



Research article**An explicit fourth-order accurate compact method for the Allen-Cahn equation****Chaeyoung Lee¹, Seokjun Ham², Youngjin Hwang², Soobin Kwak² and Junseok Kim^{2,*}**¹ Department of Mathematics, Kyonggi University, Suwon 16227, Republic of Korea² Department of Mathematics, Korea University, Seoul, 02841, Republic of Korea

* Correspondence: Email: cfdkim@korea.ac.kr.

Abstract: In this paper, we propose an explicit spatially fourth-order accurate compact scheme for the Allen-Cahn equation in one-, two-, and three-dimensional spaces. The proposed method is based on the explicit Euler time integration scheme and fourth-order compact finite difference method. The proposed numerical solution algorithm is highly efficient and simple to implement because it is an explicit scheme. There is no need to solve implicitly a system of discrete equations as in the case of implicit numerical schemes. Furthermore, when we consider the temporally accurate numerical solutions, the time step restriction is not severe because the governing equation is a second-order parabolic partial differential equation. Computational tests are conducted to demonstrate the superior performance of the proposed spatially fourth-order accurate compact method for the Allen-Cahn equation.

Keywords: fourth-order accurate compact method; Allen-Cahn equation; explicit Euler method**Mathematics Subject Classification:** 65D25, 65N06

1. Introduction

In this article, we present an explicit spatially fourth-order accurate compact method for the Allen-Cahn (AC) equation [1,2] in one-, two-, and three-dimensional spaces. The AC equation is as follows:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} + \Delta\phi(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (1.1)$$

where the order parameter $\phi(\mathbf{x}, t)$ is the difference of the local concentrations of the two components in a domain Ω , $F(\phi) = 0.25(\phi^2 - 1)^2$ is a double well free energy density, and ϵ is a small parameter. The homogeneous Neumann boundary condition is assumed:

$$\frac{\partial \phi}{\partial \mathbf{n}} = 0 \quad \text{on } \partial\Omega,$$

where $\partial/\partial\mathbf{n}$ is the normal derivative on the boundary of domain. The AC equation (1.1) is L^2 -gradient flow of the Ginzburg–Landau functional:

$$\mathcal{E}(\phi) = \int_{\Omega} \left(\frac{F(\phi)}{\epsilon^2} + \frac{1}{2} |\nabla \phi|^2 \right) d\mathbf{x}. \quad (1.2)$$

Developing high-order accurate numerical schemes is important, therefore, it has been widely studied. In studying high-order accurate numerical schemes in space, wide stencils are required. In this case, we should eliminate the inefficiency caused by calculating high-order derivatives with respect to space, such as the Laplace operator. Therefore, the compact scheme has been actively developed for high-order accurate schemes with relatively less wide stencils.

Abide [3] presented finite difference preconditioning for compact scheme discretizations of the Poisson equation with variable coefficients. Li and Liao [4] developed an explicit high-order compact finite difference method (FDM) to solve three-dimensional (3D) acoustic wave equations with spatially variable acoustic velocity. In [5], based on FDM, a high-order accurate compact method for the heat equation was studied on the inverse Lax-Wendroff boundary treatment. In [6], the authors developed a highly accurate local one-dimensional (1D) explicit compact method for the acoustic wave equation in two-dimensional (2D) space. Patel and Mehra [7] studied the fourth-order compact method for space fractional advection-diffusion reaction equations using the alternating direction implicit scheme. An explicit fourth-order compact scheme was developed for heat transfer of boundary layer flow [8], which has the advantage of finding solutions of nonlinear and linear convection-diffusion type equations that cannot be solved with the existing implicit compact scheme. Qiu et al. [9] developed a general conservative eighth-order compact finite difference method to solve the coupled Schrödinger-KdV equations, this method is decoupled and preserves several physical invariants in a discrete sense. Elmahdi and Huang [10] presented linearized finite difference and fourth-order compact finite difference schemes for the time fractional nonlinear diffusion-wave equations based on their equivalent partial integro-differential equations. Abdi et al. [11] presented high-order compact finite difference schemes for the fractional Black-Scholes equation in the case of European option pricing. Using implicit-type schemes, spatially fourth-order accurate compact schemes were presented in [12–14]. There are some studies for solving partial differential equation (PDE), including the Laplace terms using a fourth-order accurate method on hexagonal grids [15–18]. Zhai et al. [12] presented a linearized, temporally second-order, spatially fourth-order accurate compact method for the 3D AC equations with different boundary conditions. Long et al. [13] described the unconditional stable linear FDM for the 3D AC equation, combined with backward differentiation and fourth-order compact schemes. Bo et al. [14] theoretically proved the discrete maximum principle and energy stability of the compact difference scheme for AC equation in the 2D space and presented numerical examples. In [19], the authors presented a fourth-order FDM for the AC equation with a stabilized term which has an effect on structure-preserving.

Moreover, the compact finite difference scheme has been applied for solving the Cahn-Hilliard (CH) equation [20, 21]:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \Delta \left(F'(\phi) - \epsilon^2 \Delta \phi \right), \quad (1.3)$$

which is a phase-field model, H^{-1} -gradient flow of the energy functional (1.2), modeling the phase separation of spinodal decomposition in a binary alloy. Li et al. [22] analyzed a three-level linearized

compact method for the CH equation as proving the unique solvability and unconditional convergence of the numerical solution. Ju et al. [23] proposed a fast and stabilized compact exponential time differencing multi-step method for the CH equation. Lee [24] presented a fourth-order spatial and second-order temporal accurate, unconditionally stable compact FDM for the CH equation. Lee and Shin [25] presented a compact scheme for solving the CH equation with a periodic boundary condition. In [26], a fourth-order compact method for the pure stream function formulation of 3D unsteady incompressible Navier-Stokes equation was presented using the Crank-Nicolson method for time discretization.

Many studies for the compact scheme have adopted implicit-type schemes to numerically solve various equations. Implicit-type schemes have the advantage that they can solve the equations efficiently by using large time-step sizes compared to explicit-type schemes. On the other hand, in the case of a nonlinear partial differential equation, we need to use efficient numerical solvers such as multigrid methods. In addition, highly accurate results require small time-step sizes. For example, when we solve the CH equation which is a fourth-order nonlinear PDE, an explicit method imposes a severe constraint on time-step size for stability. A very small time-step size makes it almost impossible to compute the fine-grid solution. In this case, the multigrid method is useful. However, in this study, we focus on the fourth-order accurate compact method for the AC equation, which is a second-order nonlinear parabolic PDE. When we solve the AC equation using an explicit scheme, the linear term restricts the time-step size as $\Delta t < 0.5h^2/d$, where h is a mesh size and d is the spatial dimension [27–29]. The constraint on the time-step size is not too severe, therefore, the fourth-order compact explicit scheme increases the accuracy without greatly losing efficiency.

The main purpose of this study is to present a fully explicit spatially fourth-order accurate compact scheme for the AC equation in 1D, 2D, and 3D spaces. To the authors' knowledge, this is the first study of a fully explicit spatially fourth-order accurate compact scheme for the AC equation. The advantages of this study focusing on space are the accuracy and efficiency of the numerical solution, and the simplicity of implementation. In the follow-up study, we will improve the proposed scheme to be higher-order accuracy in time.

The layout of this paper is as follows. In Section 2, it is presented that the description of the fourth-order compact FDM for the AC equation. In Section 3, we perform various computational experiments to demonstrate the basic properties and advantages of the proposed method. Conclusions are made in Section 4.

2. Numerical method

2.1. Discretization

Let N_x , N_y , and N_z be the number of grid points in the x , y , and z direction, respectively, $\Delta t = T/N_t$ be the time-step size, T be the final time, and N_t be the total number of time steps.

2.1.1. One-dimensional space

In a 1D domain $\Omega = (a_x, b_x)$, let $h = (b_x - a_x)/N_x$ be the uniform mesh size. A discrete numerical domain is denoted by $\Omega_h = \{x_i : x_i = a_x + (i - 0.5)h, 1 \leq i \leq N_x\}$. Let ϕ_i^n be the approximation of $\phi(x_i, n\Delta t)$. The homogeneous Neumann boundary conditions for ϕ are given as follows: $\phi_0 = \phi_1$, $\phi_{-1} =$

$\phi_2, \phi_{N_x+1} = \phi_{N_x}, \phi_{N_x+2} = \phi_{N_x-1}$. The discrete differentiation operator is $D_x\phi_{i+\frac{1}{2}} = (\phi_{i+1} - \phi_i)/h$.

2.1.2. Two-dimensional space

We discretize the AC equation in a 2D domain $\Omega = (a_x, b_x) \times (a_y, b_y)$. Let $h = (b_x - a_x)/N_x = (b_y - a_y)/N_y$ be the uniform mesh size. Let $\Omega_h = \{(x_i, y_j) : x_i = a_x + (i - 0.5)h, y_j = a_y + (j - 0.5)h, 1 \leq i \leq N_x, 1 \leq j \leq N_y\}$ be a discrete numerical domain. Let ϕ_{ij}^n be the approximation of $\phi(x_i, y_j, n\Delta t)$. The homogeneous Neumann boundary conditions for ϕ are given as follows:

$$\begin{aligned} & \text{for } 1 \leq j \leq N_y, \\ & \quad \phi_{0,j} = \phi_{1,j}, \phi_{-1,j} = \phi_{2,j}, \phi_{N_x+1,j} = \phi_{N_x,j}, \phi_{N_x+2,j} = \phi_{N_x-1,j}, \\ & \text{for } -1 \leq i \leq N_x + 2, \\ & \quad \phi_{i,0} = \phi_{i,1}, \phi_{i,-1} = \phi_{i,2}, \phi_{i,N_y+1} = \phi_{i,N_y}, \phi_{i,N_y+2} = \phi_{i,N_y-1}. \end{aligned}$$

The discrete differentiation, gradient, and divergence operators are

$$\begin{aligned} D_x\phi_{i+\frac{1}{2},j} &= \frac{1}{12} \frac{\phi_{i+1,j+1} - \phi_{i,j+1}}{h} + \frac{5}{6} \frac{\phi_{i+1,j} - \phi_{i,j}}{h} + \frac{1}{12} \frac{\phi_{i+1,j-1} - \phi_{i,j-1}}{h}, \\ D_y\phi_{i,j+\frac{1}{2}} &= \frac{1}{12} \frac{\phi_{i+1,j+1} - \phi_{i+1,j}}{h} + \frac{5}{6} \frac{\phi_{i,j+1} - \phi_{i,j}}{h} + \frac{1}{12} \frac{\phi_{i-1,j+1} - \phi_{i-1,j}}{h}, \end{aligned}$$

$\nabla_c\phi_{ij} = (D_x\phi_{i+\frac{1}{2},j}, D_y\phi_{i,j+\frac{1}{2}})$, and $\nabla_d \cdot (u, v)_{ij} = (u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j})/h + (v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}})/h$, respectively. The discrete l_2 -inner products and norms are defined as

$$(\phi, \psi)_h := h^2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \phi_{ij} \psi_{ij}, \quad (2.1)$$

$$(\nabla_c\phi, \nabla_c\psi)_e := h^2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} (D_x\phi_{i+\frac{1}{2},j} D_x\psi_{i+\frac{1}{2},j} + D_y\phi_{i,j+\frac{1}{2}} D_y\psi_{i,j+\frac{1}{2}}), \quad (2.2)$$

$$\|\phi\|^2 = (\phi, \phi)_h, \text{ and } \|\nabla\phi\|_e^2 = (\nabla_c\phi, \nabla_c\phi)_e. \quad (2.3)$$

We define the discrete total energy functional by

$$\mathcal{E}^h(\phi^n) = (F(\phi^n), \mathbf{1})_h + \frac{\epsilon^2}{2} \|\nabla\phi^n\|_e^2, \quad (2.4)$$

where $\mathbf{1}$ is a vector with all entries are 1.

2.1.3. Three-dimensional space

We discretize the AC equation in a 3D domain $\Omega = (a_x, b_x) \times (a_y, b_y) \times (a_z, b_z)$. Let $h = (b_x - a_x)/N_x = (b_y - a_y)/N_y = (b_z - a_z)/N_z$ and $\Omega_h = \{(x_i, y_j, z_k) : x_i = a_x + (i - 0.5)h, y_j = a_y + (j - 0.5)h, z_k = a_z + (k - 0.5)h, 1 \leq i \leq N_x, 1 \leq j \leq N_y, 1 \leq k \leq N_z\}$. Let ϕ_{ijk}^n be the approximation of $\phi(x_i, y_j, z_k, n\Delta t)$. The homogeneous Neumann boundary conditions for ϕ are given as follows:

$$\text{for } 1 \leq j \leq N_y, 1 \leq k \leq N_z,$$

$$\begin{aligned} \phi_{0jk} &= \phi_{1jk}, \quad \phi_{-1,jk} = \phi_{2jk}, \quad \phi_{N_x+1,jk} = \phi_{N_x,jk}, \quad \phi_{N_x+2,jk} = \phi_{N_x-1,jk}, \\ \text{for } -1 \leq i \leq N_x + 2, \quad 1 \leq k \leq N_z, \quad & \\ \phi_{i0k} &= \phi_{i1k}, \quad \phi_{i,-1,k} = \phi_{i2k}, \quad \phi_{i,N_y+1,k} = \phi_{i,N_y,k}, \quad \phi_{i,N_y+2,k} = \phi_{i,N_y-1,k}, \\ \text{for } -1 \leq i \leq N_x + 2, \quad -1 \leq j \leq N_y + 2, \quad & \\ \phi_{ij0} &= \phi_{ij1}, \quad \phi_{ij,-1} = \phi_{ij2}, \quad \phi_{ij,N_z+1} = \phi_{ijN_z}, \quad \phi_{ij,N_z+2} = \phi_{ij,N_z-1}. \end{aligned}$$

The discrete differentiation operators are

$$\begin{aligned} D_x \phi_{i+\frac{1}{2},jk} &= \left[128(\phi_{i+1,jk} - \phi_{ijk}) + 11(\phi_{i+1,j+1,k} + \phi_{i+1,j-1,k} + \phi_{i+1,j,k+1} \right. \\ &\quad \left. + \phi_{i+1,j,k-1} - \phi_{i,j+1,k} - \phi_{i,j-1,k} - \phi_{i,j,k+1} - \phi_{i,j,k-1}) \right. \\ &\quad \left. + 2(\phi_{i+1,j+1,k+1} + \phi_{i+1,j-1,k+1} + \phi_{i+1,j+1,k-1} + \phi_{i+1,j-1,k-1} \right. \\ &\quad \left. - \phi_{i,j+1,k+1} - \phi_{i,j-1,k+1} - \phi_{i,j+1,k-1} - \phi_{i,j-1,k-1}) \right] / (180h), \end{aligned}$$

and then $D_y \phi_{i,j+\frac{1}{2},k}$ and $D_z \phi_{i,j,k+\frac{1}{2}}$ are similarly defined above. We then define the discrete l_2 -inner products as

$$(\phi, \psi)_h := h^3 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \phi_{ijk} \psi_{ijk}, \quad (2.5)$$

$$\begin{aligned} (\nabla_c \phi, \nabla_c \psi)_e &:= h^3 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \sum_{k=1}^{N_z} \left(D_x \phi_{i+\frac{1}{2},jk} D_x \psi_{i+\frac{1}{2},jk} \right. \\ &\quad \left. + D_y \phi_{i,j+\frac{1}{2},k} D_y \psi_{i,j+\frac{1}{2},k} + D_z \phi_{i,j,k+\frac{1}{2}} D_z \psi_{i,j,k+\frac{1}{2}} \right). \end{aligned} \quad (2.6)$$

2.2. Compact finite difference scheme

2.2.1. One-dimensional space

In a 1D space, a compact Laplace operator Δ_c is defined as follows:

$$\Delta_c \phi_i = \Delta \phi_i = \frac{1}{h^2} (\phi_{i+1} - 2\phi_i + \phi_{i-1}). \quad (2.7)$$

By the Taylor expansion, we can obtain

$$\phi(x + \Delta x, t) = \sum_{k=0}^5 \frac{(\Delta x)^k}{k!} \left(\frac{\partial}{\partial x} \right)^k \phi(x, t) + O((\Delta x)^6). \quad (2.8)$$

Then, we can derive the following equation from Eq (2.8) by replacing Δx with $\pm h$.

$$\phi(x + h, t) - 2\phi(x, t) + \phi(x - h, t) = h^2 \phi_{xx} + \frac{h^4}{12} \phi_{xxxx} + O(h^6). \quad (2.9)$$

By dividing both sides of Eq (2.9) by h^2 , we obtain

$$\Delta_c \phi_i^n = \Delta \phi(x_i, t_n) + \frac{h^2}{12} \Delta^2 \phi(x_i, t_n) + O(h^4), \quad (2.10)$$

where $\Delta^2\phi = \Delta(\Delta\phi)$ is the biharmonic operator. Note that the standard fourth-order 5-point Laplace operator Δ_s is defined as

$$\Delta_s\phi_i^n = \frac{1}{12h^2}(-\phi_{i-2}^n + 16\phi_{i-1}^n - 30\phi_i^n + 16\phi_{i+1}^n - \phi_{i+2}^n).$$

From Eq (2.8), we can derive $\Delta_s\phi_i = \Delta\phi(x_i) + O(h^4)$ in the similar manner. Using Eq (2.10), we discretize the governing Eq (1.1) in a 1D domain as follows:

$$\begin{aligned} \frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} + O(\Delta t) &= -\frac{F'(\phi_i^n)}{\epsilon^2} + \Delta\phi(x_i, t_n) \\ &= -\frac{F'(\phi_i^n)}{\epsilon^2} + \Delta_c\phi_i^n - \frac{h^2}{12}\Delta^2\phi(x_i, t_n) + O(h^4) \\ &= -\frac{F'(\phi_i^n)}{\epsilon^2} + \Delta_c\phi_i^n - \frac{h^2}{12}\Delta\left(\frac{\partial\phi(x_i, t_n)}{\partial t} + \frac{F'(\phi(x_i, t_n))}{\epsilon^2}\right) + O(h^4) \\ &= -\frac{F'(\phi_i^n)}{\epsilon^2} + \Delta_c\phi_i^n - \frac{h^2}{12}\left(\frac{\partial\Delta\phi(x_i, t_n)}{\partial t} + \Delta\frac{F'(\phi(x_i, t_n))}{\epsilon^2}\right) + O(h^4) \\ &= -\frac{F'(\phi_i^n)}{\epsilon^2} + \Delta_c\phi_i^n - \frac{h^2}{12}\left(\frac{\partial\Delta_c\phi(x_i, t_n)}{\partial t} + \Delta_c\frac{F'(\phi(x_i, t_n))}{\epsilon^2} + O(h^2)\right) + O(h^4) \\ &= -\frac{F'(\phi_i^n)}{\epsilon^2} + \Delta_c\phi_i^n - \frac{h^2}{12}\left(\frac{\Delta_c\phi_i^n - \Delta_c\phi_i^{n-1}}{\Delta t} + O(\Delta t) + \Delta_c\frac{F'(\phi(x_i, t_n))}{\epsilon^2}\right) + O(h^4). \end{aligned} \quad (2.11)$$

Therefore, up to $O(\Delta t)$ and $O(h^4)$, we have

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = -\frac{F'(\phi_i^n)}{\epsilon^2} + \Delta_c\phi_i^n - \frac{h^2}{12}\left(\frac{\Delta_c\phi_i^n - \Delta_c\phi_i^{n-1}}{\Delta t} + \Delta_c\frac{F'(\phi_i^n)}{\epsilon^2}\right), \quad (2.12)$$

for $n \geq 2$. Given ϕ^0, ϕ^1 is computed by using the standard fourth-order Laplacian Δ_s .

2.2.2. Two-dimensional space

In a 2D space, the compact Laplace operator Δ_c [30] is defined as

$$\begin{aligned} \Delta_c\phi_{ij} &= \nabla_d \cdot \nabla_c \phi_{ij} = \frac{1}{6h^2} \left(\phi_{i-1,j+1} + 4\phi_{i,j+1} + \phi_{i+1,j+1} + 4\phi_{i-1,j} \right. \\ &\quad \left. - 20\phi_{ij} + 4\phi_{i+1,j} + \phi_{i-1,j-1} + 4\phi_{i,j-1} + \phi_{i+1,j-1} \right). \end{aligned}$$

By the Taylor series in two variables, we can obtain

$$\begin{aligned} \phi(x + \Delta x, y + \Delta y) &= \sum_{k=0}^5 \frac{1}{k!} \left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y} \right)^k \phi(x, y) \\ &\quad + O((\Delta x)^6 + (\Delta y)^6). \end{aligned}$$

By replacing Δx and Δy with different values $\pm h$, we get

$$\phi(x + h, y) + \phi(x - h, y) + \phi(x, y - h) + \phi(x, y + h)$$

$$= 4\phi + h^2\phi_{xx} + h^2\phi_{yy} + \frac{h^4}{12}\phi_{xxxx} + \frac{h^4}{12}\phi_{yyyy} + O(h^6), \quad (2.13)$$

$$\begin{aligned} & \phi(x-h, y-h) + \phi(x-h, y+h) + \phi(x+h, y-h) + \phi(x+h, y+h) \\ & = 4\phi + 2h^2\phi_{xx} + 2h^2\phi_{yy} + \frac{h^4}{6}\phi_{xxxx} + h^4\phi_{xxyy} + \frac{h^4}{6}\phi_{yyyy} + O(h^6). \end{aligned} \quad (2.14)$$

From Eqs (2.13) and (2.14), we have

$$\begin{aligned} & \phi(x-h, y-h) + \phi(x-h, y+h) + \phi(x+h, y-h) + \phi(x+h, y+h) \\ & + 4[\phi(x+h, y) + \phi(x-h, y) + \phi(x, y-h) + \phi(x, y+h)] - 20\phi(x, y) \\ & = 6h^2(\phi_{xx} + \phi_{yy})(x, y) + \frac{h^4}{2}(\phi_{xxxx} + 2\phi_{xxyy} + \phi_{yyyy})(x, y) + O(h^6). \end{aligned}$$

Finally, we have

$$\Delta_c\phi_{ij}^n = \Delta\phi(x_i, y_j, t_n) + \frac{h^2}{12}\Delta^2\phi(x_i, y_j, t_n) + O(h^4), \quad (2.15)$$

where $\Delta^2\phi = \Delta(\Delta\phi)$ is the biharmonic operator. Note that another standard fourth-order 9-point Laplace operator Δ_s is defined as

$$\begin{aligned} \Delta_s\phi_{ij}^n &= \frac{1}{12h^2}(-\phi_{i-2,j}^n + 16\phi_{i-1,j}^n - 30\phi_{ij}^n + 16\phi_{i+1,j}^n - \phi_{i+2,j}^n) \\ &\quad + \frac{1}{12h^2}(-\phi_{i,j-2}^n + 16\phi_{i,j-1}^n - 30\phi_{ij}^n + 16\phi_{i,j+1}^n - \phi_{i,j+2}^n). \end{aligned}$$

In a similar way, $\Delta_s\phi_{ij} = \Delta\phi(x_i, y_j) + O(h^4)$. Figure 1 shows the nodes used by the compact and standard Laplace operators at a point, respectively.

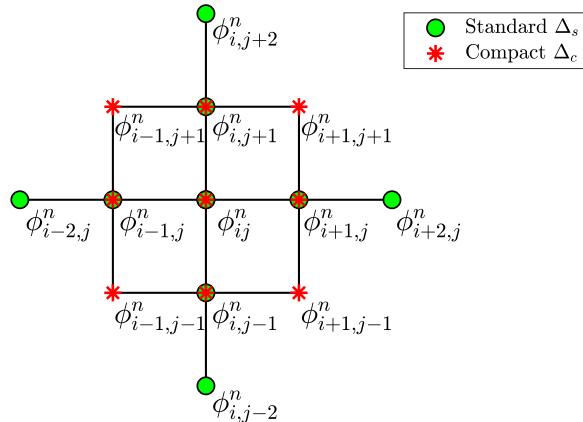


Figure 1. Schematic diagram of nodes for both the compact and standard Laplace operators at one point.

Using Eq (2.15), similar to Eq (2.11), we discretize the governing Eq (1.1). Therefore, up to $O(\Delta t)$ and $O(h^4)$, we have

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} = -\frac{F'(\phi_{ij}^n)}{\epsilon^2} + \Delta_c\phi_{ij}^n - \frac{h^2}{12} \left(\frac{\Delta_c\phi_{ij}^n - \Delta_c\phi_{ij}^{n-1}}{\Delta t} + \Delta_c \frac{F'(\phi_{ij}^n)}{\epsilon^2} \right), \quad (2.16)$$

for $n \geq 2$. Given ϕ^0, ϕ^1 is computed by using the standard fourth-order Laplacian Δ_s .

2.2.3. Three-dimensional space

In a 3D space, a compact Laplace operator Δ_c [31] is defined as

$$\begin{aligned}\Delta_c \phi_{ijk} &= \nabla_d \cdot \nabla_c \phi_{ijk} \\ &= \frac{1}{30h^2} \left[14(\phi_{i+1,jk} + \phi_{i-1,jk} + \phi_{i,j+1,k} + \phi_{i,j-1,k} + \phi_{ij,k+1} + \phi_{ij,k-1}) \right. \\ &\quad + 3(\phi_{i+1,j,k+1} + \phi_{i+1,j,k-1} + \phi_{i+1,j+1,k} + \phi_{i+1,j-1,k} + \phi_{i-1,j+1,k} \\ &\quad + \phi_{i-1,j-1,k} + \phi_{i,j+1,k+1} + \phi_{i,j-1,k-1} + \phi_{i-1,j,k+1} + \phi_{i-1,j,k-1} \\ &\quad + \phi_{i-1,j+1,k+1} + \phi_{i-1,j-1,k-1} + \phi_{i,j+1,k-1} + \phi_{i,j-1,k+1}) \\ &\quad + \phi_{i-1,j+1,k-1} + \phi_{i-1,j-1,k+1} + \phi_{i+1,j+1,k+1} + \phi_{i+1,j-1,k-1} \\ &\quad \left. + \phi_{i+1,j+1,k-1} + \phi_{i+1,j-1,k+1} - 128\phi_{ijk} \right].\end{aligned}$$

By the Taylor expansion in three variables, we can obtain

$$\begin{aligned}\phi(x + \Delta x, y + \Delta y, z + \Delta z, t) &= \sum_{k=0}^5 \frac{1}{k!} \left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y} + \Delta z \frac{\partial}{\partial z} \right)^k \phi(x, y, z, t) \\ &\quad + O((\Delta x)^6) + O((\Delta y)^6) + O((\Delta z)^6).\end{aligned}\tag{2.17}$$

Then, we can derive the following equations from Eq (2.17) by replacing Δx , Δy , and Δz with different values $\pm h$.

$$\begin{aligned}&\phi(x + h, y, z, t) + \phi(x - h, y, z, t) + \phi(x, y + h, z, t) \\ &\quad + \phi(x, y - h, z, t) + \phi(x, y, z + h, t) + \phi(x, y, z - h, t) \\ &= 6\phi + h^2 (\phi_{xx} + \phi_{yy} + \phi_{zz}) + \frac{h^4}{12} (\phi_{xxxx} + \phi_{yyyy} + \phi_{zzzz}) + O(h^6),\end{aligned}\tag{2.18}$$

$$\begin{aligned}&\phi(x - h, y - h, z, t) + \phi(x - h, y + h, z, t) + \phi(x + h, y - h, z, t) \\ &\quad + \phi(x + h, y + h, z, t) + \phi(x - h, y, z - h, t) + \phi(x - h, y, z + h, t) \\ &\quad + \phi(x + h, y, z - h, t) + \phi(x + h, y, z + h, t) + \phi(x, y - h, z - h, t) \\ &\quad + \phi(x, y - h, z + h, t) + \phi(x, y + h, z - h, t) + \phi(x, y + h, z + h, t) \\ &= 12\phi + 4h^2 (\phi_{xx} + \phi_{yy} + \phi_{zz}) + \frac{h^4}{3} (\phi_{xxxx} + \phi_{yyyy} + \phi_{zzzz}) \\ &\quad + h^4 (\phi_{xxyy} + \phi_{yyzz} + \phi_{xxzz}) + O(h^6),\end{aligned}\tag{2.19}$$

$$\begin{aligned}&\phi(x - h, y - h, z - h, t) + \phi(x - h, y - h, z + h, t) \\ &\quad + \phi(x - h, y + h, z - h, t) + \phi(x - h, y + h, z + h, t) \\ &\quad + \phi(x + h, y - h, z - h, t) + \phi(x + h, y - h, z + h, t) \\ &\quad + \phi(x + h, y + h, z - h, t) + \phi(x + h, y + h, z + h, t) \\ &= 8\phi + 4h^2 (\phi_{xx} + \phi_{yy} + \phi_{zz}) + \frac{h^4}{3} (\phi_{xxxx} + \phi_{yyyy} + \phi_{zzzz}) \\ &\quad + 2h^4 (\phi_{xxyy} + \phi_{yyzz} + \phi_{xxzz}) + O(h^6).\end{aligned}\tag{2.20}$$

Multiplying both sides of Eqs (2.18)–(2.20) by 14, 3, and 1, respectively, and adding the equations, we have

$$\begin{aligned} 30h^2\Delta_c\phi_{ijk} &= 30h^2(\phi_{xx} + \phi_{yy} + \phi_{zz}) + \frac{5}{2}h^4(\phi_{xxxx} + \phi_{yyyy} + \phi_{zzzz} \\ &\quad + 2\phi_{xxyy} + 2\phi_{yyzz} + 2\phi_{xxzz}) + O(h^6). \end{aligned} \quad (2.21)$$

By dividing both the sides of Eq (2.21) by $30h^2$, we obtain

$$\Delta_c\phi_{ijk}^n = \Delta\phi(x_i, y_j, z_k, t_n) + \frac{h^2}{12}\Delta^2\phi(x_i, y_j, z_k, t_n) + O(h^4). \quad (2.22)$$

Note that the 3D standard fourth-order 13-point Laplace operator Δ_s is defined as

$$\begin{aligned} \Delta_s\phi_{ijk}^n &= \frac{1}{12h^2}(-\phi_{i-2,jk}^n + 16\phi_{i-1,jk}^n - 30\phi_{ijk}^n + 16\phi_{i+1,j}^n - \phi_{i+2,j}^n) \\ &\quad + \frac{1}{12h^2}(-\phi_{i,j-2,k}^n + 16\phi_{i,j-1,k}^n - 30\phi_{ijk}^n + 16\phi_{i,j+1,k}^n - \phi_{i,j+2,k}^n) \\ &\quad + \frac{1}{12h^2}(-\phi_{ij,k-2}^n + 16\phi_{ij,k-1}^n - 30\phi_{ijk}^n + 16\phi_{ij,k+1}^n - \phi_{ij,k+2}^n). \end{aligned}$$

In a similar way, $\Delta_s\phi_{ijk} = \Delta\phi(x_i, y_j, z_k) + O(h^4)$. Figure 2 shows the grid point usages of standard Δ_h and compact Δ_c in a 3D space.

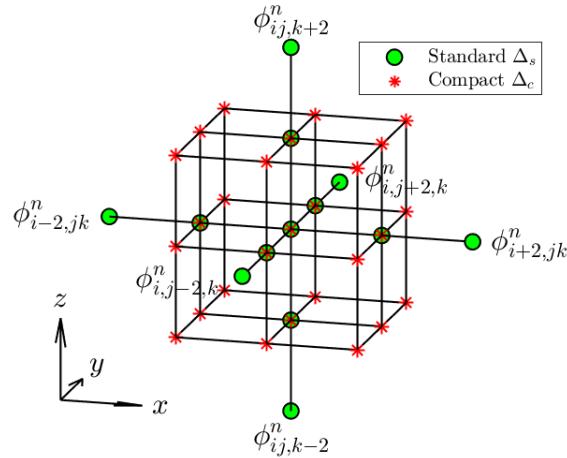


Figure 2. Schematic diagram of nodes for both the compact and standard Laplace operators at one point in a 3D space.

Using Eq (2.15), we discretize the governing Eq (1.1). Therefore, up to $O(\Delta t)$ and $O(h^4)$, we have

$$\begin{aligned} &\frac{\phi_{ijk}^{n+1} - \phi_{ijk}^n}{\Delta t} \\ &= -\frac{F'(\phi_{ijk}^n)}{\epsilon^2} + \Delta_c\phi_{ijk}^n - \frac{h^2}{12}\left(\frac{\Delta_c\phi_{ijk}^n - \Delta_c\phi_{ijk}^{n-1}}{\Delta t} + \Delta_c\frac{F'(\phi_{ijk}^n)}{\epsilon}\right), \end{aligned} \quad (2.23)$$

for $n \geq 2$. Given ϕ^0, ϕ^1 is computed by using the standard fourth-order Laplacian Δ_s .

3. Numerical results

We demonstrate the representative properties of the explicit compact AC equation: phase separation, fourth-order accuracy in space, non-increase of total energy, and motion by mean curvature. We show the efficiency of our method using numerical comparison and measuring computational time. In addition, we perform the computational experiments for a non-convex initial condition with and without an obstacle to highlight the advantage of the proposed scheme. It can easily handle the computation on arbitrary complex domains.

3.1. Phase separation

We consider one of the most basic computational simulations, that is, the dynamics of phase separation. First, we conduct the numerical experiment in a 1D computational domain $\Omega = (0, 1)$ with the initial condition given as

$$\phi_i^0 = 0.1\text{rand}, \quad \text{for } 1 \leq i \leq N_x,$$

where rand is a random number in $[-1, 1]$. To observe the equilibrium profile, we cease the iterative computation when the following criterion is met:

$$\|\phi^n - \phi^{n-1}\|_2 < 1.0e-5, \quad (3.1)$$

where n is a positive integer, and ϕ^n is called a numerical equilibrium solution. The parameters are taken as $N_x = 100$, $h = 1/N_x$, $\Delta t = 0.1h^2$, and $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$. Figure 3 indicates the initial condition ($t = 0$), numerical solution ($t = 30\Delta t$), and numerical equilibrium solution ($t = 2242\Delta t$). From this result, we can also numerically confirm that the numerical solutions are bounded between -1 and 1 .

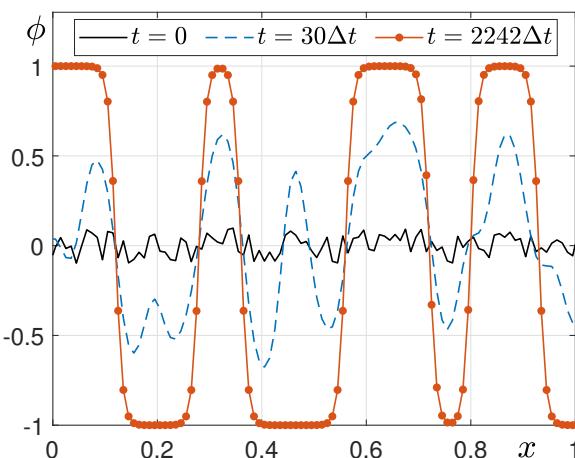


Figure 3. Temporal evolution of the initial and numerical solutions which are bounded between the interval $[-1, 1]$. The numerical solution at $t = 2242\Delta t$ is an equilibrium solution.

We next observe the phase separation in a 2D domain $\Omega = (0, 1)^2$. The initial condition is given as

$$\phi_{ij}^0 = 0.1\text{rand}, \quad \text{for } 1 \leq i \leq N_x, \quad 1 \leq j \leq N_y. \quad (3.2)$$

The parameters are taken as $N_x = N_y = 100$, $h = 1/N_x$, $\Delta t = 0.1h^2$, and $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$. Figure 4 indicates the initial condition ($t = 0$) and numerical solutions ($t = 30\Delta t$, $100\Delta t$). From this result, we can also numerically confirm that the numerical solutions are bounded between -1 and 1 .

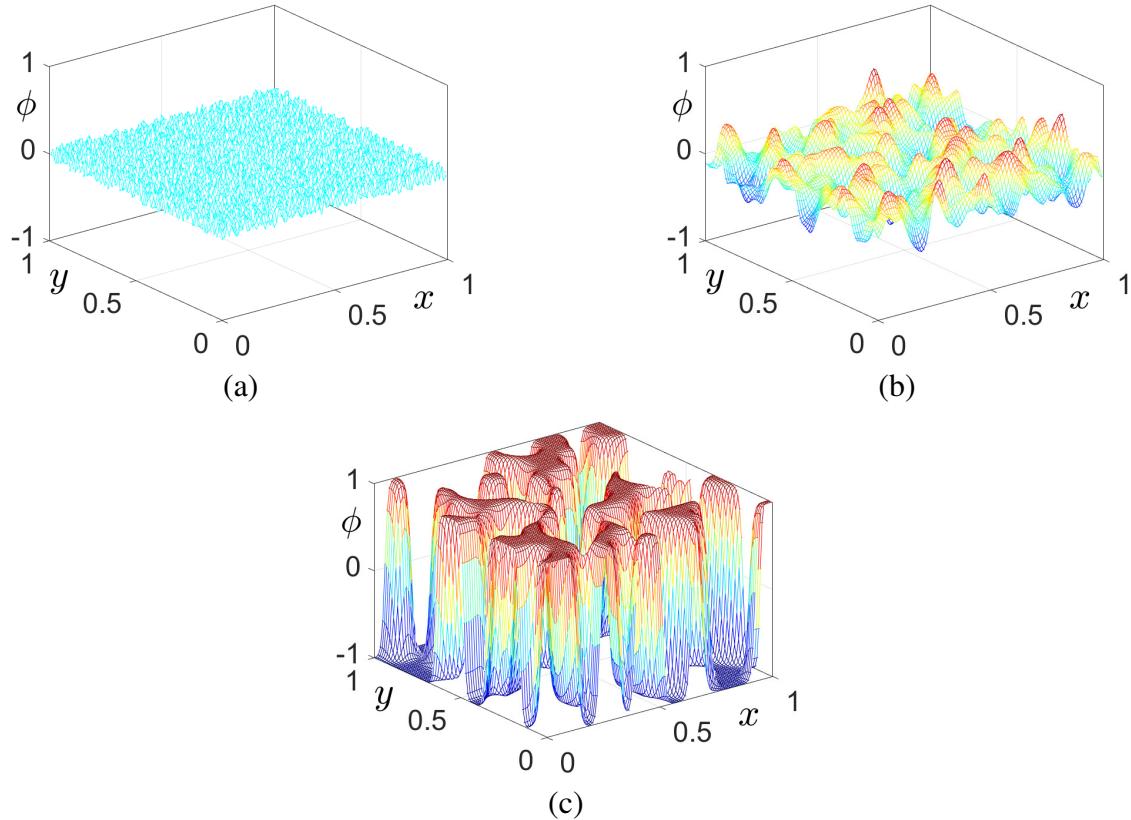


Figure 4. (a)–(c) are snapshots of 2D initial and numerical solutions at $t = 0$, $30\Delta t$, and $100\Delta t$, respectively.

3.2. Convergence test

We investigate a quantitative estimate of the rate of convergence. Note that the discrete l_2 -norms of numerical solutions is defined as

$$\|\phi^n\|_2 = \sqrt{\frac{1}{N_x} \sum_{i=1}^{N_x} (\phi_i^n)^2}, \quad \|\phi^n\|_2 = \sqrt{\frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} (\phi_{ij}^n)^2},$$

in 1D and 2D domains, respectively. First, in a 1D domain $\Omega = (0, 2)$, we consider the traveling wave solution of the AC equation [32, 33]:

$$\phi(x, t) = \frac{1}{2} \left[1 - \tanh \left(\frac{x - 0.7 - st}{2\sqrt{2}\epsilon} \right) \right], \quad (3.3)$$

where $s = 3\epsilon/\sqrt{2}$. At $t = 0$, the initial profile is

$$\phi(x, 0) = \frac{1}{2} \left[1 - \tanh \left(\frac{x - 0.7}{2\sqrt{2}\epsilon} \right) \right]. \quad (3.4)$$

Figure 5 shows the initial, numerical, and analytic profiles. Here, the parameters used are $N_x = 100$, $h = 2/N_x$, $\epsilon = 8h/(2\sqrt{2}\tanh^{-1}(0.9))$, $\Delta t = 0.1h^2$, and the final $T = 120\Delta t$.

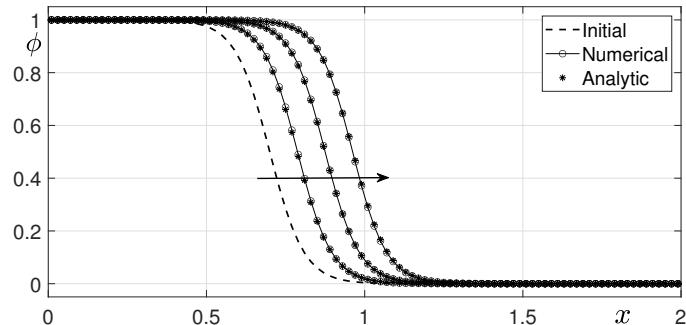


Figure 5. Numerical and analytic traveling wave solutions with the initial profile in a 1D domain.

We perform a convergence test with the initial condition (3.4) on a set of increasingly finer mesh grid sizes. The computational solutions are computed on the uniform grids and time steps, $h = 2/N_x$ for $N_x = 40$, 80 , and 160 . The fixed time-step size is $\Delta t = 5.0e-9$, the final time is $T = 50000\Delta t$, and $\epsilon = 0.025$. The convergence rate is defined as the ratio of successive errors, that is, $\log_2(\|e_{N_x}\|_2/\|e_{2N_x}\|_2)$, where the discrete l_2 -norm error is defined as $\|e_{N_x}\|_2 = \sqrt{\frac{1}{N_x} \sum_{i=1}^{N_x} (\phi(x_i, n\Delta t) - \phi_i^n)^2}$ and $\|e_{2N_x}\|_2 = \sqrt{\frac{1}{2N_x} \sum_{i=1}^{2N_x} (\phi(x_i, n\Delta t) - \phi_i^n)^2}$. Table 1 lists the errors and rates of convergence and demonstrates that the proposed scheme is spatially fourth-order accurate.

Table 1. Convergence results in a 1D space.

h	0.5000	0.0250	0.0125
l_2 -error	1.6363e-4	9.8470e-6	6.2480e-7
Rate	4.05	3.98	

In a 2D domain $\Omega = (0, 2)^2$, we consider the following solution:

$$\phi(x, y, t) = \frac{1}{2} \left[1 - \tanh \left(\frac{0.8x + 0.6y - 0.88 - st}{\sqrt{2}\epsilon} \right) \right]. \quad (3.5)$$

The parameters are taken as $N = N_x = N_y = 100$, $h = 2/N$, $\Delta t = 0.1h^2$, $\epsilon = 8h/(2\sqrt{2}\tanh^{-1}(0.9))$, and the final time $T = 90\Delta t$. In this test, the Dirichlet boundary condition is used and the values of order parameter on the boundary are set as the analytic solutions. Figures 6(a) and 6(b) show the initial profile at $t = 0$ and the numerical solution at $t = 90\Delta t$, respectively. Figure 6(c) shows the contours of the initial, numerical, and analytic solutions at the $\phi = 0.5$ level.

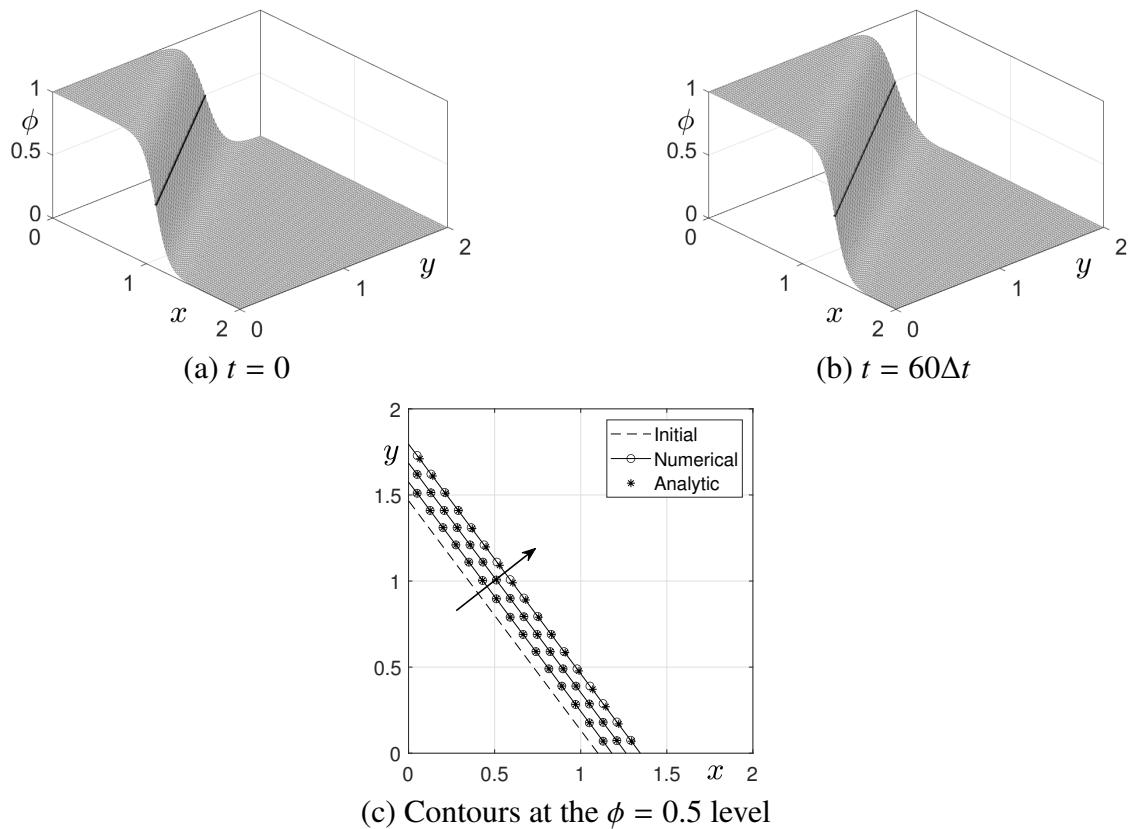


Figure 6. (a) and (b) are 2D mesh plots of the initial condition and the numerical solution at $t = 60\Delta t$, respectively. (c) is the contours of the numerical and analytic solutions with the initial condition at the $\phi = 0.5$ level.

We conduct a convergence test with the initial condition (3.5) on a set of increasingly finer mesh grid sizes. The computational solutions are computed using $h = 2/N$ for $N = 40, 80, 160$, the fixed time-step size $\Delta t = 1.0e-9$, the final time $T = 250000\Delta t$, and $\epsilon = 0.025$. Table 2 lists the errors and rates of convergence. Here, the discrete l_2 -norm error in a 2D domain is defined as $\|e_N\|_2 = \sqrt{\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N (\phi(x_i, y_j, n\Delta t) - \phi_{ij}^n)^2}$. The results in the 2D domain suggest that the scheme is indeed fourth-order accurate in space.

Table 2. Convergence results in a 2D space.

h	0.5000	0.0250	0.0125
l_2 -error	9.3779e-5	6.2659e-6	3.9393e-7
Rate	3.90	3.99	

3.3. Equilibrium profile

We verify that a numerical equilibrium solution in which the profile is a hyperbolic tangent is consistent with a theoretical solution when the time is large enough. The initial condition on a domain

$\Omega = (-1, 1)$ is

$$\phi(x, 0) = 0.5 \tanh\left(\frac{x}{\sqrt{2}\epsilon}\right). \quad (3.6)$$

The parameters are taken as $N_x = 200$, $h = 2/N_x$, $\Delta t = 0.1h^2$, $t = 5$, and $\epsilon = 4h/(2\sqrt{2} \tanh^{-1}(0.9))$. In Figure 7, the numerical equilibrium profile agrees well with the analytic one

$$\phi(x, t) = \tanh\left(\frac{x}{\sqrt{2}\epsilon}\right).$$

Here, the l_2 -norm error between two equilibrium profiles is less than 1.0e-16.

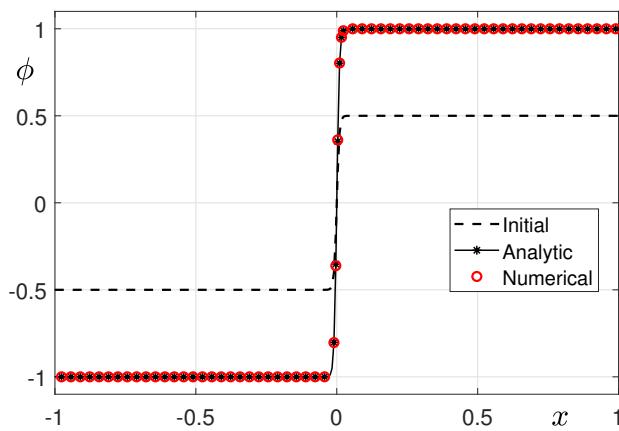


Figure 7. Analytic and numerical equilibrium solutions are in good agreement, which are denoted by a black line with stars and red circles. Here, an initial condition is $\phi(x, 0) = 0.5 \tanh(x/\sqrt{2}\epsilon)$ on a domain $\Omega = (-1, 1)$ with a black dashed line.

3.4. Total energy decrease

We numerically validate that the total energy functional (1.2) decreases over time. In a 2D domain, the discrete total energy is defined as

$$\mathcal{E}^h(\phi^n) = \frac{h^2}{\epsilon^2} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} F(\phi_{ij}^n) + \frac{h^2}{2} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |\nabla_c \phi_{ij}^n|^2.$$

The initial condition is defined as Eq (3.2) on $\Omega = (0, 1)^2$. The parameters are used as follows: $N = 100$, $h = 1/N$, $\Delta t = 0.1h^2$, $T = 2000\Delta t$, and $\epsilon = 4h/(2\sqrt{2} \tanh^{-1}(0.9))$. Figure 8 shows the time evolution of the discrete total energy and numerical solution. The normalized discrete total energy $\mathcal{E}(\phi^n)/\mathcal{E}(\phi^0)$ is non-increasing over time t . The snapshots of the numerical solutions are at $t = 0, 0.002, 0.006$, and 0.02 , respectively.

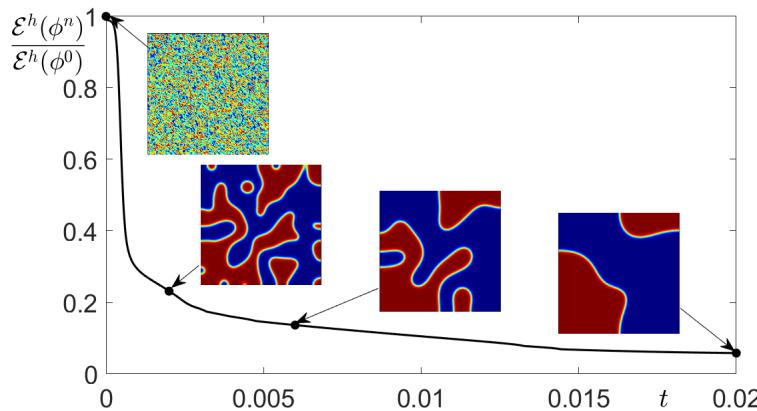


Figure 8. Discrete total energy dissipation over time t .

3.5. Efficiency

In this section, we verify the efficiency of the proposed method. First, we present numerical comparisons between the proposed scheme and previous work [13], and compare the performance using their computational time. In [13], the authors used the linear multigrid method to solve the AC equation using the fourth-order compact FDM. One V-cycle in the linear multigrid procedure consists of three parts: a pre-smoothing, a coarse grid correction, and a post-smoothing [34]. For the sake of convenience, we define the cost of performing one smoothing sweep on the finest grid as a work unit (WU). We assume that each number of pre- and post-smoothing sweeps is equal to the positive integer ν larger than 1. The smoothing steps mainly adopt the Gauss-Seidel type method, however, we apply the explicit method whose computational cost is less than that. The computational cost of the proposed method is estimated to be less than 2 WU, whereas that of the multigrid method in a 3D domain is

$$2\nu\left(1 + 2^{-3} + 2^{-6} + 2^{-9} + \dots + 2^{-3N}\right) < \frac{2}{1 - 2^{-3}}\nu\text{WU} = \frac{16\nu}{7}\text{WU},$$

where $N = N_x = N_y = N_z$. In general, one V-cycle is not enough even though it depends on the given tolerance [35, 36]. Hence, the computational cost of our method is estimated to be less than that of the multigrid method.

We use the initial condition as $\phi(x, y, z, 0) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$ on a domain $\Omega = (0, 1)^3$, the mesh size $N = 32, 64, 128, 256$, and the other parameters as $h = 1/N$, $\Delta t = 0.1h^2$, $T = 10\Delta t$, and $\epsilon = 0.1$. Our computation is performed using MATLAB 2022b on an Intel Core i9-12900H CPU @ 2.50 GHz with 32 GB of RAM. We measure total CPU time (sec) for 10 time iterations and calculate it divided by the number of time iterations. Table 3 lists the average CPU times. Our method is more than 10 times faster than the implicit-type method.

Table 3. Average CPU time (sec) according to mesh sizes N^3 . Here, the average CPU time is defined as the total CPU time over all time iterations.

Mesh size	32^3	64^3	128^3	256^3
Long et al. [13]	0.0373	0.2766	2.2953	17.8828
Ours	0.0029	0.0201	0.1674	1.3946

3.6. Motion by mean curvature

Motion by mean curvature, the representative property of the AC equation, means that the interface $\Gamma(t)$ of the solution evolves in the direction of the normal velocity proportional to the magnitude of the mean curvature [37]. The interface $\Gamma(t)$ is defined by the zero-level set of the solution ϕ , i.e., $\Gamma(t) = \{\mathbf{x} \in \Omega \mid \phi(\mathbf{x}, t) = 0\}$. The normal velocity V of $\Gamma(t)$ at each point \mathbf{x} is

$$V(\mathbf{x}, t) = -(\kappa_1 + \kappa_2) = -\left(\frac{1}{R_1} + \frac{1}{R_2}\right),$$

where κ_1 and κ_2 are the principal curvatures of the interface, and R_1 and R_2 are the principal radii. In this section, we only consider the case where the two principal radii are the same, therefore, we set the radius as $R(t)$. In a 2D space, there is one principal radius, therefore, $V = -1/R(t)$. In a 3D space, $V = -2/R(t)$. Thus, for the spatial dimension d ,

$$V = \frac{1-d}{R(t)}. \quad (3.7)$$

Because the normal velocity V is the change in radius with time t ,

$$V = \frac{dR(t)}{dt}. \quad (3.8)$$

From Eqs (3.7) and (3.8) and the initial radius $R_0 = R(0)$, the analytic radius is

$$R(t) = \sqrt{R_0^2 + 2(1-d)t}.$$

For the details, refer to the references [37, 38]. We demonstrate the motion by mean curvature in 2D and 3D domains. First, we perform a 2D simulation with the initial condition

$$\phi(x, y, 0) = \tanh\left(\frac{R_0 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon}\right),$$

in which the shape of zero-level contour is a circle with center $(0.5, 0.5)$ and radius $R_0 = 0.35$, on $\Omega = (0, 1)^2$. The parameters used are $N = 100$, $h = 1/N$, $\Delta t = 0.1h^2$, $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$, and $T = 6000\Delta t$. Figure 9(a) shows that the given circle shrinks in the direction of the normal vector according to motion by mean curvature. The zero-level contours are drawn every $600\Delta t$ from $t = 0$ to $t = 6000\Delta t$. From the contour spacing, it can be observed that the circle shrinks faster as the radius of the circle becomes smaller. Figure 9(b) indicates that the analytic and numerical radii agree with each other. Figure 9(c)–9(e) show the snapshots of the computational solution at time $t = 0$, $3600\Delta t$, and $6000\Delta t$, respectively.

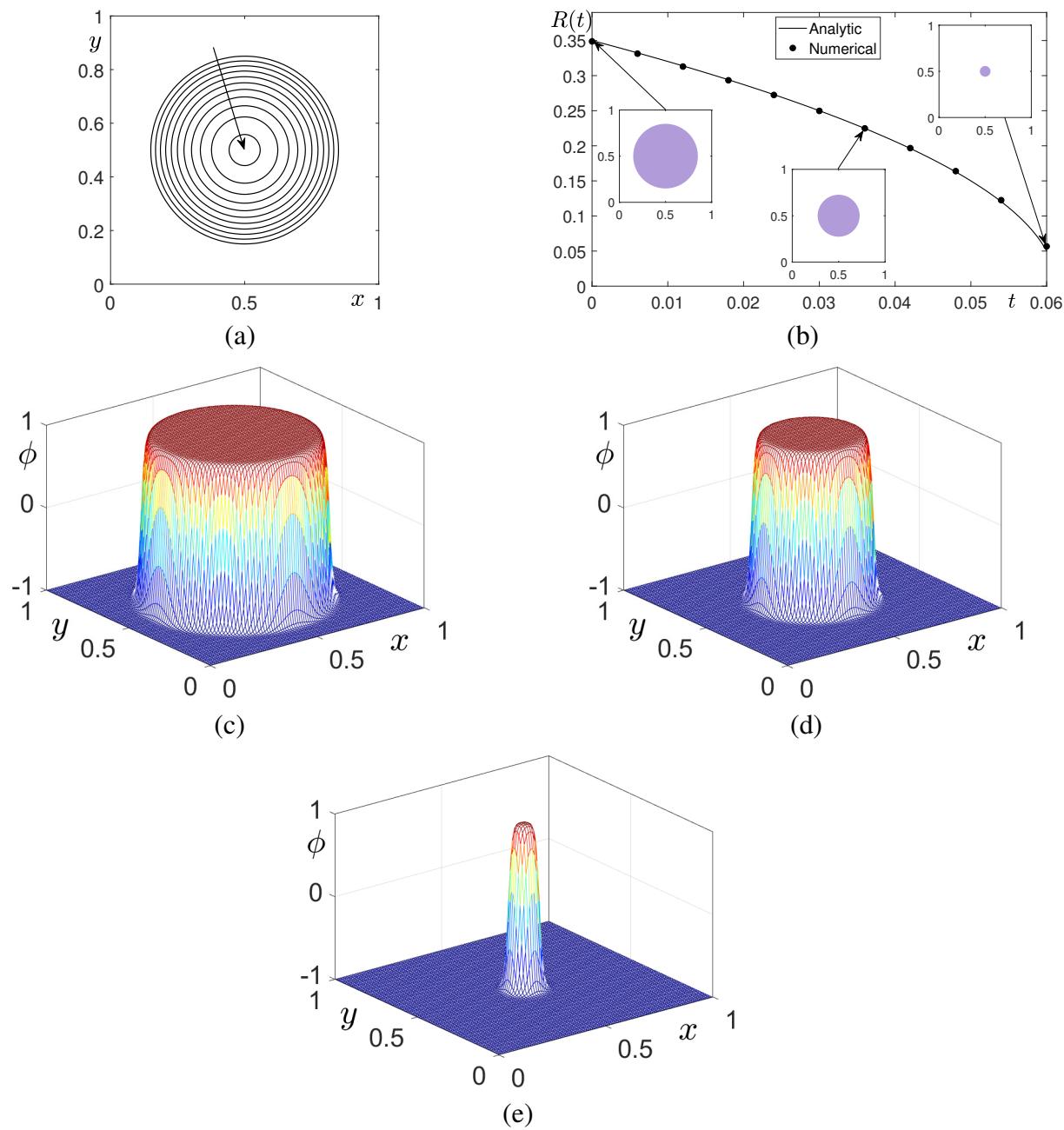


Figure 9. (a) With a circle initial condition, contours at the zero-level set of numerical solution ϕ every $600\Delta t$ from $t = 0$ to $t = 6000\Delta t$. (b) Analytic and numerical radii $R(t)$ decrease over time t . (c)–(e) Mesh plots of the numerical solution at $t = 0$, $3600\Delta t$, and $6000\Delta t$.

We consider three circles with different centers and radii as shown in Figure 10(a). The initial condition is

$$\phi(x, y, 0) = \tanh\left(\frac{0.27 - \sqrt{(x - 0.35)^2 + (y - 0.65)^2}}{\sqrt{2}\epsilon}\right)$$

$$\begin{aligned}
& + \tanh\left(\frac{0.17 - \sqrt{(x-0.72)^2 + (y-0.25)^2}}{\sqrt{2}\epsilon}\right) \\
& + \tanh\left(\frac{0.1 - \sqrt{(x-0.8)^2 + (y-0.55)^2}}{\sqrt{2}\epsilon}\right) + 2.
\end{aligned}$$

Here, the parameters, N , h , ϵ , Δt , are used as described above. Figures 10(b) and 10(c) are the snapshots of the computational solutions at $t = 1200\Delta t$ and $t = 3000\Delta t$, respectively. Figure 10(d) shows the zero-level contours every $300\Delta t$ from $t = 0$ to $t = 3000\Delta t$. The interface of the computational solution shrinks with the different velocities depending on the mean curvature of each circle. In other words, because the curvature of a circle with a smaller radius is large, the smaller-radius circle disappears first, and the larger-radius circle shrinks slowly.

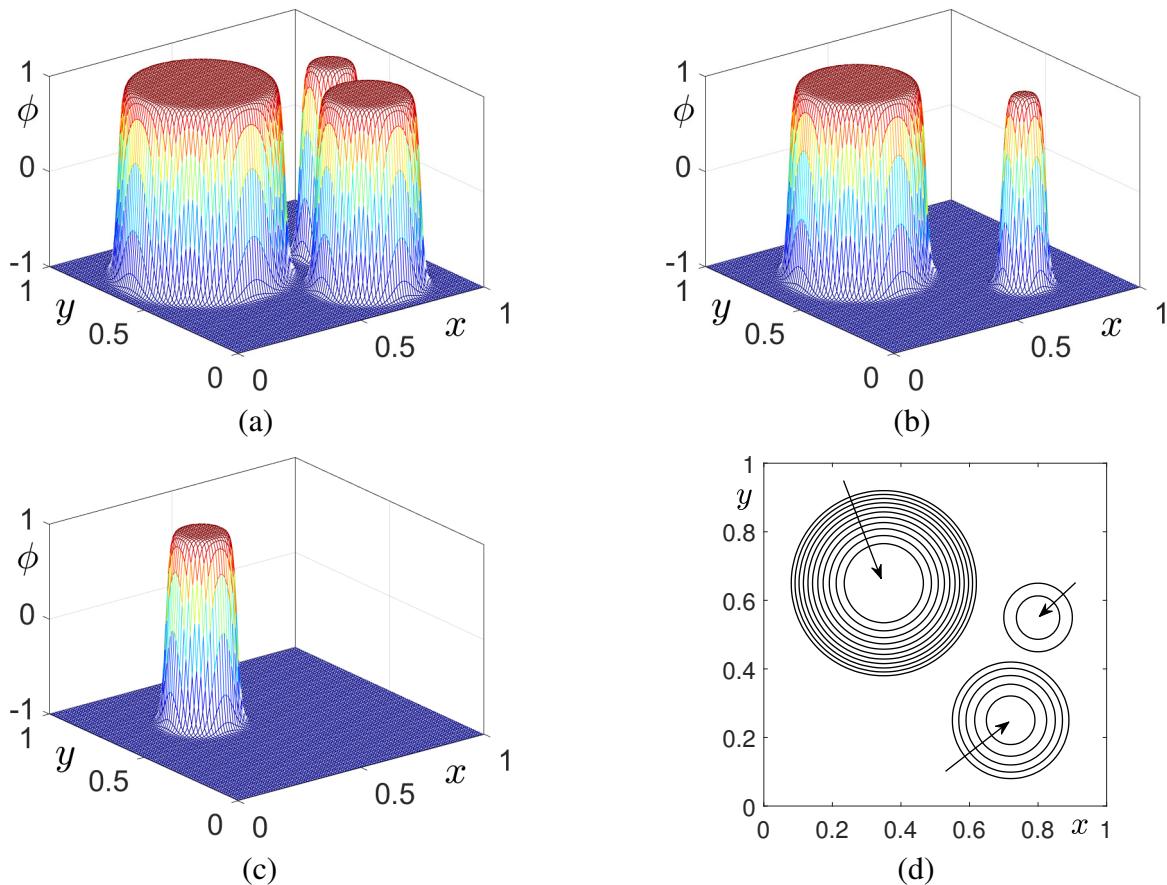


Figure 10. When the initial condition is defined as three different circles, (a)–(c) are mesh plots of the numerical solution at $t = 0$, $1200\Delta t$, and $3000\Delta t$, and (d) is the zero-level contours every $300\Delta t$ from $t = 0$ to $t = 3000\Delta t$.

Figure 11 shows the snapshots of the initial and numerical solution with the torus-type initial condition:

$$\phi(x, y, 0) = \tanh\left(\frac{0.4 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon}\right)$$

$$-\tanh\left(\frac{0.3 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon}\right) - 1,$$

on $\Omega = (0, 1)^2$. Here, the parameters, N , h , ϵ , Δt , are used as described above. In Figure 11, the top and bottom rows show mesh plots and zero-level contours of the solution, respectively, where the time t is 0, $3000\Delta t$, and $5000\Delta t$. It is observed that the radius of the inner contour is smaller than that of the outer contour as shown in Figure 11. Hence, with motion by mean curvature, the inner contour decreases faster than the outer contour.

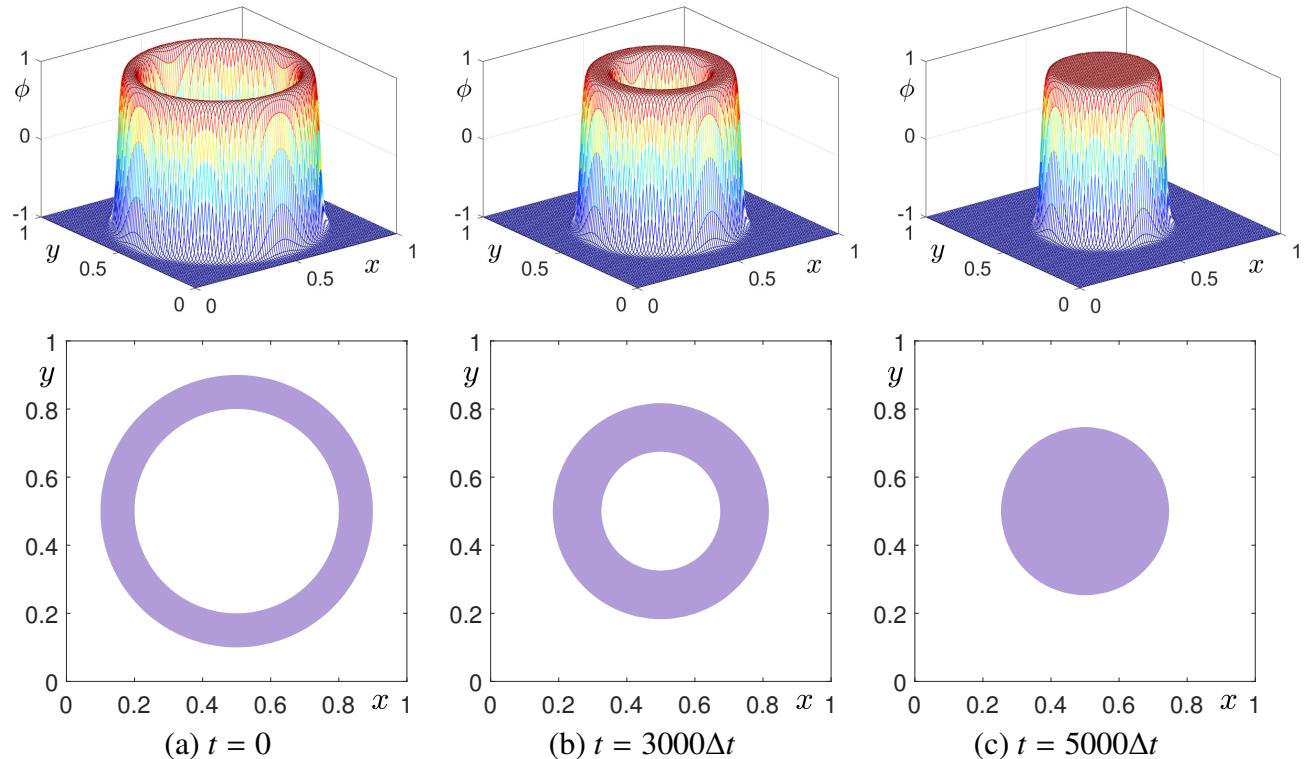


Figure 11. When the torus-type initial condition is given, (a)–(c) are snapshots of the numerical solution and zero-level set at time $t = 0$, $3000\Delta t$, and $5000\Delta t$, respectively.

In addition, we observe the results of 2D simulation with the star-shaped initial condition on $\Omega = (0, 1)^2$ as shown in Figure 12(a). We use $N = 150$, $h = 1/N$, $\Delta t = 0.1h^2$, and $\epsilon = 8h/(2\sqrt{2}\tanh^{-1}(0.9))$. Even though an initially complex shape is given, the part with the largest mean curvature is smoothed first, and then becomes circular. Figure 12(b) and 12(c) show mesh and contour plots of the computational solution at $t = 0$, $100\Delta t$, and $3000\Delta t$.

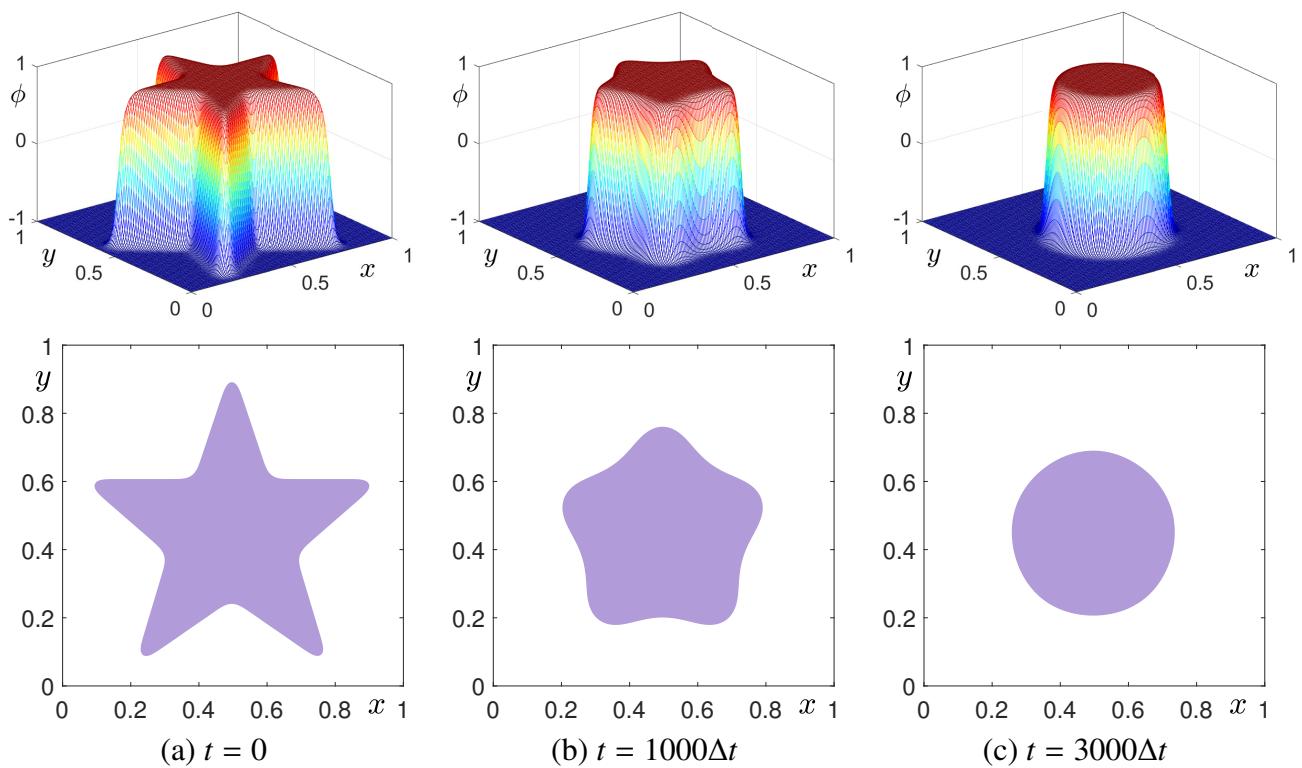


Figure 12. When the star-shaped initial condition is given, (a)–(c) are snapshots of the computational solution and zero-level set at $t = 0$, $1000\Delta t$, and $3000\Delta t$, respectively.

Next, we consider several computational simulations in a 3D domain to demonstrate motion by mean curvature. First of all, we verify the decreasing trend in the radius of a sphere. We conduct the 3D simulation with the following initial condition:

$$\phi(x, y, z, 0) = \tanh\left(\frac{R_0 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2}}{\sqrt{2}\epsilon}\right),$$

i.e., a sphere with center $(0.5, 0.5, 0.5)$ and radius $R_0 = 0.35$, on the computational domain $\Omega = (0, 1)^3$. The parameters used are $N = N_x = N_y = N_z = 80$, $h = 1/N$, $\Delta t = 0.1h^2$, and $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$. Figure 13 shows that the given sphere shrinks in the direction of the normal vector according to motion by mean curvature.

Figure 14 shows the temporal evolution of a torus in a domain $\Omega = (0, 1) \times (0, 1) \times (0, 0.5)$. The initial torus with center $(0.5, 0.5, 0.25)$, large radius $R_1 = 0.25$, and small radius $R_2 = 0.12$ is defined as:

$$\phi(x, y, z, 0) = \tanh\left(\frac{R_2 - \sqrt{(R_1 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2})^2 + (z - 0.25)^2}}{\sqrt{2}\epsilon}\right).$$

The parameters are used as $N_x = N_y = 100$, $N_z = 50$, $h = 0.01$, $\epsilon = 8h/(2\sqrt{2}\tanh^{-1}(0.9))$, $\Delta t = 0.1h^2$, and $T = 700\Delta t$.

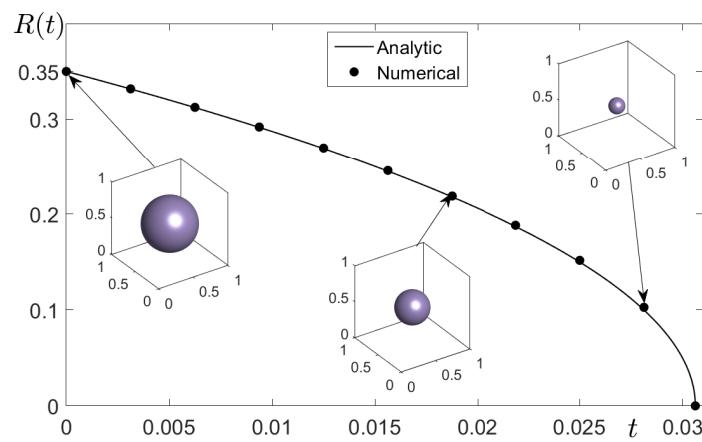


Figure 13. With a sphere initial condition, the analytic and numerical radii $R(t)$ decrease over time t . The dot plots are displayed every $200\Delta t$ from $t = 0$.

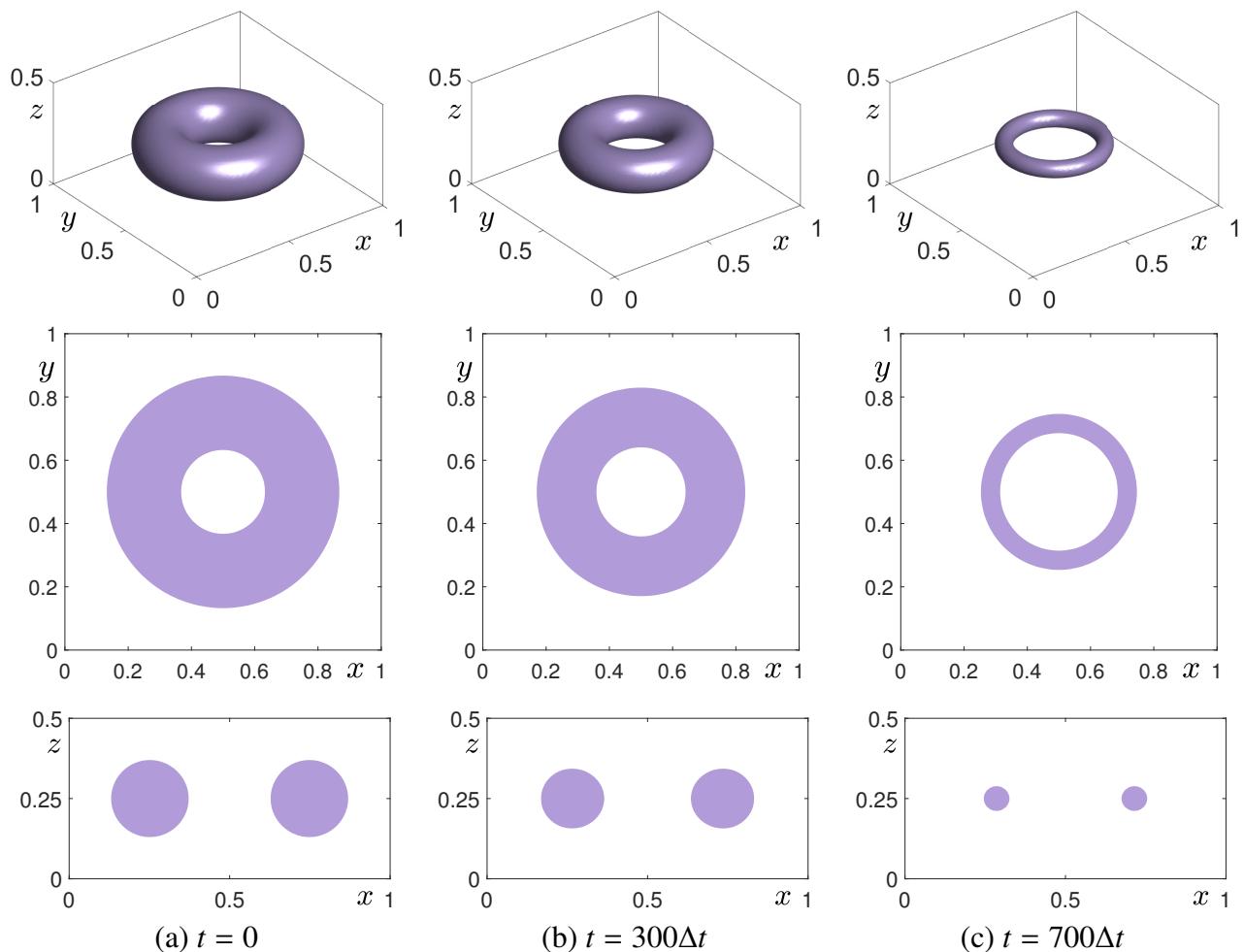


Figure 14. With a torus initial condition, (a)–(c) are snapshots of the numerical solution at $t = 0$, $300\Delta t$, and $700\Delta t$, respectively. The top row is the zero-level isosurface. The second and bottom rows are contours of horizontal ($z = 0.25$) and vertical ($y = 0.5$) sections of the top figures, respectively.

On a computational domain $\Omega = (-1, 1)^3$, we consider a sphere of center $(0, 0, 0)$ and radius R_0 perturbed by a spherical harmonic $Y_{l,m}$ [39,40]:

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

where A denotes amplitude, θ is the polar angle (measured from z -axis), and φ is the azimuthal angle (measured counterclockwise from the x -axis to the y -axis). Here, $l = 10$, $m = 7$, $R_0 = 0.7$, and $A = 0.7$. The parameters are used as $N = 100$, $h = 2/N$, $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$, $\Delta t = 0.1h^2$, and $T = 1000\Delta t$. Figure 15 shows the snapshots at $t = 0$, $200\Delta t$, and $1000\Delta t$. From the results, the temporal evolution works according to motion by mean curvature.

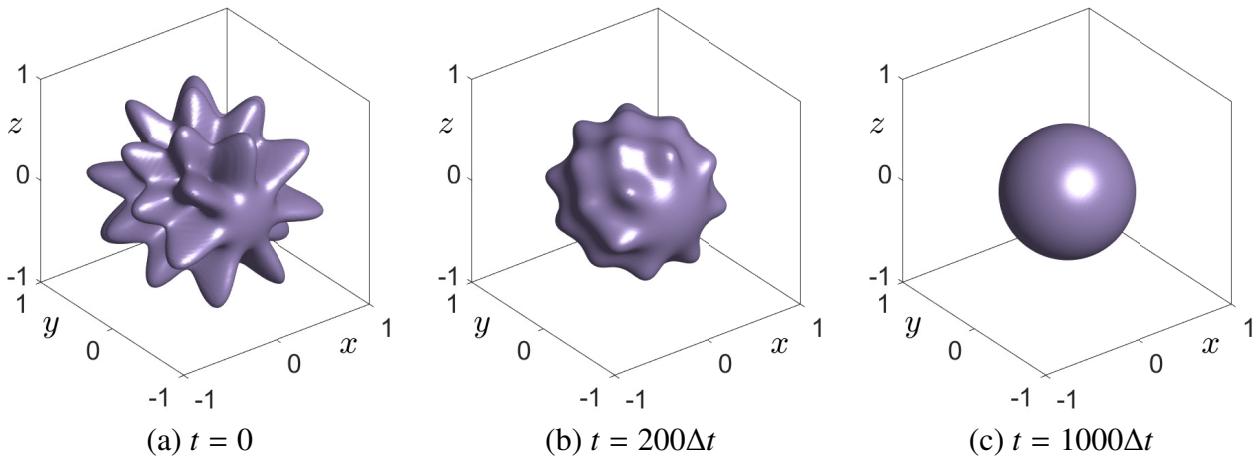


Figure 15. With a sphere initial condition perturbed by a spherical harmonic, (a)–(c) are snapshots of the numerical solution at time $t = 0$, $200\Delta t$, and $1000\Delta t$, respectively.

3.7. Temporal evolution with obstacles

The implicit-type methods, compared to the explicit method, can use a relatively large Δt . However, it is difficult to implement their algorithms for the computational simulation. For example, the multigrid method is one of the most famous algorithms for solving implicitly discretized PDEs. If the computational domain is complex or there are obstacles that interfere with the evolution of order parameter ϕ , it gets more complicated. In contrast, our proposed explicit compact method can obtain the spatially fourth-order accurate solution for the AC equation efficiently even in complex domains or in the presence of obstacles. To demonstrate these advantages, we perform the computational simulations considering obstacles that are solid and fixed structures in 2D and 3D domains.

We use the non-convex initial condition and the rectangular bar-shaped obstacle in the 2D domain as shown in Figure 16(a). In Figure 16, from top to bottom, each row represents without the obstacle, with the obstacle, and overlapping the two cases. Here, the solid line and the purple area are the zero-level set of the solutions with and without the obstacle, respectively, and the green area inside the dotted line is the obstacle. The parameters used for the test are $N = 256$, $h = 1/N$, $\Delta t = 0.1h^2$, $T = 39000\Delta t$, and $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$ in the domain $\Omega = (0, 1)^2$. The first row of Figure 16 shows the temporal evolution by mean curvature flow in the case without the obstacle as the numerical simulations presented in previous subsections. To perform the simulations with the obstacle, we perform iterative

calculations for numerical solutions only at grid points in the whole domain except for obstacles. For the numerical solution near the obstacle, we always use the fixed solution, $\phi = -1$, at points inside the obstacle. The second row of Figure 16 shows the temporal evolution of the zero-level set of numerical solutions with the obstacle. The mean curvature of the inward curves near the obstacle becomes larger than in the case without the obstacle. Therefore, the part with larger mean curvature, such as near the obstacle, shrinks faster according to motion by mean curvature. It is observed that the different mean curvature causes different results of temporal evolution depending on whether the obstacle exists as shown in the bottom row of Figure 16.

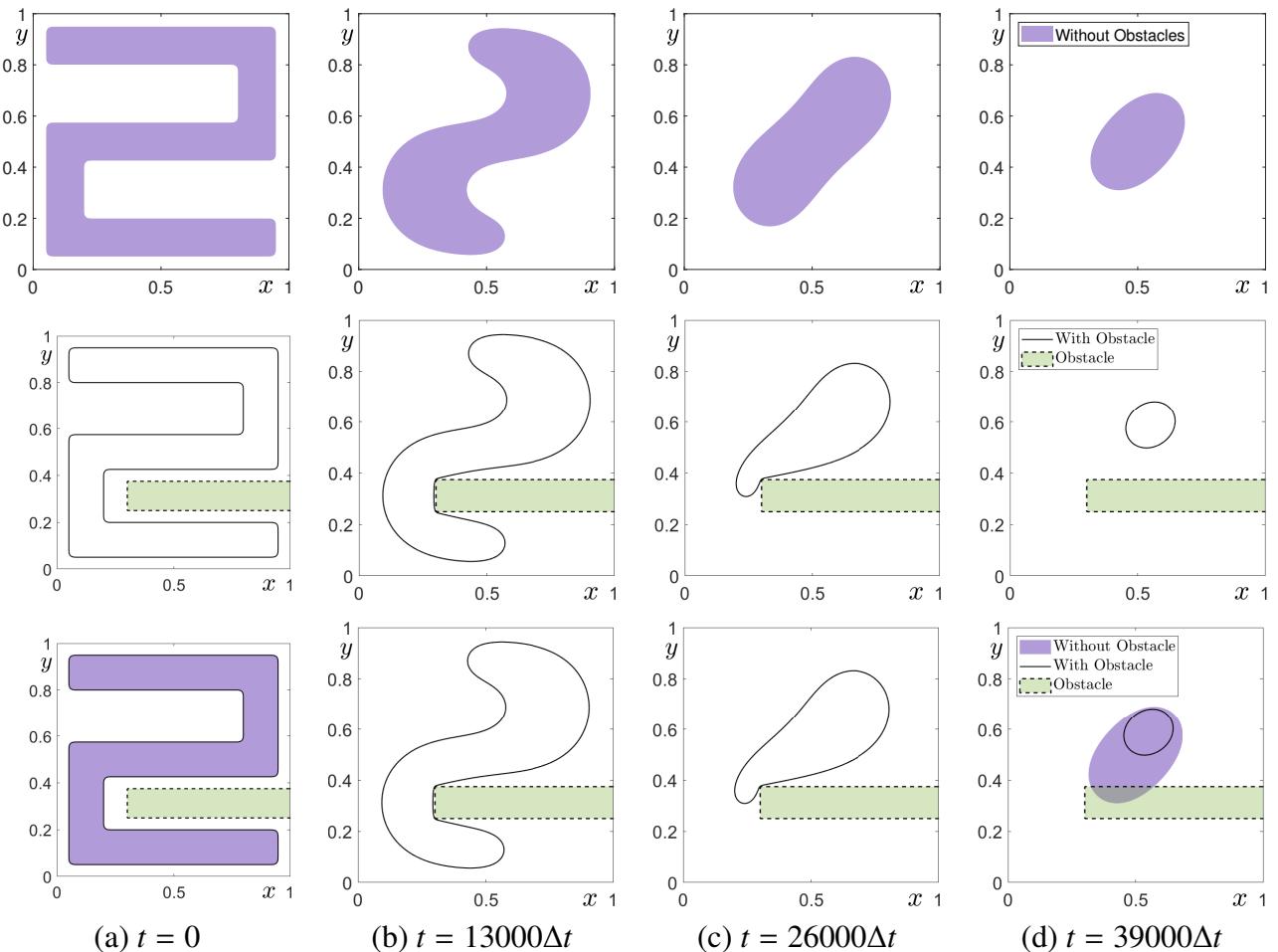


Figure 16. (a)–(d) Snapshots of zero-level contours of the numerical solution without and with an obstacle at $t = 0, 13000\Delta t, 26000\Delta t$, and $39000\Delta t$. The first and second rows imply the numerical solution without and with an obstacle, respectively. The third row shows the overlapped contours of the two cases.

Next, we conduct the 3D experiments using the non-convex initial condition, where the values inside the surface are $\phi = 1$ and the outside is $\phi = -1$ as shown in Figure 17(a). Figures 17(a)–17(d) illustrate the temporal evolutions of zero-level isosurface of the numerical solutions without and with the obstacle. In the second row of Figure 17, the green object represents the obstacle where the inside and boundary of the obstacle are fixed to $\phi = -1$. Here, we use the numerical parameters as $N = 150$,

$h = 1/N$, $\Delta t = 0.1h^2$, and $\epsilon = 4h/(2\sqrt{2}\tanh^{-1}(0.9))$ in the domain $\Omega = (0, 1)^3$. Even in 3D space, the mean curvature of the area with the obstacle is maintained to be large, and the solution contracts faster than when there is no obstacle. The difference between 2D and 3D simulations is that there are two principal curvatures in the 3D domain, which complicates the analysis of the numerical results more than in the 2D cases. However, the basic mechanism is the same as the 2D simulations.

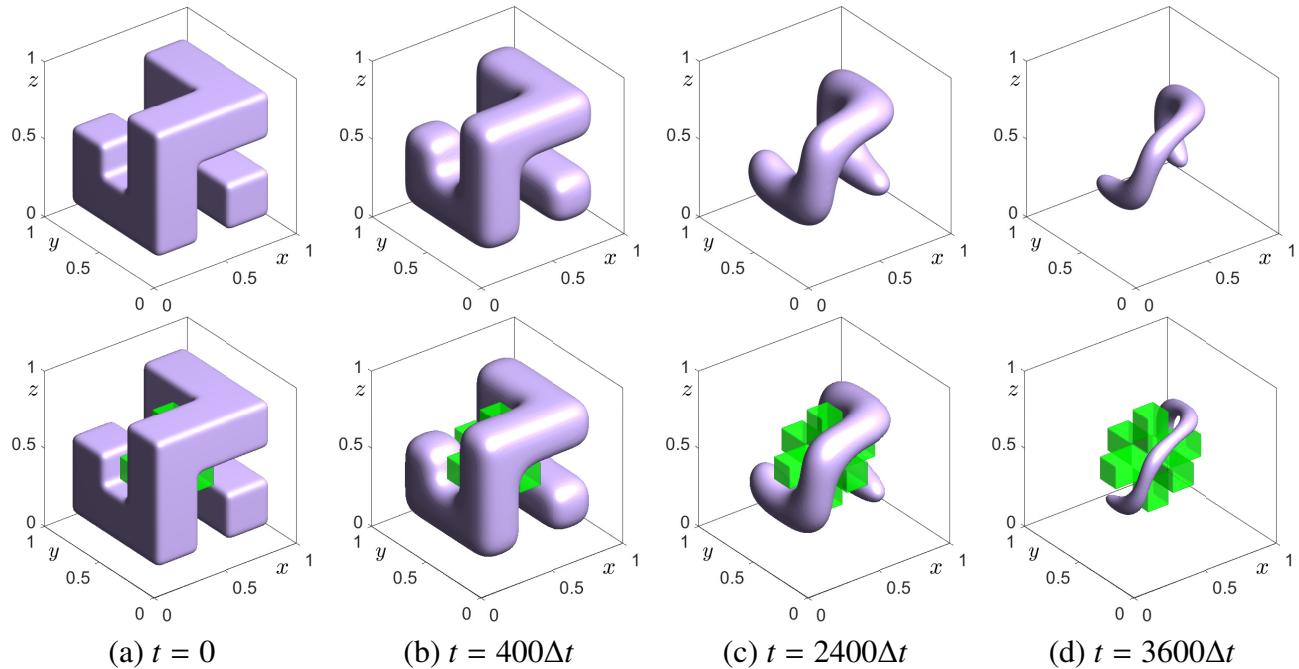


Figure 17. (a)–(d) Snapshots of isosurface of the numerical solutions without and with an obstacle at $t = 0, 400\Delta t, 2400\Delta t$, and $3600\Delta t$. The first and second rows imply the numerical solution without and with an obstacle, respectively.

4. Conclusions

We presented the explicit spatially fourth-order accurate compact method for the AC equation. We use a fourth-order compact FDM, not only being more accurate but needing relatively less wide stencils for the discrete Laplace operator. The time step restriction that is the typical problem of the explicit Euler scheme is not severe because the governing equation is a second-order parabolic PDEs, and small temporal step sizes should be used for an accurate numerical solution. Moreover, it is simple to implement. Therefore, the proposed numerical algorithm is fast and efficient. We observed the phase separation phenomena using random number initial conditions, calculated the rate of convergence to verify the fourth-order accuracy in space, verified the hyperbolic tangent equilibrium profile, and numerically demonstrated the dissipation of discrete total energy. We verified the efficiency of the proposed method by comparing the computational cost between our method and the existing method. We performed various numerical simulations for motion by mean curvature, the representative property of the AC equation. Moreover, experiments comparing the numerical solutions with and without solid obstacles were performed to indicate that the proposed numerical algorithm is efficient and simple to implement. In previous research [41], the maximum bound

principle and energy dissipation were proven for a one-step explicit method to the AC equation. On the other hand, this is a non-trivial and challenging task for the current proposed scheme, which involves a two-step method using the $(n - 1)$ -, n -, and $(n + 1)$ -th time step solutions. In this study, we focused on proposing a fully explicit fourth-order compact scheme and presenting numerical experiments of the proposed method for the AC equation. In future work, theoretical proof for the maximum bound principle, energy dissipation, and error estimates will be considered, drawing insights from the studies conducted by [42, 43].

Use of AI tools declaration

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

Acknowledgments

The first author (C. Lee) was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1C1C2003896). The authors extend their thanks to the reviewers for the valuable and constructive input they provided during the revision of the article.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. S. M. Allen, J. W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, *Acta metall.*, **27** (1979), 1085–1095. [https://doi.org/10.1016/0001-6160\(79\)90196-2](https://doi.org/10.1016/0001-6160(79)90196-2)
2. N. Takada, J. Matsumoto, S. Matsumoto, K. Kurihara, Phase-field model-based simulation of two-phase fluid motion on partially wetted and textured solid surface, *J. Comput. Sci.*, **17** (2016), 315–324. <https://doi.org/10.1016/j.jocs.2016.05.009>
3. S. Abide, Finite difference preconditioning for compact scheme discretizations of the Poisson equation with variable coefficients, *J. Comput. Appl. Math.*, **379** (2020), 112872. <https://doi.org/10.1016/j.cam.2020.112872>
4. K. Li, W. Liao, An efficient and high accuracy finite-difference scheme for the acoustic wave equation in 3D heterogeneous media, *J. Comput. Sci.*, **40** (2020), 101063. <https://doi.org/10.1016/j.jocs.2019.101063>
5. T. Li, J. Lu, C. W. Shu, Stability analysis of inverse Lax-Wendroff boundary treatment of high order compact difference schemes for parabolic equations, *J. Comput. Appl. Math.*, **400** (2022), 113711. <https://doi.org/10.1016/j.cam.2021.113711>
6. M. Wu, Y. Jiang, Y. Ge, A high accuracy local one-dimensional explicit compact scheme for the 2D acoustic wave equation, *Adv. Math. Phys.*, **2022** (2022), 9743699. <https://doi.org/10.1155/2022/9743699>

7. K. S. Patel, M. Mehra, Fourth order compact scheme for space fractional advection-diffusion reaction equations with variable coefficients, *J. Comput. Appl. Math.*, **380** (2020), 112963. <https://doi.org/10.1016/j.cam.2020.112963>
8. Y. Nawaz, M. S. Arif, W. Shatanawi, A. Nazeer, An explicit fourth-order compact numerical scheme for heat transfer of boundary layer flow, *Energies*, **14** (2021), 3396. <https://doi.org/10.3390/en14123396>
9. J. Qiu, D. Han, H. Zhou, A general conservative eighth-order compact finite difference scheme for the coupled Schrödinger-KdV equations, *AIMS Math.*, **8** (2023), 10596–10618. <https://doi.org/10.3934/math.2023538>
10. E. G. M. ElMahdi, J. Huang, Two linearized finite difference schemes for time fractional nonlinear diffusion-wave equations with fourth order derivative, *AIMS Math.*, **6** (2021), 6356–6376. <https://doi.org/10.3934/math.2021373>
11. N. Abdi, H. Aminikhah, A. R. Sheikhani, High-order compact finite difference schemes for the time-fractional Black-Scholes model governing European options, *Chaos Soliton. Fract.*, **162** (2022), 112423. <https://doi.org/10.1016/j.chaos.2022.112423>
12. S. Zhai, X. Feng, Y. He, Numerical simulation of the three dimensional Allen-Cahn equation by the high-order compact ADI method, *Comput. Phys. Commun.*, **185** (2014), 2449–2455. <https://doi.org/10.1016/j.cpc.2014.05.017>
13. J. Long, C. Luo, Q. Yu, Y. Li, An unconditional stable compact fourth-order finite difference scheme for three dimensional Allen-Cahn equation, *Comput. Math. Appl.*, **77** (2019), 1042–1054. <https://doi.org/10.1016/j.camwa.2018.10.028>
14. 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. <https://doi.org/10.1155/2022/8522231>
15. S. C. Buranay, N. Arshad, A. H. Matan, Hexagonal grid computation of the derivatives of the solution to the heat equation by using fourth-order accurate two-stage implicit methods, *Fractal Fract.*, **5** (2021), 203. <https://doi.org/10.3390/fractfract5040203>
16. S. C. Buranay, N. Arshad, Hexagonal grid approximation of the solution of the heat equation on special polygons, *Adv. Differ. Equ.*, **2020** (2020), 309. <https://doi.org/10.1186/s13662-020-02749-z>
17. A. A. Dosiyev, S. C. Buranay, On solving the cracked-beam problem by block method, *Commun. Numer. Meth. En.*, **24** (2008), 1277–1289. <https://doi.org/10.1002/cnm.1032>
18. A. A. Dosiyev, S. C. Buranay, D. Subasi, The block-grid method for solving Laplace's equation on polygons with nonanalytic boundary conditions, *Bound. Value Probl.*, **2010** (2010), 468594. <https://doi.org/10.1155/2010/468594>
19. K. Poochinapan, B. Wongsajai, 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. <https://doi.org/10.1016/j.amc.2022.127374>
20. J. W. Cahn, J. E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, *J. Chem. Phys.*, **28** (1958), 258–267. <https://doi.org/10.1063/1.1744102>

21. Y. Li, R. Liu, Q. Xia, C. He, Z. Li, First-and second-order unconditionally stable direct discretization methods for multi-component Cahn-Hilliard system on surfaces, *J. Comput. Appl. Math.*, **401** (2022), 113778. <https://doi.org/10.1016/j.cam.2021.113778>
22. J. Li, Z. Sun, X. Zhao, A three level linearized compact difference scheme for the Cahn-Hilliard equation, *Sci. China Math.*, **55** (2012), 805–826. <https://doi.org/10.1007/s11425-011-4290-x>
23. L. Ju, J. Zhang, Q. Du, Fast and accurate algorithms for simulating coarsening dynamics of Cahn-Hilliard equations, *Comp. Mater. Sci.*, **108** (2015), 272–282. <https://doi.org/10.1016/j.commatsci.2015.04.046>
24. S. Lee, Fourth-order spatial and second-order temporal accurate compact scheme for Cahn-Hilliard equation, *Int. J. Nonlin. Sci. Num.*, **20** (2019), 137–143. <https://doi.org/10.1515/ijnsns-2017-0278>
25. S. Lee, J. Shin, Energy stable compact scheme for Cahn-Hilliard equation with periodic boundary condition, *Comput. Math. Appl.*, **77** (2019), 189–198. <https://doi.org/10.1016/j.camwa.2018.09.021>
26. Z. Xiao, P. Yu, H. Ouyang, J. Zhang, A parallel high-order compact scheme for the pure streamfunction formulation of the 3D unsteady incompressible Navier-Stokes equation, *Commun. Nonlinear Sci.*, **95** (2021), 105631. <https://doi.org/10.1016/j.cnsns.2020.105631>
27. D. Jeong, J. Kim, An explicit hybrid finite difference scheme for the Allen-Cahn equation, *J. Comput. Appl. Math.*, **340** (2018), 247–255. <https://doi.org/10.1016/j.cam.2018.02.026>
28. C. Lee, J. Park, S. Kwak, S. Kim, Y. Choi, S. Ham, et al., An adaptive time-stepping algorithm for the Allen-Cahn equation, *J. Funct. Space.*, **2022** (2022), 2731593. <https://doi.org/10.1155/2022/2731593>
29. D. Jeong, S. Lee, D. Lee, J. Shin, J. Kim, Comparison study of numerical methods for solving the Allen-Cahn equation, *Comp. Mater. Sci.*, **111** (2016), 131–136. <https://doi.org/10.1016/j.commatsci.2015.09.005>
30. C. Lee, D. Jeong, J. Shin, Y. Li, J. Kim, A fourth-order spatial accurate and practically stable compact scheme for the Cahn-Hilliard equation, *Physica A*, **409** (2014), 17–28. <https://doi.org/10.1016/j.physa.2014.04.038>
31. 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 Commun.*, **200** (2016), 108–116. <https://doi.org/10.1016/j.cpc.2015.11.006>
32. J. W. Choi, H. G. Lee, D. Jeong, J. Kim, An unconditionally gradient stable numerical method for solving the Allen-Cahn equation, *Physica A*, **388** (2009), 1791–1803. <https://doi.org/10.1016/j.physa.2009.01.026>
33. C. Lee, H. Kim, S. Yoon, S. Kim, D. Lee, J. Park, et al., An unconditionally stable scheme for the Allen-Cahn equation with high-order polynomial free energy, *Commun. Nonlinear Sci.*, **95** (2021), 105658. <https://doi.org/10.1016/j.cnsns.2020.105658>
34. U. Trottenberg, C. Oosterlee, A. Schuller, *Multigrid*, Elsevier, 2000.
35. W. L. Briggs, V. E. Henson, S. F. McCormick, *A multigrid tutorial*, Society for Industrial and Applied Mathematics, 2000.

-
36. J. Yang, C. Lee, S. Kwak, Y. Choi, J. Kim, A conservative and stable explicit finite difference scheme for the diffusion equation, *J. Comput. Sci.*, **56** (2021), 101491. <https://doi.org/10.1016/j.jocs.2021.101491>
37. D. Lee, J. Kim, Mean curvature flow by the Allen-Cahn equation, *Eur. J. Appl. Math.*, **26** (2015), 535–559. <https://doi.org/10.1017/S0956792515000200>
38. C. Lee, Y. Choi, J. Kim, An explicit stable finite difference method for the Allen-Cahn equation, *Appl. Numer. Math.*, **182** (2022), 87–99. <https://doi.org/10.1016/j.apnum.2022.08.006>
39. V. Cristini, J. Lowengrub, Three-dimensional crystal growth-I: Linear analysis and self-similar evolution, *J. Cryst. Growth*, **240** (2022), 267–276. [https://doi.org/10.1016/S0022-0248\(02\)00831-X](https://doi.org/10.1016/S0022-0248(02)00831-X)
40. M. A. Wieczorek, M. Meschede, SHTools: Tools for working with spherical harmonics, *Geochem. Geophys. Geosy.*, **19** (2018), 2574–2592. <https://doi.org/10.1029/2018GC007529>
41. S. Ham, J. Kim, Stability analysis for a maximum principle preserving explicit scheme of the Allen-Cahn equation, *Math. Comput. Simulat.*, **207** (2023), 453–465. <https://doi.org/10.1016/j.matcom.2023.01.016>
42. Q. Du, L. Ju, X. Li, Z. Qiao, Maximum bound principles for a class of semilinear parabolic equations and exponential time-differencing schemes, *SIAM Rev.*, **63** (2021), 317–359. <https://doi.org/10.1137/19M1243750>
43. Y. Gong, B. Ji, H. L. Liao, A maximum bound principle preserving iteration technique for a class of semilinear parabolic equations, *Appl. Numer. Math.*, **184** (2023), 482–495. <https://doi.org/10.1016/j.apnum.2022.11.002>



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>)