International Journal of Mechanical Sciences 217 (2022) 106985

Contents lists available at ScienceDirect

International Journal of Mechanical Sciences

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

An explicit conservative Saul'yev scheme for the Cahn–Hilliard equation

Junxiang Yang [a], Yibao Li [b], Chaeyoung Lee [c], Hyun Geun Lee [d], Soobin Kwak [c], Youngjin Hwang [c], Xuan Xin [e], Junseok Kim [c,*]

[a] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China
[b] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China
[c] Department of Mathematics, Korea University, Seoul, 02841, Republic of Korea
[d] Department of Mathematics, Kwangwoon University, Seoul, 01897, Republic of Korea
[e] Department of Mathematics, Harbin Institute of Technology at Weihai, Weihai, 264209, China

ARTICLE INFO

Keywords:
Conservative finite difference method
Cahn–Hilliard equation
Phase separation

ABSTRACT

We present an explicit conservative Saul'yev finite difference scheme for the Cahn–Hilliard (CH) equation, which models a phase separation phenomenon in binary alloys. The CH equation has been successfully used in various scientific and practical applications. A variety of numerical algorithms were developed to efficiently calculate the CH equation. Because of the highly nonlinear term and the biharmonic operator, numerical methods were mostly implicit schemes. Although a fully explicit scheme is very simple, the time-step restriction is very stringent and the stable time step size is not practicable in high-dimensional spaces. To overcome this severe time-step restriction and retain the simplicity of the explicit method for the CH model, we develop an explicit conservative numerical method based on the Saul'yev method. The proposed scheme has four main merits: (i) the phase-field variable can be directly updated without iterative algorithms; (ii) the numerical solution remains stable even if relatively larger time steps are used; (iii) the mass conservation of the CH equation can be satisfied; and (iv) the simulations in complex domains are easy to implement. The computational experiments confirm the superior performance of the proposed algorithm.

1. Introduction

In this study, we present an explicit conservative computational scheme for the Cahn–Hilliard (CH) equation [1]. The CH equation was originally developed for modeling a phase separation phenomenon in binary alloys. Recently, the CH model has been successfully applied to a variety of scientific and practical applications such as phase-field model with logarithmic free energy [2], block copolymer [3,4], vector-valued dynamics [5,6], multi-phase fluid flows [7–13], phase separation on curved surfaces [14,15], image inpainting [16], and volume reconstruction, etc. The CH equation is given as follows:

$$\begin{aligned} \frac{\partial \phi(\mathbf{x},t)}{\partial t} &= \Delta\left[F'(\phi(\mathbf{x},t)) - \epsilon^2 \Delta\phi(\mathbf{x},t)\right], \quad \mathbf{x}\in\Omega,\ t>0,\\ \mathbf{n}\cdot\nabla\phi(\mathbf{x},t) &= 0,\ \mathbf{n}\cdot\nabla\Delta\phi(\mathbf{x},t)=0, \quad \mathbf{x}\in\partial\Omega,\ t>0, \end{aligned}$$

where $\phi(\mathbf{x},t)$ is a phase-field, which generally takes the values 1 and -1 in two immiscible components, ϵ is a parameter, $F(\phi)=(\phi^2-1)^2/4$, and $\mathbf{n}$ is outer unit normal vector.

The physical phenomena with interfaces extensively exist in material, biology, and fluid fields. The CH equation is a practical model to simulate interfacial problems because of its flexibility in handling the topological changes of interface. Please refer to [17–22] for the many applications of the CH equation. In particular, the immiscible multi-component fluid flows are very common in fluid mechanical engineering and some typical works are as follows. Zhang et al. [23] studied the transitions of compound sessile droplets by using the ternary CH system. Xu et al. [24] investigated the effect of temperature on the self-rewetting droplets on a solid substrate, where the CH equation was adopted to capture the interfacial position. Mu et al. [25] numerically investigated the interfacial instability of a single liquid thread in a co-flow focusing device. Yang and Kim [26] used the axisymmetric CH model to study the dynamics of binary Rayleigh instability, where the effect of solid fiber was considered. Later, Yang et al. [27] studied the dynamics of compound liquid threads by using the three-component CH equations. By considering the effect of solid wall boundary of water–oil separator, Yang et al. [28] investigated the different dynamics of Rayleigh–Taylor instability. Liu et al. [29] studied the maximal spreading of compound droplets impacting on a spherical substrate.

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

https://doi.org/10.1016/j.ijmecsci.2021.106985
Received 18 July 2021; Received in revised form 8 September 2021; Accepted 13 September 2021
Available online 27 December 2021

It is well known that the exact solution of the CH model is hard to find in general. A variety of computational techniques have been developed to efficiently solve the CH type equations in recent years. Dehghan and Abbaszadeh [30] proposed a local collocation radial basis functions (RBF) method to efficiently solve the multi-dimensional CH model. Later, Abbaszadeh et al. [31] developed a direct meshless local collocation method to treat the stochastic CH model in space and the temporal discretization was performed by using a finite difference scheme. Liu et al. [32] numerically compared the different dynamics of the time-fractional CH and AC equations, where an efficient finite difference and Fourier spectral methods were adopted to treat the increased memory requirement. Because of the highly nonlinear term and the biharmonic operator, the numerical methods were mostly implicit schemes [33–37]. The linear stabilization and convex splitting methods are two typical approaches to treat the CH equation in implicit manners. In the linear stabilization method, the nonlinear term is treated explicitly and an appropriate stabilization term is added to stabilize the computation. Please refer to [38–41] for the theoretical investigations of the linear stabilization method. In the convex splitting method, the nonlinear term is split into convex and concave parts; and then we solve the convex part implicitly and concave part explicitly. By using the Taylor expansion, we can prove that the convex splitting method leads to unconditionally energy-stable schemes for the CH equation, see [42] for some details. Moreover, the nonlinear convex splitting method was successfully applied to no-slope-selection thin film models, the temporally first-, second-, and third-order accurate schemes, detailed stability and error estimations can be found in [43–49]. For all implicit schemes, fast and accurate iterative techniques are needed. If the improper iterative algorithm is adopted, the accuracy of the numerical solution will be significantly affected. Moreover, the existence and uniqueness of the solution of nonlinearly implicit methods should be analytically proved, which obviously bring extra and tedious works for researchers in industrial fields. Different from the implicit schemes, the explicit scheme provides a straightforward approach to obtain the numerical solution without any iterative methods. In this sense, the most desirable and simplest numerical method for the partial differential equation is the explicit scheme.

Although a fully explicit scheme is very simple, the time-step restriction is very stringent and it has only practically been used for one-dimensional benchmark test [50]. Therefore, the application of the fully explicit scheme in high-dimensional spaces is not practicable. To overcome this time-step restriction and retain the simplicity of the explicit method for the CH equation, we propose an explicit conservative numerical scheme based on the Saul'yev method [51].

In Section 2, we describe the proposed explicit scheme. In Section 3, extensive numerical tests are performed to validate our proposed scheme. In Section 4, we discuss a possible extension of the proposed method. The conclusion and future applications are given in Section 5.

2. Numerical solution algorithm

In this section, we introduce the fully discrete numerical algorithms for solving the CH equation in detail. In Section 2.1, the two-dimensional algorithm will be discussed. The straightforward extension to the three-dimensional case will be given in Section 2.2.

2.1. Two-dimensional algorithm

We present the numerical solution algorithm for the CH equation in the two-dimensional domain $\Omega = (L_x, R_x) \times (L_y, R_y)$. Let $\Omega_h = \{(x_i = L_x + h(i-0.5),\ y_j = L_y + h(j-0.5)) | i = 1, \dots, N_x,\ j = 1, \dots, N_y\}$ be the discrete computational domain, where $h = (R_x - L_x)/N_x = (R_y - L_y)/N_y$ is the uniform step size; N_x and N_y are the numbers of the grid points in the x- and y-direction, respectively. Let ϕ_{ij}^n be the numerical approximation of $\phi(x_i, y_j, n\Delta t)$, where Δt is the time step. Let us start with the linear convex splitting scheme [52]:

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} = \Delta_d((\phi_{ij}^n)^3 - 3\phi_{ij}^n) + 2\Delta_d\phi_{ij}^{n+1} - \epsilon^2\Delta_d^2\phi_{ij}^{n+1}, \tag{1}$$

where

$$\begin{aligned}
\Delta_d\phi_{ij} &= \frac{\phi_{i+1,j} + \phi_{i-1,j} - 4\phi_{ij} + \phi_{i,j+1} + \phi_{i,j-1}}{h^2},\\
\Delta_d^2\phi_{ij} &= \frac{\Delta_d\phi_{i+1,j} + \Delta_d\phi_{i-1,j} - 4\Delta_d\phi_{ij} + \Delta_d\phi_{i,j+1} + \Delta_d\phi_{i,j-1}}{h^2}\\
&= \frac{1}{h^4}\big[\phi_{i+2,j} + \phi_{i-2,j} + \phi_{i,j+2} + \phi_{i,j-2} + 2(\phi_{i+1,j+1} + \phi_{i+1,j-1}\\
&\quad + \phi_{i-1,j+1} + \phi_{i-1,j-1}) - 8(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}) + 20\phi_{ij}\big].
\end{aligned}$$

Here, $(\phi^{n+1} - \phi^n)/\Delta t = \partial\phi/\partial t + O(\Delta t)$ is first-order accurate approximation in time, $\Delta_d\phi = \Delta\phi + O(h^2)$ and $\Delta_d^2\phi = \Delta^2\phi + O(h^2)$ are second-order accurate approximations in space because standard five-point difference stencil is adopted to discretize the Laplacian term. As we refine time and spatial steps, i.e., $\Delta t \to 0$ and $h \to 0$, it is easy to check that Eq. (1) and continuous partial differential equation are consistent.

For simplicity of exposition, we use the following Neumann boundary condition for all n:

$$\begin{aligned}
&\phi_{-1,j}^n = \phi_{2j}^n,\ \phi_{0j}^n = \phi_{1j}^n,\ \phi_{N_x+1,j}^n = \phi_{N_xj}^n,\ \phi_{N_x+2,j}^n = \phi_{N_x-1,j}^n,\\
&\qquad \text{for } j = 1, \dots, N_y,\\
&\phi_{i,-1}^n = \phi_{i2}^n,\ \phi_{i0}^n = \phi_{i1}^n,\ \phi_{i,N_y+1}^n = \phi_{iN_y}^n,\ \phi_{i,N_y+2}^n = \phi_{i,N_y-1}^n,\\
&\qquad \text{for } i = 1, \dots, N_x.
\end{aligned}$$

By using the definitions of $\Delta_d\phi_{ij}$ and $\Delta_d^2\phi_{ij}$, the fully discrete form of Eq. (1) can be expressed as:

$$\begin{aligned}
\frac{\phi_{ij}^{n+1}}{\Delta t} &= \frac{\phi_{ij}^n}{\Delta t} + \Delta_d((\phi_{ij}^n)^3 - 3\phi_{ij}^n) + 2\Big(\frac{\phi_{i+1,j}^{n+1} - \phi_{ij}^{n+1}}{h^2} - \frac{\phi_{ij}^{n+1} - \phi_{i-1,j}^{n+1}}{h^2}\\
&\quad + \frac{\phi_{i,j+1}^{n+1} - \phi_{ij}^{n+1}}{h^2} - \frac{\phi_{ij}^{n+1} - \phi_{i,j-1}^{n+1}}{h^2}\Big) - \frac{\epsilon^2}{h^2}\Big(\frac{\phi_{i+2,j}^{n+1} - \phi_{i+1,j}^{n+1}}{h^2}\\
&\quad - \frac{\phi_{i+1,j}^{n+1} - \phi_{ij}^{n+1}}{h^2} + \frac{\phi_{i+1,j+1}^{n+1} - \phi_{i+1,j}^{n+1}}{h^2} - \frac{\phi_{i+1,j}^{n+1} - \phi_{i+1,j-1}^{n+1}}{h^2}\\
&\quad + \frac{\phi_{ij}^{n+1} - \phi_{i-1,j}^{n+1}}{h^2} - \frac{\phi_{i-1,j}^{n+1} - \phi_{i-2,j}^{n+1}}{h^2} + \frac{\phi_{i-1,j+1}^{n+1} - \phi_{i-1,j}^{n+1}}{h^2}\\
&\quad - \frac{\phi_{i-1,j}^{n+1} - \phi_{i-1,j-1}^{n+1}}{h^2} + \frac{\phi_{i+1,j+1}^{n+1} - \phi_{i,j+1}^{n+1}}{h^2} - \frac{\phi_{i,j+1}^{n+1} - \phi_{i-1,j+1}^{n+1}}{h^2}\\
&\quad + \frac{\phi_{i,j+2}^{n+1} - \phi_{i,j+1}^{n+1}}{h^2} - \frac{\phi_{i,j+1}^{n+1} - \phi_{ij}^{n+1}}{h^2} + \frac{\phi_{i+1,j-1}^{n+1} - \phi_{i,j-1}^{n+1}}{h^2}\\
&\quad - \frac{\phi_{i,j-1}^{n+1} - \phi_{i-1,j-1}^{n+1}}{h^2} + \frac{\phi_{ij}^{n+1} - \phi_{i,j-1}^{n+1}}{h^2} - \frac{\phi_{i,j-1}^{n+1} - \phi_{i,j-2}^{n+1}}{h^2}\Big).
\end{aligned}$$

In two-dimensional space, the proposed Saul'yev method uses 8 cases of nested loops. Let us describe one case as an example.

$$\text{For } j = 1, 2, \dots, N_y, \text{ for } i = 1, 2, \dots, N_x, \tag{2}$$

$$\begin{aligned}
\phi_{ij}^{n+1} &= \frac{1}{r}\Big[\frac{\phi_{ij}^n}{\Delta t} + \Delta_d((\phi_{ij}^n)^3 - 3\phi_{ij}^n) + \frac{2}{h^2}\big(\phi_{i+1,j}^n + \phi_{i-1,j}^{n+1} - 2\phi_{ij}^n\\
&\quad + \phi_{i,j+1}^n + \phi_{i,j-1}^{n+1}\big) - \frac{\epsilon^2}{h^4}\big[\phi_{i+2,j}^n + \phi_{i-2,j}^{n+1} + \phi_{i,j+2}^n + \phi_{i,j-2}^{n+1}\\
&\quad + 2(\phi_{i+1,j+1}^n + \phi_{i+1,j-1}^n + \phi_{i-1,j+1}^n + \phi_{i-1,j-1}^{n+1})\\
&\quad - 8(\phi_{i+1,j}^n + \phi_{i-1,j}^{n+1} + \phi_{i,j+1}^n + \phi_{i,j-1}^{n+1}) + 10\phi_{ij}^n\big]\Big],
\end{aligned}$$

where $r = 1/\Delta t + 4/h^2 + 10\epsilon^2/h^4$. We should note that we use a nested loop in Eq. (2). For each outer loop j, the inner loop i runs from 1 to N_x. Fig. 1 shows a schematic diagram of two-dimensional stencil.

The other 7 cases are as follows:

$$\begin{aligned}
&\text{For } j = 1, 2, \dots, N_y, \quad \text{for } i = N_x, N_x - 1, \dots, 1,\\
&\text{For } j = N_y, N_y - 1, \dots, 1, \quad \text{for } i = 1, 2, \dots, N_x,
\end{aligned}$$

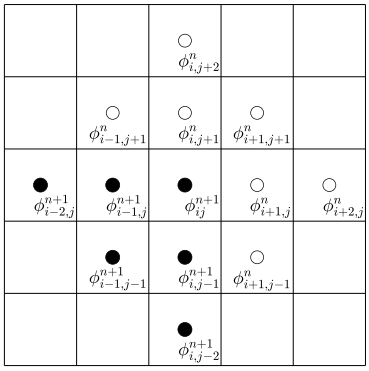

Fig. 1. Schematic diagram of two-dimensional stencil.

$$
\begin{aligned}
&\text{For } j = N_y, N_y-1, \dots, 1, \quad \text{for } i = N_x, N_x-1, \dots, 1,\\
&\text{For } i = 1, 2, \dots, N_x, \quad \text{for } j = 1, 2, \dots, N_y,\\
&\text{For } i = N_x, N_x-1, \dots, 1, \quad \text{for } j = 1, 2, \dots, N_y,\\
&\text{For } i = 1, 2, \dots, N_x, \quad \text{for } j = N_y, N_y-1, \dots, 1,\\
&\text{For } i = N_x, N_x-1, \dots, 1, \quad \text{for } j = N_y, N_y-1, \dots, 1.
\end{aligned}
$$

Because the algorithm of Saul'yev-type method cannot strictly preserve the total mass of the system, we adopt the following mass correction technique [15] after each temporal iteration:

$$
\phi^{n+1}_{ij} = \phi^{n+1}_{ij} + \frac{\sum_{p=1}^{N_x}\sum_{q=1}^{N_y}(\phi^0_{pq} - \phi^{n+1}_{pq})}{\sum_{p=1}^{N_x}\sum_{q=1}^{N_y}\sqrt{F(\phi^{n+1}_{pq})}}\sqrt{F(\phi^{n+1}_{ij})}, \tag{3}
$$

which means the mass is added or subtracted at interface transition region.

2.2. Three-dimensional algorithm

We propose numerical solution algorithm for the CH equation in three-dimensional domain $\Omega = (L_x, R_x)\times(L_y, R_y)\times(L_z, R_z)$. Let $\Omega_h = \{(x_i = L_x + h(i-0.5),\ y_j = L_y + h(j-0.5),\ z_k = L_z + h(k-0.5))|i = 1, \dots, N_x,\ j = 1, \dots, N_y,\ k = 1, \dots, N_z\}$ be the discrete computational domain, where $h = (R_x - L_x)/N_x = (R_y - L_y)/N_y = (R_z - L_z)/N_z$ is the uniform step size; N_x, N_y and N_z are the numbers of the grid points in the x-, y- and z-direction, respectively. Let ϕ^n_{ijk} be the numerical approximation of $\phi(x_i, y_j, z_k, n\Delta t)$. Let us start with the linear convex splitting scheme [52]:

$$
\frac{\phi^{n+1}_{ijk} - \phi^n_{ijk}}{\Delta t} = \Delta_d((\phi^n_{ijk})^3 - 3\phi^n_{ijk}) + 2\Delta_d\phi^{n+1}_{ijk} - \epsilon^2\Delta_d^2\phi^{n+1}_{ijk}, \tag{4}
$$

where

$$
\begin{aligned}
\Delta_d\phi_{ijk} &= \frac{1}{h^2}\big(\phi_{i+1,jk} + \phi_{i-1,jk} + \phi_{i,j+1,k} + \phi_{i,j-1,k} + \phi_{ij,k+1} + \phi_{ij,k-1} - 6\phi_{ijk}\big),\\
\Delta_d^2\phi_{ijk} &= \frac{1}{h^2}\big(\Delta_d\phi_{i+1,jk} + \Delta_d\phi_{i-1,jk} + \Delta_d\phi_{i,j+1,k} + \Delta_d\phi_{i,j-1,k} + \Delta_d\phi_{ij,k+1}\\
&\quad + \Delta_d\phi_{ij,k-1} - 6\Delta_d\phi_{ijk}\big)\\
&= \frac{1}{h^4}\big[\phi_{i+2,jk} + \phi_{i-2,jk} + \phi_{i,j+2,k} + \phi_{i,j-2,k} + \phi_{ij,k+2} + \phi_{ij,k-2}\\
&\quad + 2(\phi_{i+1,j+1,k} + \phi_{i+1,j-1,k} + \phi_{i+1,j,k+1} + \phi_{i+1,j,k-1} + \phi_{i-1,j+1,k}\\
&\quad + \phi_{i-1,j-1,k} + \phi_{i-1,j,k+1} + \phi_{i-1,j,k-1} + \phi_{i,j+1,k+1} + \phi_{i,j+1,k-1}\\
&\quad + \phi_{i,j-1,k+1} + \phi_{i,j-1,k-1}) - 12(\phi_{i+1,jk} + \phi_{i-1,jk} + \phi_{i,j+1,k}\\
&\quad + \phi_{i,j-1,k} + \phi_{ij,k+1} + \phi_{ij,k-1}) + 42\phi_{ijk}\big].
\end{aligned}
$$

We use the following Neumann boundary condition for all n:

$$
\begin{aligned}
&\phi^n_{-1,jk} = \phi^n_{2,jk},\ \phi^n_{0,jk} = \phi^n_{1,jk},\ \phi^n_{N_x+1,jk} = \phi^n_{N_x,jk},\ \phi^n_{N_x+2,jk} = \phi^n_{N_x-1,jk},\\
&\qquad \text{for } j = 1, \dots, N_y, \ \text{for } k = 1, \dots, N_z,\\
&\phi^n_{i,-1,k} = \phi^n_{i,2,k},\ \phi^n_{i,0,k} = \phi^n_{i,1,k},\ \phi^n_{i,N_y+1,k} = \phi^n_{i,N_y,k},\ \phi^n_{i,N_y+2,k} = \phi^n_{i,N_y-1,k},\\
&\qquad \text{for } i = 1, \dots, N_x, \ \text{for } k = 1, \dots, N_z,\\
&\phi^n_{ij,-1} = \phi^n_{ij,2},\ \phi^n_{ij,0} = \phi^n_{ij,1},\ \phi^n_{ij,N_z+1} = \phi^n_{ij,N_z},\ \phi^n_{ij,N_z+2} = \phi^n_{ij,N_z-1},\\
&\qquad \text{for } i = 1, \dots, N_x, \ \text{for } j = 1, \dots, N_y.
\end{aligned}
$$

By using the definitions of $\Delta_d\phi_{ijk}$ and $\Delta_d^2\phi_{ijk}$, Eq. (4) can be recast to be:

$$
\begin{aligned}
\frac{\phi^{n+1}_{ijk}}{\Delta t} &= \frac{\phi^n_{ijk}}{\Delta t} + \Delta_d((\phi^n_{ijk})^3 - 3\phi^n_{ijk}) + \frac{2}{h^2}\big(\phi^{n+1}_{i+1,jk} + \phi^{n+1}_{i-1,jk} + \phi^{n+1}_{i,j+1,k} + \phi^{n+1}_{i,j-1,k}\\
&\quad + \phi^{n+1}_{ij,k+1} + \phi^{n+1}_{ij,k-1} - 6\phi^{n+1}_{ijk}\big) - \frac{\epsilon^2}{h^4}\big[\phi^{n+1}_{i+2,jk} + \phi^{n+1}_{i-2,jk} + \phi^{n+1}_{i,j+2,k} + \phi^{n+1}_{i,j-2,k}\\
&\quad + \phi^{n+1}_{ij,k+2} + \phi^{n+1}_{ij,k-2} + 2(\phi^{n+1}_{i+1,j+1,k} + \phi^{n+1}_{i+1,j-1,k} + \phi^{n+1}_{i+1,j,k+1} + \phi^{n+1}_{i+1,j,k-1}\\
&\quad + \phi^{n+1}_{i-1,j+1,k} + \phi^{n+1}_{i-1,j-1,k} + \phi^{n+1}_{i-1,j,k+1} + \phi^{n+1}_{i-1,j,k-1} + \phi^{n+1}_{i,j+1,k+1}\\
&\quad + \phi^{n+1}_{i,j+1,k-1} + \phi^{n+1}_{i,j-1,k+1} + \phi^{n+1}_{i,j-1,k-1}) - 12(\phi^{n+1}_{i+1,jk} + \phi^{n+1}_{i-1,jk} + \phi^{n+1}_{i,j+1,k}\\
&\quad + \phi^{n+1}_{i,j-1,k} + \phi^{n+1}_{ij,k+1} + \phi^{n+1}_{ij,k-1}) + 42\phi^{n+1}_{ijk}\big]
\end{aligned}
$$

In three-dimensional space, the proposed Saul'yev method uses 48 cases of nested loops including Eq. (5). Let us describe one case as an example.

$$
\begin{aligned}
&\text{For } i = 1, 2, \dots, N_x, \quad \text{for } j = 1, 2, \dots, N_y, \quad \text{for } k = 1, 2, \dots, N_z, \qquad (5)\\
&\phi^{n+1}_{ijk} = \frac{1}{r}\Big[\frac{\phi^n_{ijk}}{\Delta t} + \Delta_d((\phi^n_{ijk})^3 - 3\phi^n_{ijk}) + \frac{2}{h^2}\big(\phi^n_{i+1,jk} + \phi^{n+1}_{i-1,jk} + \phi^n_{i,j+1,k}\\
&\quad + \phi^{n+1}_{i,j-1,k} + \phi^n_{ij,k+1} + \phi^{n+1}_{ij,k-1} - 3\phi^n_{ijk}\big) - \frac{\epsilon^2}{h^4}\big[\phi^n_{i+2,jk} + \phi^{n+1}_{i-2,jk}\\
&\quad + \phi^n_{i,j+2,k} + \phi^{n+1}_{i,j-2,k} + \phi^n_{ij,k+2} + \phi^{n+1}_{ij,k-2} + 2(\phi^n_{i+1,j+1,k} + \phi^n_{i+1,j-1,k}\\
&\quad + \phi^n_{i+1,j,k+1} + \phi^n_{i+1,j,k-1} + \phi^n_{i-1,j+1,k} + \phi^{n+1}_{i-1,j-1,k} + \phi^n_{i-1,j,k+1}\\
&\quad + \phi^{n+1}_{i-1,j,k-1} + \phi^n_{i,j+1,k+1} + \phi^n_{i,j+1,k-1} + \phi^n_{i,j-1,k+1} + \phi^{n+1}_{i,j-1,k-1})\\
&\quad - 12(\phi^n_{i+1,jk} + \phi^{n+1}_{i-1,jk} + \phi^n_{i,j+1,k} + \phi^{n+1}_{i,j-1,k} + \phi^n_{ij,k+1} + \phi^{n+1}_{ij,k-1}) + 21\phi^n_{ijk}\big]\Big],
\end{aligned}
$$

where $r = 1/\Delta t + 6/h^2 + 21\epsilon^2/h^4$. Fig. 2(a) shows the direction of the three-dimensional nested loop. In Fig. 2(b), open circles and bullets are the positions where the numerical solutions are defined at n and $n+1$ times, respectively. We should note that we use a nested loop in Eq. (5).

The other 47 cases are as follows:

$$
\begin{aligned}
&\text{For } i = 1, 2, \dots, N_x, \quad \text{for } j = 1, 2, \dots, N_y, \quad \text{for } k = 1, 2, \dots, N_k,\\
&\text{For } i = 1, 2, \dots, N_x, \quad \text{for } j = 1, 2, \dots, N_y, \quad \text{for } k = N_k, N_k-1, \dots, 1,\\
&\text{For } i = 1, 2, \dots, N_x, \quad \text{for } j = N_y, N_y-1, \dots, 1, \quad \text{for } k = 1, 2, \dots, N_k,\\
&\text{For } i = 1, 2, \dots, N_x, \quad \text{for } j = N_y, N_y-1, \dots, 1, \quad \text{for } k = N_k, N_k-1, \dots, 1,\\
&\text{For } i = N_x, N_x-1, \dots, 1, \quad \text{for } j = 1, 2, \dots, N_y, \quad \text{for } k = 1, 2, \dots, N_k,\\
&\text{For } i = N_x, N_x-1, \dots, 1, \quad \text{for } j = 1, 2, \dots, N_y, \quad \text{for } k = N_k, N_k-1, \dots, 1,\\
&\text{For } i = N_x, N_x-1, \dots, 1, \quad \text{for } j = N_y, N_y-1, \dots, 1 \quad \text{for } k = 1, 2, \dots, N_k,\\
&\text{For } i = N_x, N_x-1, \dots, 1, \quad \text{for } j = N_y, N_y-1, \dots, 1 \quad \text{for } k = N_k, N_k-1, \dots, 1,\\
&\qquad \vdots\\
&\text{For } k = N_z, N_z-1, \dots, 1, \quad \text{for } j = N_y, N_y-1, \dots, 1, \quad \text{for } i = N_x, N_x-1, \dots, 1.
\end{aligned}
$$

Because the algorithm of Saul'yev-type method cannot strictly preserve the total mass of the system, we adopt the following mass correction technique [15] after each temporal iteration:

$$
\phi^{n+1}_{ijk} = \phi^{n+1}_{ijk} + \frac{\sum_{p=1}^{N_x}\sum_{q=1}^{N_y}\sum_{s=1}^{N_z}(\phi^0_{pqs} - \phi^{n+1}_{pqs})}{\sum_{p=1}^{N_x}\sum_{q=1}^{N_y}\sum_{s=1}^{N_z}\sqrt{F(\phi^{n+1}_{pqs})}}\sqrt{F(\phi^{n+1}_{ijk})},
$$

which means the mass is added or subtracted at the interface transition region.

Remark 2.1. We briefly describe the technical contribution of fully discrete schemes on the CH equation. For temporal discretization, the well-known linear convex splitting approach is adopted to treat the nonlinear potential. More precisely, the nonlinear part is explicitly treated and the linear part is implicitly treated. Thus, we avoid the linearization and iteration in each time step. To avoid iteratively calculate

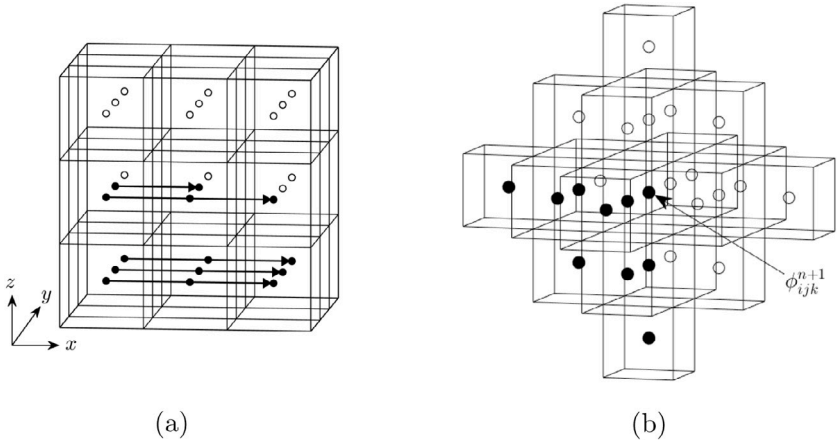

Fig. 2. Schematic illustration of (a) 3D nested loop and (b) 3D stencil.

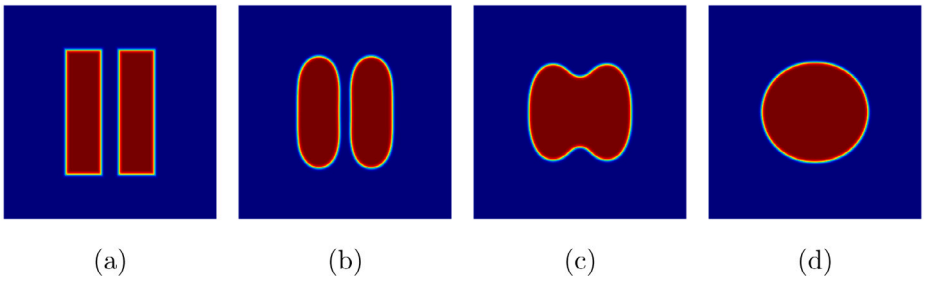

Fig. 3. Evolution of initially separated components with sharp shapes. From (a)–(d), the snapshots are at $t = 0,\ 1.56,\ 3.13$, and 11.72. Because the CH model dissipates the free energy by minimizing the interfacial length, the sharp corners become smooth and finally merge with each other to form the circular shape.

the linear terms with Laplacian operators, we consider the Saul'yev approach which is an alternating direction explicit (ADE) type method. It should be noted that the Saul'yev method has been widely used for the diffusion equation and it allows relatively large time steps to perform stable numerical computation [53,54]. The spatial discretization is performed by using the standard finite difference method with second-order accuracy in space. In each nested loop, the computation follows a step-by-step manner because all unknown variables are explicitly treated. In this sense, the numerical implementation is highly efficient.

Remark 2.2. In the existing literature, extensive stability and convergence analysis of the numerical schemes for the CH-type equations have been investigated. Please refer to [42,55–57] for some details. In the present work, we propose a novel explicit scheme for the CH equation by using the linear stabilization method and the Saul'yev finite difference discretization. Although the stability and error estimations are interesting and meaningful for the CH equation, it is not trivial for the proposed scheme because an extra mass correction technique is adopted after each computation. Herein, we aim to investigate the good performance of the proposed scheme, a detailed analysis will be conducted in future work.

3. Computational tests

In this section, we perform various numerical experiments to validate the proposed scheme. In Section 3.1, we first investigate the growth rate in short-time simulation. In Section 3.2, we show the advantage of the proposed scheme by comparing with the fully explicit scheme. The temporally accuracy and numerical stability are considered in Section 3.3. The 2D phase separation in regular and complex

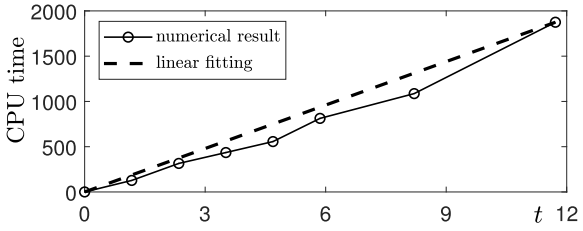

Fig. 4. Required CPU time from the initial state to the equilibrium state. Here, the solid line with open circle markers represents the CPU cost with time evolution. The open circle markers represent the specific moments. The dashed line is a linear fitting, which indicates the CPU cost is approximately linear.

domains are investigated in Sections 3.4 and 3.5. The 3D simulations are performed in Sections 3.6 and 3.7. Before the start of following contents, we use the notation [50]:

$$\epsilon = \epsilon_m = \frac{mh}{2\sqrt{2}\tanh^{-1}(0.9)} \approx 0.24015mh,$$

which indicates the diffuse interface approximately occupies m grids.

3.1. Growth rate

First of all, we consider the linearization of the one-dimensional CH equation around the solution $\phi = 0$, i.e.,

$$\phi_t = \Delta(-\phi - \epsilon^2\Delta\phi). \tag{6}$$

Let $\phi(x,t) = \alpha(t)\cos(k\pi x)$. From Eq. (6), we have

$$\alpha'(t) = (k\pi)^2(1 - (\epsilon k\pi)^2)\alpha(t). \tag{7}$$

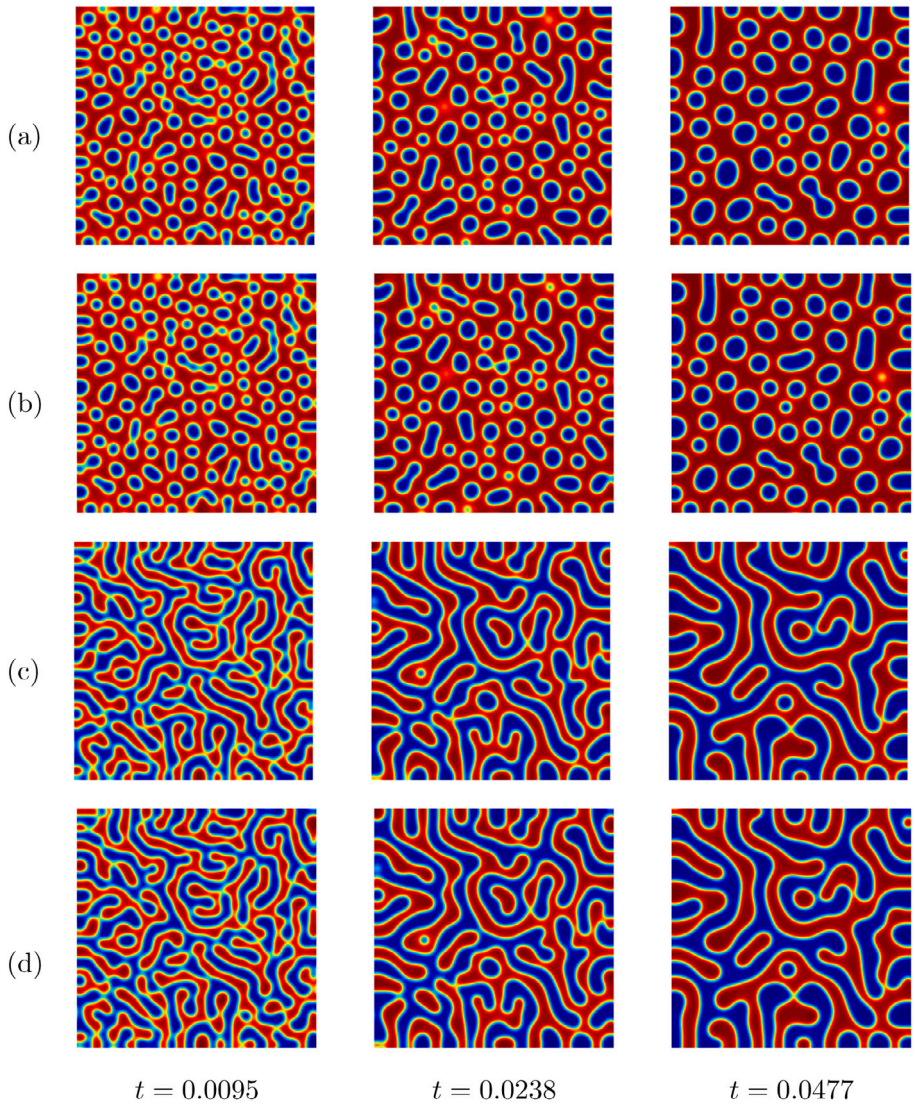

Fig. 5. Snapshots computed by the fully explicit scheme with respect to (a) $\bar{\phi} = 0.3$ and (c) $\bar{\phi} = 0$. The corresponding results computed by our proposed scheme are shown in (b) and (d). From the left to right in each column, the results are at $t = 0.0095$, 0.0238, and 0.0477. From the comparisons, it can be observed that the results obtained by the proposed scheme and fully explicit scheme are very consistent.

The solution of Eq. (7) is $\alpha(t) = \alpha(0)e^{\lambda t}$, where the analytical growth rate λ is expressed as $\lambda = (k\pi)^2(1 - (\epsilon k\pi)^2)$. Then, let

$$\hat{\lambda} = \frac{1}{T}\log\left(\frac{\|\phi^{N_t}\|_\infty^d}{\alpha(0)}\right)$$

be the computational growth rate. Here, ϕ^{N_t} is the solution at final time, $\|\cdot\|_\infty^d$ represents the discrete maximum norm. In this simulation, we use

$$\phi(x, 0) = 0.01\cos(k\pi x),$$

in the domain $\Omega = (0, 1)$. Here, $\epsilon = \epsilon_{16}$, $\alpha(0) = 0.01$, $h = 1/200$, and $\Delta t = 0.001h^2$ are used. The computations are performed until $t = 10000\Delta t$ for $k = 1, 2, \ldots, 20$. The growth rates listed in Table 1 indicate that the computational results are in close conformity with the closed-form solutions.

Table 1
Analytical and numerical growth rates with respect to various values of k.

Mode k	2	6	10	14	18
Analytical	38.9032	308.7097	627.4228	553.2427	−576.5304
Numerical	37.6525	302.7789	630.5030	571.1664	−466.6141

3.2. Comparison with the fully explicit scheme

It is well known that the restriction of the time step of the fully explicit scheme for the CH equation is of $O(h^4)$. Therefore, the fully explicit scheme is very inefficient for the CH equation. To show the efficiency of the proposed explicit scheme, we investigate the possible maximum time steps $\Delta t_{\max}$ which allow the stable computation with respect to various mesh sizes. The initial condition is

$$\phi(x, y, 0) = \tanh\left(-\frac{\max(|x + 0.15| - 0.1, |y| - 0.35)}{\sqrt{2}\epsilon}\right) + \tanh\left(-\frac{\max(|x - 0.15| - 0.1, |y| - 0.35)}{\sqrt{2}\epsilon}\right) + 1$$

in the domain $\Omega = (-1, 1)^2$. The increasingly finer mesh sizes $h = 1/16,\ 1/32,\ 1/64$, and $1/128$ are considered. For each mesh size, we use $\epsilon = \epsilon_4$. All computations are performed until the numerical equilibrium state is reached, i.e., $\|\phi^n - \phi^{n-1}\|_{L^2}^d \le 10^{-5}$, where $\|\cdot\|_{L^2}^d$ is the discrete L^2-norm. Fig. 3 displays the snapshots at specific computational

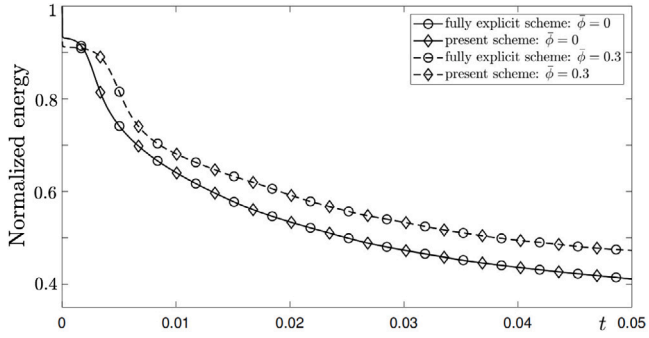

Fig. 6. Energy curves computed by the fully explicit scheme and our proposed scheme. More precisely, the solid line with open circle markers represents the energy computed by the fully explicit scheme with $\bar{\phi} = 0$. The dashed line with open circle markers represents the energy computed by the fully explicit scheme with $\bar{\phi} = 0.3$. The solid line with open diamond markers represents the energy computed by the proposed scheme with $\bar{\phi} = 0$. The dashed line with open diamond markers represents the energy computed by the proposed scheme with $\bar{\phi} = 0.3$. It can be observed that the energy curves computed by two different schemes are very consistent.

Table 2
Values of maximum time step $\Delta t_{\max}$ guaranteeing stable computation.

Mesh size h	1/16	1/32	1/64	1/128
Present scheme	7.80e−3	9.77e−4	7.32e−4	2.44e−4
Fully explicit scheme	3.05e−5	1.91e−6	1.19e−7	7.45e−9

moments with a finer mesh size $h = 1/128$, we find that the system approaches the equilibrium state by shrinking the length of interface. In Fig. 4, the required CPU time from initial state to equilibrium state is plotted by the solid line with open circle markers. The linear relationship is represented by the dashed line. It can be observed that the CPU cost is approximately linear.

Table 2 shows $\Delta t_{\max}$ for different mesh sizes. We can observe that the maximum time step of our proposed scheme is of $O(h^2)$, while the restriction of time step of fully explicit scheme is of $O(h^4)$. In actual computation, we note that $\Delta t \approx O(h^2)$ is a possible time step to perform long-time simulation.

Next, we compare the CPU time required by the proposed scheme and the fully explicit scheme. The domain $\Omega = (-1, 1)^2$ is used and the following initial condition is considered: $\phi(x, y, 0) = 0.2\cos(\pi x)\cos(\pi y)$. The other parameters are unchanged. Let a finer time step be $\delta t = 0.5h^4$, we use δt and $\Delta t = 100\delta t$ for the fully explicit scheme and our proposed scheme, respectively. The computations are conducted until $t = 32000\delta t$. We define the ratio as $Rt = CPU^e/CPU$, where CPU^e and CPU represent the total CPU time required by the fully explicit scheme and the proposed scheme, respectively. From the values listed in Table 3, we can find that our proposed scheme almost saves 100 times computational time compared to the fully explicit scheme. If mesh size is not enough, the computational cost for the grid points in the computational domain may be not dominant, thus the ratio is less than 100. However, the ratio converges to 100 as the refinement of mesh size increases.

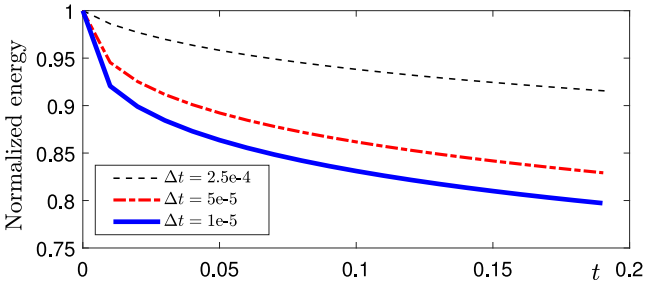

Fig. 8. Evolutions of energy curves with respect to different time steps. Here, the black dashed line, red dot–dashed line, and blue solid line corresponds to $\Delta t = 2.5$e-4, 5.0e-5, and 1.0e-5, respectively. As we can see, the energy curves are non-increasing even if larger time steps are used.

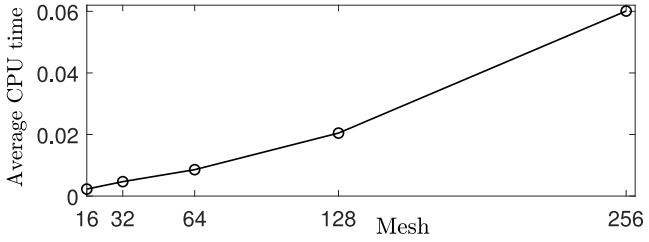

Fig. 9. Average CPU cost with respect to different mesh sizes.

Table 3
CPU costs of our proposed scheme and the fully explicit scheme.

Mesh size h	1/16	1/32	1/64	1/128
CPU^e	4.157	12.952	51.734	228.754
Rt	33.256	42.327	84.810	101.308
CPU	0.125	0.306	0.610	2.258

Then, we simulate the pattern of phase separation in the domain $\Omega = (-1, 1)^2$. The initial condition is defined to be $\phi(x, y, 0) = \bar{\phi} + 0.2\,\mathrm{rand}(x, y)$, where $\mathrm{rand}(x, y)$ represents the random number between -1 and 1. Here, $h = 1/64$ and $\epsilon = \epsilon_4$ are used. For the fully explicit scheme, we use $\delta t = h^4$ and $\Delta t = 25\delta t$ is used for our proposed scheme. In Fig. 5(a) and (c), we display the snapshots calculated by the fully explicit scheme with respect to $\bar{\phi} = 0.3$ and $\bar{\phi} = 0$, respectively. The corresponding results calculated by our proposed scheme are displayed in Fig. 5(b) and (d). As we can observe, the patterns are similar with each other. Let us define the discrete energy at nth time level as

$$\mathcal{E}^d(\phi^n) = h^2 \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left[F(\phi_{ij}^n) + \frac{\epsilon^2}{2}\left(\frac{(\phi_{i+1,j}^n - \phi_{ij}^n)^2}{h^2} + \frac{(\phi_{i,j+1}^n - \phi_{ij}^n)^2}{h^2}\right)\right].$$

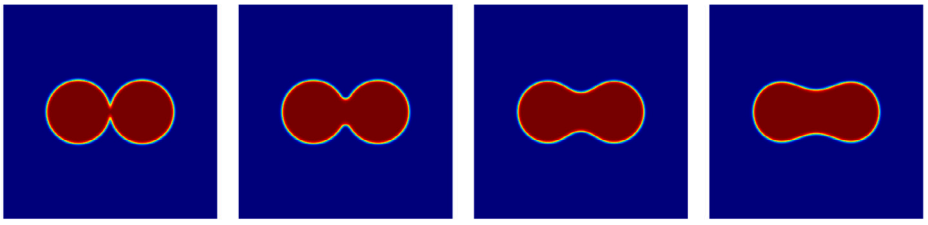

Fig. 7. (a) is the initial state. The snapshots at $t = 0.2$ with respect to $\Delta t = 2.5$e-4, 5.0e-5, and 1.0e-5 are shown in (b), (c), and (d), respectively. Here, we can find that the numerical solution is stable even if larger time steps are used. To obtain an accurate result, a smaller time step is still necessary.

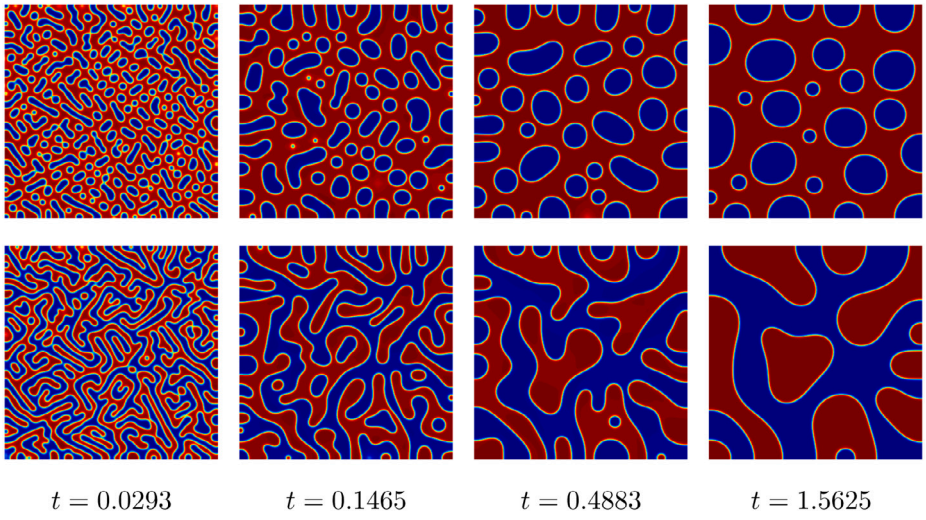

Fig. 10. Snapshots with respect to $\bar{\phi} = 0.2$ (the top row) and $\bar{\phi} = 0$ (the bottom row). The computational moments are illustrated under each figure. With even concentration (i.e., $\bar{\phi} = 0$), the co-continuous pattern evolves. On the contrary, the droplet patterns can be observed when one component is dominant. These coarsening processes occur due to the minimization of free energy of the system.

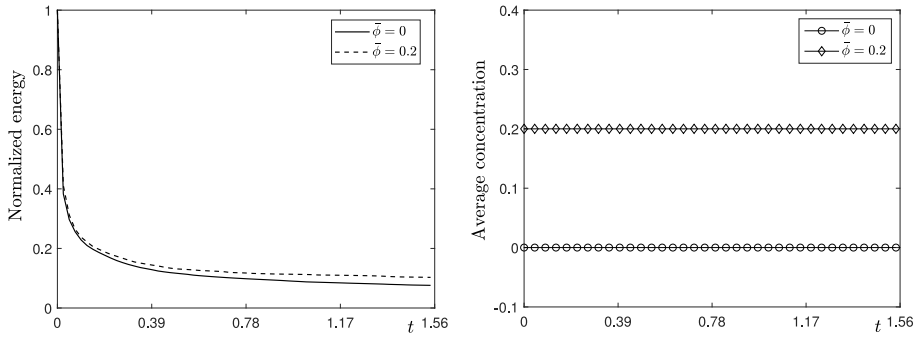

Fig. 11. Evolutions of (a) normalized energy curves and (b) average concentrations. More precisely, the solid and dashed lines represent the energy curves with respect to $\bar{\phi} = 0$ and $\bar{\phi} = 0.2$ in (a). In (b), the average concentrations for $\bar{\phi} = 0$ and $\bar{\phi} = 0.2$ are represented by the solid line with open circle markers and solid line with open diamond markers, respectively. We observe that the free energy curves are non-increasing and concentrations are conserved in time.

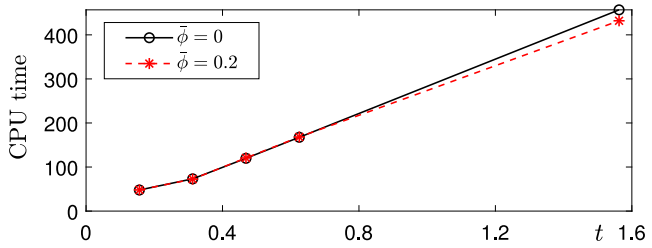

Fig. 12. Required CPU time for 2D phase separation at specific moments. Here, the black solid line with open circle markers represents the CPU cost with respect to $\bar{\phi} = 0$. The red dashed line with star markers represents the CPU cost with respect to $\bar{\phi} = 0.2$. It can be observed that the CPU costs are similar and approximately linear.

The evolutions of normalized energy $\mathcal{E}^d(\phi^n)/\mathcal{E}^d(\phi^0)$ are shown in Fig. 6. The results indicate the energy evolutions are almost same even if the time step for our proposed scheme is 25 times larger than that for the fully explicit scheme. We list the CPU times required by the fully explicit scheme and the proposed scheme in Table 4. It can be observed that the proposed scheme obviously saves computational costs.

Table 4
Required CPU costs for the fully explicit scheme and the proposed scheme.

Average concentration	$\bar{\phi} = 0$	$\bar{\phi} = 0.3$
Fully explicit scheme	8072.8398	8103.7432
Proposed scheme	1257.1004	1301.8472

3.3. Accuracy, stability, and computational complexity

Although the linear convex splitting method in Eq. (1) has first-order accuracy in time, the theoretical accurate analysis for the proposed scheme is still an open question because we add an extra mass correction technique, Eq. (3), after each computation. We herein perform a numerical experiment to verify the temporal accuracy. The domain $\Omega = (-1, 1)^2$ is used. The initial condition is defined as

$$\phi(x, y, 0) = \tanh\left(\frac{0.3 - \sqrt{(x-0.3)^2 + y^2}}{\sqrt{2}\epsilon}\right) + \tanh\left(\frac{0.3 - \sqrt{(x+0.3)^2 + y^2}}{\sqrt{2}\epsilon}\right) + 1. \tag{8}$$

We set $h = 1/128$ and $\epsilon = \epsilon_6$. Because the exact solution is hard to find, we use the reference solution by using a small enough time step $\Delta t^f = 0.01h^2$. We use increasingly coarser time steps $\Delta t = 16\Delta t^f$, $32\Delta t^f$,

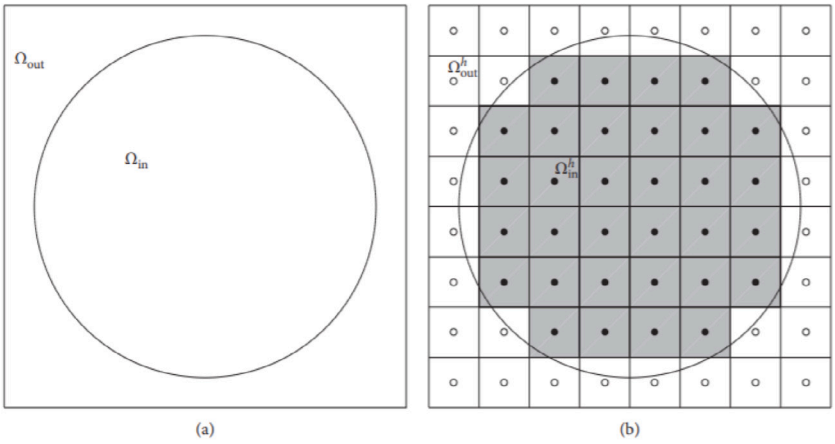

Fig. 13. Schematic illustrations of actually computational domain Ω_{in}. The cube region is the whole domain Ω. The exterior region is $\Omega_{out} = \Omega/\Omega_{in}$. We only update ϕ in Ω_{in} (shaded region) and set $\phi = 0$ in Ω_{out}.
Source: The schematic diagrams are adopted from Shin et al. [58] with the permission of Hindawi press.

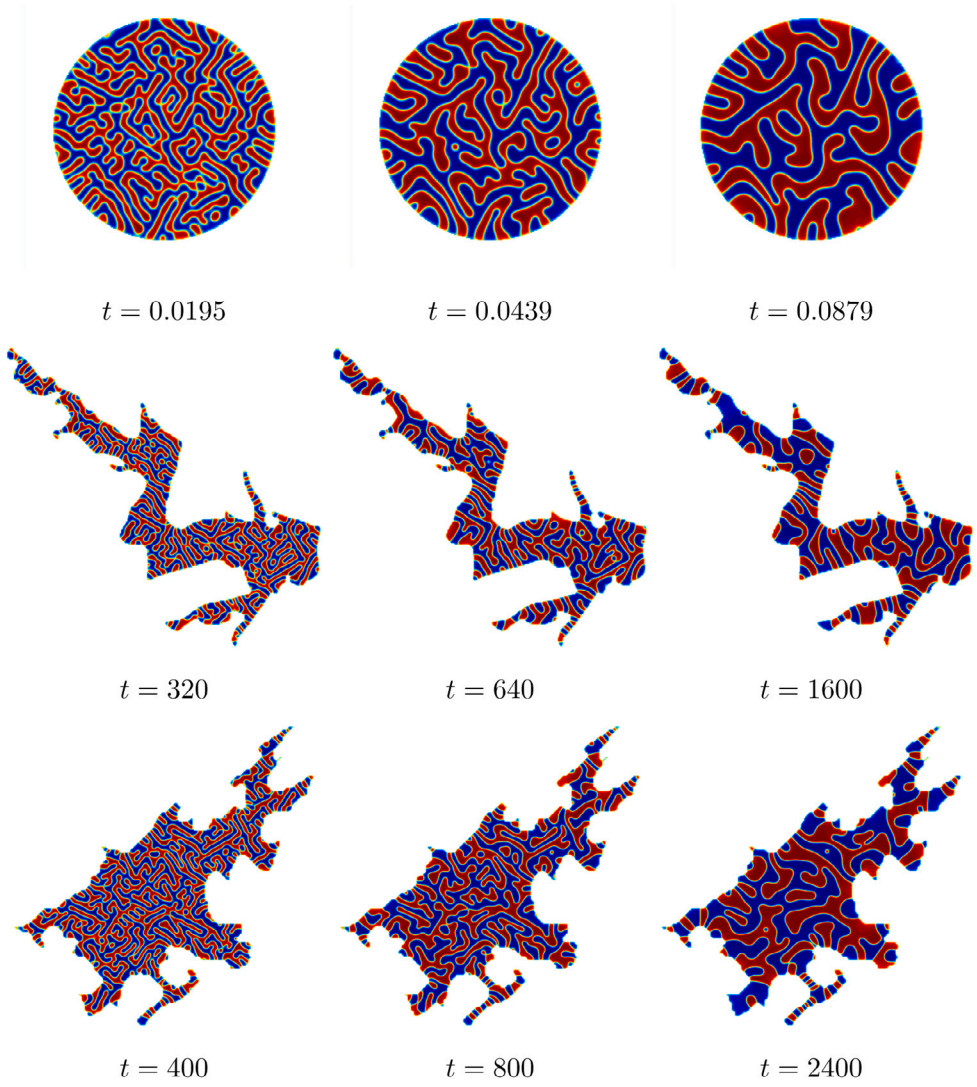

Fig. 14. Snapshots of phase separation in complex regions. The top, middle, and bottom rows illustrate the results in a disk region, Kielder Water in northern Britain, and Lake Inari in Fennoscandia, respectively.

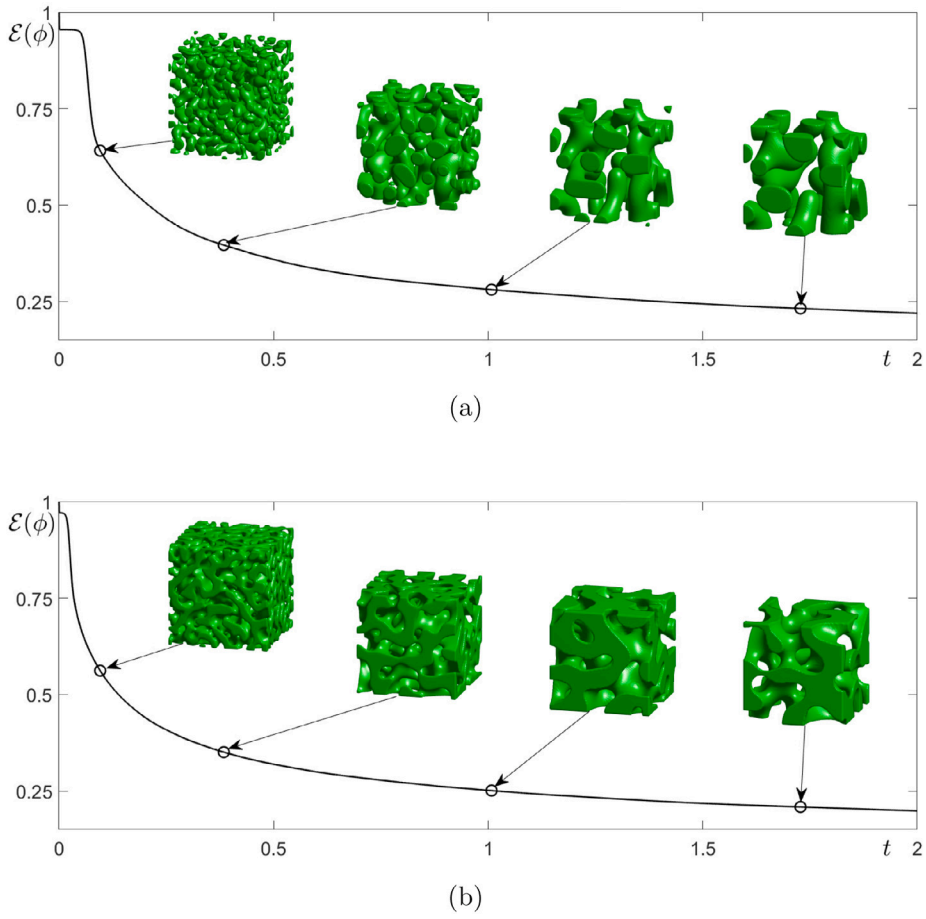

Fig. 15. Evolutions of discrete energy functional with respect to different average concentrations: (a) $\bar{\phi} = -0.4$ and (b) $\bar{\phi} = 0$. Here, the solid lines represent the energy curves, the open circle markers represent the specific moments. The insets show the snapshots of zero level set of ϕ at $t = 0.096$, 0.384, 1.008, and 1.728 (from the left to right). The energies are decreasing with the evolutions of phase coarsening.

Table 5
L^2-errors and convergence rates with respect to different time steps at $t = 0.0013$. The reference numerical solution is obtained by using $\Delta t^f = 0.01h^2$.

Time step Δt	$64\Delta t^f$		$32\Delta t^f$		$16\Delta t^f$
L^2-error	2.45e−2		1.44e−2		6.30e−3
rate		0.77		1.19	

and $64\Delta t^f$ to perform the simulations until $t = 0.0013$. Table 5 shows the L^2-errors and convergence rates. As we can see, the proposed scheme still achieves first-order accuracy in time.

For the proposed Saul'yev scheme with mass correction, the unconditional energy stability of the CH equation is not trivial to estimate. In this sense, we investigate the numerical stability by performing the simulation with the initial condition, Eq. (8), and different time steps. Fig. 7(a) shows the initial state, (b), (c), and (d) show the snapshots at $t = 0.2$ with Δt = 2.5e-4, 5.0e-5, and 1.0e-5, respectively. As we can observe, the numerical results do not blow up and this indicates that the proposed scheme is stable even if larger time steps are adopted. Fig. 8 plots the evolutions of free energy, we can observe that the energies are decreasing for all different time steps.

Next, we estimate the computational complexity of the proposed scheme which is a Gauss–Seidel type method. For the convenience of description, the cost of performing one relaxation sweep is defined as a work unit (WU) [59]. In the whole computation, the computational complexity of the proposed scheme is estimated to be less than 2 WU because the cost of the mass correction step is expected to be less than WU. We also perform the simulations with initial condition (8) and

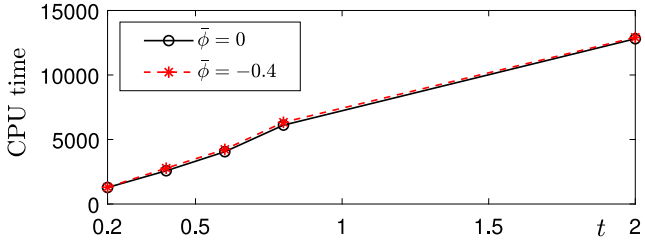

Fig. 16. Required CPU costs for the 3D phase separation at specific moments. Here, the black solid line with open circle markers represents the CPU cost with respect to $\bar{\phi} = 0$. The red dashed line with star markers represents the CPU cost with respect to $\bar{\phi} = -0.4$. It can be observed that the CPU costs are similar and approximately linear.

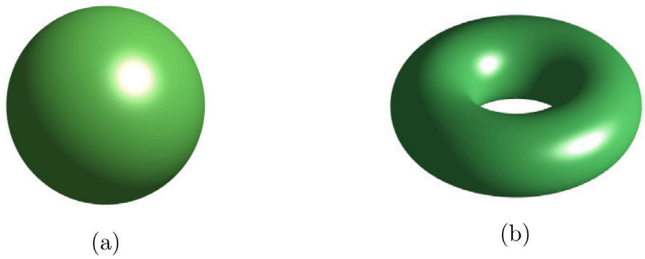

Fig. 17. Computational domains with (a) sphere and (b) torus shapes.

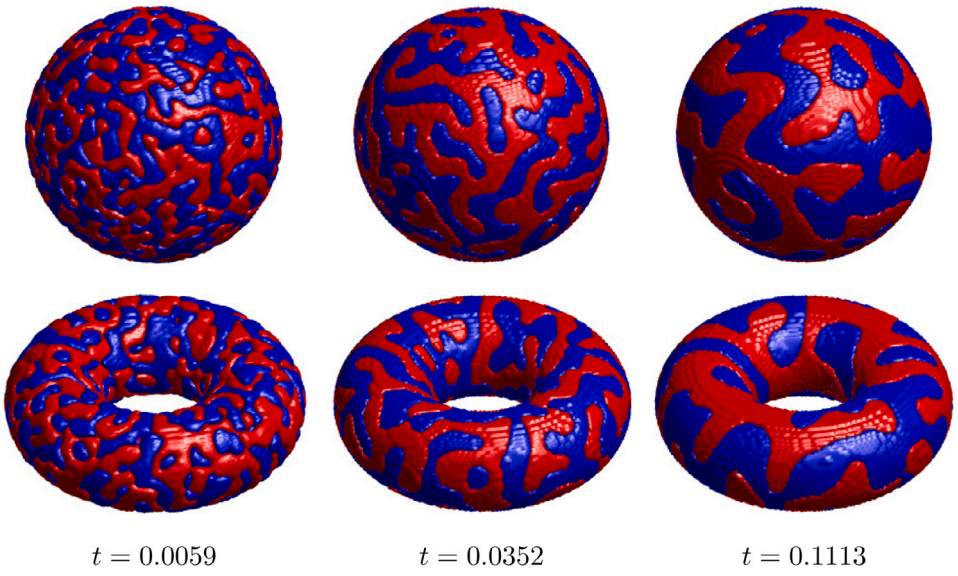

Fig. 18. Snapshots of 3D phase separation in sphere domain (top row) and torus domain (bottom row). The computational moments are illustrated under each figure. The red and blue regions are occupied by two different moments. With time evolution, the co-continuous patterns can be observed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

different mesh sizes: 16, 32, 64, 128, and 256. The average CPU cost with respect to different mesh sizes is plotted in Fig. 9. With the refinement of mesh size, the average CPU cost obviously increases.

3.4. Coarsening dynamics with different concentrations

Next, we investigate the phase separations with different concentrations by using our proposed scheme. In this subsection, we consider the following initial condition $\phi(x, y, 0) = \bar{\phi} + 0.01\text{rand}(x, y)$ in the domain $\Omega = (-1, 1)^2$. The mesh size $h = 1/128$ and the time step $\Delta t = h^2$ are used.

The simulations are performed until $t = 1.5625$. The top and the bottom rows in Fig. 10 illustrate the snapshots with respect to $\bar{\phi} = 0.2$ and $\bar{\phi} = 0$, respectively. As we can see, the lamella patterns form as $\bar{\phi} = 0$ and the droplet patterns form as $\bar{\phi} = 0.2$, these phenomena are typical in the CH dynamics and have been reported in previous works [60]. In Fig. 11(a) and (b), we plot the evolutions of normalized energy and average concentrations, it can be found that the discrete total energy is dissipative and the mean of ϕ^n is constant. In Fig. 12, we plot the evolution of CPU costs with respect to $\bar{\phi} = 0$ (black solid line with open circle markers) and $\bar{\phi} = 0.2$ (red dashed line with star markers). We can observe that the CPU costs are approximately linear. The difference of concentration does not obviously affect the CPU time.

3.5. Coarsening dynamics in complex domains

Next, we investigate the phase separation in various domains with complex shapes. Here, we consider three different regions, such as a disk with radius $R = 0.8$, Kielder Water in northern Britain, and Lake Inari in northern Fennoscandia, which are embedded in the whole domains $\Omega_1 = (-1, 1)^2$, $\Omega_2 = (0, 500) \times (0, 467)$, and $\Omega_3 = (0, 500) \times (0, 494)$, respectively. We use $h = 1/128$ for the simulation in the disk region and use $h = 1$ for the simulations in the rest two regions. The initial condition is defined to be $\phi(x, y, 0) = 0.01\text{rand}(x, y)$. The other parameters are $\Delta t = h^2$ and $\epsilon = \epsilon_4$. In the exterior regions, we fix $\phi(x, y, t) = 0$, which effectively plays a role of the Dirichlet boundary condition and the computations are only updated inside the irregular regions. The schematic illustrations are shown in Fig. 13. Fig. 13(a) illustrates the actually computational domain Ω_{in} and exterior region Ω_{out}. The solid line indicates the boundary of complex region. We only update ϕ in the shaded regions and set $\phi = 0$ in the white regions (see Fig. 13(b)). The top, middle, and bottom rows in Fig. 14 display the snapshots of phase separation in three different regions. It can be observed that the coarsening phenomena appear in various complex regions. In Table 6, the required CPU costs with respect to the phase separation in disk region, Kielder Water in northern Britain, and Lake Inari in Fennoscandia are listed.

Table 6
Required CPU costs for the 2D phase separation in disk regions, Kielder Water in northern Britain, and Lake Inari in Fennoscandia.

Complex regions:	disk	Kielder Water	Lake Inari
CPU time:	9.8553	46.5872	90.4428

3.6. Three-dimensional coarsening dynamics with different concentrations

In this subsection, we investigate three-dimensional phase separation with different concentrations. We use the following initial condition $\phi(x, y, z, 0) = \bar{\phi} + 0.1\ \text{rand}(x, y, z)$ in the domain $\Omega = (-1, 1)^3$. We use $h = 1/50$, $\Delta t = 0.01h$, $\epsilon = \epsilon_4$ and simulations are performed until $t = 2$. Fig. 15 shows the time discrete energy and numerical solutions with (a) $\bar{\phi} = -0.4$ and (b) $\bar{\phi} = 0$, respectively. In Fig. 16, we plot the required CPU costs with respect to $\bar{\phi} = 0$ (black solid line with open circle markers) and $\bar{\phi} = -0.4$ (red dashed line with star markers). We find that the CPU costs are approximately linear and the difference of concentration does not obviously affect the computational costs.

3.7. Three-dimensional phase separation in arbitrarily shaped domains

In this subsection, we consider the phase separation in various shaped domains in three-dimensional space. The computational domain is set as $\Omega = (0, 1)^3$. The initial condition is defined as

$$\phi(x, y, z, 0) = 0.1\text{rand}(x, y, z). \tag{9}$$

Here, we use $h = 1/128$, $\Delta t = 0.2h^2$, $\epsilon = \epsilon_4$. First, we consider the sphere and torus domains which are defined by

$$\psi(x, y, z) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2} - 0.4 < 0, \tag{10}$$

$$\psi(x, y, z) = \sqrt{(\sqrt{(x - 0.5)^2 + (y - 0.5)^2} - 0.3)^2 + (z - 0.5)^2} - 0.15 < 0 \tag{11}$$

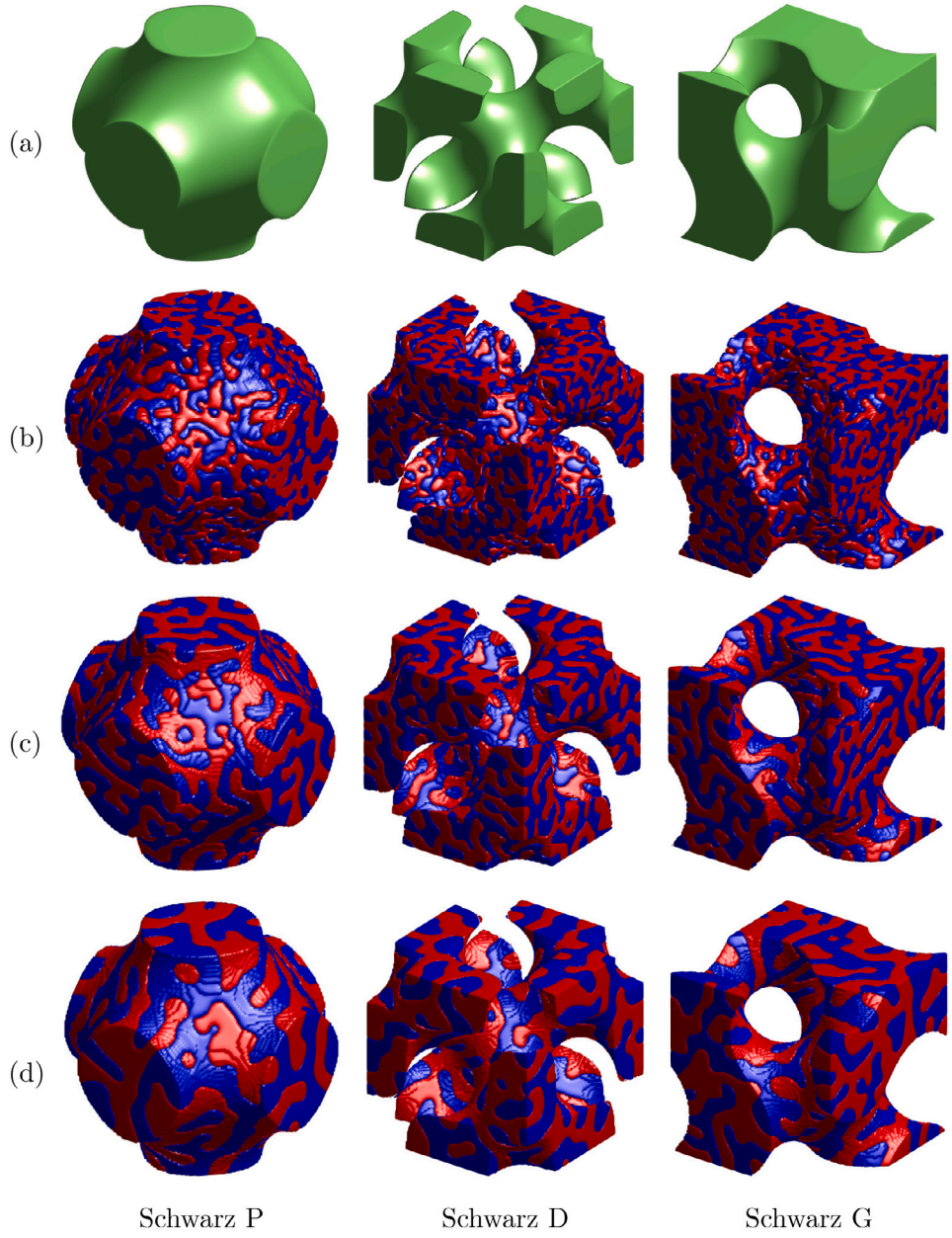

Fig. 19. 3D phase separation in complex domains. Here, the Schwarz P, Schwarz D, and Schwarz G domains are shown in the left, middle, and right columns of (a). From (b)–(d), the snapshots are at $t = 0.0059$, 0.0352, and 0.1113. The red and blue regions are occupied by two different components. With time evolution, the co-continuous patterns can be observed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The schematic illustrations of sphere and torus domains are shown in Fig. 17(a) and (b), respectively.

The top and bottom rows of Fig. 18 display the phase evolutions at different moments. In the simulations, we set $\phi(x, y, z, t) = 0$ outside the sphere and torus domains.

Next, we consider the following three marker functions:

$$\psi(x, y, z) = \cos(2\pi x) + \cos(2\pi y) + \cos(2\pi z) < 0,$$
$$\psi(x, y, z) = \cos(2\pi x)\cos(2\pi y)\cos(2\pi z) - \sin(2\pi x)\sin(2\pi y)\sin(2\pi z) < 0,$$
$$\psi(x, y, z) = \sin(2\pi x)\cos(2\pi y) + \sin(2\pi z)\cos(2\pi x) + \sin(2\pi y)\cos(2\pi z) < 0,$$

which represent the Schwarz P, Schwarz D, and Schwarz G domains, respectively [61]. In the simulations, we set $\phi(x, y, z, t) = 0$ outside the complex domains. The top row of Fig. 19 illustrates the Schwarz P, Schwarz D, and Schwarz G domains. As we can observe, the coarsening dynamics can be well simulated in various complex domains. In Table 7, we plot the required CPU costs with respect to the 3D phase separation in sphere, torus, Schwarz P, Schwarz D, and Schwarz G domains.

Table 7
Required CPU costs for the 3D phase separation in sphere, torus, Schwarz P, Schwarz D, and Schwarz G domains.

Complex regions:	sphere	torus	Schwarz P	Schwarz D	Schwarz G
CPU time:	1389.4363	949.2869	3226.2981	3396.3449	3252.1819

4. Discussion

In practical computation, a temporally first-order accurate scheme is not satisfactory. Therefore, it is natural to consider temporally

higher-order accurate numerical schemes such as a temporally second-order accurate explicit conservative Saul'yev finite difference numerical scheme. We suggest a possible extension of the current approach to a temporally second-order method based on the Crank–Nicolson/Adams–Bashforth (CNAB) scheme. To derive a temporally second-order accurate explicit conservative Saul'yev finite difference scheme, let us start from a two-dimensional CNAB scheme for the CH equation:

$$\frac{\phi_{ij}^{n+1}-\phi_{ij}^{n}}{\Delta t}=\Delta_d\left[\frac{3}{2}\left((\phi_{ij}^{n})^3-\phi_{ij}^{n}\right)-\frac{1}{2}\left((\phi_{ij}^{n-1})^3-\phi_{ij}^{n-1}\right)+\frac{\epsilon^2}{2}\left(\Delta_d\phi_{ij}^{n+1}+\Delta_d\phi_{ij}^{n}\right)\right]. \tag{12}$$

Then, we have one case of eight cases of nested loops:

$$\text{For } j=1,2,\dots,N_y, \text{ for } i=1,2,\dots,N_x, \tag{13}$$

$$\begin{aligned}\phi_{ij}^{n+1} = \frac{1}{r}\Bigg\{ & \frac{\phi_{ij}^{n}}{\Delta t}+\Delta_d\left[\frac{3}{2}\left((\phi_{ij}^{n})^3-\phi_{ij}^{n}\right)-\frac{1}{2}\left((\phi_{ij}^{n-1})^3-\phi_{ij}^{n-1}\right)+\frac{\epsilon^2}{2}\Delta_d\phi_{ij}^{n}\right]\\ & +\frac{\epsilon^2}{2h^4}\Big[\phi_{i+2,j}^{n}+\phi_{i-2,j}^{n+1}+\phi_{i,j+2}^{n}+\phi_{i,j-2}^{n+1}+2(\phi_{i+1,j+1}^{n}+\phi_{i+1,j-1}^{n}\\ & +\phi_{i-1,j+1}^{n}+\phi_{i-1,j-1}^{n+1})-8(\phi_{i+1,j}^{n}+\phi_{i-1,j}^{n+1}+\phi_{i,j+1}^{n}+\phi_{i,j-1}^{n+1})+10\phi_{ij}^{n}\Big]\Bigg\},\end{aligned}$$

where $r=1/\Delta t+5\epsilon^2/h^4$. In addition, we take the mass conservative correction step, Eq. (3). We note that this scheme is a two-step scheme, i.e., we need the numerical solutions (ϕ^n, ϕ^{n-1}) at times $t=n\Delta t$ and $t=(n-1)\Delta t$ to obtain the numerical approximation (ϕ^{n+1}) at time $t=(n+1)\Delta t$. Given the initial condition (ϕ^0), we compute (ϕ^1) at time $t=\Delta t$ by using the first-order scheme with a subcycling technique. For example, we take four time step iterations with a smaller time $\Delta t/4$ to obtain the numerical solution at $t=\Delta t$ from the given initial condition.

5. Conclusions

The CH equation was extensively used for simulating phase separation of binary mixtures and fluid flows with the interface. The most popular numerical methods for solving the CH equation were fully or semi-implicit. Therefore, iterative techniques were needed in general. Because iterative algorithms required extra computational costs and lead to tedious theoretical works, we herein proposed a novel explicit scheme for solving the CH equation.

First of all, the linear convex splitting approach was adopted to treat the nonlinear part explicitly. In this sense, we avoided the nonlinear iteration. Then, the Saul'yev method was used to discrete the linear Laplacian terms. In each nested loop, the computation was straightforward because all unknown variables were explicitly treated. The main advantages of the proposed scheme are as follows:

(i) The phase-field variable can be directly computed without any iterative calculation;

(ii) The numerical stability can be satisfied even if larger time steps are used;

(iii) The mass conservation of the CH equation can be satisfied;

(iv) It is easy to implement the simulations in complex domains.

Various numerical experiments indicated that the proposed scheme not only allowed stable computation for relatively large time steps but also worked well for simulating the phase separation in regular and complex domains.

Note that the CH equation has been extensively used to capture the interface in multi-phase fluid flows [7,8,11,12]. In upcoming works, the proposed scheme will be adopted to investigate the CH fluids. In addition, the proposed scheme can be used to solve the CH-type equations in image inpainting [16], multi-component systems [62–65], and volume reconstruction, etc.

CRediT authorship contribution statement

Junxiang Yang: Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review and editing. **Yibao Li:** Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Chaeyoung Lee:** Formal analysis, Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review and editing. **Hyun Geun Lee:** Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review and editing. **Soobin Kwak:** Investigation, Visualization, Writing – original draft, Writing – review and editing. **Youngjin Hwang:** Visualization, Writing – original draft, Writing – review and editing. **Xuan Xin:** Investigation, Writing – original draft, Writing – review and editing. **Junseok Kim:** Conceptualization, Formal analysis, Funding acquisition, Investigation Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review and editing.

Declaration of competing interest

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

Acknowledgment

Y.B. Li is supported by the Fundamental Research Funds for the Central Universities, China (No. XTR042019005). C. Lee was supported by the National Research Foundation (NRF), Korea, under project BK21 FOUR. H.G. Lee was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1C1C1011112). The corresponding author (J.S. Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A2C1003053). The authors are grateful to the reviewers whose valuable suggestions and comments significantly improved the quality of this article.

References

[1] Cahn JW, Hilliard JE. Free energy of a non-uniform system I. Interfacial free energy. J Chem Phys 1958;28(2):258–67.
[2] Chen W, Wang C, Wang X, Wise SM. Positivity-preserving, energy stable numerical schemes for the Cahn–Hilliard equation with logarithmic potential. J Comput Phys X 2019;3:100031.
[3] van den Berg JB, Williams JF. Validation of the bifurcation diagram in the 2D Ohta–Kawasaki problem. Nonlinearity 2017;30(4):1584.
[4] Li Q, Mei L. Efficient, decoupled, and second-order unconditionally energy stable numerical schemes for the coupled Cahn–Hilliard system in copolymer/homopolymer mixtures. Comput Phys Commun 2021;260:107290.
[5] Guo Z, Yu F, Wise S, Lowengrub J. A diffuse domain method for two-phase flows with large density ratio in complex geometries. J Fluid Mech 2021;907:A38.
[6] Chen W, Wang C, Wang S, Wang X, Wise SM. Energy stable numerical schemes for ternary Cahn–Hilliard system. J Sci Comput 2020;84(2):1–36.
[7] Chiu PH. A coupled phase field framework for solving incompressible two-phase flows. J Comput Phys 2019;392(1):115–40.
[8] Li HL, Liu HR, Ding H. A fully 3D simulation of fluid–structure interaction with dynamic wetting and contact angle hysteresis. J Comput Phys 2020;420(1):109709.
[9] Zhang JT, Liu HR, Ding H. Head-on collision of two immiscible droplets of different components. Phys Fluid 2020;32(8):082106.
[10] Liang H, Shi BC, Chai ZH. Lattice Boltzmann modeling of three-phase incompressible flows. Phys Rev E 2016;93(1):013308.
[11] Budiana EP, Pranowo, Indrato, Deendarlianto. Meshless numerical model based on radial basis function (RBF) method to simulate the Rayleigh–Taylor instability (RTI). Comput Fluid 2020;201:104472.
[12] Jia HE, Guo YY, Li M, Huang YQ, Feng GR. Decoupled, energy stable numerical scheme for the Cahn–Hilliard–Hele–Shaw system with logarithmic Flory–Huggins potential. Commun Comput Phys 2020;27(4):1053–75.
[13] Gui Z, Xu K. Progress of discrete unified gas-kinetic scheme for multiscale flows. Adv Aerodyn 2021;3(1):1–42.
[14] Amiri F, Ziaei-Rad S, Valizadeh N, Rabczuk T. On the use of local maximum entropy approximants for Cahn–Hilliard phase-field models in 2D domains and on surfaces. Comput Methods Appl Mech Engrg 2019;346:1–24.

[15] Jeong D, Li Y, Lee C, Yang J, Kim J. A conservative numerical method for the Cahn–Hilliard equation with generalized mobilities on curved surfaces in three-dimensional space. Commun Comput Phys 2020;27:412–30.

[16] Brkic AL, Mitrovic D, Novak A. On the image inpainting problem from the viewpoint of a nonlocal Cahn–Hilliard type equation. J Adv Res 2020;25:67–76.

[17] Kim J. Phase-field models for multi-component fluid flows. Commun Comput Phys 2012;12:613–61.

[18] Kim J, Lee S, Choi Y, Lee SM, Jeong D. Basic principles and practical applications of the Cahn–Hilliard equation. Math Probl Eng 2016;2016:9532608.

[19] Wise SM, Lowengrub J, Cristini V. An adaptive multigrid algorithm for simulating solid tumor growth using mixture models. Math Comput Modelling 2011;53:1–20.

[20] Asle Zaeem M, El Kadiri H, Horstemeyer MF, Khafizov M. Effects of internal stresses and intermediate phases on the coarsening of coherent precipitates: a phase-field study. Curr Appl Phys 2012;12:570–80.

[21] Farshbaf-Shaker MH, Heinemann C. A phase field approach for optimal boundary control of damage processes in two-dimensional viscoelastic media. Math Model Method Appl Sci 2015;25(14):2749–93.

[22] Jeong D, Lee S, Kim J. An efficient numerical method for evolving microstructures with strong elastic inhomogeneity. Model Simulat Mater Sci Eng 2015;23:045007.

[23] Zhang C, Gao P, Li E, Ding H. On the compound sessile drops: configuration boundaries and transitions. J Fluid Mech 2021;917:A37.

[24] Xu ZL, Chen JY, Liu HR, Chandra Sahu K, Ding H. Motion of self-rewetting drop on a substrate with a constant temperature gradient. J Fluid Mech 2021;915:A116.

[25] Mu K, Qiao R, Si T, Cheng X, Ding H. Interfacial instability and transition of jetting and dripping modes in a co-flow focusing process. Phys Fluid 2021;33:052118.

[26] Yang J, Kim J. Phase-field simulation of Rayleigh instability on a fibre. Int J Multiph Flow 2018;105:84–90.

[27] Yang J, Li Y, Kim J. Numerical simulations of the dynamics of axisymmetric compound liquid threads with a phase-field model. Eur J Mech B Fluids 2021;89:203–16.

[28] Yang J, Lee HG, Kim J. Side wall boundary effect on the Rayleigh–Taylor instability. Eur J Mech B Fluids 2021;85:361–74.

[29] Liu HR, Zhang C, Gao P, Lu XY, Ding H. On the maximal spreading of impacting compound drops. J Fluid Mech 2018;854:R6.

[30] Dehghan M, Addaszadeh M. The meshless local collocation method for solving multi-dimensional Cahn–Hilliard, swift–hohenberg and phase field crystal equations. Eng Anal Bound Elem 2017;78:49–64.

[31] Addaszadeh M, Khodadadian A, Parvizi M, Dehghan M, Heitzinger C. A direct meshless local collocation method for solving stochastic Cahn–Hilliard–Cook and stochastic Swift–Hohenberg equations. Eng Anal Bound Elem 2019;98:253–64.

[32] Liu H, Cheng A, Wang H, Zhao J. Time-fractional Allen–Cahn and Cahn–Hilliard phase-field models and their numerical investigation. Comput Math Appl 2018;76(8):1876–92.

[33] Baumann H, Hanschke T. Computation of solutions to linear difference and differential equations with a prescribed asymptotic behavior. Adv Differ Equ 2021;2021(1):1–27.

[34] Luo F, Tang T, Xie H. Parameter-free time adaptivity based on energy evolution for the Cahn–Hilliard equation. Commun Comput Phys 2016;19(5):1542–63.

[35] Liu Z, Li X. The exponential scalar auxiliary variable (e-SAV) approach for phase field models and its explicit computing. SIAM J Sci Comput 2020;42(3):B630–55.

[36] Liu Z, Li X. Efficient modified techniques of invariant energy quadratization approach for gradient flows. Appl Math Lett 2019;98:206–14.

[37] Dong L, Wang C, Zhang H, Zhang Z. A positivity-preserving second-order BDF scheme for the Cahn–Hilliard equation with variable interfacial parameters. Commun Comput Phys 2020;28(3):967–98.

[38] Shen J, Yang X. Numerical approximations of Allen–Cahn and Cahn–Hilliard equations. Discrete Cont Dyn-B 2010;28(4):1669–91.

[39] Li D, Qiao Z, Tang T. Characterizing the stabilization size for semi-implicit Fourier-spectral method to phase field equations. SIAM J Numer Anal 2016;54(3):1653–81.

[40] Li D, Qiao Z. On second order semi-implicit Fourier spectral methods for 2D Cahn–Hilliard equations. J Sci Comput 2017;70:301–41.

[41] Li D, Qiao Z. On the stabilization size of semi-implicit Fourier-spectral methods for 3D Cahn–Hilliard equations. Commun Math Sci 2017;15(6):1489–506.

[42] Guo J, Wang C, Wise S, Yue X. An H^2 convergence of a second-order convex-splitting, finite difference scheme for the three-dimensional Cahn–Hilliard equation. Commun Math Sci 2016;14:489–515.

[43] Chen W, Conde S, Wang C, Wang X, Wise S. A linear energy stable scheme for a thin film model without slope selection. J Sci Comput 2012;52:546–62.

[44] Li W, Chen W, Wang C, Yan Y, He R. A second order energy stable linear scheme for a thin film model without slope selection. J Sci Comput 2018;76:1905–37.

[45] Cheng K, Qiao Z, Wang C. A third order exponential time differencing numerical scheme for no-slope-selection epitaxial thin film model with energy stability. J Sci Comput 2019;81:154–85.

[46] Chen W, Li W, Luo Z, Wang C, Wang X. A stabilized second order exponential time differencing multistep method for thin film growth model without slope selection. ESAIM: M2AN 2020;54:727–50.

[47] Chen W, Li W, Wang C, Wang S, Wang X. Energy stable high order linear ETD multi-step methods for gradient flows: application to thin film epitaxy. Res Math Sci 2020;7:13.

[48] Meng X, Qiao Z, Wang C, Zhang Z. Artificial regularization parameter analysis for the no-slope-selection epitaxial thin film model. CSIAM Trans Appl Math 2020;1:441–62.

[49] Hao Y, Huang Q, Wang C. A third order BDF energy stable linear scheme for the no-slope-selection thin film model. Commun Comput Phys 2020;29(3):905–29.

[50] Jeong D, Choi Y, Kim J. A benchmark problem for the two- and three-dimensional Cahn–Hilliard equations. Commun Nonlinear Sci Numer Simul 2018;61:149–59.

[51] Dehghan M. The one-dimensional heat equation subject to a boundary integral specification. Chaos Solitons Fractals 2017;32(2):661–75.

[52] Li Y, Choi Y, Kim J. Computationally efficient adaptive time step method for the Cahn–Hilliard equation. Comput Math Appl 2017;73:1855–64.

[53] Chin SA. Understanding Saul'yev-type unconditionally stable schemes from exponential splitting. Numer Meth Part Differ Equ 2014;30(6):1961–83.

[54] Samalerk P, Pohai N. A saulyev explicit scheme for an one-dimensional advection-diffusion-reaction equation in an opened uniform flow stream. Thai J Math 2020;18(2):677–83.

[55] Diegel A, Wang C, Wise S. Stability and convergence of a second order mixed finite element method for the Cahn–Hilliard equation. J Numer Anal 2016;36:1867–97.

[56] Feng W, Guan Z, Lowengrub J, Wang C, Wise S, et al. A uniquely solvable,energy stable numerical scheme for the functionalized Cahn–Hilliard equation and its convergence analysis. J Sci Comput 2018;76:1938–67.

[57] Cheng K, Wang C, Wise S. A weakly nonlinear energy stable scheme for the strongly anisotropic Cahn–Hilliard system and its convergence analysis. J Comput Phys 2020;405:109109.

[58] Shin J, Choi Y, Kim J. The Cahn–Hilliard equation with generalized mobilities in complex geometries. Math Probl Eng 2019;2019:1710270.

[59] De Oliveira F, Franco SR, Pinto V. The effect of multigrid parameters in a 3D heat diffusion equation. Appl Mech Rev 2018;23(1):213–21.

[60] Cheng K, Feng W, Wang C, Wise SM. An energy stable fourth order finite difference scheme for the Cahn–Hilliard equation. J Comput Appl Math 2019;362:574–95.

[61] Jeong D, Yang J, Kim J. A practical and efficient numerical method for the Cahn–Hilliard equation in complex domains. Commun Nonlinear Sci Numer Simul 2019;73:217–28.

[62] Yuan X, Liang H, Chai Z, Shi B. Phase-field-based lattice Boltzmann model for immiscible incompressible N-phase flows. Phys Rev E 2020;101:063310.

[63] Zhou S, Xie YM. Numerical simulation of three-dimensional multicomponent Cahn–Hilliard systems. Int J Mech Sci 2021;198:106349.

[64] Yang J, Kim J. A variant of stabilized-scalar auxiliary variable (s-SAV) approach for a modified phase-field surfactant model. Comput Phys Commun 2021;261:107825.

[65] Tavakoli R. Unconditionally energy stable time stepping scheme for Cahn–Morral equation: Application to multi-component spinodal decomposition and optimal space tiling. J Comput Phys 2016;304:441–64.