delta t} = \frac{1}{2}G(\phi^n)\frac{\phi^{n+1} - \phi^n}{\Delta t}. \tag{20}$$

The periodic or zero Neumann boundary condition (*i.e.*, $\mathbf{n} \cdot \nabla\phi^{n+1}|_{\partial\Omega} = 0$) is used on the domain boundary $\partial\Omega$. For two functions f_a and f_b, their L^2-inner product is defined as $(f_a, f_b) = \int_\Omega f_a f_b \, d\mathbf{x}$. The associated L^2-norm is defined as $(f_a, f_a) = \|f_a\|^2$.

Theorem 1. *Eqs. (19) and (20) lead to a time-discretized energy dissipation law with respect to the following modified energy*

$$E_1^{n+1} = \|\psi^{n+1}\|^2 + \frac{1}{2}\|\nabla\phi^{n+1}\|^2 - C|\Omega|. \tag{21}$$

Proof. Taking the L^2-inner product of Eq. (19) with $-(\phi^{n+1}-\phi^n)$, we get

$$\begin{aligned} -\frac{1}{\Delta t}\|\phi^{n+1}-\phi^n\|^2 &= \int_\Omega G(\phi^n)\psi^{n+1}(\phi^{n+1}-\phi^n)\,d\mathbf{x} + \frac{1}{2}\|\nabla\phi^{n+1}\|^2 - \frac{1}{2}\|\nabla\phi^n\|^2 \\ &\quad + \frac{1}{2}\|\nabla(\phi^{n+1}-\phi^n)\|^2. \end{aligned} \tag{22}$$

Multiplying Eq. (20) with $2\psi^{n+1}$, we get

$$\|\psi^{n+1}\|^2 - \|\psi^n\|^2 + \|\psi^{n+1}-\psi^n\|^2 = \int_\Omega G(\phi^n)\psi^{n+1}(\phi^{n+1}-\phi^n)\,d\mathbf{x}. \tag{23}$$

Combining Eqs. (22) and (23), we obtain

$$\begin{aligned} &\|\psi^{n+1}\|^2 - \|\psi^n\|^2 + \frac{1}{2}\|\nabla\phi^{n+1}\|^2 - \frac{1}{2}\|\nabla\phi^n\|^2 \\ &= -\frac{1}{\Delta t}\|\phi^{n+1}-\phi^n\|^2 - \frac{1}{2}\|\nabla(\phi^{n+1}-\phi^n)\|^2 - \|\psi^{n+1}-\psi^n\|^2 \le 0. \end{aligned} \tag{24}$$

The proof of time-discretized energy dissipation law is completed. □

Similarly, the second order IEQ method can be developed using the second order backward difference formula. Please refer to [59] and references therein for more details.

2.6. Scalar auxiliary variable method

The scalar auxiliary variable (SAV) method was originally proposed by Shen et al. [60] to efficiently construct energy-stable time-marching schemes for gradient flows. The fundamental concept of the SAV method is to introduce a time-dependent auxiliary variable that theoretically corresponds to the square root of the sum of a nonlinear energy term and a constant. Here, the nonlinear energy contributes to the total free energy functional of a gradient flow system. Based on the evolution equation of the auxiliary variable and the equivalent governing equations of phase-field function, energy-stable first- or second-order schemes can be easily developed. Consider the AC equation, the time-dependent auxiliary variable is defined as

$$r = r(t) = \sqrt{\int_\Omega \frac{F(\phi(\mathbf{x},t))}{\epsilon^2}\,d\mathbf{x} + C}. \tag{25}$$

Here, C is a constant. The evolution equation for r reads as

$$\frac{dr}{dt} = \frac{1}{2}\int_\Omega \left(\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2\sqrt{\int_\Omega \frac{F(\phi(\mathbf{x},t))}{\epsilon^2}\,d\mathbf{x} + C}} \right) \frac{\partial\phi(\mathbf{x},t)}{\partial t}\,d\mathbf{x}. \tag{26}$$

It is easy to find that $\frac{r}{\sqrt{\int_\Omega \frac{F(\phi(\mathbf{x},t))}{\epsilon^2}\,d\mathbf{x}+C}} \equiv 1$, the AC equation can be modified to be

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = \frac{-r}{\sqrt{\int_\Omega \frac{F(\phi(\mathbf{x},t))}{\epsilon^2}\,d\mathbf{x} + C}} \frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t),\ \mathbf{x}\in\Omega,\ t>0. \tag{27}$$

Combining Eqs. (26) and (27), the equivalent AC model reads as

$$\frac{\partial\phi(\mathbf{x},t)}{\partial t} = \frac{-r}{\sqrt{\int_\Omega \frac{F(\phi(\mathbf{x},t))}{\epsilon^2}\,d\mathbf{x} + C}} \frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t),\ \mathbf{x}\in\Omega,\ t>0, \tag{28}$$

$$\frac{dr}{dt} = \frac{1}{2}\int_\Omega \left(\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2\sqrt{\int_\Omega \frac{F(\phi(\mathbf{x},t))}{\epsilon^2}\,d\mathbf{x} + C}} \right) \frac{\partial\phi(\mathbf{x},t)}{\partial t}\,d\mathbf{x}. \tag{29}$$

To simplify the expression, we use ϕ to replace $\phi(\mathbf{x})$ in this subsection. The temporally first-order accurate scheme based on the backward Euler method is as follows:

$$\frac{\phi^{n+1}-\phi^n}{\Delta t} = \frac{-r^{n+1}}{\sqrt{\int_\Omega \frac{F(\phi^n)}{\epsilon^2}\,d\mathbf{x} + C}} \frac{F'(\phi^n)}{\epsilon^2} + \Delta\phi^{n+1} - \frac{S}{\epsilon^2}(\phi^{n+1}-\phi^n), \tag{30}$$

$$\frac{r^{n+1}-r^n}{\Delta t} = \frac{1}{2}\int_\Omega \left(\frac{F'(\phi^n)}{\epsilon^2\sqrt{\int_\Omega \frac{F(\phi^n)}{\epsilon^2}\,d\mathbf{x} + C}} \right) \frac{\phi^{n+1}-\phi^n}{\Delta t}\,d\mathbf{x}. \tag{31}$$

Here, $S > 0$ is a stabilization parameter. For two functions f_a and f_b, their L^2-inner product is defined as $(f_a, f_b) = \int_\Omega f_a f_b \, d\mathbf{x}$. The associated L^2-norm is defined as $(f_a, f_a) = \|f_a\|^2$.

Theorem 2. *Eqs. (30) and (31) lead to a time-discretized energy dissipation law with respect to the following modified energy*

$$E_1^{n+1} = \frac{1}{2}\|\nabla\phi^{n+1}\|^2 + (r^{n+1})^2 - C. \tag{32}$$

Proof. Taking the L^2-inner product of Eq. (30) with $-(\phi^{n+1} - \phi^n)$, we get

$$\begin{aligned} -\frac{1}{\Delta t}\|\phi^{n+1} - \phi^n\|^2 &= \int_\Omega \left(\frac{F'(\phi^n)}{\epsilon^2\sqrt{\int_\Omega \frac{r^{n+1}F(\phi^n)}{\epsilon^2}\, d\mathbf{x} + C}} \right)(\phi^{n+1} - \phi^n) d\mathbf{x} \\ &+ \frac{1}{2}\|\nabla\phi^{n+1}\|^2 - \frac{1}{2}\|\nabla\phi^n\|^2 + \frac{1}{2}\|\nabla\phi^{n+1} - \nabla\phi^n\|^2 + \frac{S}{\epsilon^2}\|\phi^{n+1} - \phi^n\|^2. \end{aligned} \tag{33}$$

Multiplying Eq. (31) with $2r^{n+1}$, we get

$$(r^{n+1})^2 - (r^n)^2 + (r^{n+1} - r^n)^2 = \int_\Omega \left(\frac{F'(\phi^n)}{\epsilon^2\sqrt{\int_\Omega \frac{r^{n+1}F(\phi^n)}{\epsilon^2}\, d\mathbf{x} + C}} \right)(\phi^{n+1} - \phi^n) d\mathbf{x}. \tag{34}$$

Combining Eqs. (33) and (34), we obtain

$$\begin{aligned} \frac{1}{2}\|\nabla\phi^{n+1}\|^2 - \frac{1}{2}\|\nabla\phi^n\|^2 + (r^{n+1})^2 - (r^n)^2 &= -\frac{1}{2}\|\nabla\phi^{n+1} - \nabla\phi^n\|^2 \\ &- \left(\frac{1}{\Delta t} + \frac{S}{\epsilon^2}\right)\|\phi^{n+1} - \phi^n\|^2 - (r^{n+1} - r^n)^2 \le 0. \end{aligned} \tag{35}$$

The proof of the time-discretized energy dissipation law is completed. □

Using the second-order backward difference formula (BDF2), the temporally second-order accurate method is as follows:

$$\begin{aligned} \frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\Delta t} &= \frac{-r^{n+1}}{\sqrt{\int_\Omega \frac{F(\phi^*)}{\epsilon^2}\, d\mathbf{x} + C}} \frac{F'(\phi^*)}{\epsilon^2} + \Delta\phi^{n+1} \\ &- \frac{S}{\epsilon^2}(\phi^{n+1} - \phi^*), \end{aligned} \tag{36}$$

$$\frac{3r^{n+1} - 4r^n - r^{n-1}}{2\Delta t} = \int_\Omega \left(\frac{F'(\phi^*)}{\epsilon^2\sqrt{\int_\Omega \frac{F(\phi^*)}{\epsilon^2}\, d\mathbf{x} + C}} \right)\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{4\Delta t}\, d\mathbf{x}. \tag{37}$$

Here, $(\cdot)^* = 2(\cdot)^n - (\cdot)^{n-1}$ is a linear extrapolation.

Theorem 3. *Eqs. (36) and (37) lead to a time-discretized energy dissipation law with respect to the following modified energy*

$$\begin{aligned} E_2^{n+1,n} &= \frac{1}{4}\left(\|\nabla\phi^{n+1}\|^2 + \|2\nabla\phi^{n+1} - \nabla\phi^n\|^2\right) + \frac{1}{4}(r^{n+1})^2 + \frac{1}{4}\left(2r^{n+1} - r^n\right)^2 \\ &+ \frac{S}{\epsilon^2}\|\phi^{n+1} - \phi^n\|^2 - C. \end{aligned} \tag{38}$$

Proof. Taking the L^2-inner product of Eq. (36) with $-\left(3\phi^{n+1} - 4\phi^n + \phi^{n-1}\right)$, we get

$$\begin{aligned} -\frac{1}{2\Delta t}\|3\phi^{n+1} - 4\phi^n + \phi^{n-1}\|^2 &= \int_\Omega \left(r^{n+1}\frac{F'(\phi^*)}{\epsilon^2\sqrt{\int_\Omega \frac{F(\phi^*)}{\epsilon^2}\, d\mathbf{x} + C}} \right)\left(3\phi^{n+1} - 4\phi^n + \phi^{n-1}\right)\, d\mathbf{x} \\ &+ \frac{1}{4}\left(\|\nabla\phi^{n+1}\|^2 - \|\nabla\phi^n\|^2 + \|2\nabla\phi^{n+1} - \nabla\phi^n\|^2 - \|2\nabla\phi^n - \nabla\phi^{n-1}\|^2\right) + \frac{S}{\epsilon^2}\|\phi^{n+1} - \phi^n\|^2 \\ &- \frac{S}{\epsilon^2}\|\phi^n - \phi^{n-1}\|^2 + \frac{2S}{\epsilon^2}\|\phi^{n+1} - 2\phi^n + \phi^{n-1}\|^2 + \frac{1}{4}\|\nabla\phi^{n+1} - 2\nabla\phi^n + \nabla\phi^{n-1}\|^2. \end{aligned} \tag{39}$$

Multiplying Eq. (37) with $2r^{n+1}$, we get

$$\begin{aligned} &\frac{1}{4}\left((r^{n+1})^2 - (r^n)^2 + (2r^{n+1} - r^n)^2 - (2r^n - r^{n-1})^2 + (r^{n+1} - 2r^n + r^{n-1})^2\right) \\ &= \int_\Omega \left(r^{n+1}\frac{F'(\phi^*)}{\epsilon^2\sqrt{\int_\Omega \frac{F(\phi^*)}{\epsilon^2}\, d\mathbf{x} + C}} \right)\left(3\phi^{n+1} - 4\phi^n + \phi^{n-1}\right)\, d\mathbf{x}. \end{aligned} \tag{40}$$

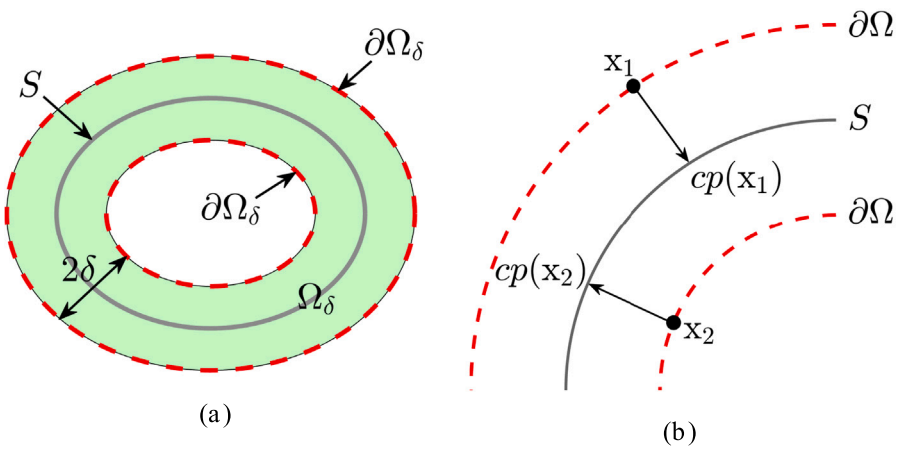

Fig. 5. (a) A graphical representation of S, Ω_δ, and $\partial\Omega_\delta$. (b) Depiction of the closest points, $\mathrm{cp}(\mathbf{x}_1)$ and $\mathrm{cp}(\mathbf{x}_2)$, corresponding to the points $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively.

Combining Eqs. (39) and (40), we obtain

$$\begin{aligned}
&\frac{1}{4}\left(\|\nabla\phi^{n+1}\|^2-\|\nabla\phi^n\|^2+\|2\nabla\phi^{n+1}-\nabla\phi^n\|^2-\|2\nabla\phi^n-\nabla\phi^{n-1}\|^2\right)+\frac{S}{\epsilon^2}\|\phi^{n+1}-\phi^n\|^2\\
&-\frac{S}{\epsilon^2}\|\phi^n-\phi^{n-1}\|^2+\frac{1}{4}\left((r^{n+1})^2+(2r^{n+1}-r^n)^2\right)-\frac{1}{4}\left((r^n)^2+(2r^n-r^{n-1})^2\right)+-\\
=&-\frac{1}{2\Delta t}\|3\phi^{n+1}-4\phi^n+\phi^{n-1}\|^2-\frac{1}{4}\|\nabla\phi^{n+1}-2\nabla\phi^n+\nabla\phi^{n-1}\|^2-\frac{1}{4}(r^{n+1}-2r^n+r^{n-1})^2\\
&-\frac{2S}{\epsilon^2}\|\phi^{n+1}-2\phi^n+\phi^{n-1}\|^2\le 0.
\end{aligned}\tag{41}$$

The proof of the time-discretized energy dissipation law is completed. □

Recently, many variants of the SAV scheme have been developed to treat gradient flow systems, hydrodynamics-coupled phase-field models, and incompressible fluid problems [61,62], nematic crystal flows [63] etc. Please refer to [64,65] and references therein for more details.

2.7. Hybrid methods and alternative approaches

A hybrid numerical method refers to a computational approach that combines two or more distinct numerical techniques to solve complex mathematical problems. These methods are often employed when a single numerical technique is inadequate for handling all aspects of a given problem, especially in cases where different techniques excel in specific domains. For instance, hybrid methods can merge closed-form solutions with finite difference methods. The motivation behind hybrid methods lies in optimizing accuracy, stability, and computational efficiency. By using the strengths of different methods, hybrid techniques can more effectively solve problems involving partial differential equations, nonlinear dynamics, or multi-physics phenomena. For example, one part of a domain may be treated with an efficient but less accurate method, while another critical part may use a more precise technique. These methods are widely applied in fields such as fluid dynamics, structural analysis, and electromagnetics, where diverse mathematical properties must be resolved. Overall, hybrid numerical methods provide a flexible and powerful means of solving intricate problems that may be difficult to handle using a single approach.

In [66], the authors presented an unconditionally stable Monte Carlo simulation (MCS) for solving the AC equation. This method is based on the Monte Carlo method with Brownian motion. Hwang et al. [67] developed an explicit computational scheme for the AC equation on effective symmetric triangular meshes, where the Laplace operator is defined based on the characteristics of the given triangular mesh. A broad spectrum of phenomena in applied and natural sciences can be represented through partial differential equations (PDEs) formulated on surfaces. There have been many applications in physics, fluid dynamics, and image processing. In particular, several studies have focused on various numerical methods for solving the AC equation on surfaces. Xiao et al. [49] proposed an unconditionally maximum principle preserving surface FEM for solving the AC equation. To solve PDEs on a triangulated curved surface, the Laplace–Beltrami operator is used. In [68], numerical simulations and error estimation for the AC equation on surfaces were presented using radial basis functions and a time-splitting scheme. Choi et al. [69] developed a fast and accurate FDM using a closest point method for the AC equation. Fig. 5 shows, in (a), a graphical representation of S, along with Ω_δ, and $\partial\Omega_\delta$, and in (b), $\mathrm{cp}(\mathbf{x}_1)$ and $\mathrm{cp}(\mathbf{x}_2)$ are shown, corresponding to the points $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively.

Fig. 6 provides the temporal evolution of the interface on a spherical surface. The motion is driven by mean curvature, and the figure captures the progression of this interface over time as it moves from left to right. This visual representation highlights the dynamic changes occurring on the surface as the curvature-driven flow influences the shape and position of the interface.

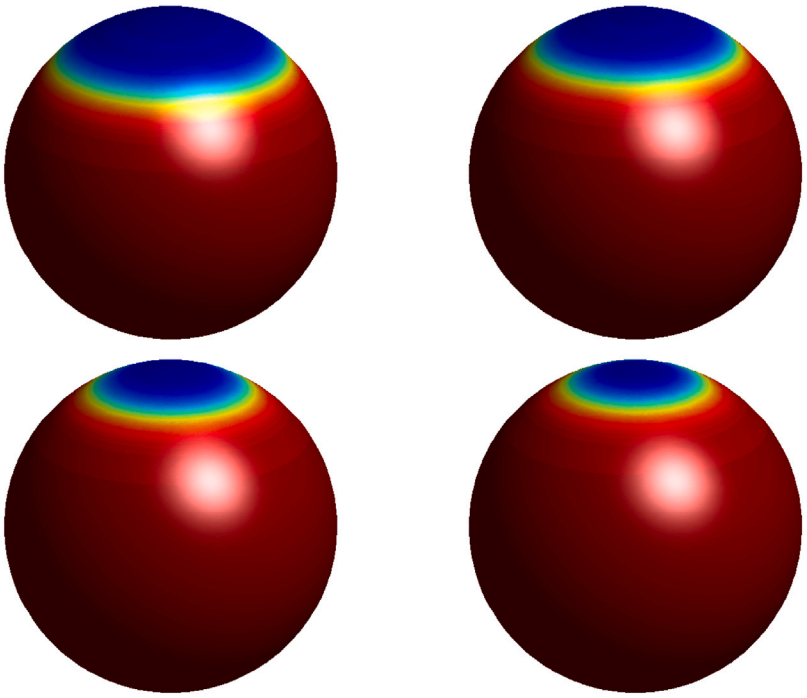

Fig. 6. From left to right and top to bottom: The temporal evolution of the interface on spherical surface driven by mean curvature motion.

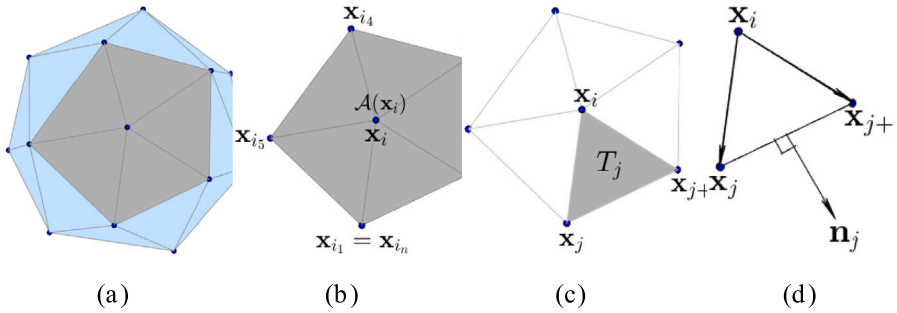

Fig. 7. Schematic diagrams.

Choi et al. [70] presented an explicit hybrid numerical method for the AC equation on curved surfaces. By applying the operator splitting technique, the heat equation is solved using a discrete Laplace–Beltrami operator, while the nonlinear equation is solved analytically.

The solution algorithm is as follows: Let M represent a triangular mesh of S, and let $\{\mathbf{x}_i\}_{i=1}^N$ denote the vertices. Define $N_1(i) = \{i_1, i_2, \dots, i_n\}$ as the set of indices corresponding to $\mathbf{x}_i$, where $i_1 = i_n$ (refer to Fig. 7(b)). Let $\phi_i = \phi(\mathbf{x}_i)$, and let T_j denote a triangle as illustrated in Fig. 7(c). The discrete gradient at $\mathbf{x}_i$ is expressed as follows:

$$\nabla_M \phi_i = \frac{1}{\mathcal{A}(\mathbf{x}_i)} \sum_{j \in N_1(i)} A_j \nabla_{T_j} \phi, \tag{42}$$

where $\mathcal{A}(\mathbf{x}_i) = \sum_{j \in N_1(i)} A_j$, see Fig. 7(b). Here, A_j is the area of T_j (Fig. 7(c)) and $\gamma(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = \left\langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_j - \mathbf{x}_k \right\rangle (\mathbf{x}_k - \mathbf{x}_i)$. Hence, a discrete Laplacian is as follows:

$$\Delta_M \phi_i = \frac{1}{2\mathcal{A}(\mathbf{x}_i)} \sum_{j \in N_1(i)} \mathbf{n}_j^T [\nabla_M \phi_j + \nabla_M \phi_{j+}] \, \|\mathbf{x}_j - \mathbf{x}_{j+}\|, \tag{43}$$

where $\mathbf{n}_j$ is normal to $\overline{\mathbf{x}_j \mathbf{x}_{j+}}$, see Fig. 7(d). Define $\phi_i^n = \phi(\mathbf{x}_i, n\Delta t)$. Then, first, we solve the discrete diffusion equation:

$$\frac{\phi_i^* - \phi_i^n}{\Delta t} = \Delta_M \phi_i^n, \quad \text{for} \quad 1 \le i \le N. \tag{44}$$

After that, we proceed to solve the next equation:

$$\phi_i^{n+1} = \frac{\phi_i^*}{\sqrt{e^{\frac{-2\Delta t}{\epsilon^2}} + (\phi_i^*)^2\left(1 - e^{\frac{-2\Delta t}{\epsilon^2}}\right)}}, \quad \text{for} \quad 1 \le i \le N. \tag{45}$$

For further details on the numerical methods for the AC equation on curved surfaces, as well as computational simulation results, please refer to [69].

Next, Hwang et al. [45] presented and analyzed the FDM for solving the AC equation on a cubic surface. Furthermore, the authors proved that the numerical solution satisfies the maximum principle property. We describe the numerical scheme for solving the AC equation on a cubic surface. Let N be a positive integer. We define the discrete sub-domains for the unfolded cubic surface as follows:

$$\begin{aligned}
\Omega_h^1 &= \{(x_i, y_j) | i = 1, 2, \dots, N,\ j = 1, 2, \dots, N\},\\
\Omega_h^2 &= \{(x_i, y_j) | i = 1, 2, \dots, N,\ j = N+1, N+2, \dots, 2N\},\\
\Omega_h^3 &= \{(x_i, y_j) | i = N+1, N+2, \dots, 2N,\ j = N+1, N+2, \dots, 2N\},\\
\Omega_h^4 &= \{(x_i, y_j) | i = 2N+1, 2N+2, \dots, 3N,\ j = N+1, N+2, \dots, 2N\},\\
\Omega_h^5 &= \{(x_i, y_j) | i = 3N+1, 3N+2, \dots, 4N,\ j = N+1, N+2, \dots, 2N\},\\
\Omega_h^6 &= \{(x_i, y_j) | i = 1, 2, \dots, N,\ j = 2N+1, 2N+2, \dots, 3N\},
\end{aligned}$$

where $(x_i, y_j) = ((i-0.5)h, (j-0.5)h)$ and $h = 1/N$. Then, the discrete global domain is defined by $\Omega_h = \cup_{k=1}^6 \Omega_h^k$. Let $\phi_{ij}^n = \phi(x_i, y_j, n\Delta t)$. For the folded cubic surface, the appropriate boundary conditions of the discrete sub-domain Ω_h^k for $k = 1, 2, \dots, 6$ are given as follows. For $s = 1, 2, \dots, N$:

$$\begin{aligned}
\phi_{s,0}^n &= \phi_{3N+1-s,N+1}^n,\ \phi_{0,s}^n = \phi_{3N+s,N+1}^n,\ \phi_{N+1,s}^n = \phi_{2N+1-s,N+1}^n \text{ on } \Omega_h^1,\\
\phi_{0,N+s}^n &= \phi_{4N,N+s}^n \text{ on } \Omega_h^2,\\
\phi_{N+s,N}^n &= \phi_{N,N+1-s}^n,\ \phi_{N+s,2N+1}^n = \phi_{N,2N+s}^n \text{ on } \Omega_h^3,\\
\phi_{2N+s,N}^n &= \phi_{N+1-s,1}^n,\ \phi_{2N+s,2N+1}^n = \phi_{N+1-s,3N}^n \text{ on } \Omega_h^4,\\
\phi_{3N+s,N}^n &= \phi_{1,s}^n,\ \phi_{3N+s,2N+1}^n = \phi_{1,3N+1,s}^n,\ \phi_{4N+1,N+s}^n = \phi_{1,N+s}^n \text{ on } \Omega_h^5,\\
\phi_{0,2N+s}^n &= \phi_{4N+1-s,2N}^n,\ \phi_{s,3N+1}^n = \phi_{3N+1-s,2N}^n,\ \phi_{N+1,2N+s}^n = \phi_{N+s,2N}^n \text{ on } \Omega_h^6.
\end{aligned}$$

The numerical solution algorithm is based on the operator splitting scheme. First, we solve the diffusion term on Ω_h.

$$\frac{\phi_{ij}^* - \phi_{ij}^n}{\Delta t} = \frac{\phi_{i-1,j}^n + \phi_{i+1,j}^n - 4\phi_{ij}^n + \phi_{i,j-1}^n + \phi_{i,j+1}^n}{h^2},$$

which can be rewritten as

$$\phi_{ij}^* = \phi_{ij}^n + \Delta t \frac{\phi_{i-1,j}^n + \phi_{i+1,j}^n - 4\phi_{ij}^n + \phi_{i,j-1}^n + \phi_{i,j+1}^n}{h^2}.$$

Next, we calculate the nonlinear term. The computational solution ϕ_{ij}^{n+1} on Ω_h is calculated as follows:

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

Fig. 8 illustrates the time evolution of the computational solutions with random initial conditions on a cubic surface domain. The left and right columns are computational solutions for the AC equation on the unfolded and folded cubic surfaces, respectively.

2.8. Adaptive numerical methods

Adaptive numerical methods dynamically adjust parameters, such as step size or grid resolution, during computations to increase accuracy and efficiency. These methods allocate more computational resources in regions requiring finer resolution, such as areas with steep gradients, while reducing resources where the solution is smoother.

2.8.1. Temporally adaptive method

A temporally adaptive method adjusts the time step size dynamically during the numerical solution process based on the solution's behavior. By using error estimators or other criteria, this approach refines the time step in regions of rapid change and coarsens it where the solution varies slowly, which improves the efficiency and accuracy of the numerical schemes. Li et al. [71] proposed a polygonal mesh adaptation method for solving the AC equation, which uses a fully implicit method that combines the discontinuous Galerkin FEM for spatial discretization with the backward Euler scheme for temporal integration. Feng and Wu [72] developed a residual-based a posteriori error estimator for the FEM of the AC equation to compute its sharp interface limit, and the mean curvature flow. Willoughby [73] introduced a high-order time-adaptive algorithm to solve the AC equation, and demonstrated

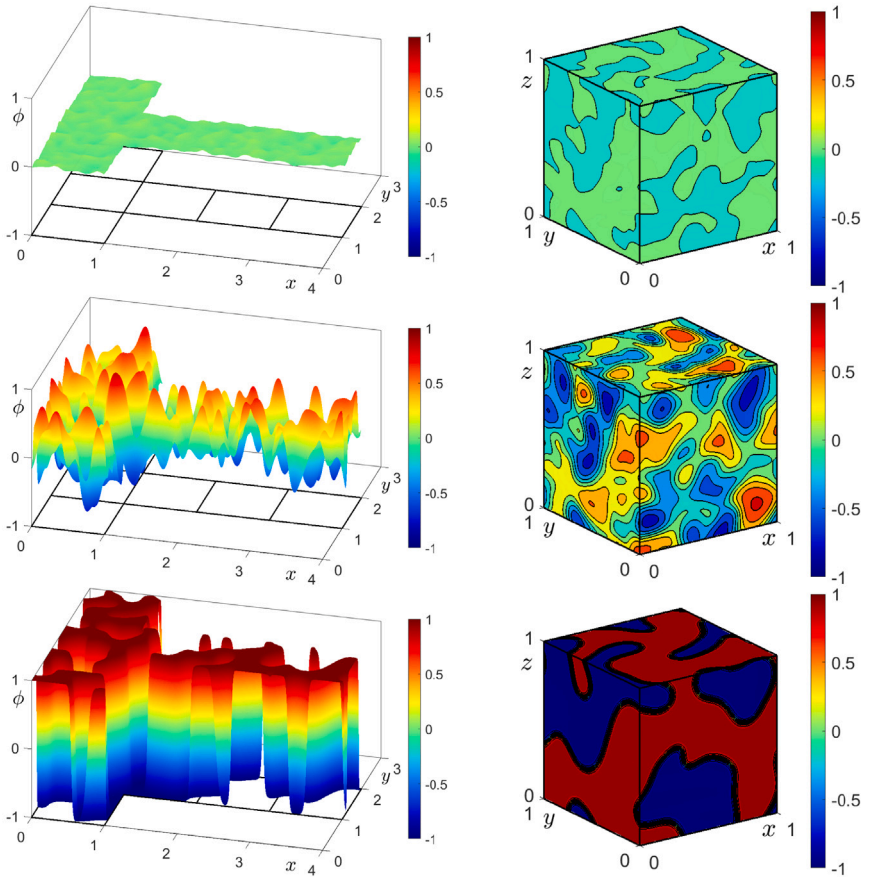

Fig. 8. The temporal evolution of the numerical solution for the AC equations on the cubic surface from top to bottom.

computational advantages over traditional techniques. Shah et al. [74] tackled a precise and efficient time-stepping method for numerically solving the AC equation, and highlighted the benefits of adaptive grids compared to uniform grids. The authors used the P1-conforming FEM and the diagonal implicit fractional-step θ method for spatial and temporal discretizations, respectively. Chen et al. [75] developed an SCR-based a posteriori error estimator to guide mesh refinement and coarsening, and proposed a time-space adaptive method for the AC equation.

2.8.2. *Spatially adaptive method*

A spatially adaptive method is a computational technique that adjusts the resolution of the computational grid or mesh based on the solution's local features. Regions requiring higher accuracy are refined with finer grids, and coarser grids are used in less critical areas, which improves efficiency without compromising the accuracy of the numerical methods. Jeong et al. [76] proposed an adaptive explicit hybrid scheme. They applied a time-adaptive method to the existing explicit hybrid finite difference scheme, see Fig. 9.

2.8.3. *Temporally and spatially adaptive method*

A temporally and spatially adaptive method is a numerical technique that adjusts the time step and mesh resolution dynamically during the simulation. The main purpose of this approach is to optimize accuracy and computational efficiency by refining the time and space discretization in regions where the solution changes rapidly and coarsening it where changes are slower.

2.9. *Other methods*

The finite volume element method (FVEM) is similar to FVM but is based on a finite element mesh for the computational domain and improves accuracy and stability by using finite element basis functions for flux approximation. In contrast to traditional FVM, the phase-field function ϕ is approximated using finite element basis functions as follows:

$$\phi_h = \sum_i \phi_i \psi_i,$$

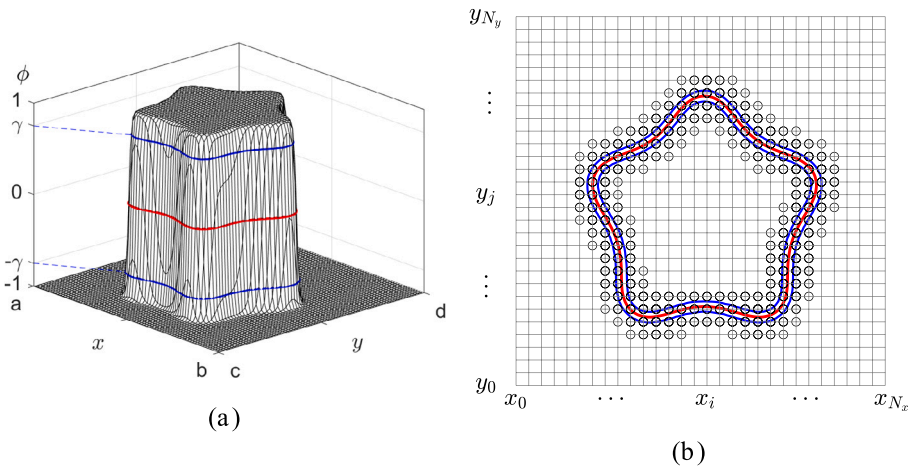

Fig. 9. Schematic illustrations of (a) ϕ and (b) $\tilde{\Omega}_h^n$.

where ψ_i are piecewise polynomial basis functions, and ϕ_i are the corresponding nodal values. Li and Wang [77] developed a modified finite volume element (MFVE) method for the AC model with a small perturbation parameter. The authors derived the optimal error estimations for the MFVE solutions and validated their results with numerical tests, which show that the MFVE algorithm is more efficient than the traditional FVE method.

An explicit numerical scheme, commonly employed in numerical simulations, is an algorithm that determines the future state of a system directly from its current state. It is simple and efficient for predicting system behavior over time. Since the explicit scheme uses the system's present conditions to extrapolate its state into the future state, it is well suited for a wide range of applications such as image processing [78] and fluid dynamics [79,80]. The explicit scheme is straightforward to implement; however, it imposes a time step restriction. In [43], the stability condition of the time-step size for the explicit FDM is analyzed for the AC equation with polynomial and logarithmic potential energy. Li and Zhang [81] proposed a high-efficiency explicit Runge–Kutta approach with energy dissipation for the AC equation. Poochinapan and Wongsaijai [82] presented a fourth-order compact difference scheme, which preserves the structure, for solving the AC equation with a stabilization term. Koohy et al. [32] solved the AC equations using stochastic differential equations and neural networks with the explicit Euler method. In the Appendix, we provide examples of fully explicit FDM MATLAB codes.

Wavelet analysis is a mathematical field that has found extensive applications in signal processing [83,84], image analysis [85,86], and numerical computation, among other areas. The term wavelets refers to small, rapidly decaying oscillations, in contrast to the larger sinusoidal waves used in Fourier analysis. Wavelets have been applied to solving partial differential equations. The examined data highlights the advantages of this method in capturing singularities, irregular structures, and transient phenomena. PDE wavelet solution algorithms are primarily based on Galerkin's method or the collocation approach. Angadi [87] presented numerical solutions of AC equation by using wavelet-based lifting schemes, introducing different wavelet filter coefficients. Wavelet-based numerical simulations of two-phase flows using the phase-field model are presented in [88,89].

2.10. Benchmark problems

Benchmark problems are standardized test cases used to evaluate and compare the performance of algorithms, methods, or models in a specific field. They serve as reference points for assessing accuracy, efficiency, and robustness, which allow researchers to validate and benchmark their solutions against widely accepted criteria or results.

2.10.1. Maximum principle-preserving property

The continuous AC equation satisfies the maximum bound principle, and this property was proved in [13]. If the initial values are bounded by 1, then the solutions of the AC equation remain bounded by 1. In [13], Evans et al. verified the preservation of the maximum boundary principle. Let us assume $\mathcal{E}^h(\boldsymbol{\phi}^n) \le \mathcal{E}^h(\boldsymbol{\phi}^{n-1})$, then for any $1 \le i \le N$ we have

$$\frac{h}{4\epsilon^2}((\phi_i^n)^2 - 1)^2 \le \mathcal{E}^h(\boldsymbol{\phi}^n) \le \mathcal{E}^h(\boldsymbol{\phi}^0). \tag{46}$$

Hence, we have

$$\|\boldsymbol{\phi}^n\|_\infty \le \sqrt{1 + 2\epsilon\sqrt{\mathcal{E}^h(\boldsymbol{\phi}^0)/h}}. \tag{47}$$

More details can be found in [90]. For instance, additional proofs regarding the maximum preserving principle for the AC equation can be found in [42,47,91,92].

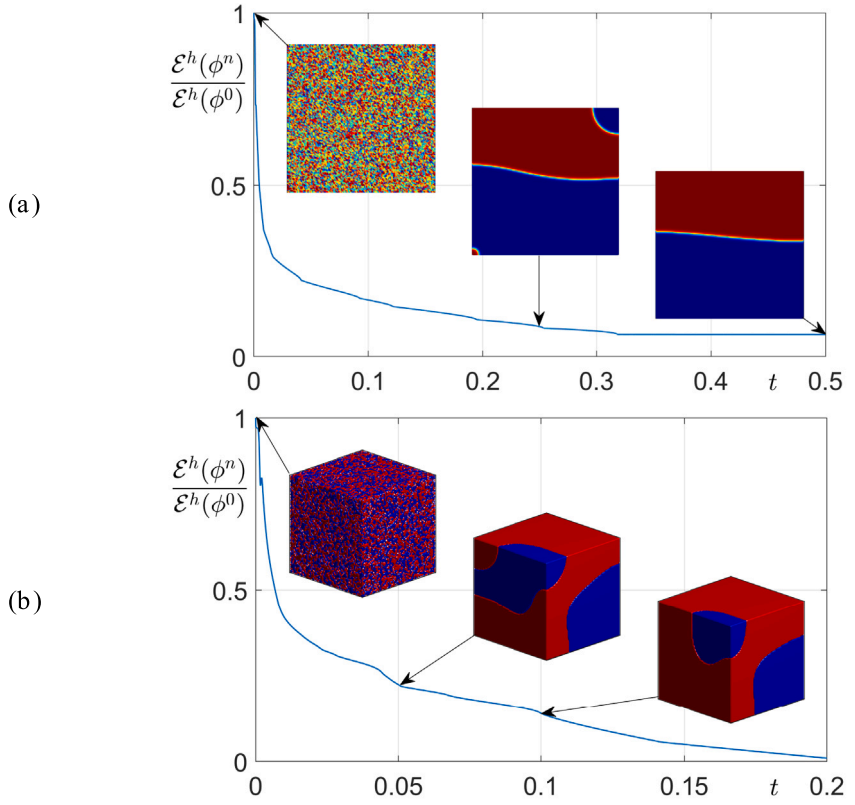

Fig. 10. (a) and (b) are the time evolutions of the dimensionless discrete total energy $\mathcal{E}^h(\boldsymbol{\phi}^n)/\mathcal{E}^h(\boldsymbol{\phi}^0)$ in two- and three-dimensional spaces, respectively.

2.10.2. Energy decrease

We now examine the progression of discrete total energy over time. The initial condition is $\phi(x, y, 0) = 0.1\text{rand}(x, y)$ on $\Omega = (-1, 1)\times(-1, 1)$ with 128×128 grid points. The parameters ϵ_4 and $\Delta t = 0.99\Delta t_{\max}$ are used, where $\Delta t_{\max} = \epsilon^2 h^2/(2h^2 + 4\epsilon^2)$ is the maximum stable time step size in two-dimensional space [43]. In Fig. 10(a), the time progression of the non-dimensional discrete total energy $\mathcal{E}^h(\boldsymbol{\phi}^n)/\mathcal{E}^h(\boldsymbol{\phi}^0)$ is shown until $t = 0.5$. We also consider a three-dimensional initial condition: $\phi(x, y, z, 0) = 0.1\text{rand}(x, y, z)$ on $\Omega = (0.5, 0.5)\times(-0.5, 0.5)\times(-0.5, 0.5)$ with $64 \times 64 \times 64$ grid points. The parameters used are the same as in the 2D test, and in the 3D space, $\Delta t_{\max} = \epsilon^2 h^2/(2h^2 + 6\epsilon^2)$. Fig. 10(b) is the temporal evolution of the discrete total energy until $t = 0.2$.

Wang et al. [91] presented a novel linear, energy-stable, and maximum principle-preserving computational scheme for approximating the AC equation using an innovative stabilization approach combined with energy factorization, and demonstrated its superior performance over conventional methods through numerical experiments. Further studies on the energy decreasing property of the AC equation can be found in [81,82].

2.10.3. Traveling wave solution

One exact solution of the AC equation is the traveling wave solution: $\phi(x, t) = 0.5 - 0.5\tanh[(x - st)/(2\sqrt{2}\epsilon_8)]$, where $s = 3/(\sqrt{2}\epsilon_8)$ [92]. Fig. 11 displays snapshots of the computational and exact solutions. Here, the initial condition and exact traveling wave solution at $t = 0.06$ are shown by the dashed and solid lines, respectively. Here, we used $\Delta t = 5 \times 10^{-6}$ and $h = 0.02$.

2.10.4. Manufactured solution

A manufactured solution for testing numerical methods involves creating an exact solution to a partial differential equation by first assuming a solution and then deriving the corresponding source terms. This approach allows for controlled testing of numerical methods and ensures accuracy and stability by comparing the numerical solution to the known exact solution. It is widely used to verify the correctness and performance of computational algorithms. In [93], the authors assumed a two-dimensional benchmark problem with the following two-dimensional AC model:

$$\begin{aligned}\frac{\partial\phi(x, y, t)}{\partial t} &= -\frac{F'(\phi(x, y, t))}{\epsilon^2} + \Delta\phi(x, y, t)\\ &= -\frac{\phi^3(x, y, t) - \phi(x, y, t)}{\epsilon^2} + \frac{\partial^2\phi(x, y, t)}{\partial x^2} + \frac{\partial^2\phi(x, y, t)}{\partial y^2},\end{aligned} \tag{48}$$

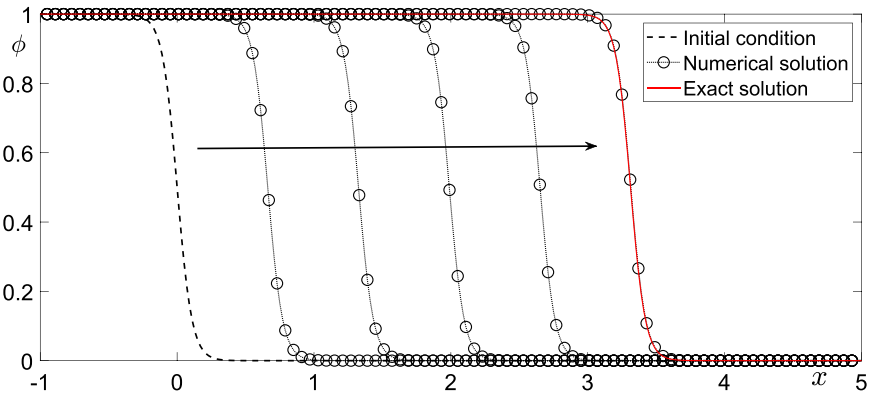

Fig. 11. Snapshots of the numerical results for the traveling wave solutions and the exact solution at the final time $t = 0.06$.

for $(x, y) \in \Omega = (2, 2\pi) \times (2, 2\pi),\ t > 0.$

To determine appropriate benchmark problems for the two-dimensional AC equation, we first linearize the nonlinear term $F'(\phi(x, y, t))$ through a Taylor expansion, which yields the approximation $F'(\phi(x, y, t)) \approx -\phi(x, y, t)$. This results in the following linearized AC equation:

$$\frac{\partial \phi(x, y, t)}{\partial t} = \frac{\phi(x, y, t)}{\epsilon^2} + \frac{\partial^2 \phi(x, y, t)}{\partial x^2} + \frac{\partial^2 \phi(x, y, t)}{\partial y^2}. \tag{49}$$

For natural numbers c_x and c_y, let us consider the following benchmark solution for the 2D AC equation (49):

$$\phi(x, y, t) = k(t) \cos(c_x x) \cos(c_y y), \tag{50}$$

where $k(t)$ is an amplitude. Substituting (50) into (49), we get:

$$\begin{aligned} \frac{\partial \phi(x, y, t)}{\partial t} &= k'(t) \cos(c_x x) \cos(c_y y) \\ &= \frac{k(t) \cos(c_x x) \cos(c_y y)}{\epsilon^2} - \left(c_x^2 + c_y^2\right) k(t) \cos(c_x x) \cos(c_y y). \end{aligned} \tag{51}$$

Dividing both sides of (51) by $\cos(c_x x) \cos(c_y y)$, we obtain

$$k'(t) = k(t) \left(\frac{1}{\epsilon^2} - \left(c_x^2 + c_y^2\right) \right).$$

Then, the solution of Eq. (51) is as follows:

$$k(t) = k(0) \exp\left[\left(\frac{1}{\epsilon^2} - \left(c_x^2 + c_y^2\right) \right) t \right]. \tag{52}$$

Let Eq. (50) serve as a benchmark solution for the two-dimensional modified AC equation, where $k(t)$ is defined by (52). The initial condition is expressed as $\phi(x, y, 0) = k(0) \cos(c_x x) \cos(c_y y)$. Finally, the modified AC equation with a source term is considered as follows:

$$\frac{\partial \phi(x, y, t)}{\partial t} = -\frac{F'(\phi(x, y, t))}{\epsilon^2} + \Delta \phi(x, y, t) + f(x, y, t), \tag{53}$$

for $(x, y) \in \Omega,\ t > 0,$

where

$$\begin{aligned} f(x, y, t) &= \frac{\partial \phi(x, y, t)}{\partial t} + \frac{F'(\phi(x, y, t))}{\epsilon^2} - \Delta \phi(x, y, t) \\ &= \left(k'(t) + \frac{1}{\epsilon^2} \left(k^3(t) \cos^2(c_x x) \cos^2(c_y y) - k(t) \right) + k(t) \left(c_x^2 + c_y^2\right) \right) \cos(c_x x) \cos(c_y y). \end{aligned}$$

2.10.5. Motion by mean curvature

Motion by mean curvature is one of the most popular benchmark problems [94]. The two-dimensional radius at time t is given as $R(t) = \sqrt{R_0^2 - 2t}$. In Fig. 12(a), the initial condition is defined as

$$\phi(x, y, 0) = \tanh\left(\frac{R_0 - \sqrt{x^2 + y^2}}{\sqrt{2}\epsilon} \right) \tag{54}$$

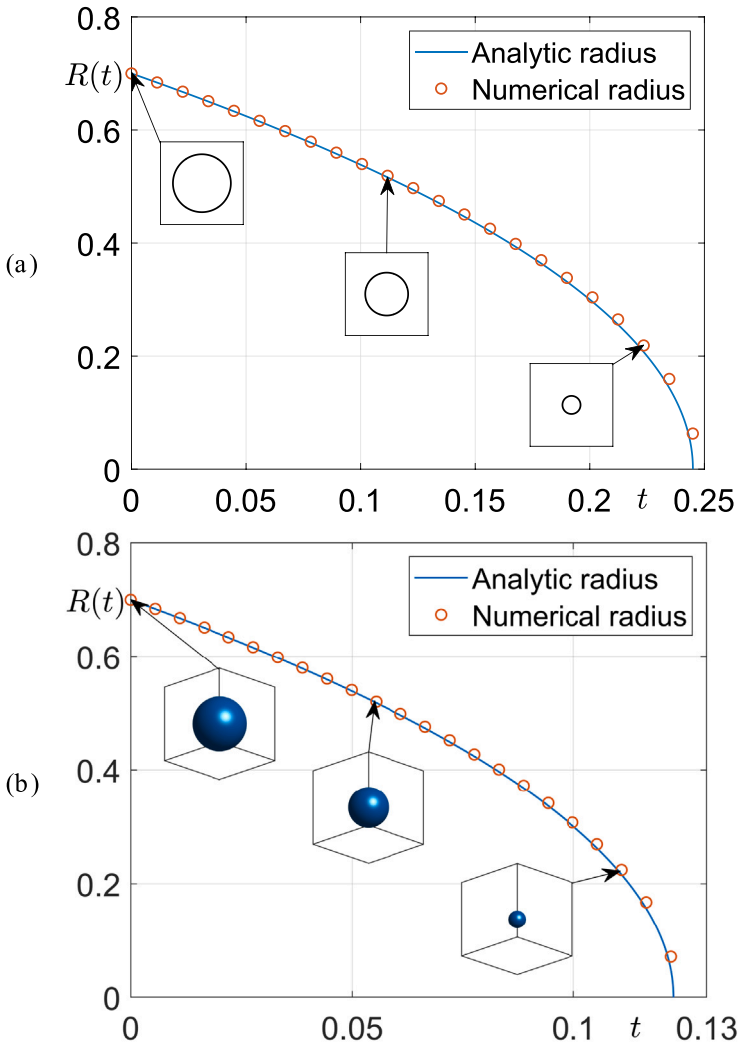

Fig. 12. Change of radius $R(t)$ of (a) a two-dimensional circle and (b) a three-dimensional sphere over time.

on $\Omega = (-1, 1) \times (-1, 1)$ with a 128×128 mesh. Here, $R_0 = 0.7$, $h = 1/64$, $\epsilon = \epsilon_8$, and $\Delta t = 0.99\Delta t_{\max}$ are used, where $\Delta t_{\max} = \frac{\epsilon^2 h^2}{2h^2 + 2d\epsilon^2}$ and d is the number of dimensions [43]. The numerical radius is calculated as the average of the distances from the coordinates of the zero-level contour to the center of the circle. We also observe motion by mean curvature in 3D spaces. The computational domain is $\Omega = (-1, 1) \times (-1, 1) \times (-1, 1)$ and

$$\phi(x, y, z, 0) = \tanh\left(\frac{R_0 - \sqrt{x^2 + y^2 + z^2}}{\sqrt{2}\epsilon}\right).$$

The parameters used are the same as those in the two-dimensional simulation. The radius of the sphere at time t is given by $R(t) = \sqrt{R_0^2 - 4t}$.

Fig. 12(a) displays $R(t)$ until $t = 0.245$. Fig. 12(b) shows the temporal evolution of $R(t)$, the radius of the sphere, until $t = 0.1225$. The spheres inside the small boxes represent the isosurface of the zero-level of $\phi(x, y, z, t)$.

Church et al. [94] presented four benchmark problems for the AC and CH equations, and validated numerical results using various computational methods and discretizations, with the objective of providing a reference for evaluating the accuracy and reliability of future software for phase field models in materials science.

3. Conclusions

In this review, we have provided various numerical methodologies used to solve the binary AC equation, which is central to modeling phase separation processes in materials science and beyond. By examining a range of methodologies, including the finite

difference, finite element, finite volume, Fourier spectral, invariant energy quadratization, and scalar auxiliary variable methods, we highlighted their respective advantages, limitations, and applicability across different scientific domains. The AC equation's capacity to model the dynamic evolution of interfaces makes it an essential tool for studying phase transitions, grain boundary motion, and pattern formation in complex systems. The advancement of numerical approaches is crucial for improving our ability to simulate and predict these phenomena accurately, and our review contributes to ongoing efforts to refine computational techniques for this important equation. In summary, the finite difference method is widely used due to its simplicity. Explicit schemes, such as the integrating factor Runge–Kutta method, provide high-order accuracy but have strict stability constraints. Implicit schemes improve stability but require higher computational costs. Finite element method is effective for complex geometries and adaptive mesh refinement. Discontinuous Galerkin methods improve accuracy, while residual-based error estimators improve precision. However, FEM is computationally more expensive than FDM. Fourier spectral methods achieve high accuracy for periodic domains. Semi-analytical Fourier spectral approaches allow large time steps but face challenges with general boundary conditions. Invariant energy quadratization and scalar auxiliary variable methods reformulate the AC equation to ensure energy stability and efficiently handle nonlinearities. SAV-based methods are energy stable but introduce additional computational overhead. Hybrid methods combine numerical techniques to improve efficiency. Monte Carlo simulations provide alternative approaches for high-dimensional problems. Adaptive methods dynamically adjust computational parameters. Temporally adaptive methods modify time-step sizes to optimize efficiency. Spatially adaptive methods refine grids where high resolution is needed. Each method's suitability depends on the complexity of problem, computational cost, and accuracy requirements. Through a better understanding of the available methods, future research can be directed toward improving accuracy, stability, and computational efficiency in simulating binary systems governed by the AC equation. In this review, we focused on the binary AC equation. In future research, we will extend our review to high-dimensional systems, such as vector-valued AC equations [95], multiphase CH equations [96–98].

CRediT authorship contribution statement

Hyun Geun Lee: Writing – review & editing, Writing – original draft, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Yibao Li:** Writing – original draft, Validation, Methodology, Investigation, Formal analysis. **Junxiang Yang:** Writing – original draft, Validation, Methodology, Investigation, Formal analysis. **Soobin Kwak:** Writing – original draft, Validation, Resources, Investigation. **Youngjin Hwang:** Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis. **Seokjun Ham:** Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Formal analysis. **Hyundong Kim:** Writing – original draft, Visualization, Validation, Software, Methodology, Funding acquisition, Formal analysis, Data curation. **Jyoti:** Writing – original draft, Validation, Resources, Investigation. **Yunjae Nam:** Writing – original draft, Validation, Resources, Investigation, Data curation. **Junseok Kim:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Use of AI tools declaration

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

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.

Acknowledgments

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 appreciate the reviewers' valuable comments and suggestions, which have greatly improved the quality of our manuscript.

Appendix A

Listing 1 Fully explicit FDM code

```
clear all;
Nx = 128; Ny = 128; Lx=1; Ly=1; h=Lx/Nx;
xx=linspace(0.5*h,Lx-0.5*h,Nx);
yy=linspace(0.5*h,Ly-0.5*h,Ny);
u = rand(Nx+2,Ny+2)-0.5; np=u;
dt = 0.1*h^2; T=10; eps2=h^2; Maxiter=200;
for iter = 1:Maxiter
```

```
u(1,:)=u(2,:); u(Nx+2,:)=u(Nx+1,:);
u(:,1)=u(:,2); u(:,Ny+2)=u(:,Ny+1);
for ii=2:Nx+1
    for jj=2:Ny+1
     np(ii,jj)=u(ii,jj)+dt*(u(ii,jj)-u(ii,jj)^3)/eps2 ...
              +dt*(u(ii-1,jj)+u(ii+1,jj)-4.0*u(ii,jj) ...
              +u(ii,jj-1)+u(ii,jj+1))/h^2;
    end
end
u=np;
if (mod(iter,10)==0)
    mesh(xx,yy,u(2:Nx+1,2:Ny+1))
    axis([0 Lx 0 Ly -1 1]); view(-47,39)
    pause(0.01)
end
end
```

Listing 2 Hybrid explicit code

```
clear all;
Nx = 128; Ny = 128; Lx=1; Ly=1; h=Lx/Nx;
xx=linspace(0.5*h,Lx-0.5*h,Nx);
yy=linspace(0.5*h,Ly-0.5*h,Ny);
u = rand(Nx+2,Ny+2)-0.5; np=u; np2=u;
dt = 0.1*h^2; T=10; eps2=h^2; Maxiter=200;
for iter = 1:Maxiter
u(1,:)=u(2,:); u(Nx+2,:)=u(Nx+1,:);
u(:,1)=u(:,2); u(:,Ny+2)=u(:,Ny+1);
for ii=2:Nx+1
    for jj=2:Ny+1
     np(ii,jj)=u(ii,jj)+dt*(u(ii-1,jj)+u(ii+1,jj) ...
               -4.0*u(ii,jj)+u(ii,jj-1)+u(ii,jj+1))/h^2;
     np2(ii,jj)=np(ii,jj)/sqrt((1-np(ii,jj)^2) ...
                *exp(-2*dt/eps2)+np(ii,jj)^2);
    end
end
u=np2;
if (mod(iter,10)==0)
    mesh(xx,yy,u(2:Nx+1,2:Ny+1))
    axis([0 Lx 0 Ly -1 1]); view(-47,39); pause(0.01)
end
end
```

Listing 3 DCT code

```
clear; Nx=128; Ny=128; Lx=1; Ly=1; hx=Lx/Nx; hy=Ly/Ny;
xx=linspace(0.5*hx,Lx-0.5*hx,Nx);
yy=linspace(0.5*hy,Ly-0.5*hy,Ny);
[X,Y]=ndgrid(xx,yy); h=xx(2)-xx(1); epsilon=h; Cahn=epsilon^2;
u=tanh((0.25-sqrt((X-0.5).^2+(Y-0.5).^2))/(sqrt(2)*epsilon));
p=pi*(0:Nx-1)/Lx; q=pi*(0:Ny-1)/Ly;
p2=p.^2; q2=q.^2; [pp2,qq2]=meshgrid(p2,q2);
dt=0.0001; T=0.03; Nt=round(T/dt); ns=Nt/20;
figure(1); clf; mesh(xx,yy,u');
axis([xx(1) xx(Nx) yy(1) yy(Ny) -1 1]); pause(0.01)
for iter=1:Nt
    u=u./sqrt(exp(-2*dt/Cahn)+u.^2*(1-exp(-2*dt/Cahn)));
    u_hat=dct2(u);
    u=idct2(exp(-dt*(pp2+qq2)).*u_hat);
    if (mod(iter,ns)==0)
     mesh(xx,yy,u'); axis([xx(1) xx(Nx) yy(1) yy(Ny) -1 1]);
     pause(0.01)
    end
end
```

Data availability

Data will be made available on 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, http://dx.doi.org/10.1016/0001-6160(79)90196-2.

[2] S. Lai, B. Jiang, Q. Xia, B. Xia, J. Kim, Y. Li, On the phase-field algorithm for distinguishing connected regions in digital model, Eng. Anal. Bound. Elem. 168 (2024) 105918, http://dx.doi.org/10.1016/j.enganabound.2024.105918.

[3] Q. Xia, J. Yang, Y. Li, On the conservative phase-field method with the N-component incompressible flows, Phys. Fluids 35 (2023) 012120, http://dx.doi.org/10.1063/5.0135490.

[4] Y. Li, K. Qin, Q. Xia, J. Kim, A second-order unconditionally stable method for the anisotropic dendritic crystal growth model with an orientation field, Appl. Numer. Math. 184 (2023) 512–526, http://dx.doi.org/10.1016/j.apnum.2022.11.006.

[5] Y. Li, Q. Yu, S. Ham, S. Kwak, C. Lee, J. Kim, A phase-field model without artificial curvature effect for the crystal growth simulation, Int. J. Heat Mass Transfer 203 (2023) 123847, http://dx.doi.org/10.1016/j.ijheatmasstransfer.2023.123847.

[6] Q. Xia, G. Sun, Q. Yu, J. Kim, Y. Li, Thermal-fluid topology optimization with unconditional energy stability and second-order accuracy via phase-field model, Commun. Nonlinear Sci. Numer. Simul. 116 (2023) 106782, http://dx.doi.org/10.1016/j.cnsns.2022.106782.

[7] Q. Xia, J. Zhu, Q. Yu, J. Kim, Y. Li, Triply periodic minimal surfaces based topology optimization for the hydrodynamic and convective heat transfer, Commun. Nonlinear Sci. Numer. Simul. 131 (2024) 107819, http://dx.doi.org/10.1016/j.cnsns.2024.107819.

[8] W. Xie, J. Feng, Q. Xia, J. Kim, Y. Li, Design of the shell-infill structures using a phase field-based topology optimization method, Comput. Methods Appl. Mech. Engrg. 429 (2024) 117138, http://dx.doi.org/10.1016/j.cma.2024.117138.

[9] W. Xie, Q. Xia, Q. Yu, Y. Li, An effective phase field method for topology optimization without the curvature effects, Comput. Math. Appl. 146 (2023) 200–221, http://dx.doi.org/10.1016/j.camwa.2023.06.037.

[10] M. Beneš, V. Chalupecký, K. Mikula, Geometrical image segmentation by the Allen–Cahn equation, Appl. Numer. Math. 51 (2004) 187–205, http://dx.doi.org/10.1016/j.apnum.2004.05.001.

[11] Z. Rong, L.L. Wang, X.C. Tai, Adaptive wavelet collocation methods for image segmentation using TV–Allen–Cahn type models, Adv. Comput. Math. 38 (2013).

[12] M. Beneš, K. Mikula, Simulation of anisotropic motion by mean curvature-comparison of phase field and sharp interface approaches, Acta Math. Univ. Comenian. 67 (1998) 17–42.

[13] L.C. Evans, H.M. Soner, P.E. Souganidis, Phase transitions and generalized motion by mean curvature, Comm. Pure Appl. Math. 45 (1992) 1097–1123, http://dx.doi.org/10.1002/cpa.3160450903.

[14] T. Ohtsuka, Motion of interfaces by an Allen–Cahn type equation with multiple-well potentials, Asymptot. Anal. 56 (2008) 87–123.

[15] J.S. Choi, T. Yamada, K. Izui, S. Nishiwaki, J. Yoo, Topology optimization using a reaction–diffusion equation, Comput. Methods Appl. Mech. Engrg. 200 (2011) 2407–2420, http://dx.doi.org/10.1016/j.cma.2011.04.013.

[16] H. Kim, S. Yoon, J. Wang, C. Lee, S. Kim, J. Park, J. Kim, Shape transformation using the modified Allen–Cahn equation, Appl. Math. Lett. 107 (2020) 106487, http://dx.doi.org/10.1016/j.aml.2020.106487.

[17] Y. Li, X. Song, S. Kwak, J. Kim, Weighted 3D volume reconstruction from series of slice data using a modified Allen–Cahn equation, Pattern Recognit. 132 (2022) 108914, http://dx.doi.org/10.1016/j.patcog.2022.108914.

[18] D. Wang, Y. Li, H. Jia, A two-grid finite element method for the Allen-Cahn equation with the logarithmic potential, Numer. Meth. Part Differ. Equ. 39 (2) (2023) 1251–1265, http://dx.doi.org/10.1002/num.22932.

[19] D. Lee, J.Y. Huh, D. Jeong, J.M. Shin, A. Yun, J. Kim, Physical, mathematical, And numerical derivations for the Cahn–Hilliard equations, Comput. Mater. Sci. 81 (2014) 216–225, http://dx.doi.org/10.1016/j.commatsci.2013.08.027.

[20] Z. Li, H. Wang, D. Yang, A space–time fractional phase-field model with tunable sharpness and decay behavior and its efficient numerical simulation, Numer. Meth. Part Differ. Equ. 347 (2017) 20–38, http://dx.doi.org/10.1016/j.jcp.2017.06.036.

[21] H. Liu, A. Cheng, H. Wang, J. Zhao, Time-fractional Allen–Cahn and Cahn–Hilliard phase-field models and their numerical investigation, Comput. Math. Appl. 76 (8) (2018) 1876–1892, http://dx.doi.org/10.1016/j.camwa.2018.07.036.

[22] J.W. Choi, H.G. Lee, D. Jeong, J. Kim, An unconditionally gradient stable numerical method for solving the Allen–Cahn equation, Phys. A 388 (9) (2009) 1791–1803, http://dx.doi.org/10.1016/j.physa.2009.01.026.

[23] D. Lee, Computing the area-minimizing surface by the Allen–Cahn equation with the fixed boundary, AIMS Math. 8 (2023) 23352–23371, http://dx.doi.org/10.3934/math.20231187.

[24] S. Wei, J. Yang, Interior interfaces with (or without) boundary intersection for an anisotropic Allen–Cahn equation, Commun. Pure Appl. Anal. 22 (7) (2023) 2040–2077, http://dx.doi.org/10.3934/cpaa.2023057.

[25] A.J. Ntsokongo, Asymptotic behavior of an Allen–Cahn type equation with temperature, Discret. Contin. Dyn. Syst.- S 16 (2023) 2452–2466, http://dx.doi.org/10.3934/dcdss.2023017.

[26] H. Alsayed, H. Fakih, A. Miranville, A. Wehbe, Optimal control of an Allen–Cahn model for tumor growth through supply of cytotoxic drugs, Integration 2 (2022) 3–13, http://dx.doi.org/10.3934/dcdss.2022003.

[27] M. Nara, Large time behavior of the solutions with spreading fronts in the Allen–Cahn equations on $\mathbb{R}^n$, Commun. Pure Appl. Anal. 21 (2022) 3605–3628, http://dx.doi.org/10.3934/cpaa.2022116.

[28] D. Li, C. Quan, J. Xu, Energy-dissipation for time-fractional phase-field equations, Commun. Pure Appl. Anal. 21 (2022) 3371–3387, http://dx.doi.org/10.3934/cpaa.2022104.

[29] N. Batangouna, A robust family of exponential attractors for a time semi-discretization of the Ginzburg–Landau equation, AIMS Math. 7 (2022) 1399–1415, http://dx.doi.org/10.3934/math.2022082.

[30] Y. Geng, Y. Teng, Z. Wang, L. Ju, A deep learning method for the dynamics of classic and conservative Allen–Cahn equations based on fully-discrete operators, J. Comput. Phys. 496 (2024) 112589, http://dx.doi.org/10.1016/j.jcp.2023.112589.

[31] B. Xia, R. Yu, X. Song, X. Zhang, J. Kim, An efficient data assimilation algorithm using the Allen–Cahn equation, Eng. Anal. Bound. Elem. 155 (2023) 511–517, http://dx.doi.org/10.1016/j.enganabound.2023.06.029.

[32] S. Koohy, G. Yao, K. Rubasinghe, Numerical solutions to low and high-dimensional Allen–Cahn equations using stochastic differential equations and neural networks, Part. Differ. Equ. Appl. Math. 7 (2023) 100499, http://dx.doi.org/10.1016/j.padiff.2023.100499.

[33] R. Mattey, S. Ghosh, A novel sequential method to train physics informed neural networks for Allen–Cahn and Cahn–Hilliard equations, Comput. Methods Appl. Mech. Engrg. 390 (2022) 114474, http://dx.doi.org/10.1016/j.cma.2021.114474.

[34] H. Xu, J. Chen, F. Ma, Adaptive deep learning approximation for Allen–Cahn equation, in: International Conference on Computational Science, Springer International Publishing, Cham, 2022, pp. 271–283, http://dx.doi.org/10.1007/978-3-031-08760-8_23.

[35] J. Wang, Z. Han, W. Jiang, J. Kim, A novel classification method combining phase-field and DNN, Pattern Recognit. 142 (2023) 109723, http://dx.doi.org/10.1016/j.patcog.2023.109723.
[36] J. Cheng, Q. Xia, J. Kim, Y. Li, An efficient linear and unconditionally stable numerical scheme for the phase field sintering model, Commun. Nonlinear Sci. Numer. Simul. 127 (2023) 107529, http://dx.doi.org/10.1016/j.cnsns.2023.107529.
[37] Y. Kim, G. Ryu, Y. Choi, Fast and accurate numerical solution of Allen–Cahn equation, Math. Probl. Eng. 2021 (1) (2021) 5263989, http://dx.doi.org/10.1155/2021/5263989.
[38] X. Teng, Z. Gao, H. Zhang, X. Qian, S. Song, Maximum-principle preserving, Delay-free parametric relaxation integrating factor Runge–Kutta schemes for the conservative nonlocal Allen–Cahn equation, Discret. Contin. Dyn. Syst.- B 30 (5) (2024) 1472–1498, http://dx.doi.org/10.3934/dcdsb.2024136.
[39] Z. Tan, J. Yang, J. Chen, J. Kim, An efficient time-dependent auxiliary variable approach for the three-phase conservative Allen–Cahn fluids, Appl. Math. Comput. 438 (2023) 127599, http://dx.doi.org/10.1016/j.amc.2022.127599.
[40] J. Wang, H. Xu, J. Yang, J. Kim, Fractal feature analysis based on phase transitions of the Allen–Cahn and Cahn–Hilliard equations, J. Comput. Sci. 72 (2023) 102114, http://dx.doi.org/10.1016/j.jocs.2023.102114.
[41] Y. Choi, J. Kim, Maximum principle preserving and unconditionally stable scheme for a conservative Allen–Cahn equation, Eng. Anal. Bound. Elem. 150 (2023) 111–119, http://dx.doi.org/10.1016/j.enganabound.2023.02.016.
[42] H. Zhang, J. Yan, X. Qian, S. Song, Numerical analysis and applications of explicit high order maximum principle preserving integrating factor Runge–Kutta schemes for Allen–Cahn equation, Appl. Numer. Math. 161 (2021) 372–390, http://dx.doi.org/10.1016/j.apnum.2020.11.022.
[43] S. Ham, J. Kim, Stability analysis for a maximum principle preserving explicit scheme of the Allen–Cahn equation, Math. Comput. Simulation 207 (2023) 453–465, http://dx.doi.org/10.1016/j.matcom.2023.01.016.
[44] J. Yang, Y. Li, C. Lee, Y. Choi, J. Kim, Fast evolution numerical method for the Allen–Cahn equation, J. King Saud Univ. Sci. 35 (1) (2023) 102430, http://dx.doi.org/10.1016/j.jksus.2022.102430.
[45] Y. Hwang, J. Yang, G. Lee, S. Ham, S. Kang, S. Kwak, et al., Fast and efficient numerical method for solving the Allen–Cahn equation on the cubic surface, Math. Comput. Simulation 215 (2024) 338–356, http://dx.doi.org/10.1016/j.matcom.2023.07.024.
[46] M. Olshanskii, X. Xu, V. Yushutin, A finite element method for Allen–Cahn equation on deforming surface, Comput. Math. Appl. 90 (2021) 148–158, http://dx.doi.org/10.1016/j.camwa.2021.03.018.
[47] X. Xiao, X. Feng, A second-order maximum bound principle preserving operator splitting method for the Allen–Cahn equation with applications in multi-phase systems, Math. Comput. Simulation 202 (2022) 36–58, http://dx.doi.org/10.1016/j.matcom.2022.05.024.
[48] X. Feng, A. Prohl, Numerical analysis of the Allen–Cahn equation and approximation for mean curvature flows, Numer. Math. 94 (2003) 33–65, http://dx.doi.org/10.1007/s00211-002-0413-1.
[49] X. Xiao, R. He, X. Feng, Unconditionally maximum principle preserving finite element schemes for the surface Allen–Cahn type equations, Numer. Methods Partial. Differ. Equ. 36 (2) (2020) 418–438, http://dx.doi.org/10.1002/num.22435.
[50] Q. Liu, K. Zhang, Z. Wang, J. Zhao, A two-level finite element method for the Allen–Cahn equation, Int. J. Comput. Math. 96 (1) (2019) 158–169, http://dx.doi.org/10.1080/00207160.2017.1420784.
[51] J. Shin, S.K. Park, J. Kim, A hybrid FEM for solving the Allen–Cahn equation, Appl. Math. Comput. 244 (2014) 606–612, http://dx.doi.org/10.1016/j.amc.2014.07.040.
[52] W. Zhao, Q. Guan, Numerical analysis of energy stable weak Galerkin schemes for the Cahn–Hilliard equation, Commun. Nonlinear Sci. Numer. Simul. 118 (2023) 106999, http://dx.doi.org/10.1016/j.cnsns.2022.106999.
[53] P. Strachota, M. Beneš, Error estimate of the finite volume scheme for the Allen–Cahn equation, Bit 58 (2018) 489–507, http://dx.doi.org/10.1007/s10543-017-0687-4.
[54] M. Cheng, A.D. Rutenberg, Maximally fast coarsening algorithms, Phys. Rev. E 72 (5) (2005) 055701, http://dx.doi.org/10.1103/PhysRevE.72.055701.
[55] R. Acar, Oscillation-free advection of interfaces with high order semi-Lagrangian schemes, Comput. & Fluids 38 (2009) 137–159, http://dx.doi.org/10.1016/j.compfluid.2008.01.006.
[56] J. Shen, X. Yang, Numerical approximations of Allen–Cahn and Cahn–Hilliard equations, Discret. Contin. Dyn. Syst. Ser. A 28 (2010) 1669–1691, http://dx.doi.org/10.3934/dcds.2010.28.1669.
[57] X. Song, Q. Xia, J. Kim, Y. Li, Unconditional energy stable data assimilation scheme for Navier–Stokes–Cahn–Hilliard equations with local discretized observed data, Comput. Math. Appl. 164 (2024) 21–33, http://dx.doi.org/10.1016/j.camwa.2024.03.018.
[58] H.G. Lee, J.Y. Lee, A semi-analytical Fourier spectral method for the Allen–Cahn equation, Comput. Math. Appl. 68 (2014) 174–184, http://dx.doi.org/10.1016/j.camwa.2014.05.015.
[59] X. Yang, Linear, first, second order, unconditionally energy stable numerical schemes for the phase field model of homopolymer blends, J. Comput. Phys. 327 (2016) 294–316, http://dx.doi.org/10.1016/j.jcp.2016.09.029.
[60] J. Shen, J. Xu, J. Yang, The scalar auxiliary variable (SAV) approach for gradient flows, J. Comput. Phys. 353 (2018) 407–416, http://dx.doi.org/10.1016/j.jcp.2017.10.021.
[61] B. Li, J. Shen, Z. Yang, Y. Zhang, Efficient energy stable schemes for incompressible flows with variable density, J. Comput. Phys. 517 (2024) 113365, http://dx.doi.org/10.1016/j.jcp.2024.113365.
[62] K. Sun, B. Wei, H. Zhang, J. Yang, Energy-stable auxiliary variable viscosity splitting (AVVS) method for the incompressible Navier–Stokes equation and turbidity current system, Comput. Methods Appl. Mech. Engrg. 431 (2024) 117295, http://dx.doi.org/10.1016/j.cma.2024.117295.
[63] Z. Zheng, G. Zou, B. Wang, W. Zhao, A fully-decoupled discontinuous Galerkin method for the nematic liquid crystal flows with SAV approach, J. Comput. Appl. Math. 429 (2023) 115207, http://dx.doi.org/10.1016/j.cam.2023.115207.
[64] Y. Zhang, J. Shen, A generalized SAV approach with relaxation for dissipative systems, J. Comput. Phys. 464 (2022) 111311, http://dx.doi.org/10.1016/j.jcp.2022.111311.
[65] Z. Liu, X. Li, A highly efficient and accurate exponential semi-implicit scalar auxiliary variable (ESI-SAV) approach for dissipative systems, J. Comput. Phys. 447 (2021) 110703, http://dx.doi.org/10.1016/j.jcp.2021.110703.
[66] Y. Hwang, I. Kim, S. Kwak, S. Ham, S. Kim, J. Kim, Unconditionally stable Monte Carlo simulation for solving the multi-dimensional Allen–Cahn equation, Electron. Res. Arch. 31 (2023) 5104–5123, http://dx.doi.org/10.3934/era.2023261.
[67] Y. Hwang, S. Ham, C. Lee, G. Lee, S. Kang, J. Kim, A simple and efficient numerical method for the Allen–Cahn equation on effective symmetric triangular meshes, Electron. Res. Arch. 31 (2023) 4557–4578, http://dx.doi.org/10.3934/era.2023233.
[68] V. Mohammadi, D. Mirzaei, M. Dehghan, Numerical simulation and error estimation of the time-dependent Allen–Cahn equation on surfaces with radial basis functions, J. Sci. Comput. 79 (2019) 493–516, http://dx.doi.org/10.1007/s10915-018-0859-7.
[69] Y. Choi, D. Jeong, S. Lee, M. Yoo, J. Kim, Motion by mean curvature of curves on surfaces using the Allen–Cahn equation, Internat. J. Engrg. Sci. 97 (2015) 126–132, http://dx.doi.org/10.1016/j.ijengsci.2015.10.002.
[70] Y. Choi, Y. Li, C. Lee, H. Kim, J. Kim, Explicit hybrid numerical method for the Allen–Cahn type equations on curved surfaces, Numer. Math. Theory Methods Appl. 14 (2021) 797–810, http://dx.doi.org/10.4208/nmtma.OA-2020-0155.
[71] R. Li, Y. Gao, Z. Chxen, Adaptive discontinuous Galerkin finite element methods for the Allen–Cahn equation on polygonal meshes, Numer. Algor. 95 (2024) 1981–2014, http://dx.doi.org/10.1007/s11075-023-01635-5.

[72] X. Feng, H. 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, http://dx.doi.org/10.1007/s10915-004-4610-1.
[73] M.R. Willoughby, High-order time-adaptive numerical methods for the Allen–Cahn and Cahn–Hilliard equations Doctoral dissertation, University of British Columbia, 2011.
[74] A. Shah, M. Sabir, M. Qasim, P. Bastian, Efficient numerical scheme for solving the Allen–Cahn equation, Numer. Methods Partial. Differ. Equ. 34 (2018) 1820–1833, http://dx.doi.org/10.1002/num.22255.
[75] 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 (2019) 204–223, http://dx.doi.org/10.1016/j.camwa.2019.02.022.
[76] D. Jeong, Y. Li, Y. Choi, C. Lee, J. Yang, J. Kim, A practical adaptive grid method for the Allen–Cahn equation, Phys. A 573 (2021) 125975, http://dx.doi.org/10.1016/j.physa.2021.125975.
[77] H. Li, D. Wang, A modified finite volume element method for solving the phase field Allen–Cahn model, Appl. Math. Lett. 127 (2022) 107860, http://dx.doi.org/10.1016/j.aml.2021.107860.
[78] C. Lee, S. Kim, S. Kwak, Y. Hwang, S. Ham, S. Kang, J. Kim, Semi-automatic fingerprint image restoration algorithm using a partial differential equation, AIMS Math. 8 (11) (2023) 27528–27541, http://dx.doi.org/10.3934/math.20231408.
[79] K. Nagata, N. Ikegaya, J. Tanimoto, Consideration of artificial compressibility for explicit computational fluid dynamics simulation, J. Comput. Phys. 443 (2021) 110524, http://dx.doi.org/10.1016/j.jcp.2021.110524.
[80] H. Yan, G. Zhang, H. Rao, H. Song, Z. Sun, An explicit velocity correction-based immersed boundary-hybrid lattice Boltzmann flux solver for fluid–structure interaction with large solid deformation, Ocean Eng. 270 (2023) 113655, http://dx.doi.org/10.1016/j.oceaneng.2023.113655.
[81] K. Li, H. Zhang, Explicit Runge–Kutta scheme with high efficiency and energy dissipation for the Allen–Cahn equation, J. Phys. Conf. Ser. 1682 (1) (2020) 012031, http://dx.doi.org/10.1088/1742-6596/1682/1/012031.
[82] K. Poochinapan, B. Wongsaijai, Numerical analysis for solving Allen–Cahn equation in 1D and 2D based on higher-order compact structure-preserving difference scheme, Appl. Math. Comput. 434 (2022) 127374, http://dx.doi.org/10.1016/j.amc.2022.127374.
[83] S. Galli, H. Koga, N. Kodama, Advanced signal processing for PLCs: Wavelet-OFDM, in: 2008 IEEE International Symposium on Power Line Communications and Its Applications, 2008, pp. 187–192, http://dx.doi.org/10.1109/ISPLC.2008.4510421.
[84] J. Rafiee, M.A. Rafiee, N. Prause, M.P. Schoen, Wavelet basis functions in biomedical signal processing, Expert Syst. Appl. 38 (5) (2011) 6190–6201, http://dx.doi.org/10.1016/j.eswa.2010.11.050.
[85] J.P. Antoine, P. Carrette, R. Murenzi, B. Piette, Image analysis with two-dimensional continuous wavelet transform, Signal Process. 31 (1993) 241–272, http://dx.doi.org/10.1016/0165-1684(93)90085-O.
[86] J.A. Dobrosotskaya, A.L. Bertozzi, A wavelet-Laplace variational technique for image deconvolution and inpainting, IEEE Trans. Image Process. 17 (5) (2008) 657–663, http://dx.doi.org/10.1109/TIP.2008.919367.
[87] L.M. Angadi, Numerical solution of Cahn–Allen equations by wavelet based lifting schemes, J. Sci. Res. 16 (1) (2024) 31–40, http://dx.doi.org/10.3329/jsr.v16i1.63085.
[88] M.J. Ahammad, J.M. Alam, M.A. Rahman, S.D. Butt, Numerical simulation of two-phase flow in porous media using a wavelet based phase-field method, Chem. Eng. Sci. 173 (2017) 230–241, http://dx.doi.org/10.1016/j.ces.2017.07.014.
[89] J.M. Alam, A wavelet based numerical simulation technique for two-phase flows using the phase field method, Comput. & Fluids 146 (2017) 143–153, http://dx.doi.org/10.1016/j.compfluid.2017.01.015.
[90] J. Kim, Phase-field models for multi-component fluid flows, Commun. Comput. Phys. 12 (3) (2012) 613–661, http://dx.doi.org/10.4208/cicp.301110.040811a.
[91] X. Wang, J. Kou, H. Gao, Linear energy stable and maximum principle preserving semi-implicit scheme for Allen–Cahn equation with double well potential, Commun. Nonlinear Sci. Numer. Simul. 98 (2021) 105766, http://dx.doi.org/10.1016/j.cnsns.2021.105766.
[92] H. Zhang, X. Qian, S. Song, Third-order accurate, large time-stepping and maximum-principle-preserving schemes for the Allen–Cahn equation, Numer. Algorithms 95 (3) (2024) 1213–1250, http://dx.doi.org/10.1007/s11075-023-01606-w.
[93] Y. Hwang, C. Lee, S. Kwak, Y. Choi, S. Ham, S. Kang, J. Yang, J. Kim, Benchmark problems for the numerical schemes of the phase-field equations, Discrete Dyn. Nat. Soc. 2022 (1) (2022) 2751592, http://dx.doi.org/10.1155/2022/2751592.
[94] J.M. Church, Z. Guo, P.K. Jimack, A. Madzvamuse, K. Promislow, B. Wetton, S.M. Wise, F. Yang, High accuracy benchmark problems for Allen–Cahn and Cahn–Hilliard dynamics, Commun. Comput. Phys. 26 (4) (2019) 947–972, http://dx.doi.org/10.4208/cicp.OA-2019-0006.
[95] S. Ham, S. Kwak, C. Lee, G. Lee, J. Kim, A second-order time-accurate unconditionally stable method for a gradient flow for the Modica–Mortola functional, J. Sci. Comput. 95 (2) (2023) 63, http://dx.doi.org/10.1007/s10915-023-02198-2.
[96] J. Yang, J. Wang, Z. Tan, J. Kim, Efficient IMEX and consistently energy-stable methods of diffuse-interface models for incompressible three-component flows, Comput. Phys. Comm. 282 (2023) 108558, http://dx.doi.org/10.1016/j.cpc.2022.108558.
[97] J. Yang, Y. Li, J. Kim, Modified multi-phase diffuse-interface model for compound droplets in contact with solid, J. Comput. Phys. 491 (2023) 112345, http://dx.doi.org/10.1016/j.jcp.2023.112345.
[98] J. Yang, Z. Tan, J. Wang, J. Kim, Modified diffuse interface fluid model and its consistent energy-stable computation in arbitrary domains, J. Comput. Phys. 488 (2023) 112216, http://dx.doi.org/10.1016/j.jcp.2023.112216.