J Sci Comput
https://doi.org/10.1007/s10915-018-0742-6

# Direct Discretization Method for the Cahn–Hilliard Equation on an Evolving Surface

**Yibao Li**[1] · **Xuelin Qi**[1] · **Junseok Kim**[2]

Received: 29 August 2017 / Revised: 22 February 2018 / Accepted: 20 May 2018


**Abstract** We propose a simple and efficient direct discretization scheme for solving the Cahn–Hilliard (CH) equation on an evolving surface. By using a conservation law and transport formulae, we derive the CH equation on evolving surfaces. An evolving surface is discretized using an unstructured triangular mesh. The discrete CH equation is defined on the surface mesh and its dual surface polygonal tessellation. The evolving triangular surfaces are then realized by moving the surface nodes according to a given velocity field. The proposed scheme is based on the Crank–Nicolson scheme and a linearly stabilized splitting scheme. The scheme is second-order accurate, with respect to both space and time. The resulting system of discrete equations is easy to implement, and is solved by using an efficient biconjugate gradient stabilized method. Several numerical experiments are presented to demonstrate the performance and effectiveness of the proposed numerical scheme.

**Keywords** Cahn–Hilliard equation · Evolving surface · Laplace–Beltrami operator · Triangular surface mesh

## 1 Introduction

The Cahn–Hilliard equation (CH) [1] has been extensively applied in a wide range of research fields, such as biological modelling and simulations [2,3], multi-phase fluid flow dynamics [4,5], volume reconstruction [6], and materials science [7–9]. Various studies have showed that phase separations can occur on surfaces, such as in the evolution of nanoporosity in

✉ Junseok Kim
cfdkim@korea.ac.kr

Yibao Li
yibaoli@xjtu.edu.cn
http://gr.xjtu.edu.cn/web/yibaoli

[1] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China

[2] Department of Mathematics, Korea University, Seoul 136-701, Republic of Korea

Published online: 29 May 2018

dealloying [10], lipid bilayer membranes [11], diblock copolymers on surfaces [12,13], crystal growth [14,15], and biological membranes [16]. The CH equation is a fourth-order nonlinear parabolic partial differential equation (PDE). In general, this equation cannot be solved analytically. Hence, we have to employ approximate techniques to solve the equation in order to understand non-equilibrium processing. Various computational algorithms have been developed to solve the CH equation numerically [17–21].

Only a few studies have solved the CH equation on curved surfaces [16,22–25]. Mercker et al. [16] presented a CH-type model for lateral membrane dynamics on a surface, and computed membrane dynamics using a finite element method (FEM). In [22], Du et al. demonstrated the convergence and well-posedness of a fully discrete FEM of the CH equation defined on a fixed surface. Elliott and Ranner [23] used a finite element technique to solve the CH equation on an evolving surface with a given velocity field. They presented a rigorous well-posedness result for the CH equation by showing convergence of the FEM. All of the above-mentioned methods are based on the finite element technique. Here, the Laplace operator is represented using the tangential gradient. However, approaches based on the FEM are deceptively difficult to implement. An alternative approach to solving PDEs on evolving surface is with implicit methods [26–31]. Such methods extend a PDE to a three-dimensional narrow band domain and then solve the extended PDE on the domain. The implicit method can be extended to solve the CH equation on an evolving surface. However, all implicit methods are computationally expensive for complex surfaces.

The main purpose of the present article is to develop a stable second-order time-accurate scheme for the Cahn–Hillard equation on an evolving surface. We use a simple method to approximate the regular evolving surface by an evolving triangular mesh, on which the gradient, divergence, and Laplacian operators can be directly defined. Unlike in the implicit method, we can reduce the additional computational costs, because we solve two-dimensional schemes on surfaces. Compared with the FEM, the proposed discretization is easy to implement. It should be noted that our methods may generally be limited according to the quality of the surface mesh. However, owing to [32,33], the mesh quality has been significantly improved.

The outline of this paper is as follows. In Sect. 2, we derive the CH equation on an evolving surface. In Sect. 3, we present the numerical solution. In Sect. 4, we present numerical experiments. Finally, conclusions are drawn in Sect. 5.

## 2 The CH Equation on Evolving Surfaces

Let $S(\mathbf{s}, t)$ be a general surface, where $\mathbf{s}$ lies on the surface $S$ and $t$ is the time. Assume that a surface $S(\mathbf{s}, t)$ evolves according to a given velocity field $\mathbf{V}(\mathbf{s}, t)$ for the time $t$. Here, $\mathbf{V}(\mathbf{s}, t)$ can be decomposed into a normal vector field $V_\nu(\mathbf{s}, t)\mathbf{N}(\mathbf{s}, t)$ and a tangential vector field $\mathbf{V}_\tau(\mathbf{s}, t)$, i.e.,

$$\mathbf{V}(\mathbf{s}, t) = V_\nu(\mathbf{s}, t)\mathbf{N}(\mathbf{s}, t) + \mathbf{V}_\tau(\mathbf{s}, t). \tag{1}$$

Then, taking the inner product with the unit normal vector $\mathbf{N}(\mathbf{s}, t)$ on both sides of Eq. (1) yields

$$V_\nu(\mathbf{s}, t) = \langle \mathbf{V}(\mathbf{s}, t), \mathbf{N}(\mathbf{s}, t) \rangle, \tag{2}$$

where $\langle\ ,\ \rangle$ is the inner product operator. Furthermore, we obtain that

$$\mathbf{V}_\tau(\mathbf{s}, t) = \mathbf{V}(\mathbf{s}, t) - \langle \mathbf{V}(\mathbf{s}, t), \mathbf{N}(\mathbf{s}, t) \rangle \mathbf{N}(\mathbf{s}, t). \tag{3}$$

For any point $\mathbf{s} \in S(\mathbf{s}, t)$, we assume that the point $\mathbf{s}$ moves under Newton's first law:

$$\frac{d\mathbf{s}(t)}{dt} = \mathbf{V}(\mathbf{s}, t). \tag{4}$$

The CH equation on an evolving surface is defined by

$$\frac{\partial^* \phi(\mathbf{s}, t)}{\partial t} + \phi \nabla_\tau \cdot \mathbf{V}(\mathbf{s}, t) = M \Delta_\tau \mu(\mathbf{s}, t), \quad \mathbf{s} \in S, \ t > 0, \tag{5}$$

$$\mu(\mathbf{s}, t) = F'(\phi(\mathbf{s}, t)) - \epsilon^2 \Delta_\tau \phi(\mathbf{s}, t), \tag{6}$$

with the initial condition

$$\phi(\mathbf{s}, 0) = \phi_0(\mathbf{s}), \ \text{in } S, \tag{7}$$

where $\partial^* \phi / \partial t = \partial \phi / \partial t + \mathbf{V} \cdot \nabla \phi$ is the material derivative of $\phi$ and $\nabla$ is the gradient operator [23]. The order parameter $\phi(\mathbf{s}, t)$ denotes the difference between the concentrations of the binary mixture. We let $M = 1$ for the sake of convenience. The operators $\Delta_\tau$ and $\nabla_\tau$ denote the tangential (surface) Laplacian and gradient operators, respectively. Furthermore, $F(\phi) = 0.25(\phi^2 - 1)^2$ is a free energy density and $\epsilon$ is a positive constant. If the evolving surface $S(\mathbf{s}, t)$ is closed, then there is no boundary condition. Otherwise, we consider the Neumann boundary condition for both the chemical potential $\mu$ and the order parameter, for a simpler presentation, i.e., $\hat{\mathbf{n}} \cdot \nabla_\tau \phi(\mathbf{s}, t) = \hat{\mathbf{n}} \cdot \nabla_\tau \mu(\mathbf{s}, t) = 0$. Here, $\hat{\mathbf{n}}$ is normal to the boundary and tangent to the surface $S$. Note that our method can also be extended to the cases of Dirichlet and periodic boundary conditions. It should be noted that the term $\phi \nabla_\tau \cdot \mathbf{V}$ in Eq. (5) can not be omitted, because of the Leibniz formula [34]:

$$\frac{d}{dt} \int_S \phi d\mathbf{s} = \int_S \left( \frac{\partial^* \phi}{\partial t} + \phi \nabla_\tau \cdot \mathbf{V} \right) d\mathbf{s}. \tag{8}$$

The CH equation on an evolving surface satisfies total mass conservation:

$$\begin{aligned} 0 &= \int_S \left( \frac{\partial^* \phi}{\partial t} + \phi \nabla_\tau \cdot \mathbf{V} - M \Delta_\tau \mu \right) d\mathbf{s} \\ &= \int_S \left( \frac{\partial^* \phi}{\partial t} + \phi \nabla_\tau \cdot \mathbf{V} \right) d\mathbf{s} - M \int_{\partial S} \hat{\mathbf{n}} \cdot \nabla_\tau \mu \, d\tau = \frac{d}{dt} \int_S \phi \, d\mathbf{s}. \end{aligned} \tag{9}$$

In order to simplify the calculation, we rewrite the term $\mathbf{V} \cdot \nabla \phi + \phi \nabla_\tau \cdot \mathbf{V}$ as

$$\begin{aligned} \mathbf{V} \cdot \nabla \phi + \phi \nabla_\tau \cdot \mathbf{V} &= (V_\nu \mathbf{N} + \mathbf{V}_\tau) \cdot \nabla \phi + \phi \nabla_\tau \cdot (V_\nu \mathbf{N} + \mathbf{V}_\tau) \\ &= V_\nu \mathbf{N} \cdot \nabla \phi + \phi V_\nu \nabla_\tau \cdot \mathbf{N} + \nabla_\tau \cdot (\phi \mathbf{V}_\tau). \end{aligned} \tag{10}$$

Furthermore, if we denote by $H(\mathbf{s})$ the sum of the principal curvatures at $\mathbf{s}$ of the surface, then $H(\mathbf{s})$ can be written as

$$H(\mathbf{s}) = \nabla_\tau \cdot \mathbf{N}(\mathbf{s}), \tag{11}$$

and $H(\mathbf{s})$ satisfies $\Delta_\tau \mathbf{s} = H(\mathbf{s}) \mathbf{N}(\mathbf{s})$ [35]. Then, we can calculate $H(\mathbf{s})$ as

$$H(\mathbf{s}) = \langle \Delta_\tau \mathbf{s}, \ \mathbf{N}(\mathbf{s}) \rangle. \tag{12}$$

Using Eqs. (10) and (11), we can rewrite Eq. (5) as

$$\frac{\partial \phi}{\partial t} + V_\nu \mathbf{N} \cdot \nabla \phi + \phi V_\nu H + \nabla_\tau \cdot (\phi \mathbf{V}_\tau) = M \Delta_\tau \mu. \tag{13}$$

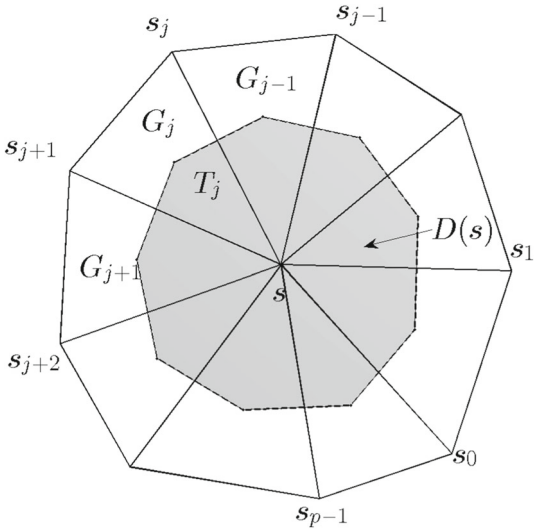


**Fig. 1** A vertex $\mathbf{s}$ with several neighbors for evaluating the divergence and Laplace–Beltrami operators

Let $\phi$ be constant in the direction normal to the surface [27]. Then, we can set $V_v\mathbf{N}\cdot\nabla\phi = 0$ and rewrite Eq. (13) as

$$\frac{\partial\phi}{\partial t} + \phi V_v H + \nabla_\tau \cdot (\phi \mathbf{V}_\tau) = M\Delta_\tau\mu. \tag{14}$$

Finally, our proposed method for the CH equation on evolving surfaces can be written as follows:

$$\frac{\partial\phi}{\partial t} + \phi V_v H + \nabla_\tau \cdot (\phi \mathbf{V}_\tau) = M\Delta_\tau\mu, \tag{15}$$

$$\mu = F'(\phi) - \epsilon^2\Delta_\tau\phi. \tag{16}$$

## 3 Numerical Solution

Instead of using the evolving surface FEM for the CH equation [23], we employ a simple approach to approximate the regular evolving surface by an evolving triangular mesh. Let us introduce basic notations on a triangular mesh, and then present our proposed method for the CH equation on evolving surfaces.

### 3.1 Discretizations of the Laplace–Beltrami Operator

Let us define a triangular surface mesh by $(S,\ F)$, where $S = \{\mathbf{s}_i | 1 \le i \le N_s\}$ is the list of vertices and $F = \{T_k | 1 \le k \le N_F\}$ is the list of triangles. Here, $N_s$ is the number of vertices and $N_F$ is the number of triangles. For $j = 0, 1, \ \ldots, p$, let $\mathbf{s}_j$ be the neighboring vertices of $\mathbf{s} \in S$, where $\mathbf{s}_0 = \mathbf{s}_p$. We label the vertices $\mathbf{s}_j$ counterclockwise relative to the outside of the shape. Let $T_j$ be the triangle with three vertices $\mathbf{s}$, $\mathbf{s}_j$, and $\mathbf{s}_{j+1}$ (see Fig. 1).

Let the normal vector $\mathbf{N}(\mathbf{s})$ at $\mathbf{s} \in S$ be

$$\mathbf{N}(\mathbf{s}) = \sum_{j=0}^{p-1} \omega_j \mathbf{N}_j \Big/ \left\| \sum_{j=0}^{p-1} \omega_j \mathbf{N}_j \right\|, \tag{17}$$

where $\mathbf{N}_j$ is the unit normal to $T_j$, $\omega_j = \|\mathbf{G}_j - \mathbf{s}\|^{-2} / \sum_{i=0}^{p-1} \|\mathbf{G}_i - \mathbf{s}\|^{-2}$, and $\mathbf{G}_j = (\mathbf{s}_j + \mathbf{s}_{j+1} + \mathbf{s})/3$ [36,37]. We defined the discrete divergence $\nabla_\tau \cdot \mathbf{X}$ by

$$\begin{aligned}\nabla_\tau \cdot \mathbf{X}(\mathbf{s}) = \frac{1}{D(\mathbf{s})} \sum_{j=0}^{p-1} \frac{\|\mathbf{G}_{j+1} - \mathbf{G}_j\|}{6} \Big( & 2\langle \mathbf{X}(\mathbf{G}_j),\ \mathbf{n}(\mathbf{G}_j)\rangle + 2\left\langle \mathbf{X}(\mathbf{G}_{j+1}),\ \mathbf{n}(\mathbf{G}_{j+1})\right\rangle \\ & + \left\langle \mathbf{X}(\mathbf{G}_j),\ \mathbf{n}(\mathbf{G}_{j+1})\right\rangle + \left\langle \mathbf{X}(\mathbf{G}_{j+1}),\ \mathbf{n}(\mathbf{G}_j)\right\rangle \Big),\end{aligned}$$

where $D(\mathbf{s}) = \sum_{i=0}^{p-1} |\hat{T}_i|$ and $|\hat{T}_i|$ is the area of $\hat{T}_i$, which is a triangle with $\mathbf{s}_j$, $\mathbf{G}_j$, and $\mathbf{G}_{j+1}$. The vectors $\mathbf{n}(\mathbf{G}_j)$ and $\mathbf{n}(\mathbf{G}_{j+1})$ are defined as

$$\mathbf{n}(\mathbf{G}_j) = \frac{(\mathbf{G}_{j+1} - \mathbf{G}_j) \times \mathbf{N}_j}{\|(\mathbf{G}_{j+1} - \mathbf{G}_j) \times \mathbf{N}_j\|} \text{ and } \mathbf{n}(\mathbf{G}_{j+1}) = \frac{(\mathbf{G}_{j+1} - \mathbf{G}_j) \times \mathbf{N}_{j+1}}{\|(\mathbf{G}_{j+1} - \mathbf{G}_j) \times \mathbf{N}_{j+1}\|}.$$

We approximate the Laplace–Beltrami operator by a second-order discretization [38]:

$$\begin{aligned}&\Delta_\tau \phi(\mathbf{s}) \approx \nabla_\tau \cdot [\nabla_h \phi(\mathbf{s})] \\ &= \frac{1}{D(\mathbf{s})} \sum_{j=0}^{p-1} \frac{\|\mathbf{G}_{j+1} - \mathbf{G}_j\|}{6} \Big( 2\left\langle \nabla_h \phi(\mathbf{G})_j,\ \mathbf{n}(\mathbf{G}_j)\right\rangle + 2\left\langle \nabla_h \phi(\mathbf{G})_{j+1},\ \mathbf{n}(\mathbf{G}_{j+1})\right\rangle \\ &\quad + \left\langle \nabla_h \phi(\mathbf{G})_j,\ \mathbf{n}(\mathbf{G}_{j+1})\right\rangle + \left\langle \nabla_h \phi(\mathbf{G})_{j+1},\ \mathbf{n}(\mathbf{G}_j)\right\rangle \Big).\end{aligned} \tag{18}$$

To compute the gradient of $\phi(\mathbf{G}_j)$ to second-order accuracy, we employ an efficient method that is second-order accurate. A similar method can be found in our previous work [39]. Because $\nabla_h \phi(\mathbf{G}_j)$, $\mathbf{s}_j - \mathbf{G}_j$, and $\mathbf{s}_{j+1} - \mathbf{G}_j$ are in the same plane, we can assume that $\nabla_h \phi(\mathbf{G}_j)$ has the following form:

$$\nabla_h \phi(\mathbf{G}_j) = \alpha_j (\mathbf{s}_j - \mathbf{G}_j) + \beta_j (\mathbf{s}_{j+1} - \mathbf{G}_j), \tag{19}$$

where $\alpha_j$ and $\beta_j$ are defined as

$$\begin{pmatrix} \alpha_j \\ \beta_j \\ \gamma_j \end{pmatrix} = B_j^{-1} \begin{pmatrix} \phi(\mathbf{s}) - \phi(\mathbf{G}_j) \\ \phi(\mathbf{s}_j) - \phi(\mathbf{G}_j) \\ \phi(\mathbf{s}_{j+1}) - \phi(\mathbf{G}_j) \end{pmatrix}, \tag{20}$$

$$B_j = \begin{pmatrix} \langle \mathbf{s}_j - \mathbf{G}_j, \mathbf{s} - \mathbf{G}_j \rangle & \langle \mathbf{s}_{j+1} - \mathbf{G}_j, \mathbf{s} - \mathbf{G}_j \rangle & \|\mathbf{s} - \mathbf{G}_j\|_2 \\ \langle \mathbf{s}_j - \mathbf{G}_j, \mathbf{s}_j - \mathbf{G}_j \rangle & \langle \mathbf{s}_{j+1} - \mathbf{G}_j, \mathbf{s}_j - \mathbf{G}_j \rangle & \|\mathbf{s}_j - \mathbf{G}_j\|_2 \\ \langle \mathbf{s}_j - \mathbf{G}_j, \mathbf{s}_{j+1} - \mathbf{G}_j \rangle & \langle \mathbf{s}_{j+1} - \mathbf{G}_j, \mathbf{s}_{j+1} - \mathbf{G}_j \rangle & \|\mathbf{s}_{j+1} - \mathbf{G}_j\|_2 \end{pmatrix}. \tag{21}$$

From Eqs. (18)–(21), $\Delta_\tau \phi(\mathbf{s})$ at the vertex $\mathbf{s}$ can be defined as

$$\Delta_\tau \phi(\mathbf{s}_i) \approx \mathcal{L}_i \cdot \begin{pmatrix} \phi(\mathbf{s}_1) \\ \phi(\mathbf{s}_2) \\ \vdots \\ \phi(\mathbf{s}_{N_S}) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \Delta_\tau \phi(\mathbf{s}_1) \\ \Delta_\tau \phi(\mathbf{s}_2) \\ \vdots \\ \Delta_\tau \phi(\mathbf{s}_{N_S}) \end{pmatrix} \approx \begin{pmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \\ \vdots \\ \mathcal{L}_{N_V} \end{pmatrix} \cdot \begin{pmatrix} \phi(\mathbf{s}_1) \\ \phi(\mathbf{s}_2) \\ \vdots \\ \phi(\mathbf{s}_{N_s}) \end{pmatrix},$$

where $\mathcal{L}_i$ is an $1 \times N_s$ matrix, for $i = 1,\ 2,\ \ldots,\ N_s$, using Eqs. (18)–(21). The $N_s \times N_s$ matrix $\mathcal{L} = (\mathcal{L}_1,\ \mathcal{L}_2,\ \ldots,\ \mathcal{L}_{N_V})'$ is defined as the Laplace matrix of the surface.

### 3.2 Proposed Scheme

Let $T$ be the final time and $N_t$ be the total number of time steps. Then, at the vertex $\mathbf{s}_i$ let $\phi_i^n$ be the approximation of $\phi(\mathbf{s}_i, n\Delta t)$ , where $\Delta t = T/N_t$. To obtain a high-order numerical solution, we apply a linearly extrapolated scheme based on the Crank–Nicolson scheme to Eqs. (15) and (16). Then, the resulting discrete system of equations is

$$\frac{\phi^{n+1}-\phi^n}{\Delta t} + \frac{3\phi^n-\phi^{n-1}}{2}\mathbf{V}_v{}^{n+\frac{1}{2}}\frac{H^n+H^{n+1}}{2} + \nabla_\tau \cdot \left(\frac{3\phi^n-\phi^{n-1}}{2}\mathbf{V}_\tau^{n+\frac{1}{2}}\right) = \Delta_\tau \mu^{n+\frac{1}{2}}, \tag{22}$$

$$\mu^{n+\frac{1}{2}} = \frac{3}{2}(F'(\phi^n)-\lambda\phi^n) - \frac{1}{2}(F'(\phi^{n-1})-\lambda\phi^{n-1}) + \frac{\lambda}{2}(\phi^{n+1}+\phi^n) - \frac{\epsilon^2}{2}(\Delta_\tau\phi^{n+1}+\Delta_\tau\phi^n). \tag{23}$$

Here, $\lambda$ is a positive stabilizing parameter. Because the velocity field $\mathbf{V}(\mathbf{s}, t)$ is known, the positions of the surface points can be updated as follows:

$$\frac{\mathbf{s}^{n+1}-\mathbf{s}^n}{\Delta t} = \mathbf{V}^{n+\frac{1}{2}}. \tag{24}$$

The discrete system (22)–(24) is formally of second-order accuracy in space and time. Although we have not given a detailed proof of the accuracy of the proposed form, the computational experiments in Sect. 4.2 demonstrate that the proposed scheme is of second-order accuracy in space and time. We describe the preprocessing step by rewriting Eqs. (22) and (23) as follows:

$$\begin{pmatrix} \mathcal{I} & \Delta t\mathcal{L} \\ -\lambda/2\mathcal{I}+\epsilon^2/2\mathcal{L} & \mathcal{I} \end{pmatrix} \begin{pmatrix} \phi_1^{n+1} \\ \vdots \\ \phi_{N_V}^{n+1} \\ \mu_1^{n+\frac{1}{2}} \\ \vdots \\ \mu_{N_V}^{n+\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & -\epsilon^2/2\mathcal{L} \end{pmatrix} \begin{pmatrix} \phi_1^n \\ \vdots \\ \phi_{N_V}^n \\ \phi_1^n \\ \vdots \\ \phi_{N_V}^n \end{pmatrix}$$

$$+ \begin{pmatrix} \phi_1^n - \Delta t\frac{3\phi_1^n-\phi_1^{n-1}}{2}V_v{}_1{}^{n+\frac{1}{2}}\frac{H_1^n+H_1^{n+1}}{2} - \Delta t\nabla_\tau\cdot(\frac{3\phi_1^n-\phi_1^{n-1}}{2}\mathbf{V}_\tau{}_1{}^{n+\frac{1}{2}}) \\ \vdots \\ \phi_{N_V}^n - \Delta t\frac{3\phi_{N_V}^n-\phi_{N_V}^{n-1}}{2}V_v{}_{N_V}{}^{n+\frac{1}{2}}\frac{H_{N_V}^n+H_{N_V}^{n+1}}{2} - \Delta t\nabla_\tau\cdot(\frac{3\phi_{N_V}^n-\phi_{N_V}^{n-1}}{2}\mathbf{V}_\tau{}_{N_V}{}^{n+\frac{1}{2}}) \\ \frac{3}{2}(F'(\phi_1^n)-\lambda\phi_1^n) - \frac{1}{2}(F'(\phi_1^{n-1})-\lambda\phi_1^{n-1}) + \frac{\lambda}{2}\phi_1^n \\ \vdots \\ \frac{3}{2}(F'(\phi_{N_V}^n)-\lambda\phi_{N_V}^n) - \frac{1}{2}(F'(\phi_{N_V}^{n-1})-\lambda\phi_{N_V}^{n-1}) + \frac{\lambda}{2}\phi_{N_V}^n \end{pmatrix}. \tag{25}$$

To solve the resulting discrete linear system (25), we employ a biconjugate gradient stabilized method [40]. For additional details regarding the solver of discrete system, we refer the reader to our previous work [39]. The biconjugate gradient method is performed until the iterative error is smaller than 1.0E−8. Note that if $\mathbf{V}(\mathbf{s}, t)$ is unknown, then in order to maintain a second-order accuracy we can interpolate $\mathbf{V}^{n+\frac{1}{2}}$ using information from the $n-1$ and $n$ levels, such as $\mathbf{V}^{n+\frac{1}{2}} = (3\mathbf{V}^n - \mathbf{V}^{n-1})/2$. The main steps of our algorithm are summarized as follows:

---

Initializing $\mathbf{s}^0$, $F$, $\phi^0$, and $\mathbf{V}^0$.
Setting $\phi^{-1} = \phi^0$ and $\mathbf{V}^{-1} = \mathbf{V}^0$.
Computing $\mathbf{N}^0$, ${V_v}^0$, $\mathbf{V}_\tau^0$, and $H^0$ by using Eqs. (17), (2), (3), (12), respectively.
for ( $n$=1; $n \leq N_t$ ; $n$++)
  Updating $\mathbf{s}^{n+1}$ by solving Eq. (24).
  Computing $\mathbf{N}^{n+1}$, ${V_v}^{n+1}$, $\mathbf{V}_\tau^{n+1}$, and $H^{n+1}$ by using Eqs. (17), (2), (3), (12), respectively.
  Updating $\phi^{n+1}$ by solving Eqs. (22) and (23).
end for

---

Because the vertices of triangles evolve according to the velocity field, some triangles may be compressed or stretched in an undesirable manner. To investigate the effect of the mesh quality on the computation of the Laplace–Beltrami operator in Eq. (18), we evaluate $\Delta_\tau \psi$ on the uniform, 10% non-uniform, and 20% non-uniform surface meshes, which are illustrated in Fig. 2a. Here, the edge of each triangle on the uniform surface mesh is almost the same. The 10 and 20% non-uniform surface meshes are obtained by introducing 10 and 20% random noises to the uniform mesh. Here, $\psi(s_x, s_y, s_z) = 0.1\cos(\pi s_x)\cos(\pi s_y)\cos(\pi s_z)$ is defined on the unit spherical surface. Figure 2b shows the plots of $\Delta_\tau \psi$ for different mesh structures. The agreement between results obtained from the uniform and 10% non-uniform surface meshes is clear. These results also suggest that the Laplace–Beltrami operator in Eq. (18) works effectively on a moderately non-uniform surface mesh. However, if a strongly non-uniform surface mesh is used, then the numerical solution is not accurate.

To maintain a high-quality triangulation, we employ the remeshing technique described in [41,42]. A brief description of the remeshing algorithm is summarized here. First, we construct a signed distance function in the whole domain using the surface mesh. Then, we calculate the net force for each surface point, and move each point under the computed net force. Here, the net forces work in the following manner. If the distance between two points on a triangular mesh is smaller than a given value, then the forces push the points away from each other. Otherwise, they attract the points towards each other. Moved surface points may not be on the original surface, in which case we project them back onto the original surface by using the signed distance function. This procedure is repeated until the mesh distribution is almost uniform. As shown in Fig. 2, the result of using our method guarantees a mesh-independent solution with 10% random noise. We adopt the remeshing procedure after every twelve time steps.

## 4 Numerical Results

In this section, we present some computational experiments to demonstrate the performance of our proposed algorithm. Unless otherwise specified, we take $\epsilon = 0.1$ and $\lambda = 2$.

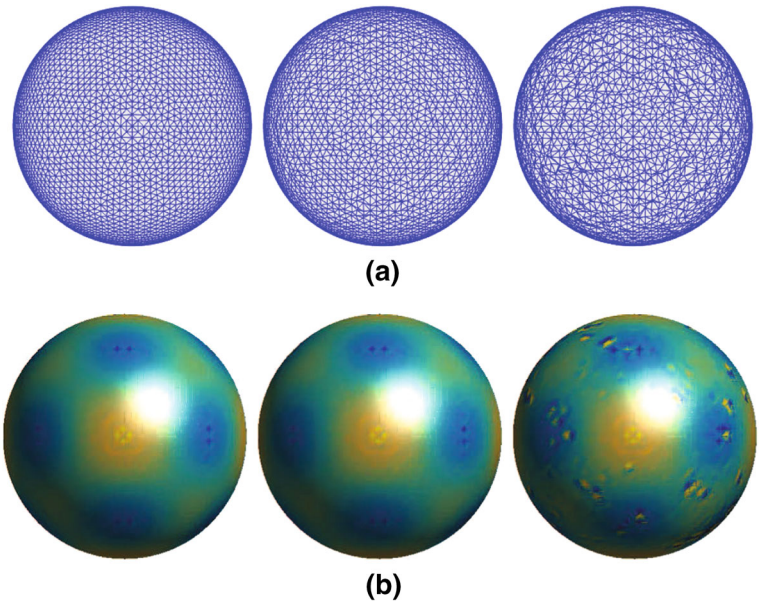


**Fig. 2** **a** Surface mesh structures. From left to right, these are uniform, 10% non-uniform, and 20% non-uniform surface meshes, respectively. **b** $\Delta_\tau \psi$ on different surface mesh structures. Here, $\psi(s_x, s_y, s_z) = 0.1\cos(\pi s_x)\cos(\pi s_y)\cos(\pi s_z)$ is defined on the unit spherical surface

### 4.1 Spinodal Decomposition on Translate Surfaces

To compare the spinodal decomposition results for a binary mixture on moving and fixed surfaces, we consider the passive advection of a unit sphere by a constant velocity, $\mathbf{V} = (0.2, 0, 0)$. Spinodal decomposition is a mechanism by which a mixture separates into different phases [1]. The initial condition is set as $\phi(s_x, s_y, s_z, 0) = 0.1\cos(\pi s_x)\cos(\pi s_y)\cos(\pi s_z)$ on the surface of the unit sphere. Here $s_x$, $s_y$, and $s_z$ are the components in the $x$-, $y$-, and $z$-directions, respectively. The edge of each triangle on the surface is almost the same as $h = 0.05$. The computation is run up to $T = 10$, with $\Delta t = 0.01$. Figure 3a, b show the spinodal decomposition of a binary mixture on a fixed surface and a moving surface, respectively, and we can observe the effect of the given velocity field.

### 4.2 Convergence Test

We carry out numerical experiments to test the spatial and temporal convergence of our proposed method. All parameters and initial conditions are the same as in Sect. 4.1. All numerical solutions are computed up to time $T = 0.02$. Because there is no analytical solution for the problem, we use a numerical reference solution $\phi^{\text{ref}}$, which is computed with a very small space step or a very small time step. To obtain the spatial convergence rate, we run a number of simulations with $h = 0.2$, 0.1 and 0.05. Here, the time step is fixed as $\Delta t = 0.0001$. The numerical reference solution is obtained with a very small space grid with $h = 0.025$. We define the error as the $l_2$-norm of the difference with the weighted average of the reference solution: $e_{hi} := \phi_{hi} - (\zeta_i \phi_p^{\text{ref}} + \eta_i \phi_q^{\text{ref}} + \theta_i \phi_r^{\text{ref}})$. Here, $p, q$, and $r$ are appropriate

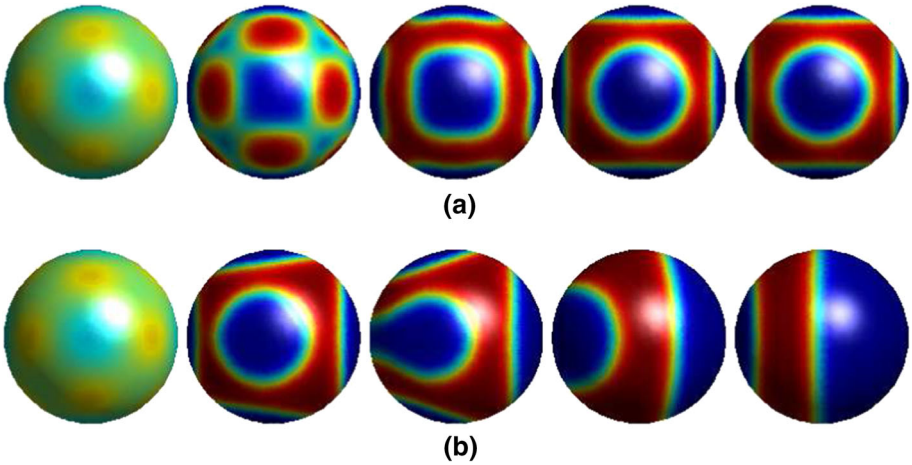


**Fig. 3** Spinodal decomposition of a binary mixture on a fixed surface (**a**) and a moving surface (**b**). The computational times of (**a**) and (**b**) are at $t = 0,\ 0.1,\ 0.3,\ 1,\ 10$ and $t = 0,\ 3,\ 5,\ 7,\ 10$, respectively, from left to right

**Table 1** Error and convergence results with various mesh grids

| $h$ | 0.2 | | 0.1 | | 0.05 |
|---|---|---|---|---|---|
| Uniform: $l_2$-error | 1.428E−3 | | 3.974E−4 | | 9.797E−5 |
| Uniform: Rate | | 1.85 | | 2.02 | |
| Non-uniform: $l_2$-error | 7.650E−3 | | 2.133E−3 | | 5.218E−4 |
| Non-uniform: Rate | | 1.84 | | 2.03 | |

Here, $\Delta t = 0.0001$ is fixed. The numerical reference solution is obtained with a very small space grid with $h = 0.025$

fine reference grid indexes in a triangle, and $\zeta_i$, $\eta_i$, and $\theta_i$ satisfy $\mathbf{s}_{hi} = \zeta_i \mathbf{s}_p^{\text{ref}} + \eta_i \mathbf{s}_q^{\text{ref}} + \theta_i \mathbf{s}_r^{\text{ref}}$. Table 1 lists the errors and the convergence rates $\log_2(\|e_h\|_2/\|e_{\frac{h}{2}}\|_2)$. To assess the effect of the mesh grid structure, we use a uniform mesh grid structure, in which the edge of each triangle is almost the same (see the first column of Fig. 2a). Here, we also consider a non-uniform mesh grid structure (see the second column of Fig. 2a), which is obtained by randomly changing the edges on the uniform mesh grid structure with 10% random amplitude. The results in Table 1 confirm that the numerical solution with the uniform mesh grid structure is better than that obtained with the non-uniform mesh grid. The results also suggest that the schemes with two mesh grid structures are indeed second-order accurate with respect to space.

To test the convergence rate for time discretization, we set the space step size to $h = 0.05$ and choose time steps $\Delta t = 1.250\text{E}{-3},\ 6.250\text{E}{-4},\ 3.125\text{E}{-4}$, and $1.563\text{E}{-4}$. The numerical reference solution is computed with a very small time size of $\Delta t = 7.813\text{E}{-5}$. The convergence rate is defined as $\log_2(\|e_{\Delta t}\|_2/\|e_{\frac{\Delta t}{2}}\|_2)$. Table 2 lists the errors and rates of convergence with uniform and non-uniform mesh structures. A second-order accuracy with respect to time is observed.

**Table 2** Error and convergence results with various time steps

| $\Delta t$ | 2.500E−3 | | 1.250E−3 | | 6.250E−4 | | 3.125E−4 |
|---|---|---|---|---|---|---|---|
| Uniform: $l_2$-error | 5.712E−4 | | 1.528E−4 | | 3.953E−5 | | 9.320E−6 |
| Uniform: Rate | | 1.91 | | 1.95 | | 2.08 | |
| Non-uniform: $l_2$-error | 2.918E−3 | | 7.934E−4 | | 2.015E−4 | | 4.823E−5 |
| Non-uniform: Rate | | 1.88 | | 1.98 | | 2.06 | |

Here, $h = 0.05$ is fixed. The reference numerical solution is obtained with a very small temporal grid with $\Delta t = 7.813\text{E}{-5}$

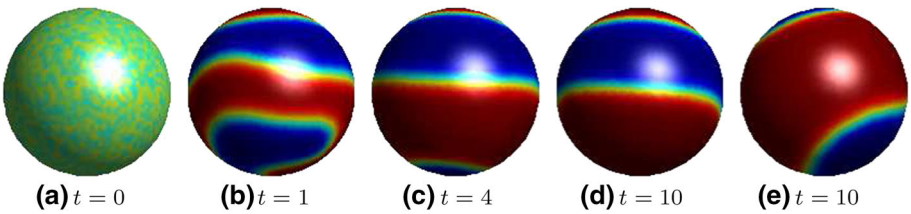


**Fig. 4** Spinodal decomposition of a binary mixture with rotational motion. **a–d** Results on a moving surface. **e** Result on a fixed surface with the same initial condition as shown in (**a**). The computational times are listed below each figure. The initial surface is the surface of a unit sphere

### 4.3 Spinodal Decomposition Associated with Rotational Motion

We consider the passive advection of a unit sphere by a given velocity field, $\mathbf{V}(\mathbf{s}) = (0.5\pi s_y, -0.5\pi s_x, 0)$. If the sphere is assumed to be centered at $(0, 0, 0)$, then the normal velocity $V_v(\mathbf{s}, t)\mathbf{N}(\mathbf{s}, t)$ is zero, owing to the rotational velocity field. Therefore, the shape of the sphere should not change as a result of the rotation. The initial condition is a random perturbation, $\phi(\mathbf{s}, 0) = 0.1\text{rand}(\mathbf{s})$, where $\text{rand}(\mathbf{s})$ is a random number between $-1$ and 1.

Figure 4 illustrates the evolution of the phase separation by the background velocity flow. The simulations are computed up to time $T = 10$ with $\Delta t = 0.01$. We can observe the alignment in the case of the velocity flow.

### 4.4 Spinodal Decomposition Under Mean Curvature Flow

Mean curvature flow is the geometric flow of a hypersurface, which has been applied to image processing [43,44] and surface reconstruction [45]. A surface evolves under mean curvature flow, in which the normal velocity of a moving hypersurface is equal to the negative mean curvature:

$$\frac{d\mathbf{s}(t)}{dt} = -\varepsilon H(\mathbf{s})\mathbf{N}(\mathbf{s}, t).$$

Here, $\varepsilon$ is a positive constant. Under the mean curvature flow, convex points move inward, while concave points move outward. We consider the CH equation under a mean curvature flow. The initial surface of a box of length eight on each side satisfies the following equation:

$$3 + \cos\left(4\cos^{-1}\left(\frac{s_z}{\sqrt{s_x^2 + s_y^2 + s_z^2}}\right)\right) - \sqrt{s_x^2 + s_y^2 + s_z^2} = 0. \tag{26}$$

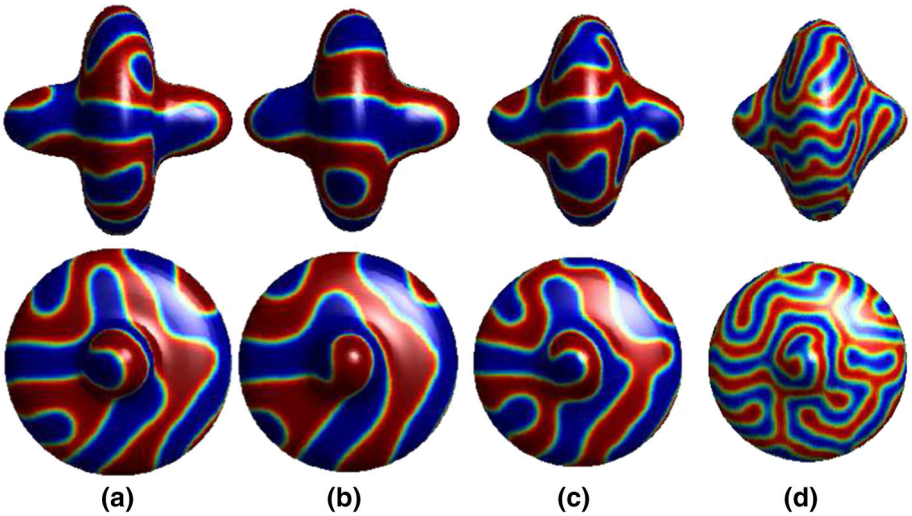


**Fig. 5** Spinodal decomposition of a binary mixture under mean curvature flow. **a** $t = 4$, **b** $t = 8$, **c** $t = 12$, and **d** $t = 16$. The results in (**a**) are obtained on the fixed surface by using the CH equation with a random perturbation. (**b–d**) are computed with a moving surface under the mean curvature flow

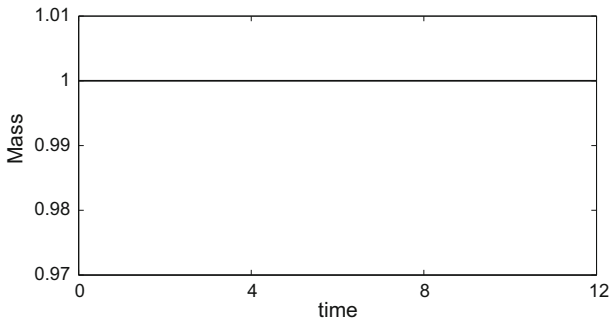


**Fig. 6** Evolution of the total mass under the mean curvature flow

First, we fix the initial surface and run the CH equation with a random initial condition up to time $t = 4$. Then, we let the surface evolve under the mean curvature flow. Here, we set$\varepsilon = 0.1$ and $\Delta t = 0.001$.

The evolution is illustrated in Fig. 5. The computational times are listed below each figure. It can be observed that the surface shrinks inward under the mean curvature flow. Furthermore, at earlier stages, the CH equation coarsens a solution that is already phase separated, and in the later stages the separated phases are mixed together. The reason for this is that in the earlier stages $H\mathbf{N}$ is small, which implies that the CH equation is dominant. Hoever, as the surface moves $H\mathbf{N}$ becomes increasingly large, which implies that the moving phase becomes dominant as a result of the mixing of the separated phases.

Figure 6 illustrates the evolution of the total mass under the mean curvature flow. Note that we normalized the total mass by the total mass at the initial time. We observe that our proposed method clearly maintains the total conservation of mass.

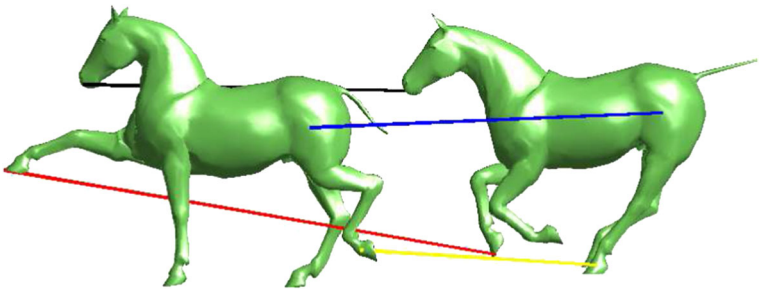

**Fig. 7** A family of controlled surfaces in which the numbers of vertices and triangles are the same for each triangular surface

### 4.5 Spinodal Decomposition on a Family of Controlled Surfaces

Surface correspondence is a fundamental notion in computer vision and related fields. Given two triangulated mesh surfaces, a correspondence is a mapping from any point on one surface to a corresponding point on the other (see Fig. 7). Here, the numbers of vertices and triangles in the triangular surfaces are the same.

In this section, we will consider the CH equation on a family of controlled surfaces. By $(S^p,\ F^p)$, denote the $p$th triangular surface mesh. Let the time evolution be $\hat{t}^p$ from $(S^p,\ F^p)$ to $(S^{p+1},\ F^{p+1})$. Using a linear interpolation, we can assume that the evolving velocity $\mathbf{V}_j^p$ of each vertex $\mathbf{s}_j$ is constant, and can be defined as

$$\mathbf{V}_j^p = \frac{\mathbf{s}_j^{p+1} - \mathbf{s}_j^p}{\hat{t}^p}. \tag{27}$$

Therefore, using the evolving velocity $\mathbf{V}^p$ we can enforce that the $p$th triangular surface mesh becomes the $(p+1)$th triangular surface mesh after time $\hat{t}^p$. Figure 8a shows the results of the CH equation at a time of 2 on a fixed horse-shaped surface with a random perturbation. Then beginning with results as shown in Fig. 8a, we evolve the surfaces from Fig. 8 a–c. Here, the settings $\hat{t}^1 = \hat{t}^2 = 2$ are used. The results suggest that our algorithm performs well on a family of controlled surfaces.

### 4.6 Spinodal Decomposition on Evolving Cell Surfaces

The growth and division of cells are fundamental phenomena that create and maintain life. In [42], Li et al. proposed a mathematical equation for cell growth and division, and simulated the governing equation using the immersed boundary method (IBM) in 3D, which extended the work of Li et al. [46]. In this section, we will consider the CH equation on an evolving cell surface by combining an immersed boundary method with a mathematical model for the growth and division. Let $\mathbf{s}(t)$ denote the cell boundary, and let $\mathbf{Y}_1(t)$ and $\mathbf{Y}_2(t)$ be the cell nuclei at time $t$. The governing equations for the cell growth and division are

$$\begin{aligned}\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} + \mathbf{u}(\mathbf{x},t)\cdot\nabla\mathbf{u}(\mathbf{x},t) &= -\nabla p(\mathbf{x},t) + \frac{1}{Re}\Delta\mathbf{u}(\mathbf{x},t) + \frac{1}{Se}\nabla S(\mathbf{x},t)\\ &\quad + \frac{1}{We}\mathbf{SF}(\mathbf{x},t) + \frac{1}{De}\mathbf{DF}(\mathbf{x},t), \qquad (28)\\ \nabla\cdot\mathbf{u}(\mathbf{x},t) &= \varsigma S(\mathbf{x},t), \qquad (29)\end{aligned}$$

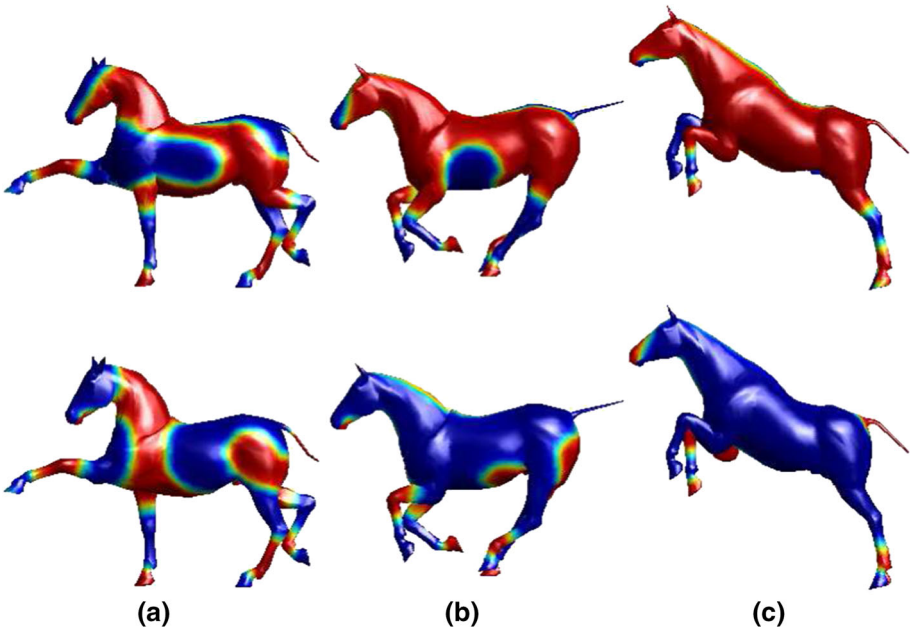


**Fig. 8** Spinodal decomposition of a binary mixture on a family of controlled surfaces. From top to bottom, these are results from the front and back views, respectively with **a** $t = 0$, **b** $t = 2$, and **c** $t = 4$. Note that the result in (**a**) is obtained at time 2 on a fixed horse-shaped surface with a random perturbation. The initial surface is in a box of size $(0, 1) \times (0, 3) \times (0, 3)$

where

$$S(\mathbf{x}, t) = \sum_{m=1}^{2} (1 - \hat{H}(t - t_0))\delta_c(\mathbf{x} - \mathbf{Y}_m(t)), \tag{30}$$

$$\mathbf{SF}(\mathbf{x}, t) = \int_{\Gamma} \mathbf{f}_1(\mathbf{s})\delta_c(\mathbf{x} - \mathbf{s}(\mathcal{A}, t))d\mathcal{A}, \tag{31}$$

$$\mathbf{f}_1(\mathbf{s}) = H(\mathbf{s})\mathbf{N}(\mathbf{s}), \tag{32}$$

$$\mathbf{DF}(\mathbf{x}, t) = \int_{\Gamma} \mathbf{f}_2(\mathbf{s}(\mathcal{A}, t_0))\delta_c(\mathbf{x} - \mathbf{s}(\mathcal{A}, t))d\mathcal{A}, \tag{33}$$

$$\mathbf{f}_2(\mathbf{X}(\mathcal{A}, t_0)) = \frac{\hat{H}(t - t_0)\mathbf{n}_{div}(\mathbf{s}(\mathcal{A}, t))}{\eta(\mathbf{s}(\mathcal{A}, t_0))}, \tag{34}$$

$$\eta(\mathbf{s}(\mathcal{A}, t_0)) = \frac{\big||\mathbf{s}(\mathcal{A}, t_0) - \mathbf{Y}_1(t_0)| - |\mathbf{s}(\mathcal{A}, t_0) - \mathbf{Y}_2(t_0)|\big|}{\epsilon|\mathbf{Y}_1(t_0) - \mathbf{Y}_2(t_0)|} + 1, \tag{35}$$

$$\mathbf{n}_{div}(\mathbf{s}(\mathcal{A}, t)) = \frac{(\mathbf{Y}_1(t_0) - \mathbf{s}(\mathcal{A}, t)) + (\mathbf{Y}_2(t_0) - \mathbf{s}(\mathcal{A}, t))}{|(\mathbf{Y}_1(t_0) - \mathbf{s}(\mathcal{A}, t)) + (\mathbf{Y}_2(t_0) - \mathbf{s}(\mathcal{A}, t))|}. \tag{36}$$

Lagrangian cell surface points move according to the following:

$$\frac{\partial \mathbf{s}}{\partial t} = \mathbf{U}(\mathbf{s}), \tag{37}$$

$$\mathbf{U}(\mathbf{s}) = \int_{\Omega} \mathbf{u}(\mathbf{x})\delta_c(\mathbf{x} - \mathbf{s})d\mathbf{x}, \tag{38}$$

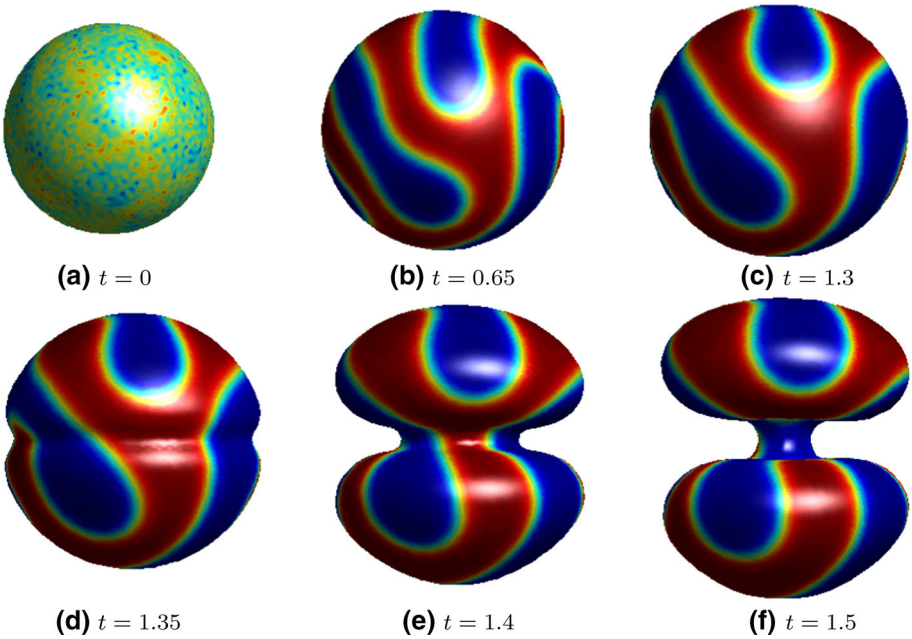


**Fig. 9** Spinodal decomposition of a binary mixture on evolving cell surfaces. The initial shape of the cell is a unit sphere on the computational domain $\Omega = (-2, 2) \times (-2, 2) \times (-2, 2)$

$$\frac{\partial \mathbf{Y}_m}{\partial t} = \mathbf{U}(\mathbf{Y}_m), \text{ for } m = 1, 2, \tag{39}$$

$$\mathbf{U}(\mathbf{Y}_m) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t)\delta_c(\mathbf{x} - \mathbf{Y}_m(t))d\mathbf{x}. \tag{40}$$

The CH equation on the evolving cell surface is combined as follows:

$$\frac{\partial \phi}{\partial t} + \phi V_v H + \nabla_\tau \cdot (\phi \mathbf{V}_\tau) = M\Delta_\tau \mu \tag{41}$$

$$\mu(\mathbf{s}, t) = F'(\phi(\mathbf{s}, t)) - \epsilon^2 \Delta_\tau \phi(\mathbf{s}, t). \tag{42}$$

Equations (28) and (29) constitute the Navier–Stokes equations, $p$ is the pressure and $\mathbf{u} = (u, v, w)$ is the fluid velocity. The non-dimensional parameters are the Reynolds number $Re$, Weber number $We$, and new parameter numbers $De$ and $Se$. In Eq. (30), $S(\mathbf{x}, t)$ is the time-dependent source, which is only positive around cell nuclei when a cell grows. When the cell volume is doubled, the sources are deactivated. Furthermore, $\hat{H}(t)$ is the Heaviside function, which satisfies $\hat{H}(t) = 1$ when $t \geq 0$ and $\hat{H}(t) = 0$ otherwise; and $\delta_c(\mathbf{x})$ is the smoothed Dirac delta function. Here, $t_0$ denotes the specific time at which the mass of the cell becomes double the size of its mother cell. For more details concerning the mathematical model for growth and division, we refer the reader to [42].

The initial shape of a cell is a unit sphere on the computational domain $\Omega = (-2, 2) \times (-2, 2) \times (-2, 2)$ and the initial condition for $\phi$ is a random perturbation. A mesh grid of size $128 \times 128 \times 128$ is used. The astral centers are $Y_1 = (0, -0.2, 0)$ and $Y_2 = (0, 0.2, 0)$. We set $Re = 0.5$, $We = 1$, $De = 2\text{E-}3$,$\eta = 0.05$, and $M = 4$. The simulation is run until time $T = 1.5$ with $\Delta t = 5\text{E}{-}5$. Figure 9a–c show that two point sources inside the cell cause it to grow. Figure 9d–e show that the cell is divided into two cells by a division force.

The computational times are listed below each figure. On the evolving cell surface, it can be seen that spinodal decomposition of a binary mixture is performed effectively. Furthermore, the framework presented here, which combines the immersed boundary method and our mathematical approach on evolving surfaces, is not limited to the CH equation. It can be extended to study diverse biological phenomena occurring on biological membranes.

## 5 Conclusions

In this paper, we proposed a second-order time-accurate scheme for the CH equation on an evolving surface. We derived a model of the CH equation on an evolving surface. The discretization is performed via a surface mesh, which consists of piecewise triangles. The proposed scheme is based on a Crank–Nicolson-type scheme, and is second-order accurate in time. The resulting numerical system of discrete equations is computed using a biconjugate gradient stabilized method. We performed several numerical tests, such as spinodal decomposition on translated surfaces, time and space convergence, spinodal decomposition associated with rotational motion, and spinodal decomposition under a mean curvature flow. These numerical experiments demonstrated the efficiency of our proposed algorithm. In particular, to demonstrate the extension of our proposed method to phenomena on evolving surfaces, we presented two applications: spinodal decompositions on a family of controlled surfaces and on evolving cell surfaces.

**Acknowledgements** 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 thank the reviewers for the constructive and helpful comments concerning the revision of this article.

## References

1. Cahn, J.W., Hilliard, J.E.: Free energy of a non-uniform system I. Interfacial free energy. J. Chem. Phys. **28**, 258–267 (1958)
2. Armstrong, N.J., Painter, K.J., Sherratt, J.A.: A continuum approach to modelling cell–cell adhesion. J. Theor. Biol. **243**, 98–113 (2006)
3. Wise, S., Lowengrub, J., Frieboes, H., Cristini, V.: Three-dimensional multispecies nonlinear tumor growth: I. model and numerical method. J. Theor. Biol. **253**, 524–543 (2008)
4. Kim, J.: Phase-field models for multi-component fluid flows. Commun. Comput. Phys. **12**, 613–661 (2012)
5. Li, Y., Choi, J.-I., Kim, J.: A phase-field fluid modeling and computation with interfacial profile correction term. Commun. Nonlinear Sci. Numer. Simul. **30**, 84–100 (2016)
6. Li, Y., Shin, J., Choi, Y., Kim, J.S.: Three-dimensional volume reconstruction from slice data using phase-field models. Comput. Vis. Image Und. **137**, 115–124 (2015)
7. Archer, A.J., Evans, R.: Dynamical density functional theory and its application to spinodal decomposition. J. Chem. Phys. **121**, 4246–4254 (2004)
8. Marconi, U.M.B., Tarazona, P.: Dynamic density functional theory of fluids. J. Chem. Phys. **110**, 8032–8044 (1999)
9. Yang, S.-D., Lee, H.G., Kim, J.S.: A phase-field approach for minimizing the area of triply periodic surfaces with volume constraint. Comput. Phys. Commun. **181**, 1037–1046 (2010)
10. Erlebacher, J., Aziz, M.J., Karma, A., Dimitrov, N., Sieradzki, K.: Evolution of nanoporosity in dealloying. Nature **410**, 450–453 (2001)
11. Baumgart, T., Hess, S., Webb, W.: Imaging coexisting fluid domains in biomembrane models coupling curvature and line tension. Nature **425**, 821–824 (2003)

12. Tang, P., Qiu, F., Zhang, H., Yang, Y.: Phase separation patterns for diblock copolymers on spherical surfaces: a finite volume method. Phys. Rev. E **72**, 016710 (2005)
13. Jeong, D., Kim, J.K.: Microphase separation patterns in diblock copolymers on curved surfaces using a nonlocal Cahn-Hilliard equation. Eur. Phys. J. E **38**, 1–7 (2015)
14. Bausch, A., et al.: Grain boundary scars and spherical crystallography. Science **299**, 1716–1718 (2003)
15. Lee, H.-G., Kim, J.: A simple and efficient finite difference method for the phase-field crystal equation on curved surfaces. Comput. Methods Appl. Mech. Eng. **307**, 32–43 (2016)
16. Mercker, M., Ptashnyk, M., Kuhnle, J., Hartmann, D., Weiss, M., Jager, W.: A multiscale approach to curvature modulated sorting in biological membranes. J. Theor. Biol. **301**, 67–82 (2012)
17. He, Y., Liu, Y., Tang, T.: On large time-stepping methods for the Cahn–Hilliard equation. Appl. Numer. Math. **57**, 616–628 (2007)
18. Gomez, H., Hughes, T.J.R.: Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models. J. Comput. Phys. **230**, 5310–5327 (2011)
19. Li, Y., Jeong, D., Shin, J., Kim, J.S.: A conservative numerical method for the Cahn–Hilliard equation with Dirichlet boundary conditions in complex domains. Comput. Math. Appl. **65**, 102–115 (2013)
20. Li, Y., Lee, H.G., Xia, B., Kim, J.: A compact fourth-order finite difference scheme for the three-dimensional Cahn–Hilliard equation. Comput. Phys. Commun. **200**, 108–116 (2016)
21. Song, H., Shu, C.-W.: Unconditional energy stability analysis of a second order implicit-explicit local discontinuous galerkin method for the Cahn–Hilliard equation. 1–26 (2017)
22. Du, Q., Ju, L., Tian, L.: Finite element approximation of the Cahn–Hilliard equation on surfaces. Comput. Methods Appl. Mech. Eng. **200**, 2458–2470 (2011)
23. Elliott, C.M., Ranner, T.: Evolving surface finite element method for the Cahn–Hilliard equation. Numer. Math. **129**, 483–534 (2015)
24. Rätz, A., Voigt, A.: PDE's on surfaces—A diffuse interface approach. Commun. Math. Sci. **4**, 575–590 (2006)
25. Witkowski, T., Backofena, R., Voigt, A.: The influence of membrane bound proteins on phase separation and coarsening in cell membranes. Phys. Chem. Chem. Phys. **14**, 14509–14515 (2012)
26. Bertalmio, M., Cheng, L.-T., Osher, S., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces. J. Comput. Phys. **174**, 759–780 (2001)
27. Xu, J.-J., Zhao, H.-K.: An Eulerian formulation for solving partial differential equations along a moving interface. J. Sci. Comput. **19**, 1–3 (2003)
28. Adalsteinsson, D., Sethian, J.A.: Transport and diffusion of material quantities on propagating interfaces via level set methods. J. Comput. Phys. **185**, 271–288 (2003)
29. Greer, J., Bertozzi, A.L., Saporo, G.: Fourth order partial differential equations on general geometries. J. Comput. Phys. **216**, 216–246 (2006)
30. Dziuk, G., Elliott, C.M.: An Eulerian approach to transport and diffusion on evolving implicit surfaces. Comput. Vis. Sci. **13**, 17–28 (2010)
31. Leung, S., Lowengrub, J., Zhao, H.K.: A grid based particle method for solving partial differential equations on evolving surfaces and modeling high order geometrical motion. J. Comput. Phys. **230**, 2540–2561 (2011)
32. Rong, G., Jin, M., Shuai, L., Guo, X.: Centroidal Voronoi tessellation in universal covering space of manifold surfaces. Comp. Aided Geom. Des. **28**, 475–496 (2011)
33. Sun, F., Choi, Y.-K., Wang, W., Yan, D.-M., Liu, Y., Lévy, B.: Obtuse triangle suppression in anisotropic meshes. Comp. Aided Geom. Des. **28**, 537–548 (2011)
34. Dziuk, G., Elliott, C.M.: Finite elements on evolving surfaces. IMA J. Numer. Anal. **27**, 262–292 (2007)
35. Willmore, T.J.: Riemannian Geometry. Clarendon Press, New York (1993)
36. Chen, S.-G., Wu, J.-Y.: Estimating normal vectors and curvatures by centroid weights. Comp. Aided Geom. Des. **21**, 447–458 (2004)
37. Chen, S.-G., Wu, J.: Discrete conservation laws on evolving surfaces. SIAM J. Sci. Comput. **38**, A1725–A1742 (2016)
38. Chen, S.-G., Wu, J.: Discrete conservation laws on curved surfaces. SIAM J. Sci. Comput. **36**, 719–39 (2013)
39. Li, Y., Wang, N., Kim, J.: An unconditionally energy-stable second-order time-accurate scheme for the Cahn-Hilliard equation on surfaces. Commun. Nonlinear. Sci. Numer. Simul. **53**, 213–227 (2017)
40. van der Vorst, H.A.: BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetriclinear systems. SIAM J. Sci. Stat. Comput. **13**(2), 631–644 (1992)
41. Hua, H., Li, Y., Shin, J., Song, H., Kim, J.: Effect of confinement on droplet deformation in shear flow. Int. J. Comput. Fluid D. **27**, 317–331 (2013)
42. Li, Y., Kim, J.: Three-dimensional simulations of the cell growth and cytokinesis using the immersed boundary method. Math. Biosci. **271**, 118–127 (2016)

43. Li, Y., Kim, J.: Multiphase image segmentation using a phase-field model. Comput. Math. Appl. **62**, 737–745 (2011)
44. Li, Y., Jeong, D., Choi, J.-I., Lee, S., Kim, J.: Fast local image inpainting based on the Allen–Cahn model. Digit. Signal Process. **37**, 65–74 (2015)
45. Li, Y., Kim, J.: Fast and efficient narrow volume reconstruction from scattered data. Pattern Recognit. **48**, 4057–4069 (2015)
46. Li, Y., Yun, A., Kim, J.: An immersed boundary method for simulating a single axisymmetric cell growth and division. J. Math. Biol. **65**, 653–675 (2012)