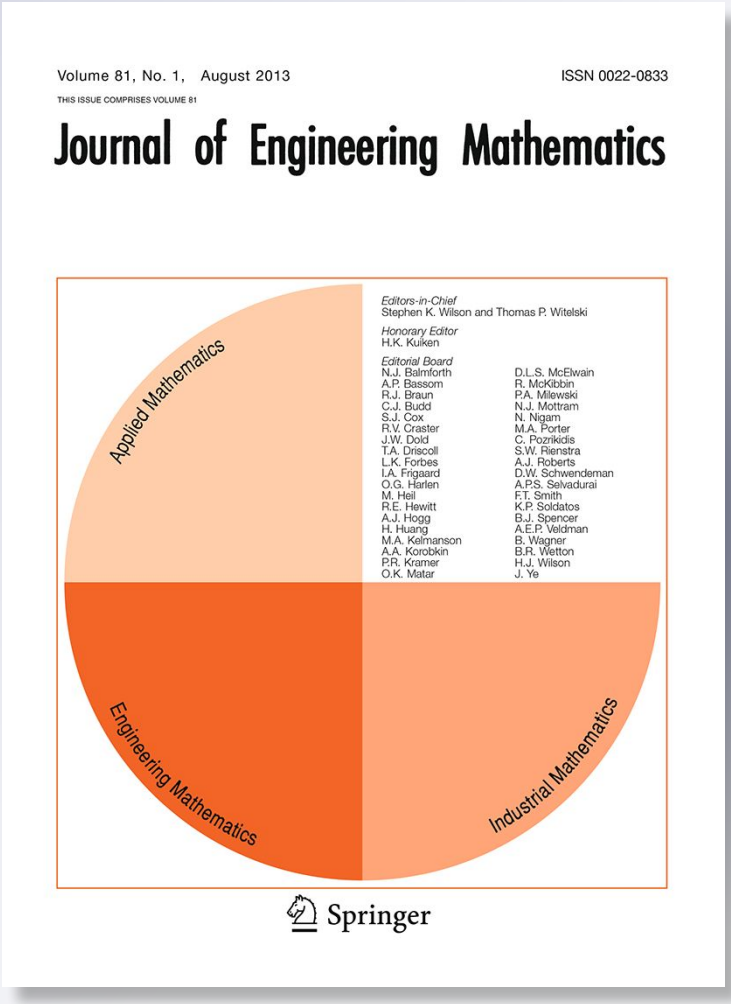
A conservative finite difference scheme for the N-component Cahn–Hilliard system on curved surfaces in 3D

Junxiang Yang, Yibao Li, Chaeyoung Lee, Darae Jeong & Junseok Kim

Journal of Engineering Mathematics

ISSN 0022-0833
Volume 119
Number 1

J Eng Math (2019) 119:149-166
DOI 10.1007/s10665-019-10023-9



Springer

J Eng Math (2019) 119:149–166
https://doi.org/10.1007/s10665-019-10023-9

A conservative finite difference scheme for the N-component Cahn–Hilliard system on curved surfaces in 3D

Junxiang Yang · Yibao Li · Chaeyoung Lee · Darae Jeong · Junseok Kim

Received: 17 March 2019 / Accepted: 1 October 2019 / Published online: 7 November 2019

Abstract This paper presents a conservative finite difference scheme for solving the N-component Cahn–Hilliard (CH) system on curved surfaces in three-dimensional (3D) space. Inspired by the closest point method (Macdonald and Ruuth, SIAM J Sci Comput 31(6):4330–4350, 2019), we use the standard seven-point finite difference discretization for the Laplacian operator instead of the Laplacian–Beltrami operator. We only need to independently solve $(N-1)$ CH equations in a narrow band domain around the surface because the solution for the Nth component can be obtained directly. The N-component CH system is discretized using an unconditionally stable nonlinear splitting numerical scheme, and it is solved by using a Jacobi-type iteration. Several numerical tests are performed to demonstrate the capability of the proposed numerical scheme. The proposed multicomponent model can be simply modified to simulate phase separation in a complex domain on 3D surfaces.

Keywords Closest point method · Conservative scheme · N-component Cahn–Hilliard equation · Narrow band domain

1 Introduction

The Cahn–Hilliard (CH) equation was originally proposed to model spinodal decomposition of binary alloys [1]. This equation can be applied to model many important phenomena such as tumor growth simulation [2,3], topology optimization [4], 3D volume reconstruction, surface diffusion motion [5], phase separation [6], surfactant dynamics [7], and multiphase fluid flows [4]. For the numerical, mathematical, and physical derivations, and the practical applications of the CH equation, please refer to [8,9] and the references therein.

The multicomponent CH system can have more applications than the binary CH equations, e.g., multicomponent buoyancy-driven mixing simulation [10], double emulsion formation [11], multicomponent Kelvin–Helmholtz

J. Yang · C. Lee · J. Kim (✉)
Department of Mathematics, Korea University, Seoul 02841, Republic of Korea
e-mail: cfdkim@korea.ac.kr

Y. Li
School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China

D. Jeong
Department of Mathematics, Kangwon National University, Gangwon-do 24341, Republic of Korea

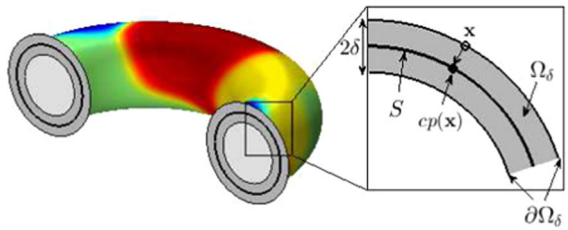


Fig. 1 Schematic representation of S, Ω_δ, $\partial\Omega_\delta$, and cp($\mathbf{x}$). (Color figure online)

instability [12], and multiphase alloys [13]. Lee and Kim [14] proposed a nonlinear multigrid method for multicomponent CH system. This approach reduces the CPU time consumed in computation. Then Lee et al. [15] presented an unconditionally gradient stable scheme for solving the N-component CH equation; this approach only needs to solve $(N-1)$ CH equations. Li et al. [16] extended the multicomponent CH system to complex domains, which can easily treat the contact angle boundary condition and can be accurately applied to the adaptive mesh method. Jeong et al. [17] proposed a practical and efficient method for solving the binary CH system in complex domains using the modified ternary CH model. Although the aforementioned methods can efficiently simulate the dynamics of a multicomponent CH system, the computation is limited to flat planes.

Recently, numerical methods for solving the binary CH equation on curved surfaces have been proposed. In [18], the authors rigorously studied a fully discrete finite element method (FEM) for solving the CH equation on a general surface. In [19], an efficient direct discretization method was developed for solving the CH equation on unstructured triangular surfaces. The authors in [20] considered the CH equation on evolving surfaces. As an extension, the authors in [21] presented a second-order unconditionally stable scheme for solving the phase-field crystal equation on various closed surfaces. In [22], the authors developed a surface FEM for obtaining the numerical solution of the CH equation on a hypersurface Γ in $\mathbb{R}^3$. The authors in [23] developed a finite difference method (FDM) for solving the CH equation on implicit surfaces defined using a level set function. In a recent study [24], the authors presented an efficient, conservative, and accurate numerical method for solving the CH equation with generalized mobilities on curved surfaces in 3D space. However, to the best of our knowledge, there are few studies that involves the multicomponent CH system on curved surfaces in a 3D space.

As an extended research of [24], we use a simple and computationally efficient conservative FDM for the N-component CH system on curved surfaces in 3D space. We employ the unconditionally gradient stable nonlinear splitting numerical scheme for solving the $(N-1)$ CH equations [25] and solve the discrete equations on a 3D narrow band domain by using a Jacobi-type iteration. For mass conservation, we apply a mass correction algorithm. We perform several numerical tests to demonstrate that the proposed method is efficient, simple, and conservative. Note that the signed distance function can only be defined for closed surfaces; therefore, all the curved surfaces in this study are closed.

The paper is organized in the following manner. We describe the N-component CH system with generalized mobilities on a 3D narrow band domain in Sect. 2. In Sect. 3, we present the numerical scheme. We provide the computational results in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 The N-component CH system

Let $\mathbb{R}^3$ and $\Omega_\delta = \{\mathbf{y} | \mathbf{y} = \mathbf{x} + \theta\mathbf{n}(\mathbf{x}) \text{ for } |\theta| < \delta,\ \mathbf{x} \in S\}$ be a narrow band domain on a surface S, where $\mathbf{n}(\mathbf{x})$ is a normal vector. Figure 1 shows the schematic illustrations of S, Ω_δ, and $\partial\Omega_\delta$.

We consider an N-component admissible system in $\mathbb{R}^3$. Let $\mathbf{c} = (c_1,\ c_2,\ \ldots,\ c_N)$ be the phase variable of each component in the system. The admissible state belongs to the Gibbs N-simplex,

$$G := \left\{ \mathbf{c} \in \mathbb{R}^N \;\middle|\; \sum_{l=1}^{N} c_l = 1,\ 0 \le c_l \le 1 \right\}, \tag{1}$$

where l is an index denoting the lth component in the entire system. The Ginzburg–Landau-type free energy can be written as

$$\mathcal{E}(\mathbf{c}) = \int_{\Omega} \sum_{l=1}^{N} \left(F(c_l) + \frac{\epsilon^2}{2} |\nabla c_l|^2 \right) \mathrm{d}\mathbf{x}, \tag{2}$$

where $F(c_l) = 0.25 c_l^2 (c_l - 1)^2$ and ϵ is constant. The governing equation for c_l is given by the N-component CH system with generalized mobilities:

$$\frac{\partial c_l(\mathbf{x}, t)}{\partial t} = \nabla \cdot [M_l(\mathbf{x}, t) \nabla \mu_l(\mathbf{x}, t)], \quad \mathbf{x} \in \Omega_\delta, \ t > 0, \tag{3}$$

$$\mu_l(\mathbf{x}, t) = F'(c_l(\mathbf{x}, t)) - \epsilon^2 \Delta c_l(\mathbf{x}, t) + \beta(\mathbf{c}), \tag{4}$$

where $c_l(\mathbf{x}, t)$ is the mass fraction of the lth component of the N-component mixture. $M_l(\mathbf{x}, t)$ is the generalized mobility, and we set it as 1 for simplicity. $\beta(\mathbf{c}) = -(1/N) \sum_{l=1}^{N} F'(c_l)$ is a Lagrange multiplier to satisfy the total mass conservation given in Eq. (1) [26]. We notice that only $(N-1)$ phase variables need to be calculated because $c_N = 1 - \sum_{l=1}^{N-1} c_l$. We use a pseudo-Neumann boundary condition:

$$c_l(\mathbf{x}, t) = c_l(\mathrm{cp}(\mathbf{x}), t) \quad \text{on} \quad \partial\Omega_\delta, \tag{5}$$

where $\mathrm{cp}(\mathbf{x}) \in S$ for $\mathbf{x} \in \partial\Omega_\delta$ is the closest point [27] (Fig. 1). For some details of the closest point method, see [28]. For a more detailed introduction and application of multicomponent CH system, please refer to [26]. Note that the equations in this study are solved in the narrow band domain around the surface instead of being solved directly on the surface. If we take a small enough domain width δ, then the numerical solution of Eqs. (3) and (4), with the pseudo-Neumann boundary condition in Eq. (5), yields a c_l that is constant along the direction normal to the surface. Thus, we can use the standard Laplacian operator instead of the Laplace–Beltrami operator [29] in the narrow band domain.

3 Numerical solution algorithm

We describe the numerical solution for the N-component CH system on Ω_δ. We discretize the N-component CH system in $\Omega = (a, b) \times (c, d) \times (e, f)$ embedding Ω_δ. Let N_x, N_y, and N_z be integers, $h = (b-a)/N_x = (d-c)/N_y = (f-e)/N_z$ be the space step, and $\Omega^h = \{\mathbf{x}_{ijk} = (x_i, y_j, z_k) = (a + hi, c + hj, e + hk) | \ 0 \le i \le N_x, 0 \le j \le N_y, 0 \le k \le N_z\}$ be the discrete domain. Let $c_{l,ijk}^n$ be approximation of $c_l(x_i, y_j, z_k, n\Delta t)$, where Δt is the temporal step. Let $S = \{\mathbf{x} \in \mathbb{R}^3 | \ \psi(\mathbf{x}) = 0\}$, where $\psi : \mathbb{R}^3 \to \mathbb{R}$ is the signed distance function to S. Let $\Omega_\delta^h = \{\mathbf{x}_{ijk} | \ |\psi_{ijk}| < \delta\}$ be the discrete domain with $\delta > \sqrt{3}h$. Note that the closest points of all the boundary points are calculated by using trilinear interpolation. Therefore, the discrete narrow band domain Ω_δ^h must include all the interpolation stencils; this means that we need to take $\delta > \sqrt{3}h$; otherwise, the boundary point may be included as an interpolation stencil. A schematic illustration is shown in Fig. 2.

Let $\|c_l\|_{L^2(\Omega_\delta^h)} = \sqrt{(1/\#\Omega_\delta^h) \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} c_{l,ijk}^2}$, where $\#\Omega_\delta^h$ is the cardinality of Ω_δ^h. Let $\partial\Omega_\delta^h = \{\mathbf{x}_{ijk} | \ I_{ijk} | \nabla_h I_{ijk}| \neq 0\}$ be the discrete boundary, where $\nabla_h I_{ijk} = (I_{i+1,jk} - I_{i-1,jk}, I_{i,j+1,k} - I_{i,j-1,k}, I_{ij,k+1} - I_{ij,k-1})/(2h)$. Here, $I_{ijk} = 0$ if $\mathbf{x}_{ijk} \in \Omega_\delta^h$; otherwise $I_{ijk} = 1$. We use the unconditionally stable discretization of the N-component CH system given by Eqs. (3) and (4) [16]:

$$\frac{c_{l,ijk}^{n+1} - c_{l,ijk}^n}{\Delta t} = \Delta_h \mu_{l,ijk}^{n+1} \quad \text{for} \quad \mathbf{x}_{ijk} \in \Omega_\delta^h, \tag{6}$$

$$\mu_{l,ijk}^{n+1} = F'\left(c_{l,ijk}^{n+1}\right) + \frac{c_{l,ijk}^{n+1}}{4} - \frac{c_{l,ijk}^n}{4} + \beta\left(\mathbf{c}_{ijk}^n\right) + \epsilon^2 \Delta_h c_{l,ijk}^{n+1}, \tag{7}$$

where $\beta\left(\mathbf{c}_{ijk}^n\right) = -(1/N) \sum_{l=1}^{N} F'\left(c_{l,ijk}^n\right) = -(1/N) \sum_{l=1}^{N} [(c_{l,ijk}^n)^3 - 1.5\left(c_{l,ijk}^n\right)^2 + 0.5 c_{l,ijk}^n]$; further, we split the nonlinear term $F'(c_{l,ijk})$ into the difference of two convex functions: $F_1 = F'(c_{l,ijk}) + c_{l,ijk}/4$ and $F_2 = c_{l,ijk}/4$;

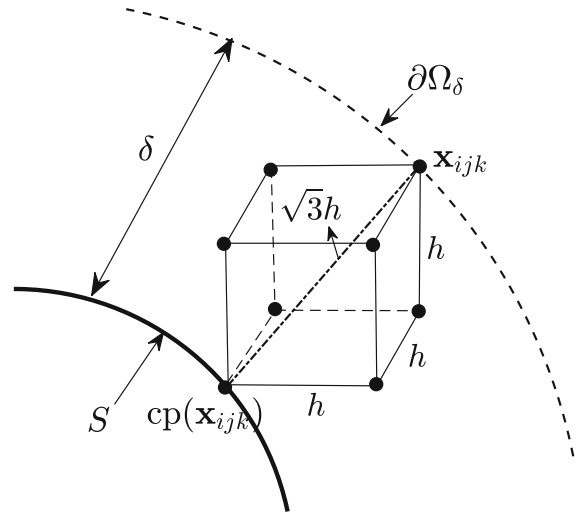


Fig. 2 Schematic illustration of boundary point $\mathbf{x}_{ijk}$ and its closest point $\mathrm{cp}(\mathbf{x}_{ijk})$. Note that the boundary point is included as an interpolation stencil. Therefore, the relationship $\delta > \sqrt{3}h$ must be satisfied

then, we treat F_1 implicitly and F_2 explicitly. The boundary condition is denoted by

$$c_{l,ijk}^{n+1} = c_l^{n+1}(\mathrm{cp}(\mathbf{x}_{ijk})) \quad \text{and} \quad \mu_{l,ijk}^{n+1} = \mu_l^{n+1}(\mathrm{cp}(\mathbf{x}_{ijk})) \quad \text{for } \mathbf{x}_{ijk} \in \partial\Omega_\delta^h.$$

Here, Δ_h is the standard seven-point discrete Laplacian operator defined as $\Delta_h c_{l,ijk} = (c_{l,ij,k-1} + c_{l,ij,k+1} + c_{l,i,j-1,k} + c_{l,i,j+1,k} + c_{l,i-1,jk} + c_{l,i+1,jk} - 6c_{l,ijk})/h^2$. Let

$$\mathrm{cp}(\mathbf{x}_{ijk}) = \mathbf{x}_{ijk} - \frac{\nabla_h|\psi_{ijk}|}{|\nabla_h|\psi_{ijk}||^2}|\psi_{ijk}|. \tag{8}$$

Since $\mathrm{cp}(\mathbf{x}_{ijk})$ is typically not a grid point, $c_l^n(\mathrm{cp}(\mathbf{x}_{ijk}))$ is obtained using the trilinear interpolation method. For each $\mathbf{x}_{ijk} \in \partial\Omega_\delta^h$, we compute $\mathrm{cp}(\mathbf{x}_{ijk})$ using Eq. (8) and derive the cube cell, $[x_p, x_{p+1}) \times [y_q, y_{q+1}) \times [z_r, z_{r+1})$, containing the point $\mathrm{cp}(\mathbf{x}_{ijk})$. Let $(\alpha_1, \alpha_2, \alpha_3) = \mathrm{cp}(\mathbf{x}_{ijk}) - (x_p, y_q, z_r)$, then

$$\begin{aligned} c_l(\mathrm{cp}(\mathbf{x}_{ijk})) = \big[&(h-\alpha_1)(h-\alpha_2)(h-\alpha_3)c_{l,pqr} + \alpha_1(h-\alpha_2)(h-\alpha_3)c_{l,p+1,qr} \\ &+ (h-\alpha_1)\alpha_2(h-\alpha_3)c_{l,p,q+1,r} + \alpha_1\alpha_2(h-\alpha_3)c_{l,p+1,q+1,r} \\ &+ (h-\alpha_1)(h-\alpha_2)\alpha_3 c_{l,pq,r+1} + \alpha_1(h-\alpha_2)\alpha_3 c_{l,p+1,q,r+1} \\ &+ (h-\alpha_1)\alpha_2\alpha_3 c_{l,p,q+1,r+1} + \alpha_1\alpha_2\alpha_3 c_{l,p+1,q+1,r+1}\big]/h^3. \end{aligned}$$

Note that we can use a high-order polynomial interpolation with a thicker narrow domain as followed in [30].

To solve the implicit discrete equations (6) and (7), we use the Jacobi iteration. For a given numerical solution $c_{l,ijk}^n$ and $\mu_{l,ijk}^n$, assume that $c_{l,ijk}^{n+1,1} = c_{l,ijk}^n$ and $\mu_{l,ijk}^{n+1,1} = \mu_{l,ijk}^n$; then, iterate the following Jacobi iterative equations (9) and (10) until a stopping criterion is satisfied, i.e., $\|c_l^{n+1,m+1} - c_l^{n+1,m}\|_{L^2(\Omega_\delta^h)} < \mathrm{tol}$. Here, $c_l^{n+1,m}$ and $c_l^{n+1,m+1}$ are the iterative solutions after the mth and $(m+1)$th rounds of Jacobi iterations. For all $\mathbf{x}_{ijk} \in \Omega_\delta^h$,

$$\begin{aligned} &\frac{c_{l,ijk}^{n+1,m+1}}{\Delta t} + \left(\frac{6}{h^2}\right)\mu_{l,ijk}^{n+1,m+1} \\ &= \frac{c_{l,ijk}^n}{\Delta t} + \frac{\mu_{l,i+1,jk}^{n+1,m} + \mu_{l,i-1,jk}^{n+1,m} + \mu_{l,i,j+1,k}^{n+1,m}}{h^2} + \frac{\mu_{l,i,j-1,k}^{n+1,m} + \mu_{l,ij,k+1}^{n+1,m} + \mu_{l,ij,k-1}^{n+1,m}}{h^2}, \\ &\quad - \left[3\big(c_{l,ijk}^{n+1,m}\big)^2 + \frac{6\epsilon^2}{h^2} - 3c_{l,ijk}^{n+1,m} + 0.75\right]c_{l,ijk}^{n+1,m+1} + \mu_{l,ijk}^{n+1,m+1} \end{aligned} \tag{9}$$

$$= -2\left(c_{l,ijk}^{n+1,m}\right)^3 + 1.5\left(c_{l,ijk}^{n+1,m}\right)^2 - \frac{\epsilon^2}{h^2}\left(c_{l,i+1,jk}^{n+1,m} + c_{l,i-1,jk}^{n+1,m} + c_{l,i,j+1,k}^{n+1,m}\right)$$
$$+ \frac{\epsilon^2}{h^2}\left(c_{l,i,j-1,k}^{n+1,m} + c_{l,ij,k+1}^{n+1,m} + c_{l,ij,k-1}^{n+1,m}\right) + \beta\left(\mathbf{c}_{ijk}^{n}\right) - 0.25c_{l,ijk}^{n}. \tag{10}$$

Then, we provide the main procedure of the Jacobi-type iteration, where Eqs. (9) and (10) are first rewritten into the following matrix forms:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}\begin{pmatrix} c_{l,ijk}^{n+1,m+1} \\ \mu_{l,ijk}^{n+1,m+1} \end{pmatrix} = \begin{pmatrix} E \\ F \end{pmatrix}.$$

Here, $A = 1/\Delta t$, $B = 6/h^2$, $C = -\left[3\left(c_{l,ijk}^{n+1,m}\right)^2 + 6\epsilon^2/h^2 - 3c_{l,ijk}^{n+1,m} + 0.75\right]$, $D = 1$, E and F represent the right-hand terms in Eqs. (9) and (10), respectively. The solutions of $c_{l,ijk}^{n+1,m+1}$ and $\mu_{l,ijk}^{n+1,m+1}$ can be calculated by

$$\begin{pmatrix} c_{l,ijk}^{n+1,m+1} \\ \mu_{l,ijk}^{n+1,m+1} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1}\begin{pmatrix} E \\ F \end{pmatrix}.$$

The Jacobi-type iteration in one time step can be described by the following three steps:
Step 1 Satisfy the initial assumption: $c_{l,ijk}^{n+1,0} = c_{l,ijk}^{n}$ and $\mu_{l,ijk}^{n+1,0} = \mu_{l,ijk}^{n}$.
Step 2 Update $c_{l,ijk}^{n+1,m}$ and $\mu_{l,ijk}^{n+1,m}$ using the matrix above to obtain $c_{l,ijk}^{n+1,m+1}$ and $\mu_{l,ijk}^{n+1,m+1}$ for the whole domain.
Step 3 Check the L^2 norm: $\left\|c_l^{n+1,m+1} - c_l^{n+1,m}\right\|_{L^2(\Omega_\delta^h)}$; if it is less than the given tolerance tol, we stop the iteration. Otherwise, *Step* 2 is continued.
The procedure given here completes the Jacobi-type iteration in one time step. To satisfy the conservation property on curved surfaces, we require

$$\frac{1}{\#\Omega_\delta^h}\sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h} c_{l,ijk}^{n+1} = \frac{1}{\#\Omega_\delta^h}\sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h} c_{l,ijk}^{n} = \cdots = \frac{1}{\#\Omega_\delta^h}\sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h} c_{l,ijk}^{0}. \tag{11}$$

Let $c_{l,ijk}^{n+1} = c_{l,ijk}^{n+1,*} + \alpha^*\sqrt{F\left(c_{l,ijk}^{n+1,*}\right)}$, where $c_{l,ijk}^{n+1,*}$ is the converged solution obtained by solving Eqs. (9) and (10). To satisfy Eq. (11), α^* is computed as following

$$\alpha^* = \sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h}\left(c_{l,ijk}^{0} - c_{l,ijk}^{n+1,*}\right) \Big/ \sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h}\sqrt{F\left(c_{l,ijk}^{n+1,*}\right)}.$$

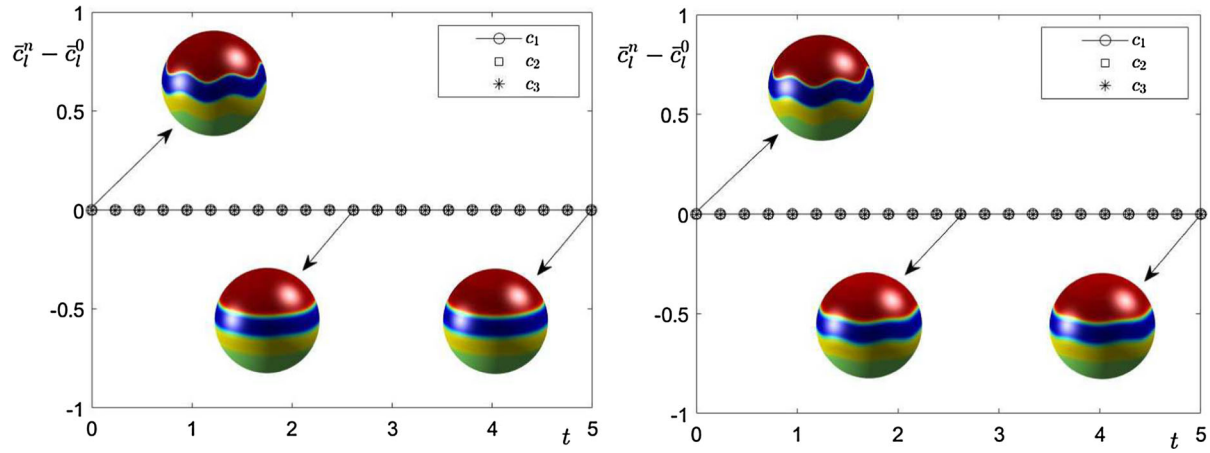
For the correction step, we have

$$c_{l,ijk}^{n+1} = c_{l,ijk}^{n+1,*} + \frac{\sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h}\left(c_{l,ijk}^{0} - c_{l,ijk}^{n+1,*}\right)}{\sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h}\sqrt{F\left(c_{l,ijk}^{n+1,*}\right)}}\sqrt{F\left(c_{l,ijk}^{n+1,*}\right)}. \tag{12}$$

Here, the form $\sqrt{F(c_l)}$ is used because we want to add or subtract the mass fraction at the interfacial transition region, and $\sqrt{F(c_l)}$ is an analytic form for the hyperbolic tangent interfacial transition profile. We note that the mass correction algorithm for binary CH equation was proposed in [24] to satisfy mass conservation. Here, we extend the correction step (12) for the N-component CH system to achieve the mass conservation of $(N-1)$ components; thus, the mass conservation of Nth component is naturally satisfied.

4 Numerical results

Several computational experiments including mass conservation are performed to demonstrate the capability of the proposed numerical scheme. If we take ϵ as $\epsilon_m = (mh)/[4\sqrt{2}\tanh^{-1}(0.9)]$, then we have a transition layer of thickness mh [31]. Unless otherwise specified, we use $M(\phi) = 1$, $\epsilon = \epsilon_3$, $\delta = 1.1\sqrt{3}h$, and tol $= 0.001$ in all numerical simulations.



(a) $\Delta t = 0.1h$ **(b)** $\Delta t = h$

Fig. 3 Temporal evolutions of $\bar{c}_l^n - \bar{c}_l^0$ with **a** $\Delta t = 0.1h$ and **b** $\Delta t = h$. (Color figure online)

4.1 Mass conservation

First, we investigate the conservation of mass. The surface is defined as the zero iso-surface of $\psi(x, y, z) = \sqrt{(x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2} - 0.4$ on the whole domain $(0, 1) \times (0, 1) \times (0, 1)$. The initial conditions are

$$c_1(x, y, z, 0) = 0.5 + 0.5 \tanh\left(\frac{-0.4\theta + \frac{3\pi r}{8} - 0.03\cos(2\pi\phi)}{\sqrt{2}\epsilon}\right),$$

$$c_2(x, y, z, 0) = 0.5 + 0.5 \tanh\left(\frac{-0.4\theta + \frac{1\pi r}{2} - 0.03\cos(2\pi\phi)}{\sqrt{2}\epsilon}\right) - c_1(x, y, z, 0),$$

$$c_3(x, y, z, 0) = 0.5 + 0.5 \tanh\left(\frac{-0.4\theta + \frac{5\pi r}{8} - 0.03\cos(2\pi\phi)}{\sqrt{2}\epsilon}\right) - c_2(x, y, z, 0),$$

$$c_4(x, y, z, 0) = 1 - c_1(x, y, z, 0) - c_2(x, y, z, 0) - c_3(x, y, z, 0),$$

where $r = \sqrt{(x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2}$, $\theta = \cos^{-1}((z-0.5)/r)$, $\phi = \tan^{-1}((y-0.5)/(x-0.5))$ and $x \neq 0.5$. We use $h = 1/80$ and two different time steps, $\Delta t = 0.1h$ and $\Delta t = h$, in our simulation. The computation is performed until $t = 5$. The average value of each mass fraction of c_1, c_2, and c_3 is defined as

$$\bar{c}_l = \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} \frac{c_{l,ijk}}{\#\Omega_\delta^h}, \quad l = 1, 2, 3.$$

Figure 3a, b shows the temporal evolutions of $\bar{c}_l^n - \bar{c}_l^0$, where $l = 1,\ 2,\ 3$ with $\Delta t = 0.1h$ and $\Delta t = h$. The embedded small figures are the corresponding phase configurations, where c_1, c_2, and c_3 are in the red, deep blue, and yellow regions, respectively. Table 1 shows the average mass fraction values for two different time steps. We can find that the mass conservation is well satisfied, and the conservative error does not appear even if a larger time step is used. By comparing the table with Fig. 3a and b, we can observe that the interfaces almost reach the equilibrium (flat) state if we use $\Delta t = 0.1h$. However, the interfaces are not flat if a larger time step $\Delta t = h$ is used. This phenomenon indicates that using a smaller time step is better for obtaining a more accurate solution.

Table 1 Average values of mass fractions with two different time steps

t	0	$1000\Delta t$	$2000\Delta t$	$3000\Delta t$	$4000\Delta t \quad (\Delta t = 0.1h)$
t	0	$100\Delta t$	$200\Delta t$	$300\Delta t$	$400\Delta t \quad (\Delta t = h)$
$\bar{c}_1$	0.3088	0.3088	0.3088	0.3088	0.3088
$\bar{c}_2$	0.1909	0.1909	0.1909	0.1909	0.1909
$\bar{c}_3$	0.1915	0.1915	0.1915	0.1915	0.1915

4.2 Four-component phase separation on a spherical surface

Inappropriate treatment of the multicomponent CH system may cause the nonphysical overlap of each phase component. To verify the proposed numerical method for solving the multicomponent CH equation on curved surfaces, we numerically study the four-component phase separation on a sphere. In this simulation, we use $h = 0.01$ and $\Delta t = 0.1$ on $\Omega = (0, 1) \times (0, 1) \times (0, 1)$. The initial condition is defined as

$$\left(c_{1,ijk}, c_{2,ijk}, c_{3,ijk}\right) = \begin{cases} (1, 0, 0) & \text{if } \mathrm{rand}_{ijk} \le \frac{1}{4}, \\ (0, 1, 0) & \text{else if } \mathrm{rand}_{ijk} \le \frac{1}{2}, \\ (0, 0, 1) & \text{else if } \mathrm{rand}_{ijk} \le \frac{3}{4}, \\ (0, 0, 0) & \text{otherwise}, \end{cases} \tag{13}$$

where rand_{ijk} are random numbers between 0 and 1 at each position $x_{ijk} \in \Omega_\delta^h$. Figure 4a–c shows the results of four-component phase separation on a spherical surface at $t = 0,\ 200\Delta t$, and $1000\Delta t$, respectively. The figures from the top to bottom in each row represent the distributions of the four components: c_1 (red), c_2 (deep blue), c_3 (yellow), and c_4 (orange), respectively. Note that the green regions here indicate the background color of the spherical surface. As we can observe, the four different components do not overlap each other, thereby indicating that our method can qualitatively simulate the process of four-component phase separation on a curved surface.

4.3 Numerical verification of unconditional stability

An unconditionally stable nonlinear splitting scheme was originally proposed by Eyre [25] to numerically solve the phase-field equations. The scheme splits the nonlinear term in the equations into the difference of two convex functions, and it is unconditionally stable. Recently, Lee et al. [15] and Li et al. [16] extended this scheme to solve the multicomponent CH system in the Cartesian coordinate system. To verify the unconditional stability of the proposed method, the numerical tests with the same initial condition as described in Sect. 4.2 are performed on a spherical surface in a series of time steps: $\Delta t = h,\ 10h$, and $100h$. $\Omega = (0, 1) \times (0, 1) \times (0, 1)$ is discretized with a uniform mesh size $h = 0.01$. The discrete total energy is defined as

$$\mathcal{E}_h(\mathbf{c}^n) = \sum_{l=1}^{N} \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} \left\{ F\left(c_{l,ijk}^n\right) + \frac{\epsilon^2}{2h^2} \left[\left(c_{l,i+1,jk}^n - c_{l,ijk}^n\right)^2 + \left(c_{l,i,j+1,k}^n - c_{l,ijk}^n\right)^2 + \left(c_{l,ij,k+1}^n - c_{l,ijk}^n\right)^2 \right] \right\}.$$

Figure 5 shows the temporal evolutions of the normalized discrete total energy $\overline{\mathcal{E}_h(\mathbf{c})} = \mathcal{E}_h(\mathbf{c}^n)/\mathcal{E}_h(\mathbf{c}^0)$ with different time steps. It can be observed that the energy stability is satisfied even when large time steps are used.

4.4 Verification of mass correction

In this part, we verify the mass correction step, given in Eq. (12), by the following two tests. First, we test the efficiency of mass correction on mass conservation. The initial conditions for c_1 (red circle), c_2 (deep blue circle),

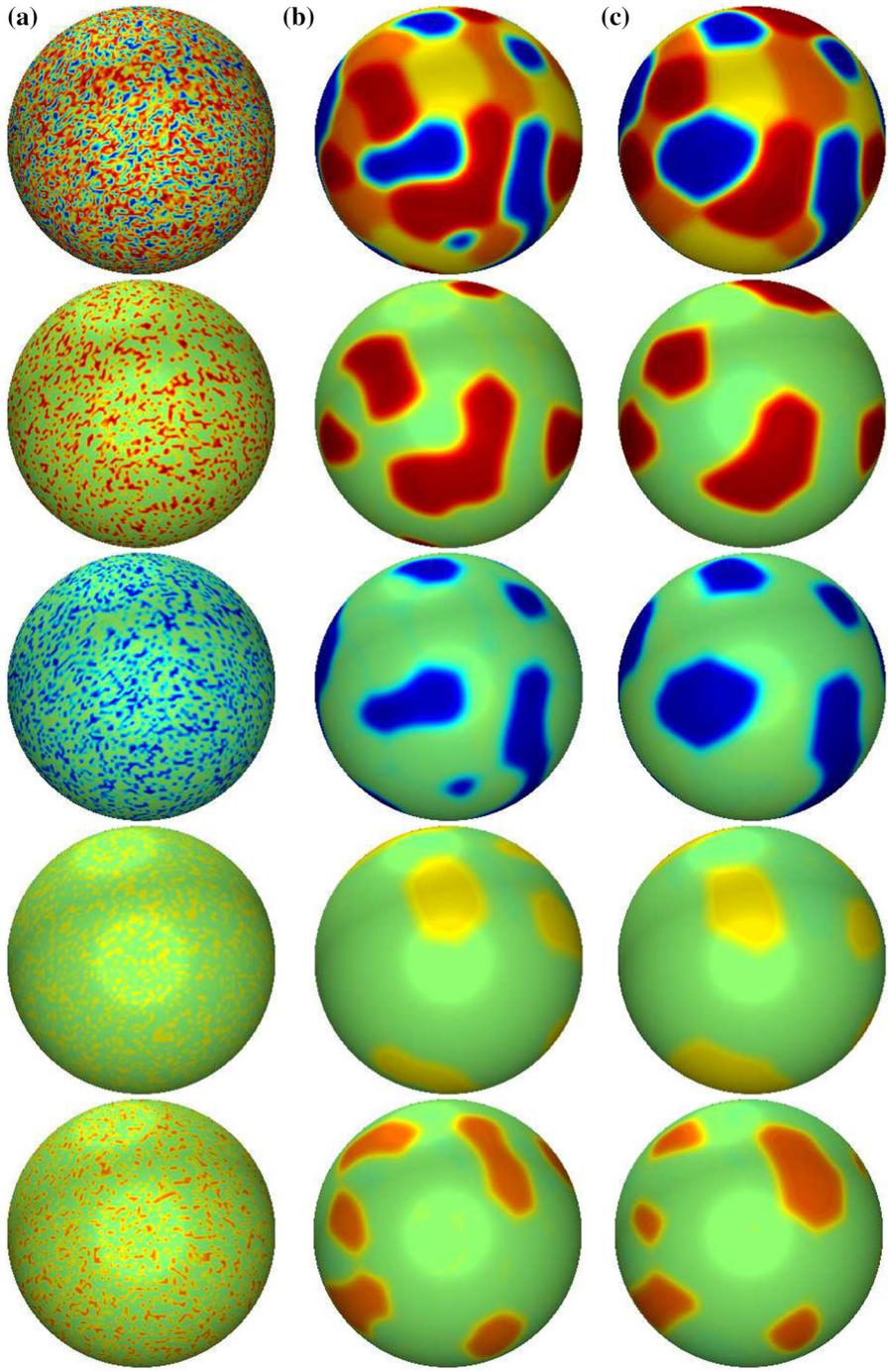


Fig. 4 Four-component phase separation on a spherical surface. **a** $t = 0$, **b** $t = 200\Delta t$, **c** $t = 1000\Delta t$. (Color figure online)

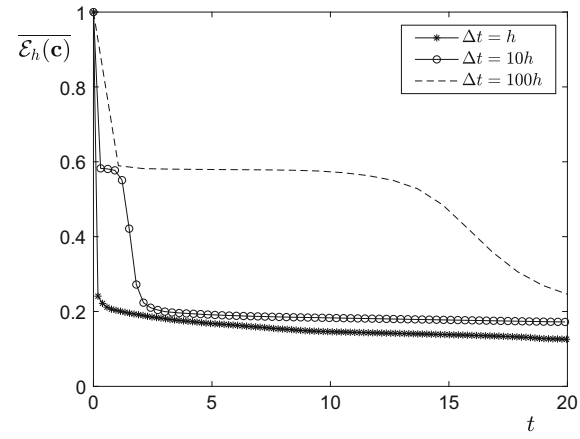


Fig. 5 Temporal evolutions of the normalized discrete total energy with large time steps

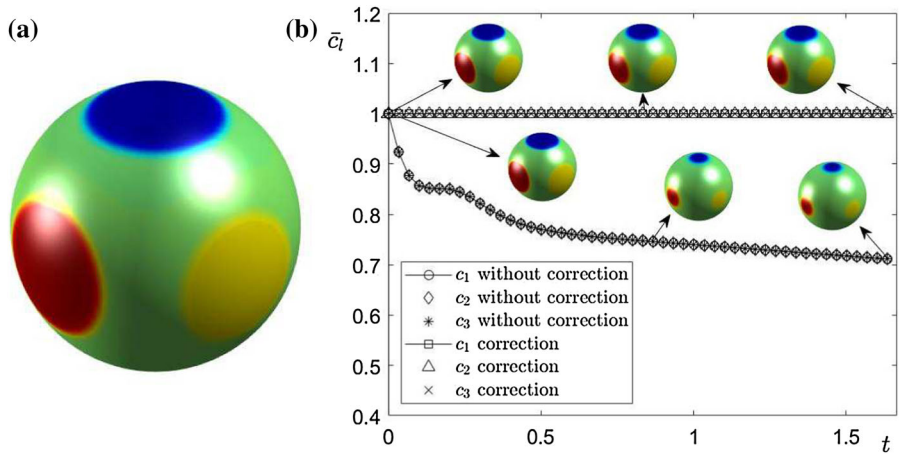


Fig. 6 **a** Initial conditions. **b** Temporal evolutions of each mass fraction $\bar{c}_l$. (Color figure online)

and c_3 (yellow circle) are shown in Fig. 6a, where c_4 is the green region. Note that each circle has the same size. The numerical tests are performed on a spherical surface in a 3D domain $\Omega = (0, 1) \times (0, 1) \times (0, 1)$. We use $h = 1/60,\ \Delta t = 0.1h$. The temporal evolutions of each mass fraction $\bar{c}_l$ are shown in Fig. 6b. In the case where mass correction is used, the mass fractions of $c_1,\ c_2$, and c_3 remain constant. In the case where mass correction is not used, the mass fractions of $c_1,\ c_2$, and c_3 decrease with time. Because mass conservation is an important problem in the simulation of N-phase CH system, therefore mass correction is necessary.

Next, we simulate the triple junction formation on a spherical surface with and without mass correction to investigate the effect of mass correction on phase dynamics. In this test, we use $h = 1/80,\ \Delta t = 0.1h$. Figure 7a and b shows the temporal evolutions with and without mass corrections, respectively. Here, c_1, c_2, and c_3 are the red, deep blue, and green regions, respectively. It is evident that the evolutional profiles are almost the same. There is no obvious effect of mass correction on the phase dynamics.

4.5 The average CPU time consumed for N-component system

In one time step, the numerical method described in Sect. 3 is used to solve the N-component system continuously (i.e., after the computation of c_1 throughout the domain, the same numerical method is then used to solve c_2, then we solve c_3. Finally, c_N is solved after the computation of c_{N-1}). In this approach, if we add one more component, the time required to solve it must be added throughout the domain; therefore, the average CPU time must exhibit a

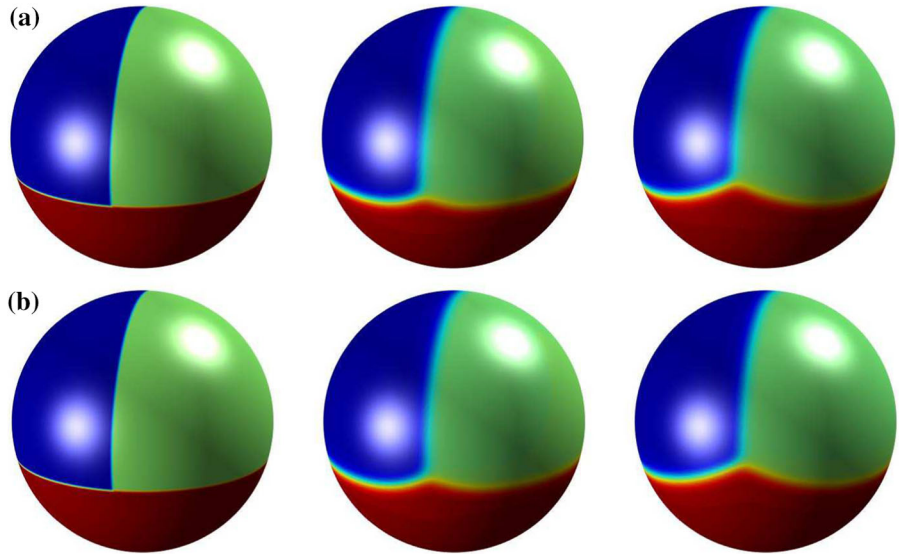


Fig. 7 Temporal evolutions of a triple junction **a** with and **b** without mass correction. The times from the left to right in each row are $\Delta t = 0,\ 1200\Delta t$, and $12000\Delta t$. (Color figure online)

linear relationship with respect to the number of components. We calculate the average CPU time (in s) consumed for solving an N-component system. The computation is performed on a sphere embedded in a unit cube. Here, $h = 0.01$ and $\Delta t = 0.1$ are used. The initial conditions for three-, four-, five-, and six-component systems are generated randomly. For $g = 1, \ldots, 100$, let $R_g = 0.12 + 0.05\,\text{rand}$ be a random radius of the given sphere, where rand is a random number between 0 and 1. Let $\mathbf{O}_g = (R_g, R_g, R_g) + (1 - 2R_g)(\text{rand}, \text{rand}, \text{rand})$ be the center of the sphere. We initialize all the concentrations as zero. Then, for $g = 1, \ldots, 100$, we assign the value 1 (inside the sphere) to a randomly chosen c_l; the other concentrations are set to zero. The main procedure for the assignment of phase-field values is described as follows:

Step 1 We first set $c_l(\mathbf{x})$ at zero in the whole domain.

Step 2 A random center point $\mathbf{O}_g$ is generated in the domain.

Step 3 We check the condition: $|\mathbf{x} - \mathbf{O}_g| \leq R_g$,, we set $c_l(\mathbf{x}) = 1$ if this condition is satisfied.

Step 4 Let $g = g + 1$.

We repeat *Step* 2–4 until $g = 100$. Note that if a new sphere overlaps the previously assigned values, we overwrite the value.

Figure 8a gives the schematic illustration of randomly distributed spheres. Table 2 provides the average CPU time that is required to advance 1000 time steps for different number of components. The plot of the average CPU time versus the number of components is shown in Fig. 8b. We observe that the change in average CPU time exhibits a linear relationship with respect to the number of components. In this simulation, the computations are performed by using a generic C program, therefore we calculate the average CPU time consumed for solving the main procedure by using a clock() function.

4.6 Triple junctions in a cube surface

In this simulation, we investigate the effect of triple junctions on a cube surface for three-, four-, and five-component phase evolution. Here, we use $h = 0.01$, $\Delta t = 0.1$, and $\Omega = (0, 1) \times (0, 1) \times (0, 1)$. The cube surface is represented by

$$\psi(x, y, z) = \min\big(\max(\hat{x}, \max(\hat{y}, \hat{z})), 0\big) + \sqrt{\max(\hat{x}, 0)^2 + \max(\hat{y}, 0)^2 + \max(\hat{z}, 0)^2}, \tag{14}$$

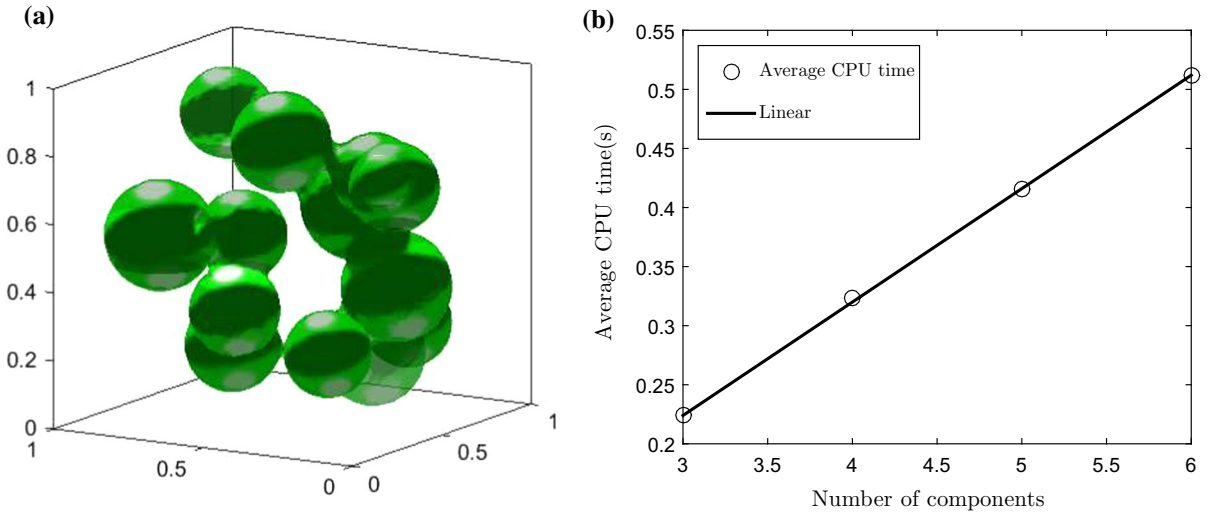


Fig. 8 **a** Schematic illustration of randomly distributed spheres. **b** Average CPU time(s) versus the number of components. (Color figure online)

Table 2 Average CPU times for different number of components

N-component	3	4	5	6
Average CPU time	0.224	0.323	0.416	0.512

where $\hat{x} = |x - 0.5| - 0.4$, $\hat{y} = |y - 0.5| - 0.4$, $\hat{z} = |z - 0.5| - 0.4$. The initial conditions for the three-, four-, and five-component evolutions are shown at the left side of Fig. 9a–c, respectively. The figures in the middle and right of each row represent the results for $t = 100\Delta t$ and $t = 1000\Delta t$, respectively. For a three-component system, the red, deep blue, and green regions represent c_1, c_2, and c_3, respectively. For a four-component system, the red, deep blue, yellow, and green regions represent c_1, c_2, c_3, and c_4, respectively. For a five-component system, the red, deep blue, yellow, light blue, and green regions represent the c_1, c_2, c_3, c_4, and c_5, respectively. We can observe that the system with more components reaches the local equilibrium state (the triple junction angles approach the true value 120°) faster. To explain this phenomenon, we consider the front of the cube. In the total energy functional Eq. (2), because $F(\phi)$ is symmetric and the interaction parameter ϵ is constant, the triple junction angles approach 120 degree as they approach local equilibrium states. If the number of components increases, the local equilibrium state can be achieved faster by moving a relatively lesser quantity of each component; therefore, the evolutional dynamics in this case is faster than in the case with fewer components.

4.7 Multicomponent evolution on various surfaces

In this test, we perform the numerical simulations of three- and six-component phase separation on a sphere, a cube, and a rabbit-shaped surface. We use $h = 0.01$ and $\Delta t = 0.1$ on $\Omega = (0, 1) \times (0, 1) \times (0, 1)$ for the sphere and cube surfaces and $h = 0.5$ and $\Delta t = 0.2$ on $\Omega = (0, 68.5) \times (0, 68.5) \times (0, 68.5)$ for the rabbit-shaped surface. The method for generating the signed distance function for a rabbit-shaped surface can be found in [32]. The initial conditions are generated randomly as done in Sect. 4.2. Figures 10, 11,and 12a, b show the temporal evolutions for three- and six-component conditions on a sphere, a cube, a rabbit-shaped surface, respectively. In the three-component condition, the red, deep blue, and green regions represent the phases c_1, c_2, and c_3, respectively. In the six-component condition, the phases c_1, c_2, c_3, c_4, c_5, and c_6 are represented by the red, deep blue, yellow, light blue, orange, and green regions, respectively. For sphere and cube surfaces, the times from the left to the right

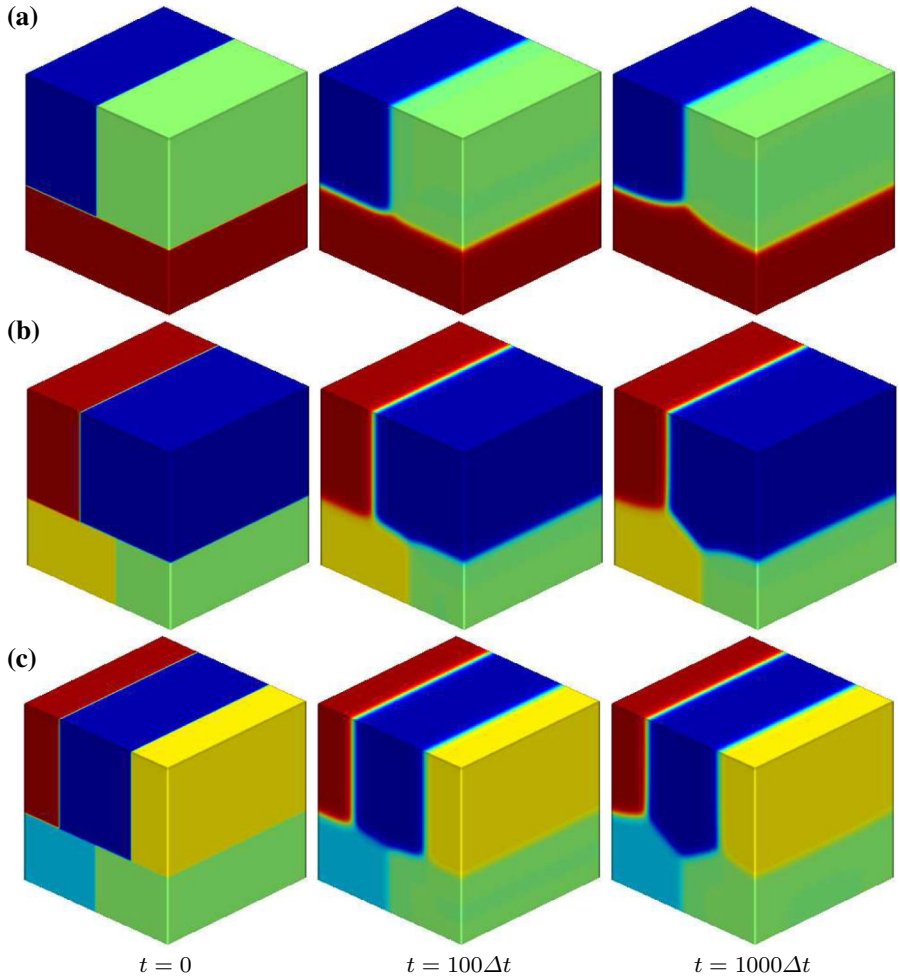


Fig. 9 Phase evolution on a cube surface for **a** three-component, **b** four-component, **c** five-component. (Color figure online)

in each row are $t = 0,\ 10\Delta t,\ 200\Delta t$, and $1000\Delta t$. For a rabbit-shaped surface, the times from the left to right in each row are $t = 0,\ 60\Delta t,\ 900\Delta t$, and $3000\Delta t$. As is clearly visible, our method can simulate multicomponent phase separation on various surfaces.

4.8 Effect of surface geometry on multicomponent phase separation

Next, we study the effect of surface shape on the dynamics of multicomponent phase separation. The initial conditions are chosen randomly for a six-component system. Here, $h = 0.01$ and $\Delta t = 0.1$ are used in the whole domain $(0, 1) \times (0, 1) \times (0, 1)$. We consider the following signed distance function for a torus shape:

$$\psi(x, y, z) = \sqrt{\left(\sqrt{(x - 0.5)^2 + (y - 0.5)^2} - 0.2\right)^2 + (z - 0.5)^2} - R. \tag{15}$$

Figure 13a, b show the temporal evolutions of a six-component system with $R = 0.1$ and $R = 0.05$, respectively. The time steps from the left to the right in each row are $t = 0,\ 10\Delta t,\ 200\Delta t$, and $1000\Delta t$. We note that a thinner torus leads to a smaller surface area. On a smaller surface, the randomly distributed components need to move just a short distance to meet and merge with themselves. This condition causes the bulk phase of each component to form

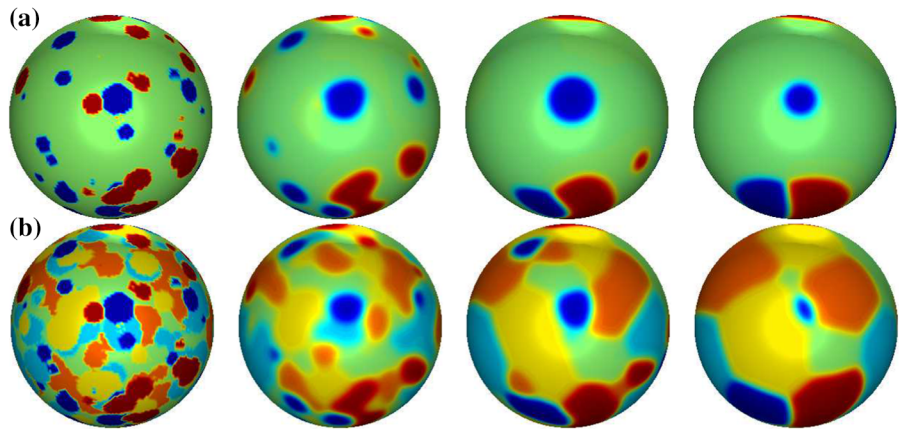


Fig. 10 Phase evolution on a sphere surface for **a** three components, **b** six components. (Color figure online)

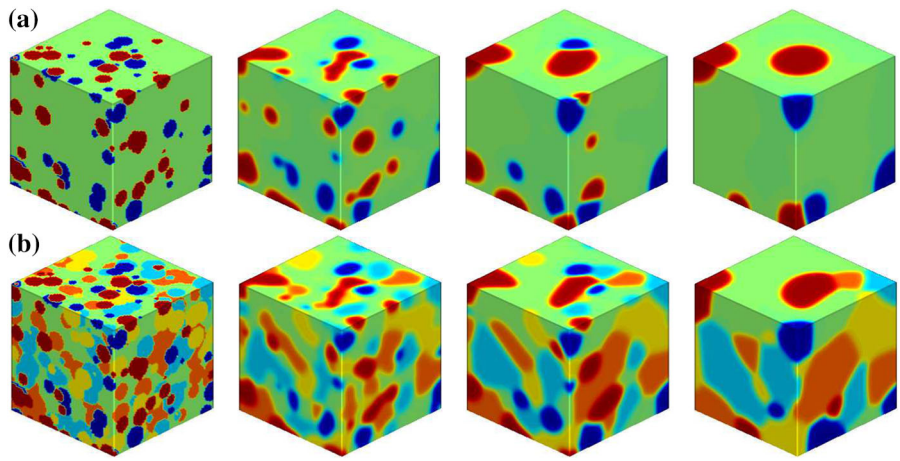


Fig. 11 Phase evolution on a cube surface for **a** three components, **b** six components. (Color figure online)

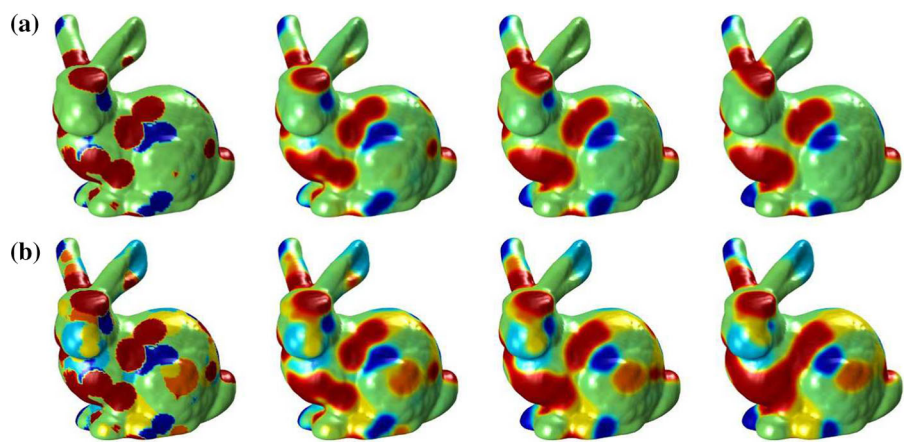


Fig. 12 Phase evolution on a rabbit-shaped surface for **a** three components and **b** six components. (Color figure online)

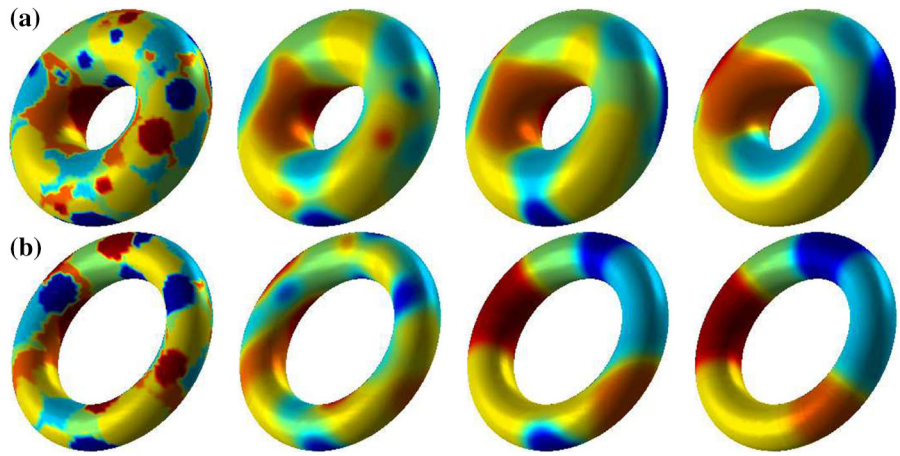


Fig. 13 Six-component phase evolution on torus surfaces. (Color figure online)

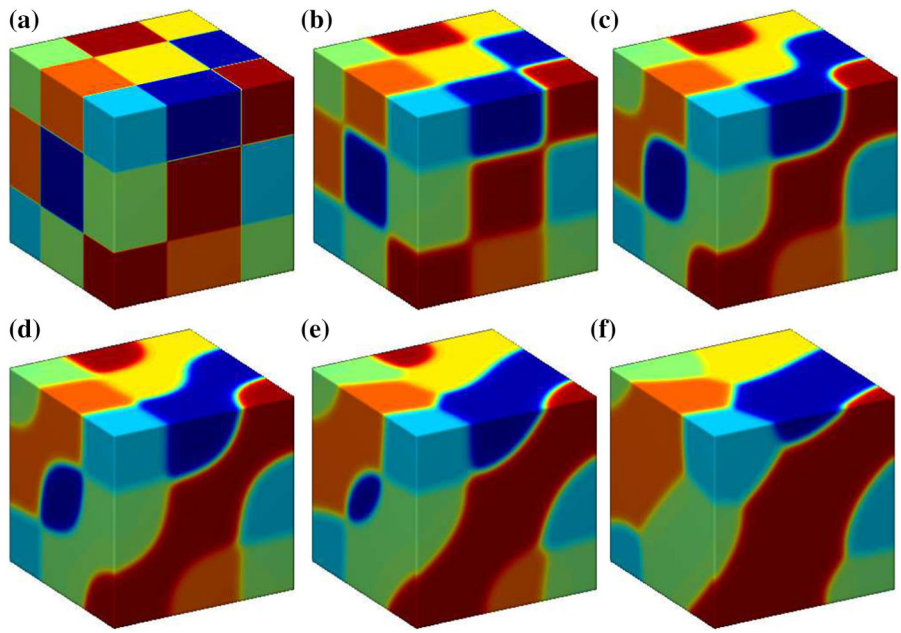


Fig. 14 Temporal evolution of a six-component system on a cube. The time steps from **a**–**f** are $t = 0,\ 30\Delta t,\ 250\Delta t,\ 660\Delta t,\ 1500\Delta t$, and $3000\Delta t$. (Color figure online)

quickly, resulting in a faster approach toward the equilibrium state on a thinner surface than on a thicker one. As is evident from the numerical results, as the torus tube gets thinner, the six-component system equilibrates faster.

The above phenomenon causes a fast formation of bulk phase of each component, and therefore, the equilibrium state can be arrived faster on a thinner surface than on a broader one.

4.9 Temporal evolution of a six-component system on a cube

In this test, we cut a cube into 27 small cubes and assign the value 1 to a randomly chosen c_l. Here, $h = 0.01$ and $\Delta t = 0.2$ are used. Figure 14a–f shows the temporal evolutions of six components on a cube. c_1, c_2, c_3, c_4, c_5, and c_6 are represented by red, deep blue, yellow, light blue, orange, and green regions, respectively. We can observe that these neighborhood regions merge with each other.

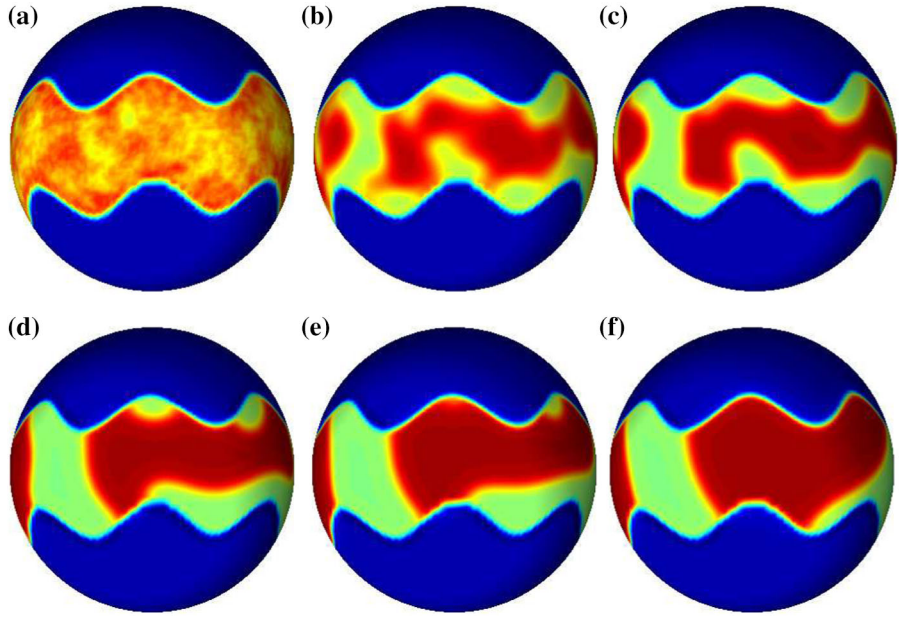


Fig. 15 Temporal evolution of phase separation in a complex domain on a sphere. The time steps from **a–f** are $t = \Delta t,\ 4\Delta t,\ 8\Delta t,\ 40\Delta t,\ 60\Delta t$, and $200\Delta t$. (Color figure online)

4.10 Phase separation in a complex domain on surfaces

Here, we simply extend our multicomponent model to simulate the two-component phase separation in a complex domain on a sphere. Here, $h = 0.01$ and $\Delta t = 0.1$ are used on $\Omega = (0, 1) \times (0, 1) \times (0, 1)$. The complex domain is represented by the initial shape of c_1 (deep blue regions) and we fix c_1 in the following computation. First, we produce a series of random numbers $rand$ lie between 0 and 1. Then, we let the initial condition be $c_2(x, y, z, 0) = 1$ if $rand \leq 0.5$; otherwise $c_2(x, y, z, 0) = 0$ and $c_3(x, y, z, 0) = 1 - c_1(x, y, z) - c_2(x, y, z, 0)$. To avoid a bias problem in the proximity of the complex domain boundary, we solve c_2 and c_3 alternately. The temporal evolutions of c_2 (red regions) and c_3 (green regions) are shown in Fig. 15a–f. It can be observed that the complex domain is well represented by c_1, and the phase separation only occurs inside the complex domain.

Next, we extend the study of the phase separation in a complex domain on a cylinder for two different mobilities: $M = 0.01 + 0.99z$ and $M = 1$. Because the mobility $M = 0.01 + 0.99z$ is not constant, $\nabla \cdot [M \nabla \mu_l] \neq M \Delta \mu_l$, and we modify Eqs. (9) and (10) to obtain the following numerical scheme for the governing equation with a typically variable mobility:

$$
\begin{aligned}
&\left(\frac{M^{n+1,m}_{i+\frac{1}{2},jk} + M^{n+1,m}_{i-\frac{1}{2},jk} + M^{n+1,m}_{i,j+\frac{1}{2},k} + M^{n+1,m}_{i,j-\frac{1}{2},k} + M^{n+1,m}_{ij,k+\frac{1}{2}} + M^{n+1,m}_{ij,k-\frac{1}{2}}}{h^2} \right) \mu^{n+1,m+1}_{l,ijk} + \frac{c^{n+1,m+1}_{l,ijk}}{\Delta t} \\
&= \frac{c^{n}_{l,ijk}}{\Delta t} + \frac{M^{n+1,m}_{i+\frac{1}{2},jk} \mu^{n+1,m}_{l,i+1,jk} + M^{n+1,m}_{i-\frac{1}{2},jk} \mu^{n+1,m}_{l,i-1,jk} + M^{n+1,m}_{i,j+\frac{1}{2},k} \mu^{n+1,m}_{l,i,j+1,k}}{h^2} \\
&\quad + \frac{M^{n+1,m}_{i,j-\frac{1}{2},k} \mu^{n+1,m}_{l,i,j-1,k} + M^{n+1,m}_{ij,k+\frac{1}{2}} \mu^{n+1,m}_{l,ij,k+1} + M^{n+1,m}_{ij,k-\frac{1}{2}} \mu^{n+1,m}_{l,ij,k-1}}{h^2}, \\
&- \left[3\left(c^{n+1,m}_{l,ijk}\right)^2 + \frac{6\epsilon^2}{h^2} - 3c^{n+1,m}_{l,ijk} + 0.75 \right] c^{n+1,m+1}_{l,ijk} + \mu^{n+1,m+1}_{l,ijk}
\end{aligned}
\tag{16}
$$

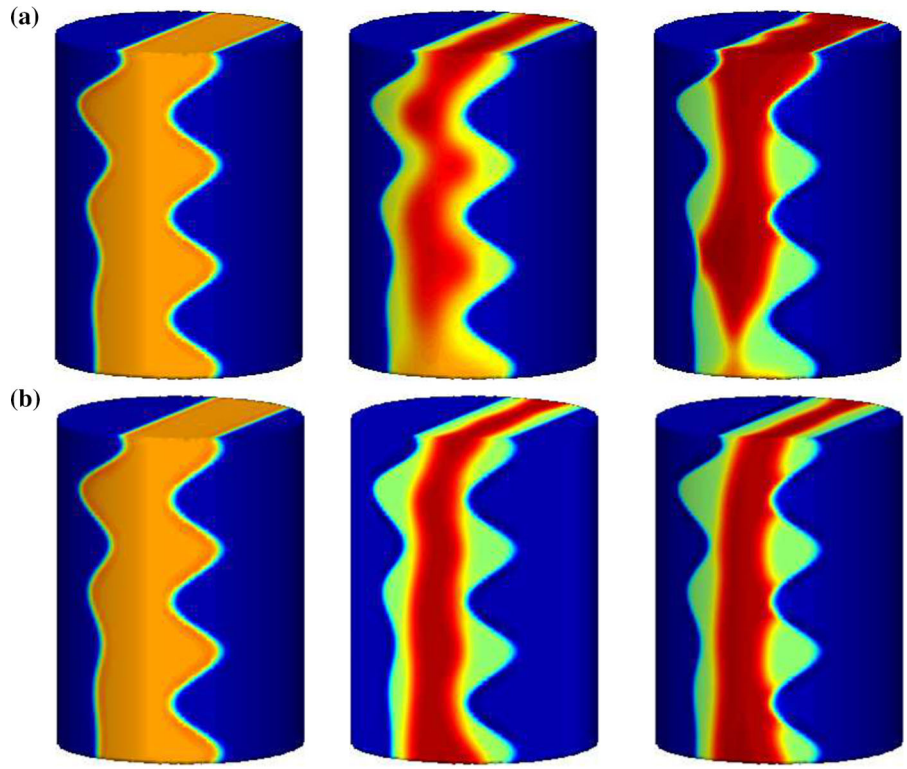


Fig. 16 Temporal evolutions of the phase separation in a wave-shaped complex domain on a cylinder for two different mobilities: **a** $M = 0.01 + 0.99z$ and **b** $M = 1$. The time steps from the left to right in each row are $t = \Delta t,\ 12\Delta t$, and $30\Delta t$. (Color figure online)

$$= -2\left(c_{l,ijk}^{n+1,m}\right)^3 + 1.5\left(c_{l,ijk}^{n+1,m}\right)^2 - \frac{\epsilon^2}{h^2}\left(c_{l,i+1,jk}^{n+1,m} + c_{l,i-1,jk}^{n+1,m} + c_{l,i,j+1,k}^{n+1,m}\right)$$
$$+\frac{\epsilon^2}{h^2}\left(c_{l,i,j-1,k}^{n+1,m} + c_{l,ij,k+1}^{n+1,m} + c_{l,ij,k-1}^{n+1,m}\right) + \beta(\mathbf{c}_{ijk}^{n}) - 0.25c_{l,ijk}^{n}, \tag{17}$$

where $M_{i+\frac{1}{2},jk} = (M_{i+1,jk} + M_{ijk})/2$ and the other terms are similarly defined. Here, $h = 0.01$ and $\Delta t = 0.2h$ are used on $\Omega = (0, 1) \times (0, 1) \times (0, 1)$. The cylinder surface is represented by

$$\psi(x, y, z) = \max\left(\sqrt{(x - 0.5)^2 + (y - 0.5)^2} - 0.3, |z - 0.5| - 0.4\right). \tag{18}$$

The complex domain is defined by the initial shape of c_1 (deep blue regions). The initial conditions for c_1, c_2, and c_3 are

$$c_1(x, y, z) = \begin{cases} 0.5 + 0.5\tanh\left(-\frac{x-0.4+0.1z^2\cos(8\pi z)}{\sqrt{2}\epsilon}\right) & \text{if } x < 0.5, \\ 0.5 + 0.5\tanh\left(\frac{x-0.65+0.05\cos(8\pi z)}{\sqrt{2}\epsilon}\right) & \text{otherwise,} \end{cases} \tag{19}$$

$$c_2(x, y, z, 0) = (1 - c_1(x, y, z))(0.5 + 0.05\,\text{rand}), \tag{20}$$

$$c_3(x, y, z, 0) = 1 - c_1(x, y, z) - c_2(x, y, z, 0). \tag{21}$$

We fix c_1 in the following computation and compute c_2 and c_3 alternatively. Figure 16a, b illustrate the temporal evolutions of c_2 (red regions) and c_3 (green regions) for a z-direction-dependent mobility ($M = 0.01 + 0.99z$) and a constant mobility ($M = 1$) in a wave-shaped complex domain on a cylinder. It is evident that a faster separation occurs due to the z-direction-dependent mobility as the height of z increases. In the case of constant mobility, the dynamics along the z-direction are independent of direction.

5 Conclusions

A conservative finite difference method for solving the N-component CH equation on 3D curved surfaces was proposed in this study. An unconditionally stable scheme was used to discretize the governing equation. We used the mass correction step to each component to achieve total mass conservation. The numerical results demonstrated the efficiency of the mass correction algorithm for solving the N-component CH equation. Various numerical experiments demonstrated that the proposed method can simulate the N-component phase evolution on various 3D surfaces: sphere, cube, rabbit-shaped, torus, and cylinder. In addition, the proposed method can be simply modified to simulate the phase separation in arbitrary complex domains on 3D surfaces.

Acknowledgements The author (D. Jeong) was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MSIP) (NRF-2017R1E1A1A03070953). Y.B. Li is supported by National Natural Science Foundation of China (Nos. 11601416, 11631012). 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-2016R1D1A1B03933243). The authors appreciate the reviewers for their constructive comments, which have improved the quality of this paper.

References

1. Cahn JW (1965) Phase separation by spinodal decomposition in isotropic systems. J Chem Phys 42(1):93–99
2. Wise SM, Lowengrub JS, Frieboes HB, Cristini V (2008) Three-dimensional multispecies nonlinear tumor growth - 1. Model and numerical method. J Theor Biol 253:524–543
3. Dehghan M, Mohammadi V (2017) Comparison between two meshless methods based on collocation technique for the numerical solution of four-species tumor growth model. Commun Nonlinear Sci Numer Simul 44:204–219
4. Deng Y, Liu Z, Wu Y (2017) Topology optimization of capillary, two-phase flow problems. Commun Comput Phys 22:1413–1438
5. Zhang Y, Ye W (2017) A flux-corrected phase-field method for surface diffusion. Commun Comput Phys 22:422–440
6. Ju L, Zhang J, Du Q (2015) Fast and accurate algorithms for simulating coarsening dynamics of Cahn–Hilliard equations. Comput Mater Sci 108:272–282
7. Kang D, Chugunova M, Nadim A, Waring AJ, Walther FJ (2018) Modeling coating flow and surfactant dynamics inside the alveolar compartment. J Eng Math 113(1):23–43
8. Lee D, Huh JY, Jeong D, Shin J, Yun A, Kim JS (2014) Physical, mathematical, and numerical derivations of the Cahn–Hilliard euqation. Comput Mater Sci 81:216–225
9. Kim J, Lee S, Choi Y, lee SM, Jeong D (2016) Basic principles and practical applications of the Cahn–Hilliard equation. Math Probl Eng 2016:9532608
10. Lee HG, Kim J (2013) Buoyancy-driven mixing of multi-component fluids in two-dimensional titled channels. Eur J Mech B 42:37–46
11. Park JM, Anderson PD (2012) A ternary model for double-emulsion formation in a capillary microfluidic device. Lab Chip 12:2672–2677
12. Lee HG, Kim J (2015) Two-dimesional Kelvin–Helmholtz instabilities of multi-component fluids. Eur J Mech B 49:77–88
13. Bhattacharyya S, Abinanadanan TA (2003) A study of phase separation in ternary alloys. Bull Mater Sci 26:193
14. Lee HG, Kim J (2008) A second-order accurate non-linear difference scheme for the N-component Cahn–Hilliard system. Physica A 387:4787–4799
15. Lee HG, Choi JW, Kim J (2012) A practically unconditionally gradient stable scheme for the N-component Cahn–Hilliard system. Physica A 391:1009–1019
16. Li Y, Choi JI, Kim J (2016) Multi-component Cahn–Hilliard system with different boundary conditions in complex domains. J Comput Phys 323:1–16
17. Jeong D, Yang J, Kim J (2019) A practical and efficient numerical method for the Cahn–Hilliard equation in complex domains. Commun Nonlinear Sci Numer Simul 73:217–228
18. Du Q, Ju L, Tian L (2011) Finite element approximation of the Cahn–Hilliard equation on surfaces. Comput Methods Appl Mech Eng 200(29–32):2458–2470
19. Li Y, Kim J, Wang N (2017) An unconditionally energy-stable second-order time-accurte scheme for the Cahn-Hilliard equation on surfaces. Commun Nonlinear Sci Numer Simul 53:213–227
20. Li Y, Qi X, Kim J (2018) Direct discretization method for the Cahn–Hilliard equation on an evolving surface. J Sci Comput 77:1147–1163
21. Li Y, Luo C, Xia B, Kim J (2019) An efficient linear second order unconditionally stable direct discretization method for the phase-field crystal equation on surfaces. Appl Math Model 67:477–490
22. Dziuk G, Elliott CM (2007) Surface finite elements for parabolic equations. J Comput Math 25(4):385–407

23. Green JB, Bertozzi AL, Sapiro G (2006) Fourth order paratial differential equations on general geometries. J Comput Phys 216(1):216–246
24. Jeong D, Li Y, Lee C, Yang J, Kim J (2019) A conservative numerical method for the Cahn–Hilliard euqation with generalized mobilities on curved surfaces in three-dimensional space. Commun Comput Phys (in press)
25. Eyre DJ (1998) Unconditionally gradient stable time marching the Cahn–Hilliard equation. In: Computational and mathematical models of microstructural evolution. MRS proceedings, vol 529, pp 39–46
26. Kim J (2009) A generalized continuous surface force formulation for phase-field models for multi-component immiscible fluid flows. Comput Methods Appl Mech Eng 198:3105–3112
27. Macdonald CB, Brandman J, Ruuth SJ (2011) Solving eigenvalue problems on curved surfaces using the closet point method. J Comput Phys 230:7944–7956
28. Ruuth SJ, Merriman B (2008) A simple embedding method for solving partial differential equations on surfaces. J Comput Phys 227(3):1943–1961
29. Greer JB (2006) An improvment of a recent Eulerian method for solving PDEs on general geometries. J Sci Comput 29:321–352
30. Macdonald CB, Ruuth SJ (2009) The implicit closest point method for the numerical solution of partial differential equations on surfaces. SIAM J Sci Comput 31(6):4330–4350
31. Choi JW, Lee HG, Jeong D, Kim J (2009) An unconditionally gradient stable numerical method for solving the Allen–Cahn equation. Physica A 338(9):1791–1803
32. Jeong D, Li Y, Choi Y, Yoo M, Kang D, Park J, Choi J, Kim J (2017) Numerical simulation of the zebra pattern formation on a three-dimensional model. Physica A 475:106–116