Journal of Computational Physics 334 (2017) 170–181

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A finite difference method for a conservative Allen–Cahn equation on non-flat surfaces

Junseok Kim, Darae Jeong, Seong-Deog Yang, Yongho Choi *

Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

ARTICLE INFO

Article history:
Received 31 August 2016
Received in revised form 20 November 2016
Accepted 30 December 2016
Available online 5 January 2017

Keywords:
Conservative Allen–Cahn equation
Narrow band domain
Closest point method
Space–time-dependent Lagrange multiplier

ABSTRACT

We present an efficient numerical scheme for the conservative Allen–Cahn (CAC) equation on various surfaces embedded in a narrow band domain in the three-dimensional space. We apply a quasi-Neumann boundary condition on the narrow band domain boundary using the closest point method. This boundary treatment allows us to use the standard Cartesian Laplacian operator instead of the Laplace–Beltrami operator. We apply a hybrid operator splitting method for solving the CAC equation. First, we use an explicit Euler method to solve the diffusion term. Second, we solve the nonlinear term by using a closed-form solution. Third, we apply a space–time-dependent Lagrange multiplier to conserve the total quantity. The overall scheme is explicit in time and does not need iterative steps; therefore, it is fast. A series of numerical experiments demonstrate the accuracy and efficiency of the proposed hybrid scheme.

1. Introduction

The Allen–Cahn (AC) equation is a second-order nonlinear parabolic partial differential equation, which was originally proposed by Allen and Cahn [1] to describe the phase separation in binary alloys. The classical AC equation is

$$\frac{\partial \phi}{\partial t}(\mathbf{x},t) = -M\left(\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} - \Delta\phi(\mathbf{x},t)\right). \tag{1}$$

The AC equation has been used to model many phenomena such as crystal growth [2,3], image inpainting [4,5], image segmentation [6,7], and tumor growth [8] on flat surfaces. Moreover, it has been studied on non-flat surfaces [9]. Although the conservative Allen–Cahn (CAC) equation has been solved and studied on flat surfaces [10–13], to the best of authors' knowledge, there is no research work that has attempted to solve this equation on non-flat surfaces.

Therefore, the main purpose of this study is to develop a fast and computationally efficient finite difference method for the CAC equation on non-flat surfaces in three-dimensional space. The problem for partial differential equations on the surfaces has been studied in various fields such as image processing [14,15] and biological modeling [15–17]. Therefore, solving the CAC equation on surfaces is an important issues, both in the geometrical and numerical sense. We employ a hybrid explicit numerical method, which is based on an operator splitting method and we solve the resulting discrete

* Corresponding author.
E-mail addresses: cfdkim@korea.ac.kr (J. Kim), poohyongho@korea.ac.kr (Y. Choi).
URL: http://math.korea.ac.kr/~cfdkim/ (J. Kim).

http://dx.doi.org/10.1016/j.jcp.2016.12.060
0021-9991/

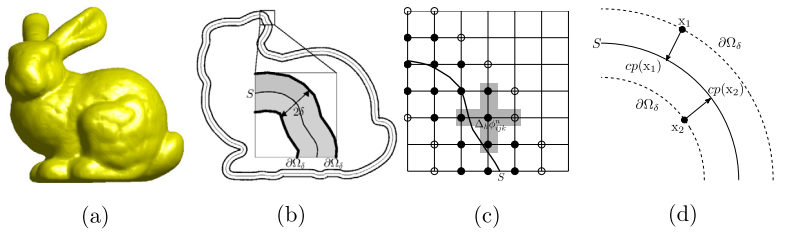


Fig. 1. (a) Bunny surface, (b) slice data of the band domain, (c) part of the bunny's ear in (b), definition of the Laplacian using the points on the shaded region, (d) closest points $cp(\mathbf{x}_1)$ and $cp(\mathbf{x}_2)$ for boundary points $\mathbf{x}_1$ and $\mathbf{x}_2$.

equations on a narrow band domain. We use the idea of the closest point method [18] to define the boundary condition. The numerical results demonstrated that the proposed algorithm is accurate and efficient.

The rest of the paper is structured as follows. In Section 2, we describe the governing equation, i.e., the CAC equation on a narrow band domain. In Section 3, we provide the numerical scheme and algorithm. We present the various numerical results in Section 4. Finally, in Section 5, we provide the conclusion.

2. The conservative Allen–Cahn equation

In this section, we describe the CAC equation [13]

$$\frac{\partial \phi}{\partial t}(\mathbf{x},t) = -\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t) + \beta\sqrt{F(\phi(\mathbf{x},t))}, \tag{2}$$

where $\phi(\mathbf{x},t)$ is the order parameter, $\sqrt{F(\phi)} = 0.5|\phi^2 - 1|$, ϵ is the thickness of the transition layer, and β is the Lagrange multiplier used for conserving the total mass. Let S be a smooth surface in $\mathbb{R}^3$ and Ω_δ be a neighborhood of S which is defined as $\Omega_\delta = \{\mathbf{y} | \mathbf{x} \in S, \mathbf{y} = \mathbf{x} + \theta\mathbf{n}(\mathbf{x}) \text{ for } |\theta| < \delta\}$ where $\mathbf{n}$ is a unit normal vector on surface S and δ is a positive constant.

We describe how to find the boundary points by using the closest point method. In the numerical simulation section, we will show a test that solves the CAC equation on the bunny. We will explain how to define the Laplacian and boundary condition by using that example. To help the reader understand the idea, we will explain how the algorithm works in two-dimensional space. Fig. 1(a) shows the surface of the bunny, and we define surface S, boundary $\partial\Omega_\delta$, band width 2δ, and band domain Ω_δ, which is denoted by the shaded region in Fig. 1(b). When we calculate $\Delta_h\phi^n_{ijk}$, the boundary values at the empty circles shown in Fig. 1(c) are needed. For each point $\mathbf{x} \in \partial\Omega_\delta$, using a trilinear interpolation, we define the closest point function $\text{cp} : \partial\Omega_\delta \to S$, which assigns the value of the closest point $\text{cp}(\mathbf{x},t)$, as shown in Fig. 1(d). Therefore, we define the boundary condition as

$$\phi(\mathbf{x},t) = \phi(\text{cp}(\mathbf{x}),t) \quad \text{on } \partial\Omega_\delta. \tag{3}$$

In the next section, we describe how to solve the CAC equation on the narrow band domain.

3. Numerical solution

In this section, we represent the numerical schemes for the CAC equation on the narrow band domain, Ω_δ. The CAC equation is discretized on the three-dimensional domain $\Omega = (a,b)\times(c,d)\times(e,f)$. The uniform spatial step size is $h = (b-a)/N_x = (d-c)/N_y = (f-e)/N_z$, where N_x, N_y, and N_z are the number of cells in the x-, y-, and z-directions, respectively. Discrete domain Ω^h is defined as $\Omega^h = \{\mathbf{x}_{ijk} = (x_i, y_j, z_k) = (a+hi, c+hj, e+hk) \big| 0 \le i \le N_x,\ 0 \le j \le N_y,\ 0 \le k \le N_z\}$, and $\Omega^h_\delta = \{\mathbf{x}_{ijk} \big| |\psi_{ijk}(\mathbf{x}_{ijk})| < \delta\}$ is the discrete narrow band domain, where ψ is a signed distance function. The narrow band domain must contain the stencil as in Fig. 1(c); therefore, we should take $\delta \ge \sqrt{3}h$. Let the boundary points be defined as $\partial\Omega^h_\delta = \{\mathbf{x}_{ijk} \big| I_{ijk}|\nabla_h I_{ijk}| \ne 0\}$, 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 $\mathbf{x}_{ijk} \in \Omega^h_\delta$; otherwise, $I_{ijk} = 1$.

Let ϕ^n_{ijk} be the approximations of $\phi(\mathbf{x}_{ijk}, n\Delta t)$, where $\Delta t = T/N_t$ is the time step, T is the final time, and N_t is the total number of time steps. In Ω^h_δ, we define a discrete L^2-norm error as $\|\phi\|_{L^2} = \sqrt{\frac{1}{\#\Omega^h_\delta}\sum_{\mathbf{x}_{ijk}\in\Omega^h_\delta}\phi^2_{ijk}}$, where $\#\Omega^h_\delta$ is the number of points on the band. We consider the discretization of the CAC Eq. (2). First, we solve the AC equation which is obtained by using an operator splitting method. We solve the diffusion term on the narrow band domain Ω^h_δ by using an explicit Euler method with the boundary condition $\phi^n_{ijk} = \phi^n(\text{cp}(\mathbf{x}_{ijk}))$ on $\partial\Omega^h_\delta$:

$$\frac{\phi^*_{ijk} - \phi^n_{ijk}}{\Delta t} = \Delta_h\phi^n_{ijk}. \tag{4}$$

Here, we use the standard Laplacian $\Delta_h\phi_{ijk} = (\phi_{i+1,jk} + \phi_{i-1,jk} + \phi_{i,j+1,k} + \phi_{i,j-1,k} + \phi_{ij,k+1} + \phi_{ij,k-1} - 6\phi_{ijk})/h^2$.

Note that we can use a fully implicit or a Crank–Nicolson-type semi-implicit numerical scheme for the diffusion equation. In this work, we use a fully explicit scheme for simplicity and accuracy. We define the closest point $\mathrm{cp}(\mathbf{x}_{ijk})$ as

$$\mathrm{cp}(\mathbf{x}_{ijk}) = \mathbf{x}_{ijk} - \frac{\nabla_h|\psi_{ijk}|}{|\nabla_h|\psi_{ijk}||}|\psi_{ijk}|, \tag{5}$$

where ψ is a signed distance function. The closet point $\mathrm{cp}(\mathbf{x}_{ijk})$ is generally not on a grid point. Therefore, we use a trilinear interpolation algorithm to obtain the values of $\phi^n(\mathrm{cp}(\mathbf{x}_{ijk}))$. Next, we solve the reaction term from the following closed-form solution on the discrete narrow band domain:

$$\phi_{ijk}^{n+1,**} = \frac{\phi_{ijk}^{*}}{\sqrt{e^{-2\Delta t/\epsilon^2} + (\phi_{ijk}^{*})^2(1-e^{-2\Delta t/\epsilon^2})}}. \tag{6}$$

Finally, we conserve the total mass. Let

$$\phi_{ijk}^{n+1} = \phi_{ijk}^{n+1,**} + \beta\sqrt{F(\phi_{ijk}^{n+1,**})}, \tag{7}$$

where

$$\beta = \sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h}(\phi_{ijk}^{0} - \phi_{ijk}^{n+1,**}) \Big/ \sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h}\sqrt{F(\phi_{ijk}^{n+1,**})}. \tag{8}$$

Here, we use the total mass conservation constraint, $\sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h}\phi_{ijk}^{0} = \sum_{\mathbf{x}_{ijk}\in\Omega_\delta^h}\phi_{ijk}^{n+1}$. Our mass-correction step is only corrected on the interfacial transition layer. However in the previous research [19], the mass was corrected by shifting all the phase-field values. Since we solve the CAC equation only on the narrow band domain and use a fully explicit scheme, the computational cost is low.

As a similar method, there is a level set-based Eulerian method for solving partial differential equations on general geometries [20]. There are a couple of differences between our proposed algorithm and the one in [20]. Whenever the stencil of the numerical schemes needs a grid point outside the narrow band, $\nabla\psi\cdot\nabla\phi = 0$ is used to prescribe the value of ϕ at that grid point. For the AC equation without the Lagrange multiplier, Eq. (1), the following modified equation is used:

$$\frac{\partial\phi}{\partial t} = -\frac{F'(\phi)}{\epsilon^2} + (\tilde{P}\nabla)\cdot(\tilde{P}\nabla\phi) + \nabla\cdot(\nabla\psi\otimes\nabla\psi\nabla\phi), \tag{9}$$

where $\tilde{P} = (I - \psi D^2\psi)^{-1}(I - \nabla\psi\otimes\nabla\psi)$ and $D^2\psi$ is the Hessian of ψ. For more details, see [20]. Compared to this level set-based method, our proposed algorithm is simpler and can naturally deal with the mass conservation property by using the discrete Lagrange multiplier, $\beta\sqrt{F(\phi_{ijk}^{n+1,**})}$, in Eq. (7).

4. Numerical experiments

We show a number of numerical experiments to demonstrate the accuracy and efficiency of the proposed numerical algorithm. In an equilibrium profile, ϕ varies from -0.9 to 0.9 over a distance of about $2\sqrt{2}\epsilon\tanh^{-1}(0.9)$. If we take ϵ as $\epsilon_m = mh/[2\sqrt{2}\tanh^{-1}(0.9)] \approx 0.24mh$, we have approximately mh transition layer [21]. Unless otherwise stated, we use the following parameters on the computational domain $\Omega = (-1.4, 1.4)\times(-1.4, 1.4)\times(-1.4, 1.4)$ in all numerical simulations: $\epsilon = \epsilon_4$, mesh size of $141\times141\times141$, spatial step size $h = 0.02$, time step size $\Delta t = 0.1h^2$, and narrow band half-width $\delta = 1.1\sqrt{3}h$.

4.1. Quantitative comparison

In this section, we quantitatively compare the numerical results of our algorithm with the analytic solutions to show the accuracy. We consider two spherical caps on a sphere of radius R (see Fig. 2). Here, O is the center of the sphere.

Let $R_1(t)$ and $R_2(t)$ be the radii of two spherical caps at time t. Let $l_1(t) = R\theta_1(t)$ and $l_2(t) = R\theta_2(t)$ be the arc lengths at time t, where $\theta_1(t) = \sin^{-1}[R_1(t)/R]$ and $\theta_2(t) = \sin^{-1}[R_2(t)/R]$. Let us assume $R_1(0) > R_2(0)$; then, we can expect that $R_1(t)$ increases until $R_2(t)$ becomes zero by the dynamics of the CAC equation. Let

$$A(t) = 2\pi R^2[1-\cos\theta_1(t)] + 2\pi R^2[1-\cos\theta_2(t)]$$

be the sum of the areas of the two spherical caps, which is constant by the conservation property, i.e., $\mathrm{d}A(t)/\mathrm{d}t = 0$. Therefore, we obtain

$$\sin\theta_1(t)\frac{\mathrm{d}\theta_1(t)}{\mathrm{d}t} + \sin\theta_2(t)\frac{\mathrm{d}\theta_2(t)}{\mathrm{d}t} = 0. \tag{10}$$

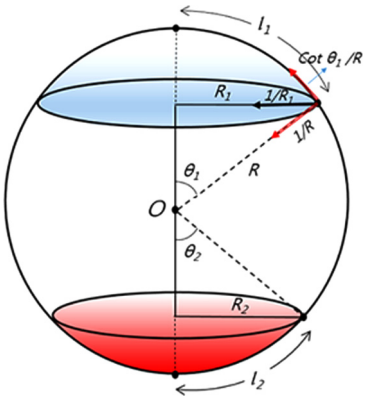
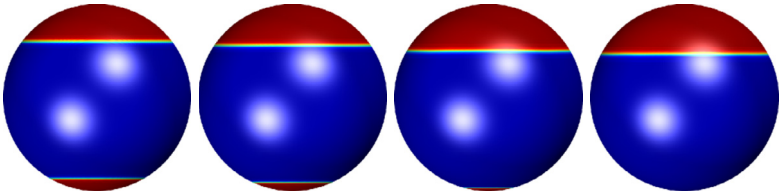


Fig. 2. Two spherical caps on a sphere of radius R.



(a) $t = 0$ (b) $t = 0.112$ (c) $t = 0.224$ (d) $t = 0.272$

Fig. 3. Temporal evolutions of two different spherical caps on the surface of the sphere with the initial condition $\phi(x, y, z, 0)$ in Eq. (15). The computational times are shown below for each figure.

We extend the evolution equations [22,23] for the case of a flat domain to the ones on a spherical domain. The curvature of the circle with the radius R_1 is $1/R_1$. The tangential component of the curvature on the sphere is $\cot\theta_1/R$. Therefore, we can write the evolution equations of the arc lengths, $l_1(t)$ and $l_2(t)$, as

$$\frac{dl_1}{dt} = -\frac{\cot\theta_1}{R} + \beta(\theta_1, \theta_2), \tag{11}$$

$$\frac{dl_2}{dt} = -\frac{\cot\theta_2}{R} + \beta(\theta_1, \theta_2), \tag{12}$$

where β is the mass conservation factor. By substituting Eqs. (11)–(12) into Eq. (10), we obtain the mass conservation factor β as follows:

$$\beta(\theta_1, \theta_2) = \frac{\cos\theta_1 + \cos\theta_2}{R(\sin\theta_1 + \sin\theta_2)}.$$

We can rewrite Eqs. (11) and (12) in terms of the radian angles:

$$\frac{d\theta_1}{dt} = -\frac{\cot\theta_1}{R^2} + \frac{\cos\theta_1 + \cos\theta_2}{R^2(\sin\theta_1 + \sin\theta_2)}, \tag{13}$$

$$\frac{d\theta_2}{dt} = -\frac{\cot\theta_2}{R^2} + \frac{\cos\theta_1 + \cos\theta_2}{R^2(\sin\theta_1 + \sin\theta_2)}. \tag{14}$$

For a quantitative comparison, we compute the radii $R_1(t)$ and $R_2(t)$ of the numerical solution with three different space and time step sizes. We set $R = 1$, and the initial condition as $R_1(0) = 0.8$ and $R_2(0) = 0.5$, i.e.,

$$\phi(x, y, z, 0) = \begin{cases} \tanh\left(\dfrac{0.8 - \sqrt{x^2 + y^2}}{\sqrt{2}\epsilon}\right), & z \geq 0 \\ \tanh\left(\dfrac{0.5 - \sqrt{x^2 + y^2}}{\sqrt{2}\epsilon}\right), & z < 0. \end{cases} \tag{15}$$

Note that Eq. (15) represents the two different spherical caps shown in Fig. 3(a). Fig. 3 shows the temporal evolution of two spherical caps on a surface. In accordance with our expection, the larger one grows and the smaller one shrinks until it disappears (Fig. 3(d)).

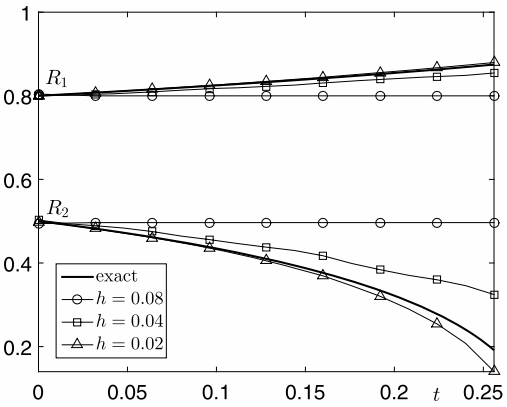
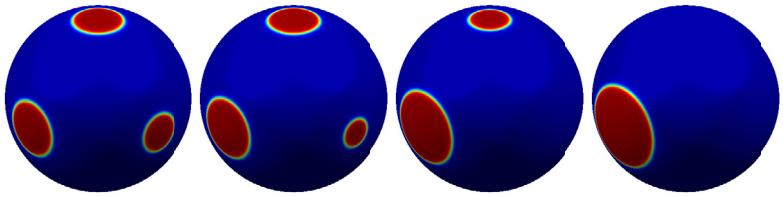


Fig. 4. Convergence of radii R_1 and R_2 with different space step size h. Here, the analytic radii are represented by the thick lines.

Table 1
Error and order of the numerical solutions by the proposed scheme with $\Delta t = 0.1h^2$ at $T = 0.176$. Note that the analytic radii are $R_1(T) = 0.84690$ and $R_2(T) = 0.35658$.

h	0.08		0.04		0.02
Radius (R_1)	0.79983		0.83522		0.84994
Error	0.04707		0.01168		0.00305
Order		2.01110		1.93827	
Radius (R_2)	0.49660		0.39890		0.34552
Error	0.14002		0.04232		0.01106
Order		1.72626		1.93601	



(a) $t = 0$ (b) $t = 0.04$ (c) $t = 0.16$ (d) $t = 2$

Fig. 5. The evolutions on the surface of the sphere with the initial condition $\phi(x, y, z)$, representing the three lens shapes in Eq. (16). The computational times are shown below for each figure.

Fig. 4 shows the temporal evolutions of the radii R_1 and R_2 with different space step size h. Here, the analytic radii, which are approximated by using a fourth-order Runge–Kutta method [24] for Eqs. (13) and (14), are represented by the thick lines. Table 1 represents the errors and orders of the numerical solutions by the proposed scheme with different space step size h and time step $\Delta t = 0.1h^2$ at $T = 0.176$.

Table 1 indicates that the numerical solutions R_1 and R_2 obtained using our scheme are almost second-order accurate in space and first-order accurate in time.

4.2. Evolution on surfaces

Next, we show the evolution of three spherical caps with different sizes on the unit sphere. In this test, the initial condition is set as follows:

$$\phi(x, y, z, 0) = \begin{cases} \tanh\left(\frac{R_1 - \sqrt{y^2+z^2}}{\sqrt{2}\epsilon}\right), & \text{if } x > 0.5, \\ \tanh\left(\frac{R_2 - \sqrt{x^2+z^2}}{\sqrt{2}\epsilon}\right), & \text{if } y > 0.5, \\ \tanh\left(\frac{R_3 - \sqrt{x^2+y^2}}{\sqrt{2}\epsilon}\right), & \text{if } z > 0.5, \end{cases} \tag{16}$$

where $R_1 = 0.3162$, $R_2 = 0.2236$, and $R_3 = 0.2774$. Fig. 5 shows the temporal evolutions. In the figure, we can see that the smaller spherical caps shrink and the largest spherical cap survives in the long run.

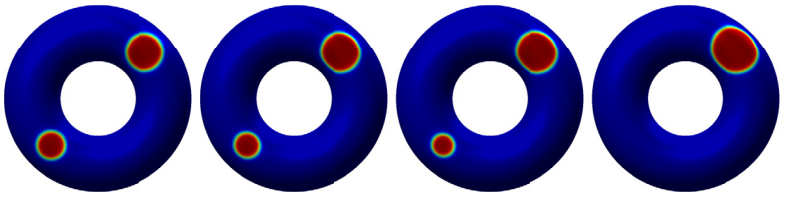


Fig. 6. The times are shown below for each figure.

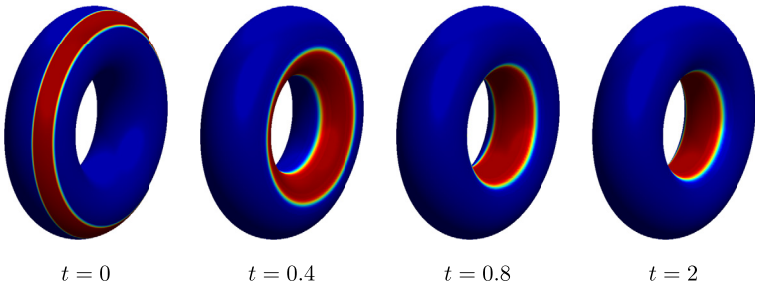


Fig. 7. The initial condition is a band on the torus. The times are shown below for each figure.

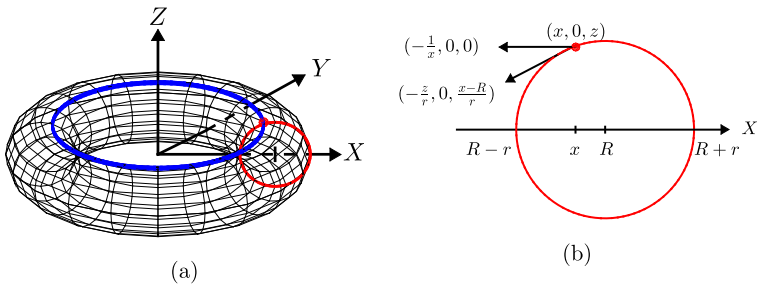


Fig. 8. (a) Shape of the torus, (b) part of (a) for the $y = 0$ plane.

We also simulate the evolutions on a torus embedded in the computational domain $\Omega = (-1.4, 1.4) \times (-1.4, 1.4) \times (-0.5, 0.5)$. The major and the minor radii of the torus are 0.7 and 0.3, respectively. The initial condition is defined as

$$\phi(x, y, z, 0) = \begin{cases} \tanh\left(\dfrac{0.2 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon}\right), & x > 0,\ y > 0,\ z > 0, \\ \tanh\left(\dfrac{0.15 - \sqrt{(x+0.5)^2 + (y+0.5)^2}}{\sqrt{2}\epsilon}\right), & x < 0,\ y < 0,\ z > 0. \end{cases} \tag{17}$$

In Fig. 6, the smaller one shrinks and the larger one grows, and then, eventually, the smaller one disappears.

Next, we consider a band on the torus. Fig. 7(a) shows the initial condition on $\Omega = (-1.4, 1.4) \times (-1.4, 1.4) \times (-0.5, 0.5)$:

$$\phi(x, y, z, 0) = \begin{cases} 1 & \text{if } 0.1 < z < 0.25, \\ -1 & \text{otherwise.} \end{cases} \tag{18}$$

Fig. 7 shows the temporal evolution of the initial band on the torus.

To understand the evolution dynamics on the torus surface, let us consider a longitude curve like that shown in Fig. 8(a). Fig. 8(b) represents a meridian curve on the xz-plane at $y = 0$. The curvature at point $(x, 0, z)$ of the longitude curve is $(-1/x, 0, 0)$. The unit tangent vector at $(x, 0, z)$ is $(-z/r, 0, (x - R)/r)$. Therefore, the tangential component of the curvature is

$$\left(-\frac{1}{x}, 0, 0\right) \cdot \left(-\frac{z}{r}, 0, \frac{x-R}{r}\right) = \frac{z}{rx}. \tag{19}$$

Fig. 9 shows the magnitude of z/rx at each point x for $R - r \le x \le R + r$, with $R = 0.7$ and $r = 0.3$. Here, $z = \sqrt{r^2 - (R - x)^2}$. Fig. 9 explains the dynamic direction of the band in Fig. 7.

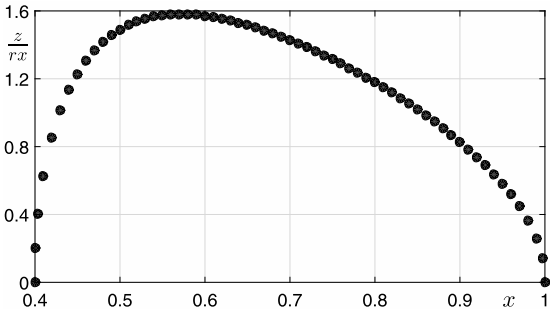


Fig. 9. Magnitude of z/rx at each point x for $R - r \le x \le R + r$, with $R = 0.7$ and $r = 0.3$. Here, $z = \sqrt{r^2 - (R - x)^2}$.

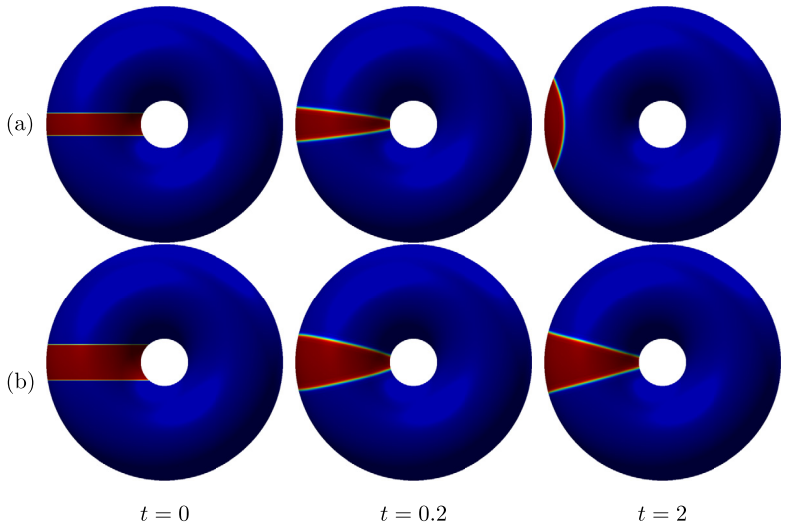


Fig. 10. Comparison of the effect of band width: (a) $b = 0.1$ and (b) $b = 0.15$.

4.3. Effect of band width on the torus

In this section, we investigate the effect of band width on the equilibrium state. For the numerical test, we only change the band width ($2b$) and fix the other parameters (i.e., 251^3 mesh and spatial step size $h = 0.0112$) on the computational domain $(-1.4, 1.4) \times (-1.4, 1.4) \times (-1.4, 1.4)$. The initial condition is defined by

$$\phi(x, y, z, 0) = \begin{cases} 1 & \text{if } -b < x < b \text{ and } y < 0, \\ -1 & \text{otherwise.} \end{cases} \tag{20}$$

When the band width is 0.2, the band is snapped and becomes a topological disc, as shown in Fig. 10(a). However, snapping does not occur in the case of a band width of 0.3, as shown in Fig. 10(b).

4.4. Phase separation on a sphere

We perform the phase separation on the surface of a unit sphere. Fig. 11 shows the time evolution of phase on the sphere. The initial conditions are $\phi(x, y, z, 0) = -0.4 + 0.1\text{rand}()$ and $\phi(x, y, z, 0) = 0.1\text{rand}(x, y, z)$ for Fig. 11(a) and Fig. 11(b), respectively, where $\text{rand}(x, y, z)$ is a uniformly distributed random number between -1 and 1.

4.5. Phase separation on the torus

We simulate the phase separation on the torus with a signed distance function defined as

$$\psi(x, y, z) = \sqrt{(C - R)^2 + z^2} - r, \tag{21}$$

where $C = \sqrt{x^2 + y^2}$, R is the distance from the center of the tube to the center of the torus, and r is the radius of the tube. We use $R = 0.7$ and $r = 0.3$. Fig. 12 shows the phase separation on the torus with the following initial conditions: $\phi(x, y, z) = -0.4 + 0.1\text{rand}(x, y, z)$ and $\phi(x, y, z) = 0.1\text{rand}(x, y, z)$.

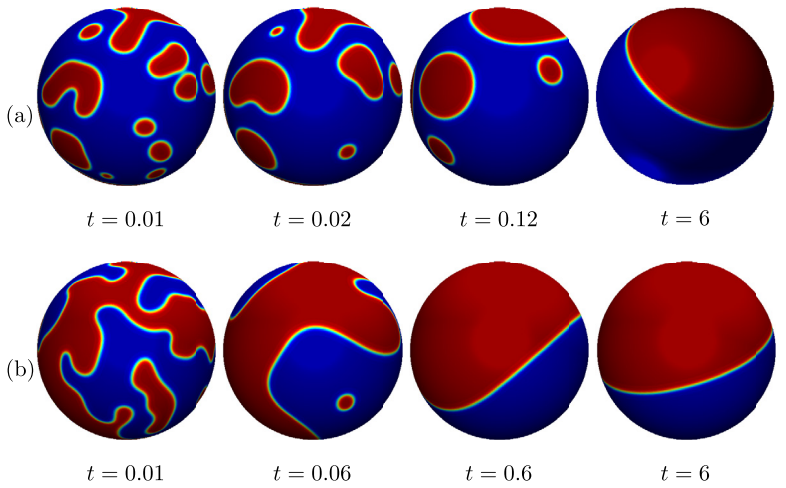


Fig. 11. Temporal evolutions on the surface of the sphere with the initial conditions $\phi(x, y, z) = -0.4 + 0.1\text{rand}(x, y, z)$ and $\phi(x, y, z) = 0.1\text{rand}(x, y, z)$. The computational times are shown below for each figure.

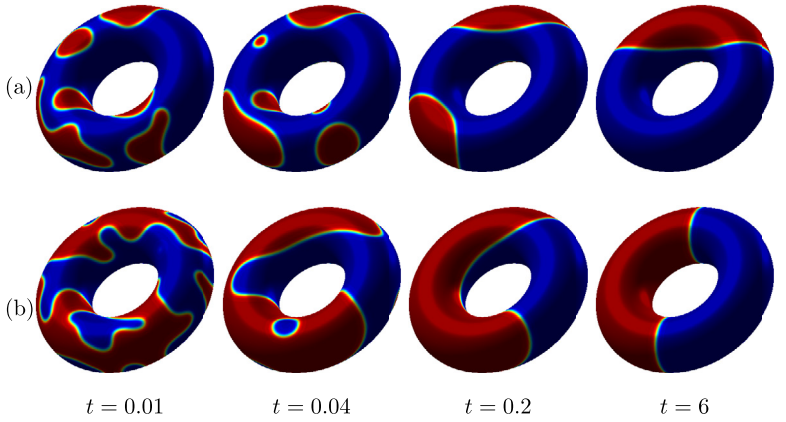


Fig. 12. The evolutions on the surface of the torus with the initial conditions $\phi(x, y, z) = -0.4 + 0.1\text{rand}(x, y, z)$ on the top row and $\phi(x, y, z) = 0.1\text{rand}(x, y, z)$ on the bottom row. The computational times are shown below for each figure.

4.6. Phase separation on a spindle

We simulate the phase separation on the surface of a spindle with different average concentrations. The spindle shape is defined by the following parametric equations: $r = \cos x$, $x(r, \theta) = x$, $y(r, \theta) = r\cos\theta$, and $z(r, \theta) = r\sin\theta$, where $|x| \leq \pi/2$ and $\theta \in [0, 2\pi)$. In this simulation, we use $h = 0.0379$ on $\Omega = (-200h, 100h) \times (-32h, 32h) \times (-32h, 32h)$. Fig. 13 shows the time evolution of the spinodal decomposition on the surface of a spindle with two different initial conditions: $\phi(x, y, z, 0) = -0.4 + 0.1\text{rand}(x, y, z)$ and $\phi(x, y, z, 0) = 0.1\text{rand}(x, y, z)$.

4.7. Phase separation on a funnel

In this section, we simulate the phase separation on the funnel shape which is defined by the following parametric equations: $x(r, \theta) = r\cos\theta$, $y(r, \theta) = r\sin\theta$, and $z(r, \theta) = \ln r$, where $r > 0$ and $\theta \in [0, 2\pi)$. We use $h = 0.0121$ on the computational domain $\Omega = (-1.2, 1.2) \times (-1.2, 1.2) \times (-2.4, 1.2)$. The initial conditions are $\phi(x, y, z, 0) = -0.4 + 0.1\text{rand}(x, y, z)$ and $\phi(x, y, z, 0) = 0.1\text{rand}(x, y, z)$. Figs. 14(a) and (b) show the results of the temporal evolutions on the funnel shape with an average concentration of -0.4 and 0, respectively.

4.8. Phase separation on a bunny

In this section, we perform the phase separation on the surface of a bunny with different average concentrations. To perform the numerical simulation, we use $h = 0.5$ on $\Omega = (0, 68.5)^3$. Fig. 15 shows the temporal evolution of the spinodal decomposition on the surface of the bunny with two different initial conditions: $\phi(x, y, z, 0) = -0.4 + 0.1\text{rand}(x, y, z)$ and $\phi(x, y, z, 0) = 0.1\text{rand}(x, y, z)$.

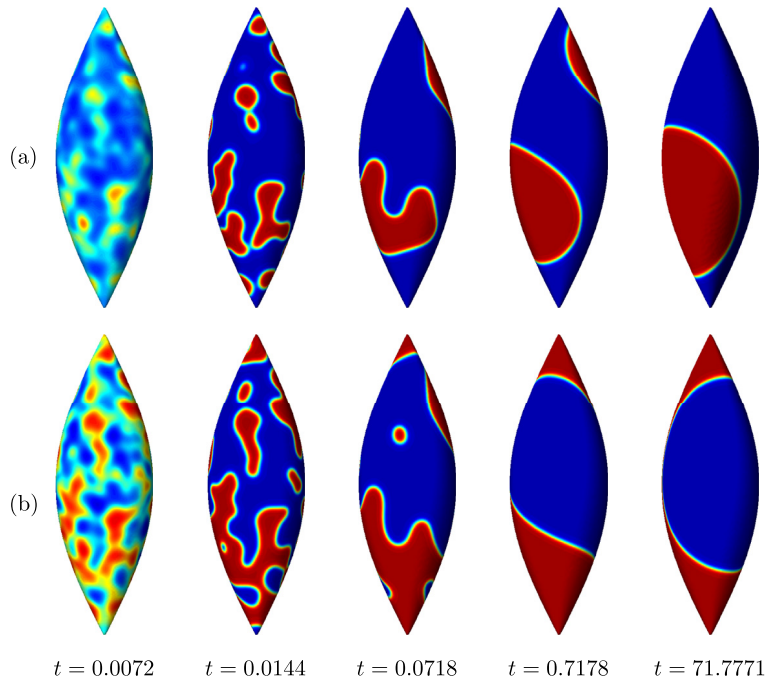


Fig. 13. Temporal evolution of the phase separation on the spindle shape. The initial conditions are (a) $\phi(x, y, z) = -0.4 + 0.1\text{rand}(x, y, z)$ and (b) $\phi(x, y, z) = 0.1\text{rand}(x, y, z)$.

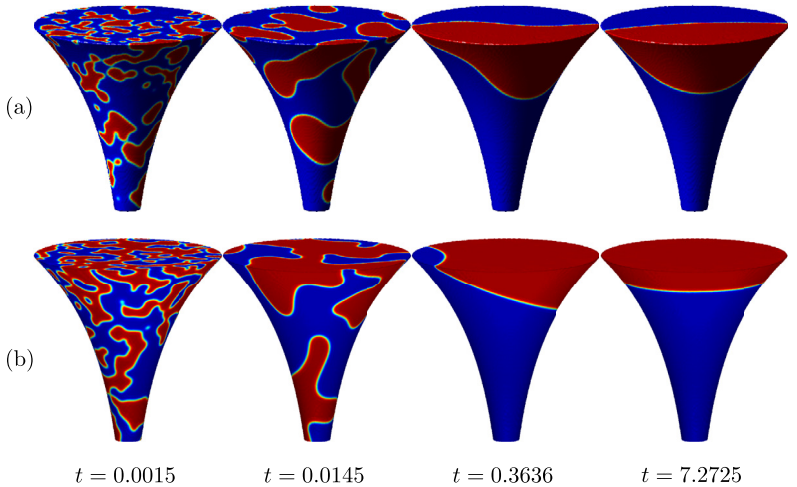


Fig. 14. Temporal evolution of the phase separation on the funnel shape. The initial conditions are (a) $\phi(x, y, z) = -0.4 + 0.1\text{rand}(x, y, z)$ and (b) $\phi(x, y, z) = 0.1\text{rand}(x, y, z)$.

4.9. Comparison study with an implicit scheme

In this section, we compare the computational costs for our proposed algorithm and an implicit algorithm in order to demonstrate the speed and efficiency of the proposed algorithm. For the comparison study, we consider a well-known nonlinear convex splitting scheme [25], which is unconditionally gradient stable. We take the following two steps to solve the CAC Eq. (2):

$$\frac{\phi_{ijk}^{n+1,*} - \phi_{ijk}^{n}}{\Delta t} = \frac{\phi_{ijk}^{n} - (\phi_{ijk}^{n+1,*})^3}{\epsilon^2} + \Delta_h \phi_{ijk}^{n+1,*}, \tag{22}$$

$$\phi_{ijk}^{n+1} = \phi_{ijk}^{n+1,*} + \beta\sqrt{F(\phi_{ijk}^{n+1,*})}, \tag{23}$$

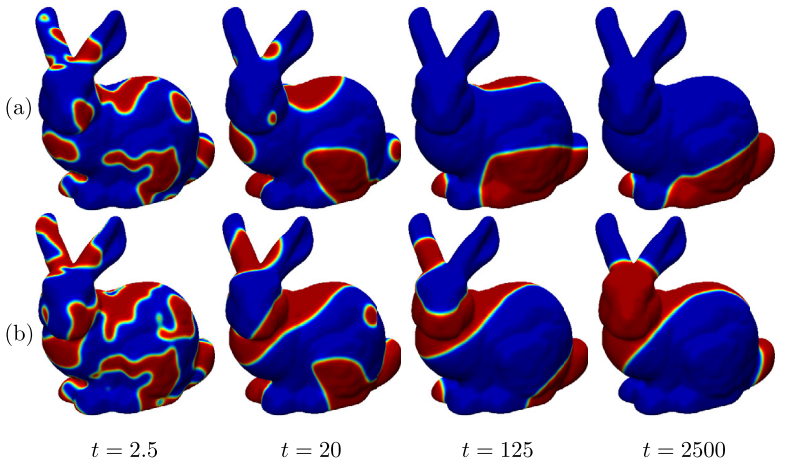


Fig. 15. Morphological evolutions on the surface of the bunny with the initial conditions (a) $\phi(x, y, z, 0) = -0.4 + 0.1\text{rand}(x, y, z)$ and (b) $\phi(x, y, z, 0) = 0.1\text{rand}(x, y, z)$. The computational times are shown below for each figure.

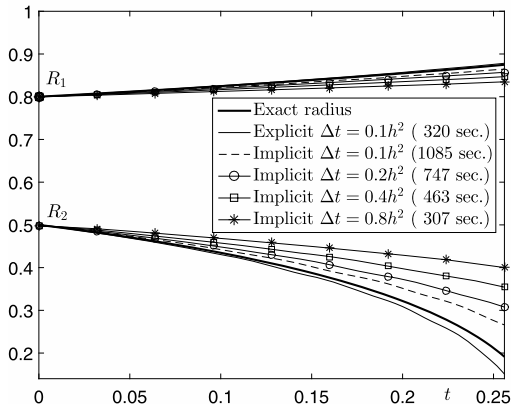


Fig. 16. Temporal evolutions of two radii by the explicit and implicit schemes. The CPU times are shown in parentheses.

where $\beta = \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} (\phi_{ijk}^0 - \phi_{ijk}^{n+1,*}) \Big/ \sum_{\mathbf{x}_{ijk} \in \Omega_\delta^h} \sqrt{F(\phi_{ijk}^{n+1,*})}$. Here, we use the Gauss–Seidel iterative method to solve Eq. (22). Let $\phi_{ijk}^{n+1,*,s}$ and $\phi_{ijk}^{n+1,*,s+1}$ be the approximation of $\phi_{ijk}^{n+1,*}$ before and after one Gauss–Seidel iteration, respectively. Since $(\phi_{ijk}^{n+1,*})^3$ in Eq. (22) is nonlinear, we linearize it as follows:

$$(\phi_{ijk}^{n+1,*,s+1})^3 = 3(\phi_{ijk}^{n+1,*,s})^2(\phi_{ijk}^{n+1,*,s+1}) - 2(\phi_{ijk}^{n+1,*,s})^3.$$

Then, we rearrange Eq. (22) as the Gauss–Seidel iterative form:

$$\left(\frac{1}{\Delta t} + \frac{3(\phi_{ijk}^{n+1,*,s})^2}{\epsilon^2} + \frac{6}{h^2}\right)\phi_{ijk}^{n+1,*,s+1} = \left(\frac{1}{\Delta t} + \frac{1}{\epsilon^2}\right)\phi_{ijk}^{n} + \frac{2(\phi_{ijk}^{n+1,*,s})^3}{\epsilon^2} + \frac{\phi_{i-1,jk}^{n+1,*,s+1} + \phi_{i+1,jk}^{n+1,*,s} + \phi_{i,j-1,k}^{n+1,*,s+1} + \phi_{i,j+1,k}^{n+1,*,s} + \phi_{ij,k-1}^{n+1,*,s+1} + \phi_{ij,k+1}^{n+1,*,s}}{h^2}.$$

In this study, we update the new solution ϕ_{ijk}^{n+1} when the l_2-norm of the consecutive error, $\|\phi_{ijk}^{n+1,*,s+1} - \phi_{ijk}^{n+1,*,s}\|_2$, is less than the tolerance value, $tol = 10^{-6}$. Now, we compare the analytic solution with the numerical results by the proposed scheme and the implicit convex splitting scheme. We use the same initial condition as in Section 4.1. Fig. 16 represents the time evolution of two radii with the explicit and implicit schemes. The analytic solution is shown with the solid line. Moreover, we put the computational times in the legend to show the efficiency of the explicit scheme.

It is well known that one of the drawbacks of an explicit scheme is the severe time step restriction for stability. For example, a fourth-order nonlinear equation such as the Cahn–Hilliard equation [25] has a $\Delta t = O(h^4)$ time step constraint for the explicit scheme. However, it turns out that a $\Delta t \le h^2/6$ time step constraint in three-dimensional space for the CAC equation is not a severe constraint if we consider the accuracy of the numerical solutions. As shown in Fig. 16, it only takes

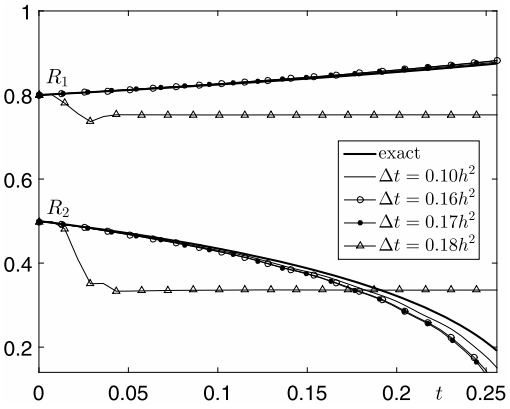


Fig. 17. Stability of the numerical solutions by the explicit scheme with different time step sizes.

320 s of CPU time for the explicit scheme, whereas it takes 1085 s of CPU time for the implicit scheme with $\Delta t = 0.1h^2$. If we compare the results from the implicit scheme ($\Delta t = 0.8h^2$) and the explicit scheme ($\Delta t = 0.1h^2$), we can see that the result from the latter is more accurate than from the former with a comparable CPU time.

In our algorithm, we have the time step restriction as $\Delta t \leq h^2/6 \approx 0.1667h^2$ when we solve the diffusion Eq. (4). Fig. 17 shows the stability of the numerical solutions by the proposed scheme with different time step sizes. To show the stability, we use four different time step sizes: $\Delta t = 0.1h^2$, $0.16h^2$, $0.17h^2$, and $0.18h^2$. As shown in Fig. 17, we can see the stable solutions when $\Delta t \leq 0.17h^2$. One of the reasons why $\Delta t = 0.17h^2$ gives a stable solution is that the second step of the proposed algorithm damps an unstable solution and stabilize the solution.

5. Conclusions

We performed numerical simulations on various three-dimensional surface shapes with the CAC equation. For improving the efficiency and lowering the computational cost, we employed a narrow band domain and solved the equation on that domain. On the narrow band domain boundary, we applied a quasi-Neumann boundary condition, which is defined by using the closest point method. To solve the CAC equation, we used the hybrid operator splitting method, which consists of three steps. First, we solved the diffusion term by using an explicit Euler method and then updated the nonlinear term by using a closed-form solution. Lastly, to conserve the total concentration, we applied the space–time-dependent Lagrange multiplier. The overall scheme is fully explicit in time and does not need iterative steps; therefore, it is fast. A series of numerical experiments with various initial conditions on the sphere, torus, spindle, funnel, and bunny shapes demonstrated the effectiveness and accuracy of the proposed hybrid scheme. In this paper, we used uniform meshes. In a future research, we plan to consider the body-fitted coordinate on a non-uniform mesh and triangular surface mesh for possible improvements.

Acknowledgements

The first author (J.S. Kim) was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2014R1A2A2A01003683). The authors greatly appreciate the reviewers for their constructive comments and suggestions, which helped us to improve the quality of this paper.

References

[1] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metall. 27 (1979) 1085–1095.
[2] L.Q. Chen, Phase-field models for microstructure evolution, Annu. Rev. Mater. Res. 32 (2002) 113–140.
[3] M. Cheng, J.A. Warren, An efficient algorithm for solving the phase field crystal model, J. Comput. Phys. 227 (2008) 6241–6248.
[4] Z. Feng, J. Yin, J. Zhou, Inpainting algorithm for jacquared image based on phase-field model, IEEE Intell. Syst. Knowl. Eng. ISKE (2008) 1203–1207.
[5] Y. Li, D. Jeong, J. Choi, S. Lee, J. Kim, Fast local image inpainting based on the Allen–Cahn model, Digit. Signal Process. 37 (2015) 65–74.
[6] M. Beneš, V. Chalupecký, K. Mikula, Geometrical image segmentation by the Allen–Cahn equation, Appl. Numer. Math. 51 (2004) 187–205.
[7] D.A. Kay, A. Tomasi, Color image segmentation by the vector-valued Allen–Cahn phase-field model: a multigrid solution, IEEE Trans. Image Process. 18 (2009) 2330–2339.
[8] X. Wu, G.J. Zwieten, K.G. Zee, Stabilized second-order convex splitting schemes for Cahn–Hilliard models with application to diffuse-interface tumor-growth models, Int. J. Numer. Methods Biomed. Eng. 30 (2014) 180–203.
[9] Y. Choi, D. Jeong, S. Lee, M. Yoo, J. Kim, Motion by mean curvature of curves on surfaces using the Allen–Cahn equation, Int. J. Eng. Sci. 97 (2015) 126–132.
[10] M.J. Ward, Metastable bubble solutions for the Allen–Cahn equation with mass conservation, SIAM J. Appl. Math. 56 (1996) 1247–1279.
[11] M. Kowalczyk, On the existence and Morse index of solutions to the Allen–Cahn equation in two dimensions, Ann. Mat. Pura Appl. 184 (2005) 17–52.
[12] J. Shen, X. Yang, A phase-field model and its numerical approximation for two-phase incompressible flows with different densities and viscosities, SIAM J. Sci. Comput. 32 (2010) 1159–1179.

[13] J. Kim, S. Lee, Y. Choi, A conservative Allen–Cahn equation with a space–time dependent Lagrange multiplier, Int. J. Eng. Sci. 84 (2014) 11–17.
[14] S. Haker, S. Angenent, A.R. Tannenbaum, R. Kikinis, Conformal 3D visualization for virtual colonoscopy, in: Med. Imag. 2000 Inter. Soc. Opt. Phot., 2000, pp. 154–164.
[15] E. Orlandini, D. Marenduzzo, A.B. Goryachev, Domain formation on curved membranes: phase separation or Turing patterns?, Soft Matter 9 (2013) 9311–9318.
[16] M. Sun, X. Yan, R.J. Sclabassi, Solving partial differential equations in real-time using artificial neural network signal processing as an alternative to finite-element analysis, in: IEEE Neur. Net. Signal Process. Proceed., 2003.
[17] S. Angenent, S. Haker, A. Tannenbaum, R. Kikinis, On the Laplace–Beltrami operator and brain surface flattening, IEEE Trans. Med. Imaging 18 (1999) 700–711.
[18] S.J. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, J. Comput. Phys. 227 (2008) 1943–1961.
[19] H.G. Lee, J. Kim, A simple and efficient finite difference method for the phase-field crystal equation on curved surfaces, Comput. Methods Appl. Math. 307 (2016) 32–43.
[20] J.B. Greer, An improvement of a recent Eulerian method for solving PDEs on general geometries, J. Sci. Comput. 29 (2006) 321–352.
[21] J.W. Choi, H.G. Lee, D. Jeong, J. Kim, An unconditionally gradient stable numerical method for solving the Allen–Cahn equation, Physica A 388 (2009) 1791–1803.
[22] H. Garcke, B. Nestler, B. Stinner, F. Wendler, Allen–Cahn systems with volume constraints, Math. Models Methods Appl. Sci. 18 (2008) 1347–1381.
[23] J. Rubinstein, P. Sternberg, Nonlocal reaction–diffusion equations and nucleation, IMA J. Appl. Math. 48 (1992) 249–264.
[24] R.L. Burden, J.D. Faires, Numerical Analysis, PWS, Boston, 1993.
[25] D.J. Eyre, Unconditionally gradient stable time marching the Cahn–Hilliard equation, in: Computational and Mathematical Models of Microstructural Evolution, San Francisco, CA, 1998, Mater. Res. Soc. Sympos. Proc. 529 (1998) 39–46.