Journal of Computational and Applied Mathematics 453 (2025) 116159

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

A fourth-order finite difference method for the Allen–Cahn equation

Seokjun Ham [a], Seungyoon Kang [a], Youngjin Hwang [a], Gyeonggyu Lee [a], Soobin Kwak [a], Jyoti [b], Junseok Kim [a,*]

[a] *Department of Mathematics, Korea University, Seoul, 02841, Republic of Korea*
[b] *The Institute of Basic Science, Korea University, Seoul, 02841, Republic of Korea*

ARTICLE INFO

Keywords:
Allen–Cahn equation
Fourth-order accurate
Finite difference method
Penta-diagonal matrix

ABSTRACT

In this study, we present a spatially fourth-order accurate hybrid numerical scheme for the Allen–Cahn (AC) equation in two-dimensional (2D) and three-dimensional (3D) spaces. The proposed hybrid numerical method splits the AC model into nonlinear and linear components using the operator splitting technique. The nonlinear component is solved by using an analytic solution. In 3D space, the linear diffusion term is solved by splitting it into the x-, y-, and z-directional single spatial variable diffusion equations. The fully implicit scheme for temporal difference and the spatially fourth-order finite difference discretization are applied. The system of discrete equations becomes a penta-diagonal matrix that can be directly solved without any iterative techniques. Stability analysis and various computational experiments are performed to verify the numerical convergence and stability of the proposed method in 2D and 3D spaces. Furthermore, we compared the convergence rate, error, and CPU time between the proposed fourth-order and standard second-order schemes.

1. Introduction

We present a finite difference method (FDM) for the Allen–Cahn (AC) equation [1]:

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

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

where $\Omega\subset\mathbb{R}^d$ for $d=2,3$; $\phi(\mathbf{x},t)$ is an order parameter which takes the values between two immiscible phases -1 and 1; $\epsilon>0$; and $F(\phi)=0.25(\phi^2-1)^2$. The AC equation can be obtained from the L^2 gradient flow of the following total free energy functional:

$$\mathcal{E}(\phi) = \int_\Omega \left(\frac{F(\phi)}{\epsilon^2} + \frac{1}{2}|\nabla\phi|^2\right) d\mathbf{x}. \tag{3}$$

Also, differentiating the total free energy functional Eq. (3) with respect to time t, we can derive as follows:

$$\begin{aligned}\frac{d}{dt}\mathcal{E}(\phi) &= \int_\Omega \left(\frac{F'(\phi)}{\epsilon^2}\phi_t + \nabla\phi\cdot\nabla\phi_t\right) d\mathbf{x}\\ &= \int_\Omega \left(\frac{F'(\phi)}{\epsilon^2} - \Delta\phi\right)\phi_t d\mathbf{x} = -\int_\Omega |\phi_t|^2\, d\mathbf{x} \le 0,\end{aligned} \tag{4}$$

using the integration by part and the homogeneous Neumann boundary condition.

[*] Corresponding author.
E-mail address: cfdkim@korea.ac.kr (J. Kim).

https://doi.org/10.1016/j.cam.2024.116159
Received 1 November 2023; Received in revised form 31 May 2024
Available online 26 July 2024

The AC equation is a model of the physical phenomenon for anti-phase domain coarsening in a binary alloy [1]. Because the AC equation can be applied to various physical study, many kind of numerical approach es to solve the AC equation are developed such as finite difference [2–4], finite element [5–7], and spectral [8] methods. A large number of studies demonstrating the AC equation contains a stability and error analysis [9], and some such as [10] focus on proving the stability.

Wang et al. [2] proposed a linear energy stable and maximum principle preserving semi-implicit scheme for the AC equation. The authors considered the AC equation with double well potential adopting a stabilized energy factorization. Xiao and Feng [11] presented a second-order operator splitting method (OSM) for the AC equation. The proposed method reduces high storage requirements and complexity preserves maximum bounded principle. Numerical tests such as multi-phase separations and crystal growth in two and three dimensions are given. He et al. [3] proposed a spatial fourth-order scheme by using a fourth-order quasi-compact scheme and Strang's splitting for the fractional AC equation. Because fractional AC equation is a popular topic, various high-order schemes for the reaction–diffusion equation have been presented. There also have been studies on stability and convergence analysis works for the OSM to various partial differential equations and phase field models. Zhai et al. [12] proposed OSM combined with pseudo-spectral method for space fractional nonlinear Schrödinger equation. Zhai et al. [13] presented the semi-implicit spectral deferred correction method with the OSM to solve the fractional Gray–Scott model. Almushaira et al. [14] proposed a high-order multi-dimensional space-fractional method for reaction–diffusion equations and provided stability, accuracy, efficiency, and some applications including AC model. Wang [15] proposed a high-order compact method for time-fractional reaction–diffusion equations with variable coefficients. The method is developed under the nonhomogeneous Neumann boundary condition. Authors proved unconditional stability and second-order convergence. In [16], a second-order Strang splitting scheme with exponential integrating factor was introduced for the AC equation with logarithmic Flory–Huggins potential. The maximum principle, energy stability, and convergence were proved for the proposed scheme. The compact scheme is a kind of high order numerical solver [17,18]. In particular, there are various studies that used the compact scheme to solve the AC equation [18–21]. However, the compact scheme is hard to calculate boundary points. Furthermore, it requires an additional approximation scheme which makes the high numerical computation and computational cost. Zhai et al. [21] proposed the compact difference scheme based on the Crank–Nicolson/Adams–Bashforth scheme combined with the Douglas–Gunn ADI method in the three-dimensional (3D) space to solve the AC equation. They presented the numerical simulation using the three boundary conditions which are Dirichlet, Neumann, and periodic boundary conditions. To maintain the accuracy and simplicity of the system, they consider the fourth-order accurate Padé approximations. Zhang et al. [22] proposed third-order explicit structure-preserving schemes for two modified conservative AC equations(with RSLM and with BBLM). Mass conservation, convergence and stability is proven in the study and they demonstrated various numerical experiments to demonstrate the advantages of the proposed method. Rizwan et al. [23] presented the high-order compact scheme to solve the AC equation coupled with the incompressible Navier–Stokes equations to simulate the two phase incompressible flow. Lee et al. [24] studied a high-order conservative AC equation that is unconditionally energy stable with nonlocal Lagrange multiplier. The unconditional stability is analytically shown and numerical experiments illustrates the accuracy and energy stability of the proposed scheme. The fourth-order compact FDM was studied to solve the AC equation in the 3D space by Long et al. [20]. To overcome the time step restriction, they used linearly stabilized splitting scheme and achieved high-order accuracy and stability of the system. Also, they took a linear multigrid solver to improve the efficiency and stability. In recent years, the fourth-order numerical algorithm to solve the AC equation is solved with additional stabilized term [25]. They consider the Crank–Nicolson/Adams–Bashforth method to deal with the non-linear term. In addition, in the iteration, they consider that the three level subiterations which are consistent with the Crank–Nicolson/Adams–Bashforth method to preserve the stability and accuracy. Bo et al. [26] solved the AC equation in the two-dimensional (2D) space using fully discrete compact difference scheme with second-order accuracy in time and fourth-order in space. They investigated the energy stability of the scheme under the condition of discrete maximum principle with reasonable spacial mesh and time steps.

In this study, we consider the fourth-order numerical scheme to solve the AC equation in 2D and 3D space. We use the OSM and separate the AC equation into a nonlinear equation and a linear equation. The nonlinear term is solved by using the closed-form solution, based on explicit Euler's method for the temporal derivative. The linear term is solved by splitting it into the x-, y- and z-directions, considering the implicit scheme for temporal difference. We note that using OSM has the advantage of allowing higher-dimensional problems to be solved with a simple, non-iterative direct solver. However, it is difficult to prove the original energy stability of the governing equation with the OSM. Stability of the proposed method is proven based on the von Neumann stability analysis. We conduct numerical experiments such as energy decrease and motion by mean curvature in 2D and 3D spaces. The convergence rate, error and cpu time is compared between the proposed method and a second-order scheme.

The outline of this paper is organized as follows. In Section 2, we present the fourth-order accurate hybrid method on grids in 2D and 3D spaces. We conduct computational tests and stability analysis in Section 3. Finally, our conclusions and future works are presented in Section 4.

2. Numerical method

Now, we present computational solution algorithms for the AC model, using the hybrid method that combines OSM and the implicit Euler method. We demonstrate the discretization theory for the 2D domain and extend from the 2D discretized algorithm to the 3D domain in Sections 2.1 and 2.2, respectively.

2.1. Two-dimensional space

We consider the AC equation on the 2D domain $\Omega = (L_x, R_x) \times (L_y, R_y)$. We discretize the domain with the cell-centered grid that consists x_i and y_j are grid points. Let $\Omega_h = \left\{(x_i,\ y_j)|\ x_i = L_x + h(i-0.5), y_j = L_y + h(j-0.5),\ \text{for } i = 1,\dots,N_x,\ j = 1,\dots,N_y\right\}$ be the discrete domain. Here, the spatial mesh size is equal in all directions: $h = (L_x - R_x)/N_x = (L_y - R_y)/N_y$. We set the value of ϵ to depend on the number of grids [27]. Let $\Delta t = T/N_t$ be time step. We denote $\phi(x_i, y_j, n\Delta t)$ by the numerical approximation ϕ^n_{ij}.

In the 2D domain Ω, the AC model (1) can be written as

$$\frac{\partial \phi(x,y,t)}{\partial t} = -\frac{F'(\phi(x,y,t))}{\epsilon^2} + \phi_{xx}(x,y,t) + \phi_{yy}(x,y,t),\ (x,y) \in \Omega,\ t > 0. \tag{5}$$

For the AC Eq. (5), we use the following temporal operator splitting discretization [28].

$$\phi(x,y,(n+1)\Delta t) = (\mathcal{L}^{\Delta t}_y \circ \mathcal{L}^{\Delta t}_x \circ \mathcal{N}^{\Delta t})\phi(x,y,n\Delta t), \tag{6}$$

which has three-step operators.

First, we obtain $\Phi_1(x,y)$ from $\phi(x,y,n\Delta t)$ and the nonlinear operator $\mathcal{N}^{\Delta t}$ as

$$\Phi_1(x,y) = \mathcal{N}^{\Delta t}\phi(x,y,n\Delta t). \tag{7}$$

Here, $\Phi_1(x,y)$ is a solution of the following nonlinear equation at time $\tau = \Delta t$:

$$\frac{\partial p(x,y,\tau)}{\partial \tau} = -\frac{F'(p(x,y,\tau))}{\epsilon^2} \tag{8}$$

with the initial condition $p(x,y,0) = \phi(x,y,n\Delta t)$, i.e., $\Phi_1(x,y) = p(x,y,\Delta t)$.

Second, $\Phi_2(x,y)$ is obtained from $\Phi_1(x,y)$ and x-direction linear operator $\mathcal{L}^{\Delta t}_x$ as

$$\Phi_2(x,y) = \mathcal{L}^{\Delta t}_x \Phi_1(x,y). \tag{9}$$

Here, $\Phi_2(x,y)$ is a solution of the linear equation at time $\tau = \Delta t$:

$$\frac{\partial q(x,y,\tau)}{\partial \tau} = q_{xx}(x,y,\tau) \tag{10}$$

with the initial condition $q(x,y,0) = \Phi_1(x,y)$, i.e., $\Phi_2(x,y) = q(x,y,\Delta t)$.

Finally, we get $\phi(x,y,(n+1)\Delta t)$ from Φ_2 and y-direction linear operator $\mathcal{L}^{\Delta t}_y$ as

$$\phi(x,y,(n+1)\Delta t) = \mathcal{L}^{\Delta t}_y \Phi_2(x,y). \tag{11}$$

Here, $\phi(x,y,(n+1)\Delta t)$ is a solution of the linear equation at time $\tau = \Delta t$:

$$\frac{\partial r(x,y,\tau)}{\partial \tau} = r_{yy}(x,y,\tau) \tag{12}$$

with the initial condition $r(x,y,0) = \Phi_2(x,y)$, i.e., $\phi(x,y,(n+1)\Delta t) = r(x,y,\Delta t)$. Eqs. (10) and (12) are solved using the fully implicit Euler method. The homogeneous Neumann boundary condition is applied to all three operators.

Now we describe each step with numerical schemes. First, we solve Eq. (8)

$$\frac{\partial p(x,y,\tau)}{\partial \tau} = -\frac{F'(p(x,y,\tau))}{\epsilon^2}. \tag{13}$$

We obtain the solution, for any time $\tau > 0$, using the separation of variables [29,30] as follows:

$$p(x,y,\tau) = \frac{p(x,y,0)}{\sqrt{e^{\frac{-2\tau}{\epsilon^2}} + (p(x,y,0))^2\left(1 - e^{\frac{-2\tau}{\epsilon^2}}\right)}}. \tag{14}$$

Here $p(x,y,0) := \phi(x,y,n\Delta t)$, then we can get $\Phi_1(x,y) = p(x,y,\Delta t)$.

Next, to solve the linear terms, we discretize Eqs. (10) and (12) using fourth-order discrete Laplacian with implicit Euler's method as follows:

$$\frac{q^{n+1}_{ij} - q^n_{ij}}{\Delta \tau} = \Delta^x_h q^{n+1}_{ij}, \tag{15}$$

$$\frac{r^{n+1}_{ij} - r^n_{ij}}{\Delta \tau} = \Delta^y_h r^{n+1}_{ij}, \tag{16}$$

where

$$\Delta^x_h q_{ij} = \frac{-q_{i-2,j} + 16q_{i-1,j} - 30q_{ij} + 16q_{i+1,j} - q_{i+2,j}}{12h^2}, \tag{17}$$

$$\Delta^y_h r_{ij} = \frac{-r_{i,j-2} + 16r_{i,j-1} - 30r_{ij} + 16r_{i,j+1} - r_{i,j+2}}{12h^2}. \tag{18}$$

Here, q_{ij}^n and r_{ij}^n denote $q(x_i, y_j, n\Delta\tau)$ and $r(x_i, y_j, n\Delta\tau)$, respectively. The fourth-order finite difference formula can be derived by the Taylor expansion. Rewriting Eqs. (15) and (16) in the following ways:

$$\alpha q_{i-2,j}^{n+1} - 16\alpha q_{i-1,j}^{n+1} + (1+30\alpha)\, q_{ij}^{n+1} - 16\alpha q_{i+1,j}^{n+1} + \alpha q_{i+2,j}^{n+1} = q_{ij}^n, \tag{19}$$

$$\alpha r_{i,j-2}^{n+1} - 16\alpha r_{i,j-1}^{n+1} + (1+30\alpha)\, r_{ij}^{n+1} - 16\alpha r_{i,j+1}^{n+1} + \alpha r_{i,j+2}^{n+1} = r_{ij}^n, \tag{20}$$

where $\alpha = \Delta\tau/(12h^2)$. We can derive the following matrix formula using Eqs. (19)–(20) and the homogeneous Neumann boundary condition.

$$A\mathbf{q}_j^{n+1} = \mathbf{q}_j^n, \text{ for } j = 1, \dots, N_y, \tag{21}$$

$$A\mathbf{r}_i^{n+1} = \mathbf{r}_i^n, \text{ for } i = 1, \dots, N_x, \tag{22}$$

where

$$A = \begin{pmatrix} 1+14\alpha & -15\alpha & \alpha & 0 & \dots & 0 & 0 & 0 & 0 \\ -15\alpha & 1+30\alpha & -16\alpha & \alpha & 0 & \dots & 0 & 0 & 0 \\ \alpha & -16\alpha & 1+30\alpha & -16\alpha & \alpha & 0 & \dots & 0 & 0 \\ 0 & \alpha & -16\alpha & 1+30\alpha & -16\alpha & \alpha & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \alpha & -16\alpha & 1+30\alpha & -16\alpha & \alpha & 0 \\ 0 & 0 & \dots & 0 & \alpha & -16\alpha & 1+30\alpha & -16\alpha & \alpha \\ 0 & 0 & 0 & \dots & 0 & \alpha & -16\alpha & 1+30\alpha & -15\alpha \\ 0 & 0 & 0 & 0 & \dots & 0 & \alpha & -15\alpha & 1+14\alpha \end{pmatrix},$$

$\mathbf{q}_j = (q_{1,j}, \dots, v_{N_x,j})$ and $\mathbf{r}_i = (r_{i,1}, \dots, r_{i,N_y})$. A is a penta-diagonal matrix that can be directly solved without applying an iterative method. Also, A is constructed so that holding the homogeneous boundary condition. We can take $\Delta\tau = \Delta t$ in Eqs. (15) and (16). Then, from Eq. (21), we can obtain $\Phi_2(x, y) = q(x, y, \Delta t)$ with an initial condition $q(x, y, 0) := \Phi_1(x, y)$. Finally, applying an initial condition $r(x, y, 0) := \Phi_2(x, y)$ to Eq. (22), we can get the computational solution $\phi(x, y, (n+1)\Delta t) = r(x, y, \Delta t)$.

We clarify the assumptions and limitations of the proposed method. According to Eq. (14), the phase grows exponentially due to the effect of the nonlinear term. If the time step is excessively large, the solution does not evolve and becomes pinned. Therefore, to ensure the accuracy of the solution, it is assumed that exceptionally large time steps are not used, despite the proposed scheme being unconditionally stable.

2.2. Three-dimensional space

In this section, we expand the AC equation into the 3D domain $\Omega = (L_x, R_x) \times (L_y, R_y) \times (L_z, R_z)$ using the hybrid numerical solution algorithm. Let $\Omega_h = \{(x_i, y_j, z_k) | x_i = L_x + h(i - 0.5),\ y_j = L_y + h(j - 0.5),\ z_k = L_z + h(k - 0.5),\ \text{for } i = 1, \dots, N_x,\ j = 1, \dots, N_y,\ k = 1, \dots, N_z\}$. Here, the spatial mesh sizes are equal in all directions: $h = (L_x - R_x)/N_x = (L_y - R_y)/N_y = (L_z - R_z)/N_z$. We denote $\phi(x_i, y_j, z_k, n\Delta t)$ by the numerical approximation ϕ_{ijk}^n.

In the three-dimensional (3D) domain $\Omega \subset \mathbb{R}^3$, the AC model (1) can be written as

$$\frac{\partial\phi(\mathbf{x}, t)}{\partial t} = -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} + \phi_{xx}(\mathbf{x}, t) + \phi_{yy}(\mathbf{x}, t) + \phi_{zz}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega,\ t > 0, \tag{23}$$

where $\mathbf{x} = (x, y, z) \in \Omega$. For the AC Eq. (23), we use the following temporal operator splitting discretization.

$$\phi(\mathbf{x}, (n+1)\Delta t) = (\mathcal{L}_z^{\Delta t} \circ \mathcal{L}_y^{\Delta t} \circ \mathcal{L}_x^{\Delta t} \circ \mathcal{N}^{\Delta t})\phi(\mathbf{x}, n\Delta t), \tag{24}$$

which has four-step operators: nonlinear operator $\mathcal{N}^{\Delta t}$, x-direction linear operator $\mathcal{L}_x^{\Delta t}$, y-direction linear operator $\mathcal{L}_y^{\Delta t}$, and z-direction linear operator $\mathcal{L}_z^{\Delta t}$.

First, we obtain $\Phi_1(\mathbf{x})$ from $\phi(\mathbf{x}, n\Delta t)$ and nonlinear operator $\mathcal{N}^{\Delta t}$ as

$$\Phi_1(\mathbf{x}) = \mathcal{N}^{\Delta t}\phi(\mathbf{x}, n\Delta t). \tag{25}$$

Here, $\Phi_1(\mathbf{x})$ is a solution of the following nonlinear equation at time $\tau = \Delta t$:

$$\frac{\partial p(\mathbf{x}, \tau)}{\partial \tau} = -\frac{F'(p(\mathbf{x}, \tau))}{\epsilon^2} \tag{26}$$

with the initial condition $p(\mathbf{x}, 0) = \phi(\mathbf{x}, n\Delta t)$, i.e., $\Phi_1(\mathbf{x}) = p(\mathbf{x}, \Delta t)$.

Second, $\Phi_2(\mathbf{x})$ is obtained from $\Phi_1(\mathbf{x})$ and x-direction linear operator $\mathcal{L}_x^{\Delta t}$ as

$$\Phi_2(\mathbf{x}) = \mathcal{L}_x^{\Delta t}\Phi_1(\mathbf{x}). \tag{27}$$

Here, $\Phi_2(\mathbf{x})$ is a solution of the linear equation at time $\tau = \Delta t$:

$$\frac{\partial q(\mathbf{x}, \tau)}{\partial \tau} = q_{xx}(\mathbf{x}, \tau) \tag{28}$$

with the initial condition $q(\mathbf{x},0) = \Phi_1(\mathbf{x})$, i.e., $\Phi_2(\mathbf{x}) = q(\mathbf{x},\Delta t)$.

Third, $\Phi_3(\mathbf{x})$ is obtained from $\Phi_2(\mathbf{x})$ and y-direction linear operator $\mathcal{L}_y^{\Delta t}$ as

$$\Phi_3(\mathbf{x}) = \mathcal{L}_y^{\Delta t}\Phi_2(\mathbf{x}). \tag{29}$$

Here, $\Phi_3(\mathbf{x})$ is a solution of the linear equation at time $\tau = \Delta t$:

$$\frac{\partial r(\mathbf{x},\tau)}{\partial \tau} = r_{yy}(\mathbf{x},\tau) \tag{30}$$

with the initial condition $r(\mathbf{x},0) = \Phi_2(\mathbf{x})$, i.e., $\Phi_3(\mathbf{x}) = r(\mathbf{x},\Delta t)$.

Finally, we get $\phi(\mathbf{x},(n+1)\Delta t)$ from Φ_3 and z-direction linear operator $\mathcal{L}_z^{\Delta t}$ as

$$\phi(\mathbf{x},(n+1)\Delta t) = \mathcal{L}_z^{\Delta t}\Phi_3(\mathbf{x}). \tag{31}$$

Here, $\phi(\mathbf{x},(n+1)\Delta t)$ is a solution of the linear equation at time $\tau = \Delta t$:

$$\frac{\partial s(\mathbf{x},\tau)}{\partial \tau} = s_{zz}(\mathbf{x},\tau) \tag{32}$$

with the initial condition $s(\mathbf{x},0) = \Phi_3(\mathbf{x})$, i.e., $\phi(\mathbf{x},(n+1)\Delta t) = r(\mathbf{x},\Delta t)$. Eqs. (28) and (30) are solved using the fully implicit Euler's method. The homogeneous Neumann boundary condition is applied to all four operators.

Now we describe each step with numerical schemes. First, we solve Eq. (26)

$$\frac{\partial p(\mathbf{x},\tau)}{\partial \tau} = -\frac{F'(p(\mathbf{x},\tau))}{\epsilon^2}. \tag{33}$$

We can obtain the analytic solution,

$$p(\mathbf{x},\tau) = \frac{p(\mathbf{x},0)}{\sqrt{e^{\frac{-2\tau}{\epsilon^2}} + (p(\mathbf{x},0))^2\left(1 - e^{\frac{-2\tau}{\epsilon^2}}\right)}}. \tag{34}$$

for any time $\tau > 0$, using the separation of variables [29], which does not affect the order of accuracy of the scheme [30]. Here $p(\mathbf{x},0) := \phi(\mathbf{x},n\Delta t)$, then we can get $\Phi_1(\mathbf{x}) = p(\mathbf{x},\Delta t)$.

Next, to solve the linear terms, we discretize Eqs. (28), (30), and (32) using a fourth-order discrete Laplacian with the implicit Euler method as follows:

$$\frac{q_{ijk}^{n+1} - q_{ijk}^{n}}{\Delta\tau} = \Delta_h^x q_{ijk}^{n+1}, \tag{35}$$

$$\frac{r_{ijk}^{n+1} - r_{ijk}^{n}}{\Delta\tau} = \Delta_h^y r_{ijk}^{n+1}, \tag{36}$$

$$\frac{s_{ijk}^{n+1} - s_{ijk}^{n}}{\Delta\tau} = \Delta_h^z s_{ijk}^{n+1}, \tag{37}$$

where

$$\Delta_h^x q_{ijk} = \frac{-q_{i-2,jk} + 16q_{i-1,jk} - 30q_{ijk} + 16q_{i+1,jk} - q_{i+2,jk}}{12h^2}, \tag{38}$$

$$\Delta_h^y r_{ijk} = \frac{-r_{i,j-2,k} + 16r_{i,j-1,k} - 30r_{ijk} + 16r_{i,j+1,k} - r_{i,j+2,k}}{12h^2}, \tag{39}$$

$$\Delta_h^z s_{ijk} = \frac{-s_{ij,k-2} + 16s_{ij,k-1} - 30s_{ijk} + 16s_{ij,k+1} - s_{ij,k+2}}{12h^2}. \tag{40}$$

Here, q_{ijk}^n, r_{ijk}^n, and s_{ijk}^n denote $q(x_i,y_j,z_k,n\Delta\tau)$, $r(x_i,y_j,z_k,n\Delta\tau)$, and $s(x_i,y_j,z_k,n\Delta\tau)$, respectively. Rewriting Eqs. (35)–(37) in the following ways:

$$\alpha q_{i-2,jk}^{n+1} - 16\alpha q_{i-1,jk}^{n+1} + (1+30\alpha)\,q_{ijk}^{n+1} - 16\alpha q_{i+1,jk}^{n+1} + \alpha q_{i+2,jk}^{n+1} = q_{ijk}^{n}, \tag{41}$$

$$\alpha r_{i,j-2,k}^{n+1} - 16\alpha r_{i,j-1,k}^{n+1} + (1+30\alpha)\,r_{ijk}^{n+1} - 16\alpha r_{i,j+1,k}^{n+1} + \alpha r_{i,j+2,k}^{n+1} = r_{ijk}^{n}, \tag{42}$$

$$\alpha s_{i,j,k-2}^{n+1} - 16\alpha s_{i,j,k-1}^{n+1} + (1+30\alpha)\,s_{ijk}^{n+1} - 16\alpha s_{i,j,k+1}^{n+1} + \alpha s_{i,j,k+2}^{n+1} = s_{ijk}^{n}, \tag{43}$$

where $\alpha = \Delta\tau/(12h^2)$. We can derive the following matrix formula using Eqs. (41)–(43) and the homogeneous Neumann boundary condition.

$$A\mathbf{q}_{jk}^{n+1} = \mathbf{q}_{jk}^{n}, \text{ for } j = 1,\dots,N_y \text{ and } k = 1,\dots,N_z, \tag{44}$$

$$A\mathbf{r}_{ik}^{n+1} = \mathbf{r}_{ik}^{n}, \text{ for } i = 1,\dots,N_x \text{ and } k = 1,\dots,N_z, \tag{45}$$

$$A\mathbf{s}_{ij}^{n+1} = \mathbf{s}_{ij}^{n}, \text{ for } i = 1,\dots,N_x \text{ and } j = 1,\dots,N_y, \tag{46}$$

where

$$A = \begin{pmatrix} 1+14\alpha & -15\alpha & \alpha & 0 & \dots & 0 & 0 & 0 & 0 \\ -15\alpha & 1+30\alpha & -16\alpha & \alpha & 0 & \dots & 0 & 0 & 0 \\ \alpha & -16\alpha & 1+30\alpha & -16\alpha & \alpha & 0 & \dots & 0 & 0 \\ 0 & \alpha & -16\alpha & 1+30\alpha & -16\alpha & \alpha & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \alpha & -16\alpha & 1+30\alpha & -16\alpha & \alpha & 0 \\ 0 & 0 & \dots & 0 & \alpha & -16\alpha & 1+30\alpha & -16\alpha & \alpha \\ 0 & 0 & 0 & \dots & 0 & \alpha & -16\alpha & 1+30\alpha & -15\alpha \\ 0 & 0 & 0 & 0 & \dots & 0 & \alpha & -15\alpha & 1+14\alpha \end{pmatrix},$$

$\mathbf{q}_{jk} = (q_{1,jk}, \dots, v_{N_x,jk})$, $\mathbf{r}_{ik} = (r_{i,1,k}, \dots, r_{i,N_y,k})$ and $\mathbf{s}_{ij} = (r_{ij,1}, \dots, r_{ij,N_z})$. A is a penta-diagonal matrix and can be directly solved without applying an iterative method. Also, A is constructed so that holding the homogeneous boundary condition. We can take $\Delta\tau = \Delta t$ into Eqs. (35), (36) and (37). Then, from Eq. (44), we can obtain $\Phi_2(\mathbf{x}) = q(\mathbf{x}, \Delta t)$ with an initial condition $q(\mathbf{x}, 0) := \Phi_1(\mathbf{x})$. Similarly, we obtain $\Phi_3(\mathbf{x}) = r(\mathbf{x}, \Delta t)$ from Eq. (45) with an initial condition $r(\mathbf{x}, 0) := \Phi_2(\mathbf{x})$. Finally, applying an initial condition $s(\mathbf{x}, 0) := \Phi_3(x, y)$ to Eq. (46), we can get the computational solution $\phi(\mathbf{x}, (n+1)\Delta t) = s(\mathbf{x}, \Delta t)$.

3. Numerical experiments

This section presents details of numerical tests such as stability tests, convergence tests, and maximum principle tests of the proposed scheme on the AC equation. Unless otherwise specified, we consider the cell-centered grid to ensure the homogeneous Neumann boundary condition.

3.1. Stability analysis

In this section, we prove the stability of the proposed numerical solution algorithm described in Section 2. For simplicity, we consider the following operator splitting discretization in Eq. (6) in 2D space. The proof for 3D is similar to that for 2D.

Theorem 1. *Suppose that ϕ_{ij}^n is bounded $|\phi_{ij}^n| \le 1$ for $i = 1, \dots, N_x$ and $j = 1, \dots, N_y$. Then, the proposed fourth-order FDM*

$$\phi_{ij}^{n+1} = (\mathcal{L}_y^{\Delta t} \circ \mathcal{L}_x^{\Delta t} \circ \mathcal{N}^{\Delta t})\phi_{ij}^n \tag{47}$$

preserves the boundedness of the numerical solution $|\phi_{ij}^{n+1}| \le 1$ for any time step Δt. Here, the proposed method is consistent with the following steps:

$$\Phi_1(x, y) = \mathcal{N}^{\Delta t}\phi(x, y, n\Delta t), \tag{48}$$

$$\Phi_2(x, y) = \mathcal{L}_x^{\Delta t}\Phi_1(x, y), \tag{49}$$

$$\phi(x, y, (n+1)\Delta t) = \mathcal{L}_y^{\Delta t}\Phi_2(x, y). \tag{50}$$

Proof. In the first step (48), $\Phi_1(x, y)$ is obtained analytically by Eq. (14) as follows:

$$\Phi_1(x, y) = \frac{\phi(x, y, n\Delta t)}{\sqrt{e^{\frac{-2\Delta t}{\epsilon^2}} + (\phi(x, y, n\Delta t))^2 \left(1 - e^{\frac{-2\Delta t}{\epsilon^2}}\right)}}, \tag{51}$$

which ensures that $|\Phi_1(x, y)| \le 1$ for any Δt, since $|\phi(x, y, n\Delta t)| \le 1$. In the second step (49), we consider the following heat equation with the given initial condition and the homogeneous Neumann boundary condition.

$$\frac{\partial q(x, y, \tau)}{\partial \tau} = q_{xx}(x, y, \tau), \tag{52}$$

$$q(x, y, 0) = \Phi_1(x, y), \tag{53}$$

$$\mathbf{n} \cdot \nabla q(x, y, \tau) = 0. \tag{54}$$

Eq. (52) can be discretized with the fourth-order accurate implicit method as follows:

$$\frac{q_i^{n+1} - q_i^n}{\Delta\tau} = \frac{-q_{i-2}^{n+1} + 16q_{i-1}^{n+1} - 30q_i^{n+1} + 16q_{i+1}^{n+1} - q_{i+2}^{n+1}}{12h^2}, \tag{55}$$

for $i = 1, \dots, N_x$. Since we only consider the x-direction, we omit subscript j, which is the index of the y-direction. The boundary condition is implemented as

$$q_{-1} := q_2, \quad q_0 := q_1, \quad q_{N_x+1} := q_{N_x}, \quad q_{N_x+2} := q_{N_x-1}. \tag{56}$$

For a given discrete value q_i^n, we apply the discrete cosine transform to obtain $\hat{q}_M^n$ and the inverse discrete cosine transform :

$$\hat{q}_M^n = \alpha_M \sum_{i=1}^{N} q_i^n \cos\big(\beta_M \pi(x_i - L_x)\big), \tag{57}$$

$$q_i^n = \sum_{M=1}^{N} \alpha_M \hat{q}_M^n \cos\big(\beta_M \pi(x_i - L_x)\big), \tag{58}$$

where

$$\alpha_M = \begin{cases} \sqrt{\dfrac{1}{N}}, & \text{for} \quad M = 1, \\ \sqrt{\dfrac{2}{N}}, \text{for} & M \geq 2, \end{cases} \qquad \beta_M = \frac{M-1}{R_x - L_x}, \text{for} \quad M = 1, \dots, N. \tag{59}$$

By using inverse discrete cosine transform (58), discrete values q_i^n can be represented as a summation of cosine functions, which satisfies the homogeneous Neumann boundary condition. Since the cosine functions in Eq. (58) are the basis functions of q_i^n with respect to M, we can consider q_i^n omitting the subscript M as follows:

$$q_i^n = \alpha \hat{q}^n \cos\big(\beta\pi(x_i - L_x)\big). \tag{60}$$

In Eq. (60), we notate the amplitude at nth time as $\xi^n := \alpha\hat{q}^n$. Substituting the discrete cosine mode $q_i^n = \xi^n \cos\big(\beta\pi(x_i - L_x)\big)$ into Eq. (55), we obtain

$$\begin{aligned} |\xi| &= \frac{1}{1 + 4\alpha\left[\cos^2\big(\beta\pi(x_i - L_x)\big) - 8\cos\big(\beta\pi(x_i - L_x)\big) + 7\right]} \\ &= \frac{1}{1 + 4\alpha\left\{[\cos\big(\beta\pi(x_i - L_x)\big) - 4]^2 - 9\right\}} \leq 1, \end{aligned} \tag{61}$$

where $\alpha = \Delta\tau/(12h^2)$. By the von Neumann stability analysis [31], for any time step $\Delta\tau$, the numerical method (55) is unconditionally stable. Taking $\Delta\tau = \Delta t$, it implies that $|\Phi_2(x, y) = q(x, y, \Delta t)| \leq 1$. Analogously, when applied to the y-direction, we can obtain $|\phi_{ij}^{n+1}| \leq 1$, which guarantees the boundedness of the numerical solution for any time step Δt. □

We demonstrate the stability of the proposed method by considering the dynamics of phase separation in the 2D domain $\Omega = (0, 1)^2$. We take the numerical parameters as $N_x = N_y = 256$, $h = 1/N_x = 1/N_y$ and the following random perturbed initial condition:

$$\phi(x, y, 0) = \text{rand}(x, y), \tag{62}$$

where $\text{rand}(x, y)$ is a random number in $[-1,\ 1]$. Fig. 1 shows temporal evolutions of the AC equation with different time steps $\Delta t = 0.1h^2$, $\Delta t = h^2$, and $\Delta t = 10h^2$. In Fig. 1 the numerical solutions of the AC equation do not blow up for the given time steps and remain bounded by -1 and 1, which confirms the stability of the proposed method.

3.2. Numerical maximum principle and total energy decreasing

The total energy non increasing property of Eq. (4) and the maximum principle property [1] of the AC equation is demonstrated by numerical experiments of the proposed scheme in 2D and 3D space. To validate this, we consider the computational domain $\Omega_h = (0, 1)^2$ and $(0, 1)^3$, respectively. The initial condition is $\phi(x, y, 0) = 0.1\text{rand}(x, y)$ and $\phi(x, y, z, 0) = 0.1\text{rand}(x, y, z)$ where $\text{rand}(x, y)$ and $\text{rand}(x, y, z)$ have random values between -1 and 1 in 2D and 3D, respectively.

Let us consider the following decomposition of the fourth-order discrete Laplacian operator:

$$\begin{aligned} \Delta_h^x \phi_{ij} &= \frac{-\phi_{i-2,j} + 16\phi_{i-1,j} - 30\phi_{ij} + 16\phi_{i+1,j} - \phi_{i+2,j}}{12h^2} \\ &= -\frac{1}{12}\frac{\phi_{i-2,j} - 2\phi_{i-1,j} + \phi_{ij}}{h^2} + \frac{14}{12}\frac{\phi_{i-1,j} - 2\phi_{ij} + \phi_{i+1,j}}{h^2} \\ &\quad - \frac{1}{12}\frac{\phi_{ij} - 2\phi_{i+1,j} + \phi_{i+2,j}}{h^2} \\ &= -\frac{1}{12h}\left(\frac{\phi_{ij} - \phi_{i-1,j}}{h} - \frac{\phi_{i-1,j} - \phi_{i-2,j}}{h}\right) \\ &\quad + \frac{14}{12h}\left(\frac{\phi_{i+1,j} - \phi_{ij}}{h} - \frac{\phi_{ij} - \phi_{i-1,j}}{h}\right) \\ &\quad - \frac{1}{12h}\left(\frac{\phi_{i+2,j} - \phi_{i+1,j}}{h} - \frac{\phi_{i+1,j} - \phi_{ij}}{h}\right). \end{aligned} \tag{63}$$

From Eq. (63), we define a discrete first derivative as

$$D_h^x \phi_{i+\frac{1}{2},j} = -\frac{1}{12}\frac{\phi_{ij} - \phi_{i-1,j}}{h} + \frac{14}{12}\frac{\phi_{i+1,j} - \phi_{ij}}{h} - \frac{1}{12}\frac{\phi_{i+2,j} - \phi_{i+1,j}}{h}. \tag{64}$$

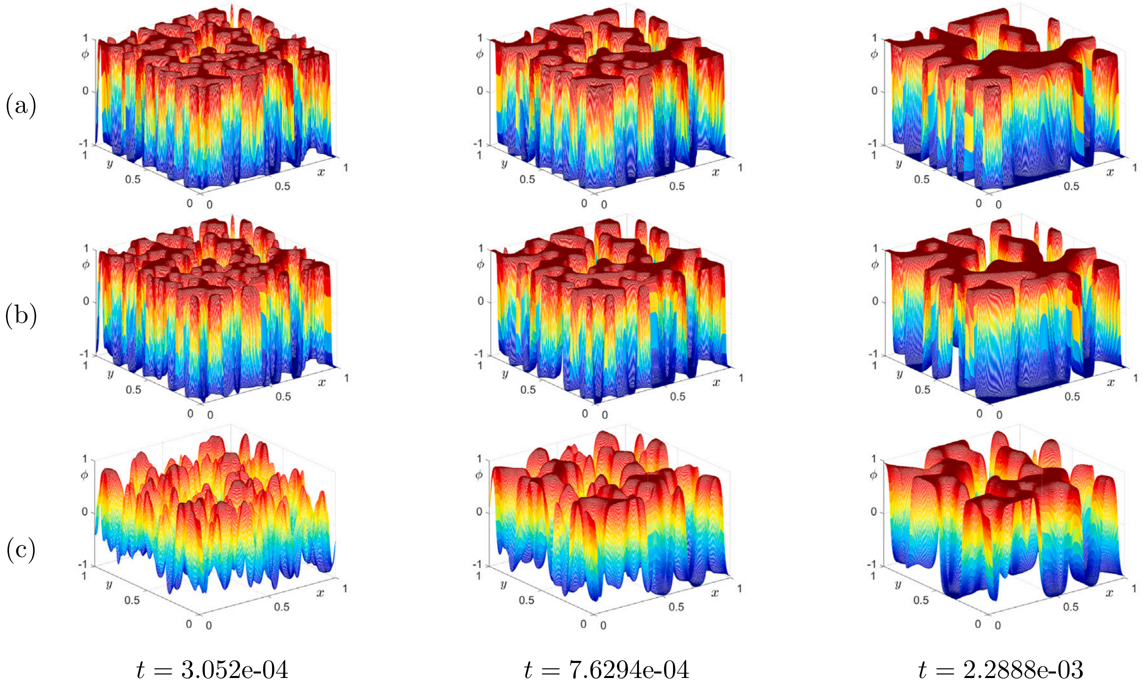


Fig. 1. Snapshots of temporal evolutions of the AC equation with random perturbed initial condition and different time steps (a) $\Delta t = 0.1h^2$, (b) $\Delta t = h^2$, and (c) $\Delta t = 10h^2$.

Other discrete first derivatives are similarly defined. Therefore,

$$\Delta_h^x \phi_{ij} = \frac{D_h^x \phi_{i+\frac{1}{2},j} - D_h^x \phi_{i-\frac{1}{2},j}}{h}, \quad \Delta_h^y \phi_{ij} = \frac{D_h^y \phi_{i,j+\frac{1}{2}} - D_h^y \phi_{i,j-\frac{1}{2}}}{h}. \tag{65}$$

Similar definitions are applied for 3D space. Now, we define the discrete energy functional $\mathcal{E}_d(\phi)$ in d-dimension for $d = 2, 3$ as follows:

$$\begin{aligned}
\mathcal{E}_2(\phi^n) = {} & h^2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \frac{F(\phi_{ij}^n)}{\epsilon^2} \\
& + \frac{h^2}{2} \sum_{j=1}^{N_y} \left(\frac{1}{2} \left(D_h^x \phi_{\frac{1}{2},j} \right)^2 + \sum_{i=1}^{N_x-1} \left(D_h^x \phi_{i+\frac{1}{2},j} \right)^2 + \frac{1}{2} \left(D_h^x \phi_{N_x+\frac{1}{2},j} \right)^2 \right) \\
& + \frac{h^2}{2} \sum_{i=1}^{N_x} \left(\frac{1}{2} \left(D_h^y \phi_{i,\frac{1}{2}} \right)^2 + \sum_{j=1}^{N_y-1} \left(D_h^y \phi_{i,j+\frac{1}{2}} \right)^2 + \frac{1}{2} \left(D_h^y \phi_{i,N_y+\frac{1}{2}} \right)^2 \right), \\
\mathcal{E}_3(\phi^n) = {} & h^3 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \frac{F(\phi_{ijk}^n)}{\epsilon^2} \\
& + \frac{h^3}{2} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \left(\frac{1}{2} \left(D_h^x \phi_{\frac{1}{2},j,k} \right)^2 + \sum_{i=1}^{N_x-1} \left(D_h^x \phi_{i+\frac{1}{2},j,k} \right)^2 + \frac{1}{2} \left(D_h^x \phi_{N_x+\frac{1}{2},j,k} \right)^2 \right) \\
& + \frac{h^3}{2} \sum_{i=1}^{N_x} \sum_{k=1}^{N_z} \left(\frac{1}{2} \left(D_h^y \phi_{i,\frac{1}{2},k} \right)^2 + \sum_{j=1}^{N_y-1} \left(D_h^y \phi_{i,j+\frac{1}{2},k} \right)^2 + \frac{1}{2} \left(D_h^y \phi_{i,N_y+\frac{1}{2},k} \right)^2 \right) \\
& + \frac{h^3}{2} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(\frac{1}{2} \left(D_h^z \phi_{i,j,\frac{1}{2}} \right)^2 + \sum_{j=1}^{N_z-1} \left(D_h^z \phi_{i,j,k+\frac{1}{2}} \right)^2 + \frac{1}{2} \left(D_h^z \phi_{i,j,N_z+\frac{1}{2}} \right)^2 \right),
\end{aligned}$$

where h is spatial mesh size for $h = 1/N_x = 1/N_y = 1/N_z$ and interfacial thickness ϵ is chosen as $6h/(2\sqrt{2}\tanh^{-1}(0.9))$. To validate the maximum principle, we define $(\cdot)_{\max}$ and $(\cdot)_{\min}$ as

$$(\phi^n)_{\max} = \max_{ij} \phi_{ij}^n \quad \text{and} \quad (\phi^n)_{\min} = \min_{ij} \phi_{ij}^n \quad \text{in} \quad 2\text{D}, \tag{66}$$

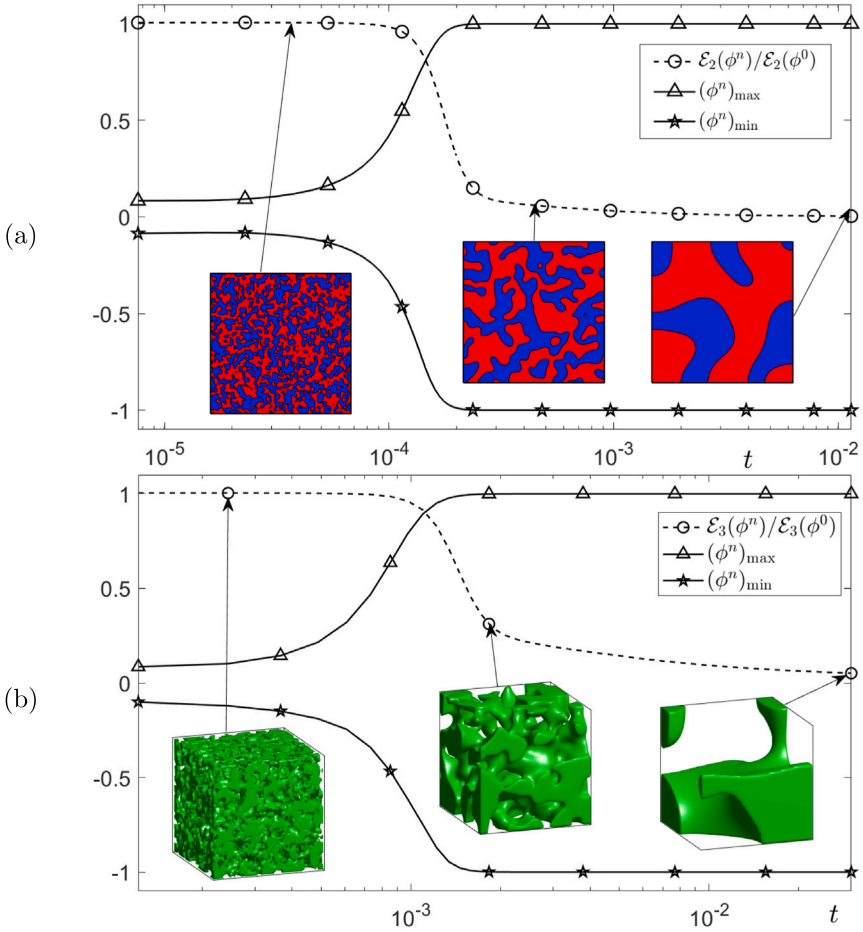


Fig. 2. Snapshots of evolutionary dynamics and discrete energy functional for the AC equation in (a) 2D and (b) 3D. $\mathcal{E}_d(\phi^n)/\mathcal{E}_d(\phi^0)$, $(\phi^n)_{\max}$ and $(\phi^n)_{\min}$ are shown on a semi-logarithmic scale.

$$(\phi^n)_{\max} = \max_{ijk} \phi^n_{ijk} \quad \text{and} \quad (\phi^n)_{\min} = \min_{ijk} \phi^n_{ijk} \quad \text{in} \quad 3\text{D}. \tag{67}$$

To demonstrate the boundedness and total energy non-increasing property, we designed numerical experiments for 2D and 3D. We consider $N_x = N_y = 256$, $\epsilon = 6h/(2\sqrt{2}\tanh^{-1}(0.9))$ for 2D and $N_x = N_y = N_z = 64$, $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$ for 3D. For both tests, the spatial step size is $h = 1/N_x$ and temporal step size is $\Delta t = 0.5h^2$. Fig. 2(a) and (b) illustrate $\mathcal{E}_d(\phi^n)/\mathcal{E}_d(\phi^0)$, $(\phi^n)_{\min}$ and $(\phi^n)_{\max}$ according to the temporal evolving of computational solution for 2D and 3D spaces, respectively. Numerical tests confirm the maximum bound principle and the total energy non-increasing property for the proposed fourth-order hybrid scheme.

3.3. Convergence test

In this section, we demonstrate the convergence rate of the AC equation using the traveling wave. For a given initial condition $\phi(x, y, 0)$, the traveling wave has an exact solution $\phi_{ext}(x, y, T)$ at time T for the AC equation:

$$\phi(x, y, 0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x}{2\sqrt{2}\epsilon}\right), \tag{68}$$

$$\phi_{ext}(x, y, T) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x - sT}{2\sqrt{2}\epsilon}\right), \tag{69}$$

where $\epsilon = 0.2/(2\sqrt{2}\tanh^{-1}(0.9))$ is a constant and s is the speed of the traveling wave [32,33]. By applying Eq. (69) into Eq. (5), we obtain the following equation:

$$\frac{\sqrt{2}s - 3\epsilon}{8\epsilon}\operatorname{sech}^2\left(\frac{x - st}{2\sqrt{2}\epsilon}\right) = 0. \tag{70}$$

Table 1
l_2-norm error and spatial convergence rate on the AC equation.

	$N_x = 16$	Rate	$N_x = 32$	Rate	$N_x = 64$
Second-order	8.9135e−3	1.9291	2.3406e−3	1.9753	5.9525e−4
Fourth-order	2.1711e−3	3.5784	1.8175e−4	3.9140	1.2057e−5

Table 2
l_2-norm error and temporal convergence rate on the AC equation.

	$\Delta t = 0.8h^2$	Rate	$\Delta t = 0.4h^2$	Rate	$\Delta t = 0.2h^2$
Fourth-order	1.3581e−2	1.0853	6.4011e−3	0.9784	3.2487e−3

Therefore, the speed of the traveling wave that satisfies Eq. (70) is $s = 3\epsilon/\sqrt{2}$. The traveling wave is applied on a rectangular domain $(-0.5, 1.5) \times (-0.5, 0.5)$ where the number of spatial steps are $N_x = 64$, $N_y = 32$ and step size $h = 1/32$. For a final time $T = 0.01$, the temporal step size is determined using the spatial step size: $\Delta t = 1\text{e}{-7}$. The number of time steps are determined accordingly, $N_t = T/\Delta t = 100\,000$. Numerical results and the exact solution is illustrated in Fig. 3(a). The white, red, and green surfaces indicate the initial condition, numerical result, and exact solution, respectively. Now that we have confirmed that the numerical solution of the traveling wave matches with the exact solution, we calculate the discrete l_2-norm error. For a final time T, the error E between the numerical solution ϕ and exact solution ϕ_{ext} is defined as

$$\begin{aligned} e_{ij} &= \phi(x_i, y_j) - \phi_{ext}(x_i, y_j) \ \text{ for } i = 1, \dots, N_x \text{ and } j = 1, \dots, N_y, \\ E &= \{e_{ij}, \ \text{ for } i = 1, \dots, N_x \text{ and } j = 1, \dots, N_y\}. \end{aligned} \tag{71}$$

The discrete l_2-error $\|\cdot\|_2$ is defined as follows:

$$\|E\|_2 = \sqrt{\frac{1}{N_x N_y} \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} (e_{ij})^2}. \tag{72}$$

Using the above defined error and l_2-norm, the spatial convergence rate is calculated by $\log_2(\|E_1\|_2/\|E_2\|_2)$ where E_1 is the error with spatial step size h and E_2 is the error with spatial step size $h/2$. Temporal convergence rate is calculated similarly by using a half decreased temporal step size. To calculate the spatial convergence rate, we set $T = 0.01$, $\Delta t = 1\text{e}{-7}$, $N_x = 2^n$ for $n = 4, 5, 6$, and $N_y = N_x/2$ on a rectangular domain $(-0.5, 1.5)\times(-0.5, 0.5)$. Fig. 3(b)–(d) illustrates the error difference for the proposed fourth-order scheme and a second-order scheme. In 2D and 3D space, the Laplace operator is approximated by using second-order discretization as follows:

$$\begin{aligned} \Delta\phi_{ij} &= \phi_{xx} + \phi_{yy} \\ &= \frac{\phi_{i-1,j} - 2\phi_{ij} + \phi_{i+1,j}}{h^2} + \frac{\phi_{i,j-1} - 2\phi_{ij} + \phi_{i,j+1}}{h^2} + \mathcal{O}(h^2), \end{aligned} \tag{73}$$

$$\begin{aligned} \Delta\phi_{ijk} &= \phi_{xx} + \phi_{yy} + \phi_{zz} \\ &= \frac{\phi_{i-1,jk} - 2\phi_{ijk} + \phi_{i+1,jk}}{h^2} + \frac{\phi_{i,j-1,k} - 2\phi_{ijk} + \phi_{i,j+1,k}}{h^2} \\ &\quad + \frac{\phi_{ij,k-1} - 2\phi_{ijk} + \phi_{ij,k+1}}{h^2} + \mathcal{O}(h^2). \end{aligned} \tag{74}$$

The error of the fourth-order scheme decreases significantly faster than the second-order scheme. Error and convergence rate results on Table 1 show that the scheme is fourth-order accurate in space.

Next, the temporal convergence rate is calculated when $N_x = 96$ $N_y = 48$, $h = 1/48$, $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$, $T = 0.002$, and $dt = 0.8h^2/2^N$, where $N = 0, 1, 2$. Results in Table 2 show that the proposed scheme is first-order accurate in time. In this study, we focus on the fourth-order space-accurate scheme for the AC equation. The proposed method can be extended using the second-order time-accurate operator splitting scheme [34,35] and alternating direction implicit method [36,37].

3.4. Boundary error

In this section, we investigate the numerical error at the boundary when using the fourth-order scheme in 2D space. Since there are no closed-form analytic solutions for the general initial and boundary conditions of the AC equation, we use a manufactured solution to compute the numerical error of the proposed numerical method. The manufactured benchmark problem solution for the AC equation is defined as follows [38]:

$$\Phi(x, y, t) = \alpha(t)\cos(k_x x)\cos(k_y y),$$

where

$$\alpha(t) = \alpha(0)\exp\left[\left(\frac{1}{\epsilon^2} - k_x^2 - k_y^2\right)t\right]$$

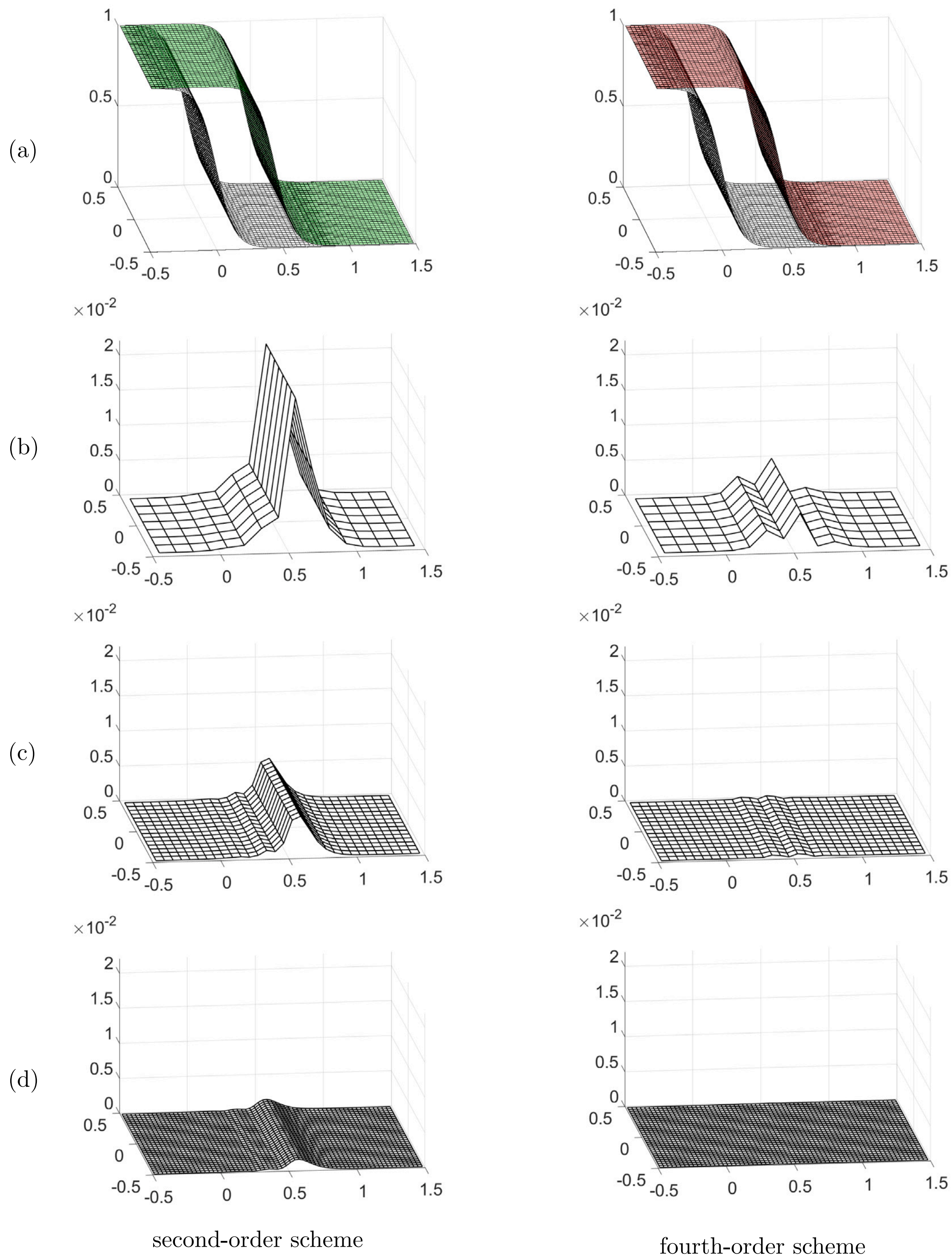


Fig. 3. (a) Initial condition (white), exact solution (green), and numerical result (red) at time $T = 0.01$. (b)–(d) Error when $N_x = 16$, 32 and 64.

is temporal amplitude for the given initial amplitude $\alpha(0)$. Then, the modified AC equation for the benchmark problem is defined as follows:

$$\phi_t(x, y, t) = -\frac{F'(\phi(x, y, t))}{\epsilon^2} + \Delta\phi(x, y, t) + s(x, y, t), \tag{75}$$

Table 3
l_2-error and convergence rate at the boundary.

	$h = 1/32$	Rate	$h = 1/64$	Rate	$h = 1/128$	Rate	$h = 1/256$
$\|E_{\text{bd}}\|_2$	3.4049e−4	1.9991	8.5175e−5	1.9998	2.1297e−5	1.9999	5.3244e−6

Table 4
Average CPU time and error using fourth-order and second-order scheme.

	$N_x \times N_y = 64 \times 32$		$N_x \times N_y = 128 \times 64$		$N_x \times N_y = 256 \times 128$	
	CPU time	Error	CPU time	Error	CPU time	Error
Second-order	1.1151	1.7529e−3	3.5223	4.4297e−6	11.5213	1.1106e−6
Fourth-order	1.7363	1.2448e−6	5.4918	8.3195e−8	18.6312	9.0309e−9

where

$$s(x, y, t) = \Phi_t(x, y, t) + \frac{F'(\Phi(x, y, t))}{\epsilon^2} - \Delta\Phi(x, y, t).$$

We solve Eq. (75) numerically by using the proposed OSM as follows:

$$\phi(\mathbf{x}, (n+1)\Delta t) = (\mathcal{S}^{\Delta t} \circ \mathcal{L}_y^{\Delta t} \circ \mathcal{L}_x^{\Delta t} \circ \mathcal{N}^{\Delta t})\phi(x, y, n\Delta t),$$

where $\mathcal{S}^{\Delta t}$ is operator for source term $s(x, y, t)$.

For the convergence test at the boundary, we consider the following initial condition on the computational domain $\Omega = (-1, 1) \times (-1, 1)$.

$$\phi(x, y, 0) = 0.1\cos(2\pi x)\cos(2\pi y).$$

The discrete l_2-error on the boundary is defined as follows:

$$\|E_{\text{bd}}\|_2 = \sqrt{\frac{e}{2(N_x + N_y)}},$$

where

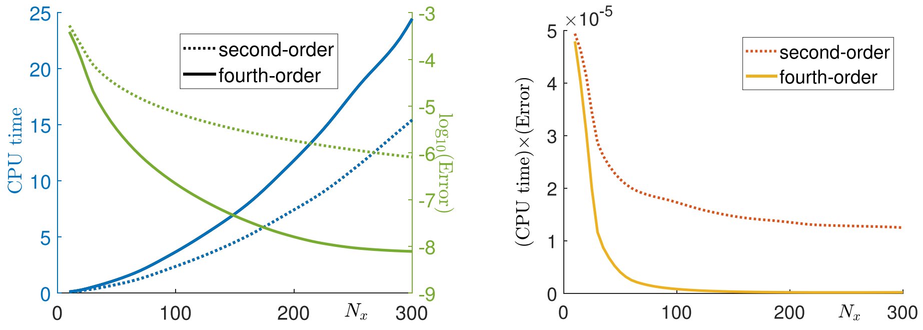
$$\begin{aligned} e = & \sum_{j=1}^{N_y}\left(\left[\phi_{1,j}^{N_t} - \Phi(L_x, y_j, T)\right]^2 + \left[\phi_{N_x,j}^{N_t} - \Phi(R_x, y_j, T)\right]^2\right) \\ & + \sum_{i=1}^{N_x}\left(\left[\phi_{i,1}^{N_t} - \Phi(x_i, L_y, T)\right]^2 + \left[\phi_{i,N_y}^{N_t} - \Phi(x_i, R_y, T)\right]^2\right). \end{aligned}$$

It exhibits second-order accuracy at the boundary, employing the discrete first derivative defined in Eq. (64) and the homogeneous Neumann boundary condition. Table 3 lists the l_2-errors and convergence rates at the boundary. Results in Table 3 demonstrate that the proposed fourth-order method achieves second-order accurate at the boundary. Consequently, while the proposed numerical method achieves second-order accuracy at the boundary, from the perspective of the entire domain, this can be considered negligible. As shown in Table 1, the proposed method achieves fourth-order accuracy in space over the entire domain.

3.5. CPU time and error comparison

Continuing from Section 3.3, we analyze the CPU time and error of the proposed fourth-order scheme and the second-order scheme. Using the initial condition (68), $\epsilon = 0.2/(2\sqrt{2}\tanh^{-1}(0.9))$ and $s = 3/(\sqrt{2}\epsilon)$ on the space $(-1, 1) \times (-0.5, 0.5)$, we ran the fourth- and second-order schemes until the final time $T = 0.0001$ and $N_t = 10000$. We constructed three different tests, each using $N_x \times N_y = 64 \times 32, 128 \times 64$, and 256×128. Each test is performed 5 times, calculating the CPU time and l_2-norm error of the fourth-order scheme and second-order scheme. Each case is tested on a 6-core workstation with 16 Gb of RAM memory. Table 4 lists the average CPU time and error for both schemes.

Note the case where $N_x = 256$ at the second-order scheme and $N_x = 64$ at the fourth-order scheme. Both cases have similar l_2-norm error, however the CPU time is remarkably smaller with the fourth-order scheme. Fig. 4(a) illustrates the CPU time(blue) and l_2-norm error(green) respect to N_x for the second-order scheme(dotted line) and fourth-order scheme(solid line). Error is shown in a logarithmic scale. The error is smaller when using the fourth-order scheme and costs longer CPU time. However, observing Fig. 4(b) where it illustrates the value of (CPU time) × (Error), we can see that it converges but smaller at the fourth-order scheme. We can conclude that the fourth-order scheme is the superior option.



(a) (b)

Fig. 4. (a) CPU time and l_2-norm error respect to N_x. (b) (CPU time) × (Error) respect to N_x.

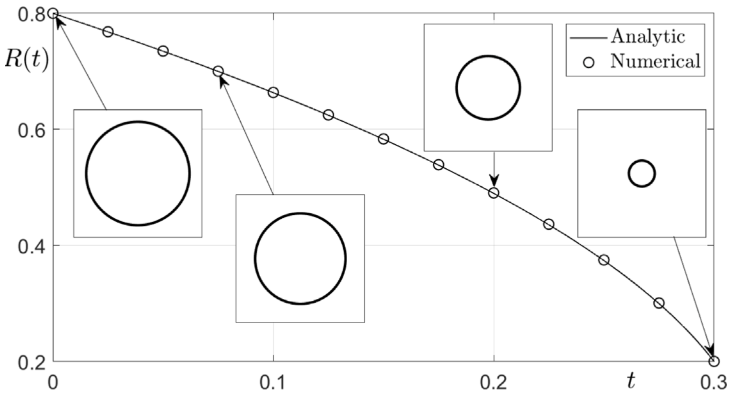


Fig. 5. Temporal evolution of the zero level contour and radius $R(t)$ in 2D.

3.6. Motion by mean curvature

We consider the motion by mean curvature with the AC equation. The zero level set of ϕ moves according to motion by mean curvature as ϵ converges to zero [39]. Model a circle with an initial radius R_0, which is given as

$$\phi(x, y, 0) = \tanh\left(\frac{R_0 - \sqrt{x^2 + y^2}}{\sqrt{2}\epsilon}\right), \tag{76}$$

on $\Omega = (-1, 1) \times (-1, 1)$ with $N_x \times N_y$ mesh. We used parameters $R_0 = 0.8$, $h = 1/64$, $\Delta t = 0.05h^2$, $T = 0.3$ and $\epsilon = \epsilon_9$. ϵ_m is defined as $\epsilon_m = hm/(2\sqrt{2}\tanh^{-1}(0.9))$ [32]. The analytic solution for this model at time t is a circle with radius $R(t) = \sqrt{R_0^2 - 2t}$. Fig. 5 shows the zero-level contours of the analytic and numerical solution with $N_x = N_y = 128$.

Next, we consider a sphere with the initial radius R_0. The initial condition is given as

$$\phi(x, y, z, 0) = \tanh\left(\frac{R_0 - \sqrt{x^2 + y^2 + z^2}}{\sqrt{2}\epsilon}\right), \tag{77}$$

on $\Omega = (-1, 1) \times (-1, 1) \times (-1, 1)$ with $N_x \times N_y \times N_z$ mesh. The used parameters are $R_0 = 0.8$, $h = 1/32$, $\Delta t = 0.1h^2$, $T = 0.15$ and $\epsilon = \epsilon_7$. The radius $R(t) = \sqrt{R_0^2 - 4t}$ is the analytic solution at time t. Fig. 6 shows the temporal evolution of the sphere with $N_x = N_y = N_z = 64$. We can observe that for both 2D and 3D, the motion by mean curvature property of the AC equations holds when applying the proposed numerical method.

Here, we compare the numerical results for second-order and fourth-order schemes and present advantages of the fourth-order scheme over the second-order scheme. First, we consider the following circle shaped initial condition

$$\phi(x, y, 0) = \tanh\left(\frac{0.8 - \sqrt{x^2 + y^2}}{\sqrt{2}\epsilon}\right)$$

on $\Omega = (-1, 1) \times (-1, 1)$ with 128 × 128 mesh. Here, we use $h = 1/64$, $\Delta t = 0.25h^2$ and $\epsilon = 2.6h/(2\sqrt{2}\tanh^{-1}(0.9))$. Fig. 7 shows temporal evolutions of the zero-level contour of the numerical solutions for the second-order scheme and fourth-order scheme. The

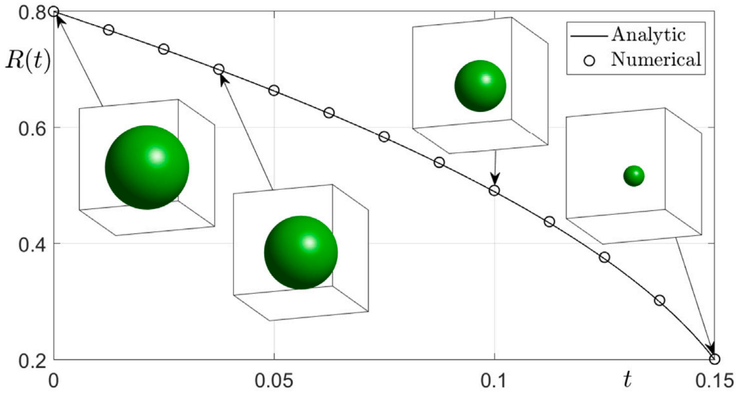
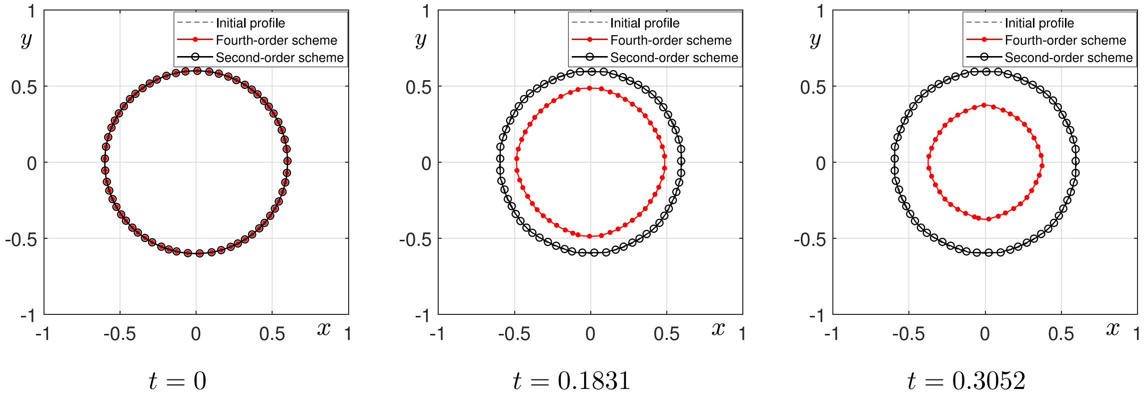


Fig. 6. Temporal evolution of the zero level contour and radius $R(t)$ in 3D.



$t = 0$ $t = 0.1831$ $t = 0.3052$

Fig. 7. Temporal evolutions of the zero-level contour of the numerical solutions for the fourth-order scheme and second-order scheme.

interface of the Allen–Cahn equation, driven by motion by mean curvature flow, should shrink as demonstrated in the results of Section 3.5. However, in the case of the second-order scheme, the solution becomes pinned and does not evolve for small interfacial parameter ϵ. Therefore, compared to a second-order scheme, the fourth-order scheme can capture sharp interface details more effectively.

4. Conclusions

In this paper, we proposed the fourth-order accurate hybrid method on grids for the AC equation. The proposed method consists of the OSM. In the proposed scheme, we used the implicit Euler's scheme and operator splitting scheme to solve discrete Laplace term in each direction in 2D and 3D respectively. And, we used the analytic solution to solve nonlinear term. We proved the stability of our scheme using the von Neumann stability analysis. Several numerical simulations demonstrated the superior performance of the proposed scheme and compared it with the second order scheme. We also showed that the our proposed scheme satisfies the motion by mean curvature flow and the total energy decreasing property by numerical simulation. Furthermore, compared to a second-order scheme, this method permits the use of a finer interfacial parameter ϵ, thereby enhancing accuracy and providing more detailed modeling at the same resolution, such as other reaction–diffusion equations.

Given in Section 3.3, our method shows high performance on spatial convergence. However, one weakness is when the error from the time derivative is the dominant error. In cases like this, the high spatial convergence rate is less effective, reducing the convergence rate of the entire scheme. This is because the scheme is fourth-order accurate in space but first-order accurate in time. Therefore, for future work, we plan to develop our numerical scheme to be unconditionally stable and second-order accurate in time. To achieve this goal we are considering semi-implicit Euler's scheme [30] and half step evolution operators [40].

Data availability

Data will be made available on request.

Acknowledgments

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). The authors express their gratitude to the reviewers for their valuable feedback during the revision process of this article.

References

[1] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metall. 27 (1979) 1085–1095.

[2] 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.

[3] D. He, K. Pan, H. Hu, A spatial fourth-order maximum principle preserving operator splitting scheme for the multi-dimensional fractional Allen–Cahn equation, Appl. Numer. Math. 151 (2020) 44–63.

[4] H. Zhang, J. Yan, X. Qian, S. Song, Temporal high-order, unconditionally maximum-principle-preserving integrating factor multi-step methods for Allen–Cahn-type parabolic equations, Appl. Numer. Math. 186 (2023) 19–40.

[5] C. Li, Y. Huang, N. Yi, An unconditionally energy stable second order finite element method for solving the Allen–Cahn equation, J. Comput. Appl. Math. 353 (2019) 38–48.

[6] F. Yan, Y. Xu, Stability analysis and error estimates of local discontinuous Galerkin methods with semi-implicit spectral deferred correction time-marching for the Allen–Cahn equation, J. Comput. Appl. Math. 376 (2020) 112857.

[7] J. Yang, N. Yi, H. Zhang, High-order, unconditionally maximum-principle preserving finite element method for the Allen–Cahn equation, Appl. Numer. Math. 188 (2023) 42–61.

[8] S. Ayub, H. Affan, A. Shah, Comparison of operator splitting schemes for the numerical solution of the Allen–Cahn equation, AIP Adv. 9 (12) (2019) 125202.

[9] Z. Weng, S. Zhai, W. Dai, Y. Yang, Y. Mo, Stability and error estimates of Strang splitting method for the nonlocal ternary conservative Allen–Cahn model, J. Comput. Appl. Math. 441 (2024) 115668.

[10] 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.

[11] 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.

[12] S. Zhai, D. Wang, Z. Weng, X. Zhao, Error analysis and numerical' simulations of Strang splitting method for space fractional nonlinear Schrödinger equation, J. Sci. Comput. 81 (2019) 965–989.

[13] S. Zhai, Z. Weng, Q. Zhuang, F. Liu, V. Anh, An effective operator splitting method based on spectral deferred correction for the fractional Gray–Scott model, J. Comput. Appl. Math. 425 (2023) 114959.

[14] M. Almushaira, H. Bhatt, A.M. Al-Rassas, Fast high-order method for multi-dimensional space-fractional reaction–diffusion equations with general boundary conditions, Math. Comput. Simulation 182 (2021) 235–258.

[15] Y.M. Wang, A high-order compact difference method on fitted meshes for Neumann problems of time-fractional reaction–diffusion equations with variable coefficients, Math. Comput. Simulation 181 (2021) 598–623.

[16] C. Wu, X. Feng, Y. He, L. Qian, A second-order Strang splitting scheme with exponential integrating factor for the Allen–Cahn equation with logarithmic Flory–Huggins potential, Commun. Nonlinear Sci. Numer. Simul. 117 (2023) 106983.

[17] Y. Li, H.G. Lee, B. Xia, J. Kim, A compact fourth-order finite difference scheme for the three-dimensional Cahn–Hilliard equation, Comput. Phys. Comm. 200 (2016) 108–116.

[18] Y. Li, J. Kim, An efficient and stable compact fourth-order finite difference scheme for the phase field crystal equation, Comput. Methods Appl. Math. 319 (2017) 194–216.

[19] H. Li, S. Xie, X. Zhang, A high order accurate bound-preserving compact finite difference scheme for scalar convection diffusion equations, SIAM J. Numer. Anal. 56 (2018) 3308–3345.

[20] J. Long, C. Luo, Q. Yu, Y. Li, An unconditional stable compact fourth-order finite difference scheme for three dimensional Allen–Cahn equation, Comput. Math. Appl. 77 (4) (2019) 1042–1054.

[21] S. Zhai, X. Feng, Y. He, Numerical simulation of the three dimensional Allen–Cahn equation by the high-order compact ADI method, Comput. Phys. Comm. 185 (2014) 449–2455.

[22] H. Zhang, J. Yan, X. Qian, X. Chen, S. Song, Explicit third-order unconditionally structure-preserving schemes for conservative Allen–Cahn equations, J. Sci. Comput. 90 (2022) 1–29.

[23] M. Rizwan, A. Shah, L. Yuan, A central compact scheme for numerical solution of two-phase incompressible flow using Allen–Cahn phase field model, J. Braz. Soc. Mech. Sci. Eng. 38 (2016) 433–441.

[24] H.G. Lee, J. Shin, J.Y. Lee, A high-order and unconditionally energy stable scheme for the conservative Allen–Cahn equation with a nonlocal Lagrange multiplier, J. Sci. Comput. 90 (2022) 1–12.

[25] 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.

[26] Y. Bo, D. Tian, X. Liu, Y. Jin, Discrete maximum principle and energy stability of the compact difference scheme for two-dimensional Allen–Cahn equation, J. Funct. space 2022 (2022) 8522231.

[27] J.J. Eggleston, G.B. McFadden, P.W. Voorhees, A phase-field model for highly anisotropic interfacial energy, Phys. D 150 (1–2) (2001) 91–103.

[28] 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.

[29] A. Stuart, A.R. Humphries, Dynamical Systems and Numerical Analysis, Vol. 2, Cambridge University Press, 1998.

[30] Y. Li, H.G. Lee, D. Jeong, J. Kim, An unconditionally stable hybrid numerical method for solving the Allen–Cahn equation, Comput. Math. Appl. 60 (6) (2010) 1591–1606.

[31] J.W. Thomas, Numerical Partial Differential Equations: Finite Difference Methods, Vol. 22, Springer Science & Business Media, 2013.

[32] 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.

[33] A.M. Wazwaz, The tanh-coth method for solitons and kink solutions for nonlinear parabolic equations, Appl. Math. Comput. 188 (2) (2007) 1467–1475.

[34] S. Zhai, Z. Weng, X. Feng, Y. He, Stability and error estimate of the operator splitting method for the phase field crystal equation, J. Sci. Comput. 86 (2021) 1–23.

[35] S. Zhai, Z. Weng, Y. Yang, A high order operator splitting method based on spectral deferred correction for the nonlocal viscous Cahn–Hilliard equation, J. Comput. Phys. 446 (2021) 110636.

[36] D. Jeong, J. Kim, A comparison study of ADI and operator splitting methods on option pricing models, J. Comput. Appl. Math. 247 (2013) 162–171.

[37] S. Lee, Non-iterative compact operator splitting scheme for Allen–Cahn equation, Comput. Appl. Math. 40 (2021) 1–9.

[38] 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 (2022) 1–10.

[39] D.S. Lee, J.S. Kim, Mean curvature flow by the Allen–Cahn equation, Eur. J. Appl. Math. 26 (4) (2015) 535–559.

[40] H.G. Lee, J.Y. Lee, A second order operator splitting method for Allen–Cahn type equations with nonlinear source terms, Phys. A 432 (2015) 24–34.