Journal of Computational Science 77 (2024) 102252

Contents lists available at ScienceDirect

Journal of Computational Science

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

The Allen–Cahn equation with a space-dependent mobility and a source term for general motion by mean curvature

Junxiang Yang [a], Seungyoon Kang [b], Soobin Kwak [b], Junseok Kim [b,*]

[a] School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macao Special Administrative Region of China
[b] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

ARTICLE INFO

Keywords:
Space-dependent mobility
Finite difference method
Operator splitting method

ABSTRACT

We propose the Allen–Cahn (AC) equation with a space-dependent mobility and a source term for general motion by mean curvature. Using the space-dependent mobility, we can control the temporal evolution dynamics. Furthermore, by using the source term, we can control the growth and shrinkage of the interfaces. To efficiently solve the governing equation, we use an operator splitting method that splits the main equation into the modified AC equation and the source term equation. The modified AC model is numerically computed using a fully explicit Euler method, and the source term equation is solved analytically. The overall numerical schemes preserve the maximum principle if the time step size satisfies a certain condition. To show the performance of the proposed mathematical model and its corresponding numerical scheme, we conduct several computational experiments. The numerical results confirm the efficiency and robust performance of the proposed model and its numerical algorithm, rendering the proposed model as a versatile tool for a wide range of applications.

1. Introduction

In this paper, we propose the Allen–Cahn (AC) equation with a space-dependent mobility and a source term for general motion by mean curvature:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = M(\mathbf{x})\left(-\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t) - \lambda|\nabla\phi(\mathbf{x},t)|\right), \ \mathbf{x} \in \Omega, \ t > 0, \tag{1}$$

where $\phi(\mathbf{x},t)$ is a phase-field at space point $\mathbf{x}$ and time t, Ω is the domain, $M(\mathbf{x})$ is the space-dependent mobility, $F(\phi) = 0.25(\phi^2-1)^2$, ϵ is the interfacial transition parameter, and λ is a growth or decay parameter. In this study, we use the homogeneous Neumann boundary condition for simplicity. We can also use Dirichlet and periodic boundary conditions without any difficulties. If $M(\mathbf{x}) = M_0$ is a constant and $\lambda = 0$, then Eq. (1) reduces to the classical AC equation as follows:

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

which models the dynamics of anti-phase domain coarsening in crystalline solids [1,2]. The AC model can be derived from the following functional:

$$\mathcal{E}(\phi) = \int_\Omega \left(\frac{F(\phi(\mathbf{x},t))}{\epsilon^2} + \frac{1}{2}|\nabla\phi(\mathbf{x},t)|^2\right) d\mathbf{x}. \tag{3}$$

The source term $\lambda|\nabla\phi(\mathbf{x},t)|$ appears as a ice melting term [3]. Wang et al. demonstrated the conditions for the interface between ice and liquid, i.e. the zero level set of the phase-field to move toward the area of low temperature. $F(\phi(\mathbf{x},t)) = 0.5\epsilon^2|\nabla\phi(\mathbf{x},t)|^2$ satisfies the desired condition. Next, the space dependent mobility $M(\mathbf{x})$ plays an important role in the governing equation. Throughout the space dependent mobility, we can significantly control the diffusion across different regions in the computational domain. The mobility can lower the long-range diffusion in particular regions. However, up to authors' knowledge, there has been no reported study concerned with the space dependent mobility in AC equation. Other studies using a mobility term tend to use concentration dependent, periodic and constant mobility terms [4–7]. Therefore, the use of the space dependent mobility and the source term is the main strength of our study.

Furthermore, it is well known that the conventional AC equation is closely related to the evolution driven by the motion by mean curvature, which describes how interfaces evolve over time to minimize their curvature [8].

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

https://doi.org/10.1016/j.jocs.2024.102252
Received 1 December 2023; Received in revised form 26 January 2024; Accepted 26 February 2024
Available online 28 February 2024

The AC equation has been used for modeling many important applications. For instance, Qiao and Zhang [9] proposed a novel edge detection method of the grayscale image using the AC equation and a nonlocal edge detection operator. Wang et al. [10] proposed a linear energy stable and maximum principle preserving semi-implicit numerical method for the AC equation and proved its properties. Hötzer et al. [11] measured the performance of PACE3D using the sweep for the phase-field equation based on the AC equation. Feng et al. [12] proposed a new linear second-order finite difference method (FDM) for the AC equations using a modified Leap-Frog FDM. Zhang et al. [13] proposed and analyzed a family of up to third-order maximum-principle-preserving schemes for solving the AC equation using the stabilization technique and fixed-point-preserving improvement of the integrating factor method. Sitompul et al. [14] developed an ordered active parameter tracking method for simulations of multiple phases based of the AC equation. Hou et al. [15] proposed a novel second-order maximum-principle preserving FDM for AC equations with periodic boundary conditions. Tan and Zhang [16] presented a new FDM for solving the AC equations which preserve the discrete maximum principle and energy stability by combining the second-order FDM and the Crank–Nicolson (CN) scheme. Bo et al. [17] studied the discrete maximum principle and energy stability of the compact FDM for the 2D AC equation. Wang et al. [18] studied the Hurst exponent $H(2)$ during the phase separation evolution of the conservative AC equation. Gao et al. [19] proposed the second-order FDM and the Runge–Kutta method for numerically solving the nonlocal AC equation. Yang et al. [20] developed fourth-order unconditionally maximum-principle preserving finite element methods for both the local and nonlocal AC equations.

Phase field models with constant and nonconstant mobility is popular subject that is gathering attention. Shen et al. [4] studied the AC equation with degenerating mobility. Stability analysis and time step restrictions for various numerical methods are introduced in the study. The stochastic AC equation with mobility term is introduced by Bertini et al. [5], where the authors proved the existence and uniqueness of the solution. Morfe [6] studied the sharp interface limit for the AC equation with periodic mobility to prove that the large-scale behavior of interfaces are determined by mobility and mean curvature flow. While Morfe studied the AC equation, sharp interface limit for the Cahn–Hilliard equation mobility is studied by Lee et al. [7]. See [21–24] for studies concerning Cahn–Hilliard equation with the mobility term.

As described in the references above, it is very important for the numerical schemes of the AC equation to have the maximum principle because it is one of the key properties of the equation. In this study, we propose the AC equation with a space-dependent mobility and a source term for general motion by mean curvature. Furthermore, we present a numerical method that preserves the maximum principle for this modified AC equation.

The structure of the paper is outlined as follows. In Section 2, numerical solution algorithm is presented. In Section 3, computational experiments are given. In Section 4, conclusions are drawn.

2. Numerical solution algorithm

To numerically solve Eq. (1), we use the operator splitting method. We first rewrite Eq. (1) as an equivalent form:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = M(\mathbf{x})\left(-\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t) - \frac{\lambda}{\sqrt{2}\epsilon}(1-\phi^2(\mathbf{x},t))\right), \tag{4}$$

where we have used $F(\phi(\mathbf{x},t)) = 0.5\epsilon^2|\nabla\phi(\mathbf{x},t)|^2$, which holds when the interfacial transition layer is locally equilibrated. The term $\lambda|\nabla\phi(\mathbf{x},t)|$ is a source term which effectively moves the interface by adding or subtracting concentration. Let us consider an one-dimensional case with constant mobility and without the source term for simplicity of exposition, i.e.,

$$\frac{\partial \phi(x,t)}{\partial t} = -\frac{F'(\phi(x,t))}{\epsilon^2} + \phi_{xx}(x,t). \tag{5}$$

If Eq. (5) is in equilibrium state, then

$$\frac{F'(\phi(x,t))}{\epsilon^2} = \phi_{xx}(x,t). \tag{6}$$

By multiply $\phi_x(x,t)$ to both the terms in Eq. (6) and taking indefinite integration, we have

$$\int \phi_x(x,t)\frac{F'(\phi(x,t))}{\epsilon^2}dx = \int \phi_x(x,t)\phi_{xx}(x,t)dx. \tag{7}$$

$$\Rightarrow \frac{F(\phi(x,t))}{\epsilon^2} = \frac{1}{2}(\phi_x(x,t))^2 + C, \tag{8}$$

where C is some constant. If we assume that $\phi(x,t) = \pm 1$ as $x \to \infty$, we have $C = 0$. Therefore, we have $F(\phi(x,t)) = 0.5\epsilon^2(\phi_x(x,t))^2$. Now, using this condition, we can rewrite the source term in Eq. (1) as an equivalent form:

$$\lambda|\nabla\phi(\mathbf{x},t)| = \lambda\sqrt{\frac{F(\phi(\mathbf{x},t))}{0.5\epsilon^2}} = \lambda\sqrt{\frac{0.25(\phi^2(\mathbf{x},t)-1)^2}{0.5\epsilon^2}} = \frac{\lambda}{\sqrt{2}\epsilon}(1-\phi^2(\mathbf{x},t)).$$

The operator splitting method is one of the standard method for solving original and modified AC equations [25–29]. We formally split Eq. (4) into two equations as follows:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = M(\mathbf{x})\left(-\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t)\right), \tag{9}$$

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = -\frac{\lambda M(\mathbf{x})}{\sqrt{2}\epsilon}(1-\phi^2(\mathbf{x},t)). \tag{10}$$

2.1. Numerical scheme

Let $\Omega = (L_x, R_x) \times (L_y, R_y)$ be a two-dimensional domain and $\Omega_h = \{(x_i, y_j)|x_i = L_x + (i-0.5)h,\ y_j = L_y + (j-0.5)h,\ 1 \le i \le N_x,\ 1 \le j \le N_y\}$ be its discrete domain, where N_x and N_y are integers and $h = (R_x - L_x)/N_x$ is the grid size. For simplicity of notation, let $M_{ij} = M(x_i, y_j)$ and $\phi^n_{ij} = \phi(x_i, y_j, n\Delta t)$, where Δt is the time step. Let us define the maximum norm as follows:

$$\|\phi^n\|_\infty = \max_{1\le i\le N_x,\ 1\le j\le N_y} |\phi^n_{ij}|. \tag{11}$$

We solve Eq. (9) using a finite difference method with a fully explicit Euler method:

$$\frac{\phi^*_{ij} - \phi^n_{ij}}{\Delta t} = M_{ij}\left(-\frac{F'(\phi^n_{ij})}{\epsilon^2} + \Delta_d\phi^n_{ij}\right),\ \text{for } 1 \le i \le N_x,\ 1 \le j \le N_y, \tag{12}$$

where $\Delta_d\phi_{ij} = (\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{ij})/h^2$. The zero Neumann boundary condition is implemented as follows:

$$\phi^n_{0j} = \phi^n_{1j},\ \phi^n_{N_x+1,j} = \phi^n_{N_xj},\ \text{for}\ \ j = 1,\dots,N_y,$$
$$\phi^n_{i0} = \phi^n_{i1},\ \phi^n_{i,N_y+1} = \phi^n_{iN_y},\ \text{for}\ \ i = 1,\dots,N_x.$$

If we rewrite Eq. (12), then we have

$$\phi^*_{ij} = \phi^n_{ij} + \Delta t M_{ij}\left(-\frac{F'(\phi^n_{ij})}{\epsilon^2} + \Delta_d\phi^n_{ij}\right),\ \text{for } 1 \le i \le N_x,\ 1 \le j \le N_y. \tag{13}$$

Next, because Eq. (10) is an ordinary differential equation with respect to time variable t, we can analytically solve it using separation of variables as follows:

$$\phi^{n+1}_{ij} = \frac{(1+\phi^*_{ij})e^{A_{ij}} - 1 + \phi^*_{ij}}{(1+\phi^*_{ij})e^{A_{ij}} + 1 - \phi^*_{ij}},\ \text{for } 1 \le i \le N_x,\ 1 \le j \le N_y, \tag{14}$$

where $A_{ij} = -\sqrt{2}\lambda M_{ij}\Delta t/\epsilon$. Therefore, the computational scheme for the two-dimensional modified AC equation consists of Eqs. (13) and (14).

The three-dimensional modified AC equation is solved in a similar manner. Let $\Omega = (L_x, R_x) \times (L_y, R_y) \times (L_z, R_z)$ be a three-dimensional domain and $\Omega_h = \{(x_i, y_j, z_k)|x_i = L_x+(i-0.5)h,\ y_j = L_y+(j-0.5)h,\ z_k = L_z + (k-0.5)h,\ 1 \le i \le N_x,\ 1 \le j \le N_y,\ 1 \le k \le N_z\}$ be its discrete domain, where N_x, N_y and N_z are integers representing the number of grid for each axis and $h = (R_x - L_x)/N_x$ is the grid size. For simplicity of notation, let $M_{ijk} = M(x_i, y_j, z_k)$ and $\phi^n_{ijk} = \phi(x_i, y_j, z_k, n\Delta t)$. Let us define the maximum norm as follows:

$$\|\phi^n\|_\infty = \max_{\substack{1\le i\le N_x,\\ 1\le j\le N_y\\ 1\le k\le N_z}} |\phi^n_{ijk}|. \tag{15}$$

We solve Eq. (9) using a finite difference method with a fully explicit Euler method:

$$\frac{\phi^*_{ijk} - \phi^n_{ijk}}{\Delta t} = M_{ijk}\left(-\frac{F'(\phi^n_{ijk})}{\epsilon^2} + \Delta_d \phi^n_{ijk}\right), \tag{16}$$
$$\text{for } 1 \le i \le N_x,\ 1 \le j \le N_y,\ 1 \le k \le N_z,$$

where $\Delta_d \phi_{ijk} = (\phi_{i+1,jk} + \phi_{i-1,jk} + \phi_{i,j+1,k} + \phi_{i,j-1,k} + \phi_{ij,k+1} + \phi_{ij,k-1} - 6\phi_{ijk})/h^2$. The zero Neumann boundary condition is implemented as follows:

$$\phi^n_{0jk} = \phi^n_{1jk},\ \phi^n_{N_x+1,jk} = \phi^n_{N_x jk},\ \text{for }\ j = 1,\dots,N_y,\ k = 1,\dots,N_z,$$
$$\phi^n_{i0k} = \phi^n_{i1k},\ \phi^n_{i,N_y+1,k} = \phi^n_{iN_y k},\ \text{for }\ i = 1,\dots,N_x,\ k = 1,\dots,N_z,$$
$$\phi^n_{ij0} = \phi^n_{ij1},\ \phi^n_{ij,N_z+1} = \phi^n_{ijN_z},\ \text{for }\ i = 1,\dots,N_x,\ j = 1,\dots,N_y.$$

If we rewrite Eq. (16), then we have

$$\phi^*_{ijk} = \phi^n_{ijk} + \Delta t M_{ijk}\left(-\frac{F'(\phi^n_{ijk})}{\epsilon^2} + \Delta_d \phi^n_{ijk}\right), \tag{17}$$
$$\text{for } i = 1,\dots,N_x,\ j = 1,\dots,\le N_y,\ k = 1,\dots,N_z.$$

Next, because Eq. (10) is an ordinary differential equation with respect to time variable t, we can analytically solve it using separation of variables as follows:

$$\phi^{n+1}_{ijk} = \frac{(1+\phi^*_{ijk})e^{A_{ijk}} - 1 + \phi^*_{ijk}}{(1+\phi^*_{ijk})e^{A_{ijk}} + 1 - \phi^*_{ijk}}, \tag{18}$$
$$\text{for } 1 \le i \le N_x,\ 1 \le j \le N_y,\ 1 \le k \le N_z,$$

where $A_{ijk} = -\sqrt{2}\lambda M_{ijk}\Delta t/\epsilon$.

Therefore, the numerical solution algorithm for the modified AC equation consists of Eqs. (17) and (18).

2.2. Stability analysis and maximum principle

We can rewrite Eq. (16) as follows:

$$\frac{\phi^*_{ij} - \phi^n_{ij}}{M_{ij}\Delta t} = -\frac{F'(\phi^n_{ij})}{\epsilon^2} + \Delta_d \phi^n_{ij},\ \text{for } 1 \le i \le N_x,\ 1 \le j \le N_y, \tag{19}$$

where we assume $M_{ij} > 0$. If $M_{ij} = 0$, then Eq. (16) becomes

$$\phi^*_{ij} = \phi^n_{ij}, \tag{20}$$

which is valid for any time step. Because the finite difference scheme used in Eq. (19) is the fully explicit Euler method, there is a time step constraint [29] for stability, i.e.,

$$M_{ij}\Delta t \le \frac{\epsilon^2 h^2}{2h^2 + 2N\epsilon^2}, \tag{21}$$

where N is the dimension of the governing equation. In this 2D case, $N = 2$. Let $M_{max} = \max_{ij} M_{ij}$ and $M_{max} = \max_{ijk} M_{ijk}$ for the two- and three-dimensional modified AC equations, respectively. Then, we have

$$\Delta t \le \frac{\epsilon^2 h^2}{M_{max}(2h^2 + 2N\epsilon^2)}, \tag{22}$$

The continuous AC equation satisfies the maximum bound principle [30–32]. If the initial value and boundary conditions are bounded by 1, then the entire solution is also bounded by 1. Therefore, the maximum principle for both continuous and discrete AC equation is verified.

Table 1
Errors and convergence rates for the spatial discretization.

$N_x \times N_y$	32 × 32		64 × 64		128 × 128
Error	7.90885e-03		2.13944e-03		5.36247e-04
Rate		1.89		2.00	

Therefore, the time step satisfying the condition (21) guarantees $\|\phi^*\|_\infty \le 1$ if $\|\phi^n\|_\infty \le 1$, where ϕ^*_{ij} is from Eq. (13). Next, from Eq. (14), we have

$$|\phi^{n+1}_{ij}| = \left|\frac{(1+\phi^*_{ij})e^{A_{ij}} - 1 + \phi^*_{ij}}{(1+\phi^*_{ij})e^{A_{ij}} + 1 - \phi^*_{ij}}\right| \le \frac{(1+\phi^*_{ij})e^{A_{ij}} + 1 - \phi^*_{ij}}{(1+\phi^*_{ij})e^{A_{ij}} + 1 - \phi^*_{ij}} = 1, \tag{23}$$
$$\text{for } 1 \le i \le N_x,\ 1 \le j \le N_y,$$

and from Eq. (18), we have

$$|\phi^{n+1}_{ijk}| = \left|\frac{(1+\phi^*_{ijk})e^{A_{ijk}} - 1 + \phi^*_{ijk}}{(1+\phi^*_{ijk})e^{A_{ijk}} + 1 - \phi^*_{ijk}}\right| \le \frac{(1+\phi^*_{ijk})e^{A_{ijk}} + 1 - \phi^*_{ijk}}{(1+\phi^*_{ijk})e^{A_{ijk}} + 1 - \phi^*_{ijk}} = 1, \tag{24}$$
$$\text{for } 1 \le i \le N_x,\ 1 \le j \le N_y,\ 1 \le k \le N_z,$$

where we have used the condition, $\|\phi^*\|_\infty \le 1$. Eqs. (23) and (24) implies $\|\phi^{n+1}\|_\infty \le 1$ for all $n \ge 0$ if the initial condition satisfies $\|\phi^0\|_\infty \le 1$. In an analytic perspective, the space dependent mobility term is in a sense degenerating term for the time step Δt and 1 is the maximum value for the AC equation [4], therefore the governing equation is bounded above by 1. Therefore, the proposed scheme satisfies the maximum principle and implies the boundedness of the numerical solution if we use the time step satisfying the condition (21).

3. Computational results

In this section, unless otherwise stated, we use $\epsilon = h$ and $\Delta t = 0.99\epsilon^2 h^2/(2h^2 + 4\epsilon^2)$ and $\Delta t = 0.99\epsilon^2 h^2/(2h^2 + 6\epsilon^2)$ for the two- and three-dimensional modified AC equations, respectively.

3.1. Convergence tests

We verify the accuracy of the numerical scheme we used. Since we do not know the exact solution for our proposed model, we use the solution obtained with a fine mesh and sufficiently small time step using a fully explicit method as the reference solution. To verify the spatial accuracy with respect to space, we double the grid size on $\Omega = (-3,3)\times(-3,3)$ and calculate the error. We set the initial condition and define the $M(x,y)$ as follow:

$$\phi(x,y,0) = \tanh\left(\frac{2 - \sqrt{x^2+y^2}}{\sqrt{2}\epsilon}\right),$$
$$M(x,y) = \frac{p(x,y) - \min_{(x,y)\in\Omega} p(x,y)}{\max_{(x,y)\in\Omega} p(x,y) - \min_{(x,y)\in\Omega} p(x,y)},$$
$$p(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)e^{-x^2-y^2} - \frac{1}{3}e^{-(x+1)^2-y^2}.$$

Here, we use $\epsilon = 3H/[2\sqrt{2}\tanh^{-1}(0.9)]$, $H = 6/32$, $\Delta t = 1.\text{e-}8$, $T = 0.02$. For the reference solution, we set the grid size $N^{ref}_x \times N^{ref}_y = 2^1 0 \times 2^1 0$ and use the time step $\Delta t = 1.\text{e-}8$. Table 1 shows the errors and convergence rates in space. The error is defined as the discrete l_2-norm error and is defined as follows:

$$\text{error} = \sqrt{\frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left(\phi_{ij} - \frac{\phi^{ref}_{mi,mj} + \phi^{ref}_{mi+1,mj} + \phi^{ref}_{mi,mj+1} + \phi^{ref}_{mi+1,mj+1}}{4}\right)},$$

where ϕ^{ref} is the reference solution, and m is the ratio of N^{ref}_x to $2N_x$.

We also consider accuracy for time. We use the same initial condition and $M(x,y)$ on $\Omega = (-3,3)\times(-3,3)$. Here, we use $\epsilon = 4h/[2\sqrt{2}\tanh^{-1}(0.9)]$, $h = 6/1024$, $T = 1.\text{e-}4$. Table 2 shows the errors and convergence rates in time. We note that the accuracy of the used scheme is first order for time and second order for space, respectively.

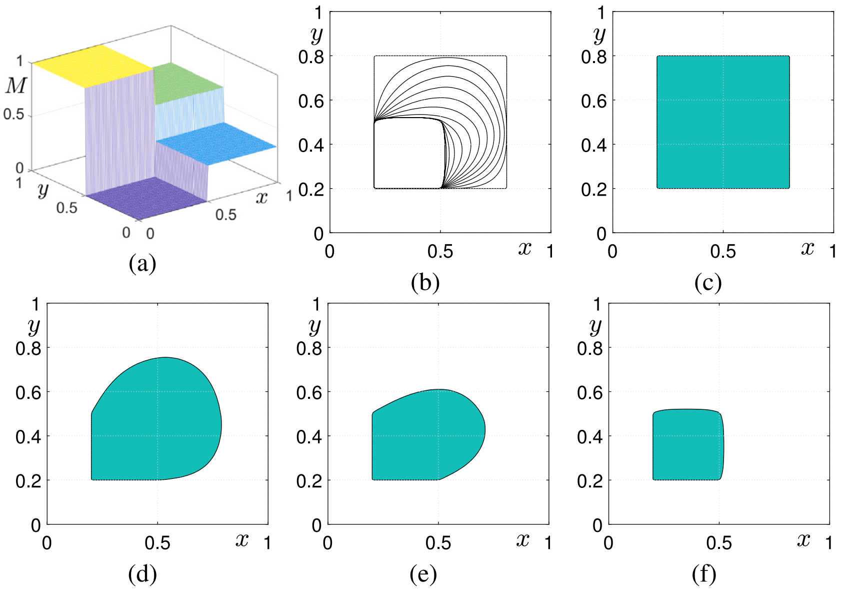

Fig. 1. (a) $M(x,y)$ is a space-dependent mobility. (b) is the temporal evolutions of contours. (c)–(f) are the temporal evolutions at times $t = 0$, $2000\Delta t$, $5000\Delta t$, and $15000\Delta t$.

Table 2
Errors and convergence rates for the temporal discretization.

N_t	100		200		400
Error	3.56418e-06		1.76366e-06		8.63723e-07
Rate		1.01		1.03	

3.2. Four different mobility values

The space-dependent mobility and initial condition on the domain $\Omega = (0,1)\times(0,1)$ are defined as follows:

$$M(x,y) = \begin{cases} 0, & \text{if } 0 < x,\ y < 0.5, \\ 0.33, & \text{if } 0.5 \le x < 1,\ 0 < y < 0.5, \\ 0.66, & \text{if } 0.5 \le x,\ y < 1, \\ 1, & \text{otherwise,} \end{cases} \tag{25}$$

$$\phi(x,y,0) = \begin{cases} +1, & \text{if } 0.2 < x,\ y < 0.8, \\ -1, & \text{otherwise,} \end{cases} \tag{26}$$

which are shown in Fig. 1(a) and (c). Fig. 1(b) shows the temporal evolution of the zero level contours of the phase-field function. Fig. 1(c)–(f) shows the snapshots the filled contours at times $t = 0$, $2000\Delta t$, $5000\Delta t$, and $15000\Delta t$, respectively. Here, we use $N_x = N_y = 100$, $h = 1/N_x$, and $\lambda = 0$. From the computational results, we can clearly see that the temporal evolutions are dependent on the space-dependent mobility. In regions with high values of $M(x,y)$, evolution is fast, while in regions with low values of $M(x,y)$, evolution is slow. As shown in Fig. 1(f), the temporal evolution is frozen in the region of $M(x,y) = 0$.

3.3. Smooth transition of mobility with several peaks

We consider a space-dependent mobility based on the several peaks function in MATLAB [33] such as

$$p(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-x^2-y^2} - \frac{1}{3} e^{-(x+1)^2-y^2}.$$

We define $M(x,y)$ on the domain $\Omega = (-3,3)\times(-3,3)$ as follows:

$$M(x,y) = \frac{p(x,y) - \min_{(x,y)\in\Omega} p(x,y)}{\max_{(x,y)\in\Omega} p(x,y) - \min_{(x,y)\in\Omega} p(x,y)}, \tag{27}$$

which is shown in Fig. 2(a). The initial condition is defined as follows:

$$\phi(x,y,0) = \begin{cases} +1, & \text{if } -2.5 < x,\ y < 2.5, \\ -1, & \text{otherwise,} \end{cases} \tag{28}$$

which is displayed in Fig. 2(c). Fig. 2(b) shows the temporal evolution of the zero level contours of the phase-field function. Fig. 2(c)–(f) shows the snapshots the filled contours at times $t = 0$, $7000\Delta t$, $13000\Delta t$, and $17000\Delta t$, respectively. Here, we use $N_x = N_y = 100$, $h = 6/N_x$, and $\lambda = 0$. We can observe the effect of the space-dependent mobility $M(x,y)$ on the evolution dynamics. As shown in Fig. 2(f), the interface shrinks at the position of the small value of $M(x,y)$.

3.4. Star-shaped space-dependent mobility

The authors in [34] proposed an image segmentation method utilizing a modified AC equation with a fractional Laplacian and presented image segmentation results with different fractional orders. We consider a similar test to a basic numerical experiment with a star-shaped initial condition. A star-shaped space-dependent mobility and initial condition on the domain $\Omega = (-3,3)\times(-3,3)$ are defined as follows:

$$M(x,y) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{1.5 + 0.4\cos(6\tan^{-1}(y/x)) - \sqrt{x^2+y^2}}{\sqrt{2}\epsilon}\right), \tag{29}$$

$$\phi(x,y,0) = \begin{cases} +1, & \text{if } 0.2 < x < 0.8,\ 0.2 < y < 0.8, \\ -1, & \text{otherwise,} \end{cases} \tag{30}$$

which are shown in Fig. 3(a) and (d), respectively. Fig. 3(b) displays the temporal evolution of the zero level contours of the phase-field function. Fig. 3(d)–(g) shows the snapshots of the filled contours at the zero level of the phase-field function at times $t = 0$, $2000\Delta t$, $5000\Delta t$, and $15000\Delta t$, respectively. Here, we use $N_x = N_y = 100$, $h = 6/N_x$, and $\lambda = 0$. Because of the space-dependent mobility and motion by mean curvature, the evolution reaches a convex hull shape as shown in Fig. 3(g). However, in the case of non-zero λ value, the evolution is a combination of motion by mean curvature and shrinkage. Therefore, the evolution goes further to reach a star-shaped morphology which is guided by the space-dependent mobility. These phenomena can be seen in Fig. 3(c), which is the temporal evolution of the zero level contours of the phase-field function. Here, we use $\lambda = 7$ and Fig. 3(h)–(k) shows the snapshots of the filled contours at the zero level of the phase-field function at times $t = 0$, $240\Delta t$, $330\Delta t$, and $1890\Delta t$, respectively.

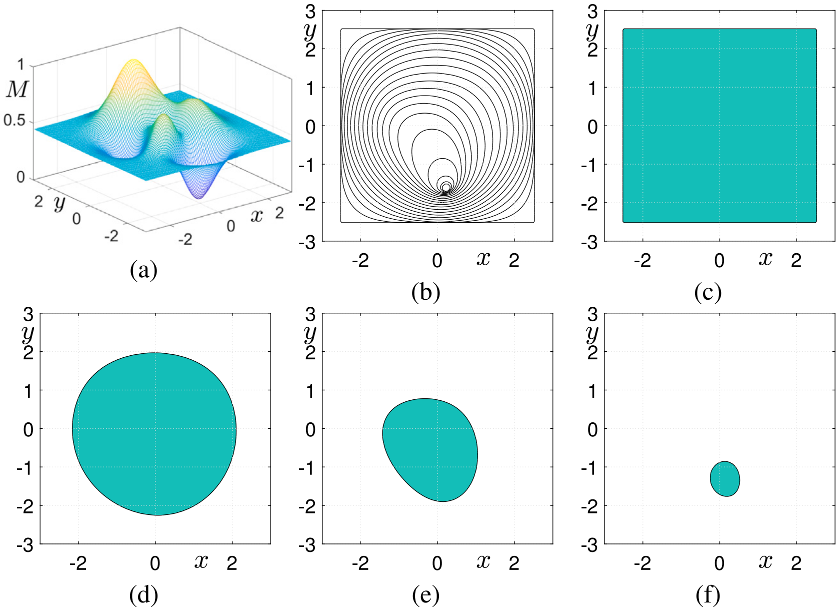

Fig. 2. (a) $M(x, y)$ is a space-dependent mobility with the several peaks. (b) is the temporal evolutions of contours. (c)–(f) are the temporal evolutions at times $t = 0$, $7000\Delta t$, $13000\Delta t$, and $17000\Delta t$, respectively.

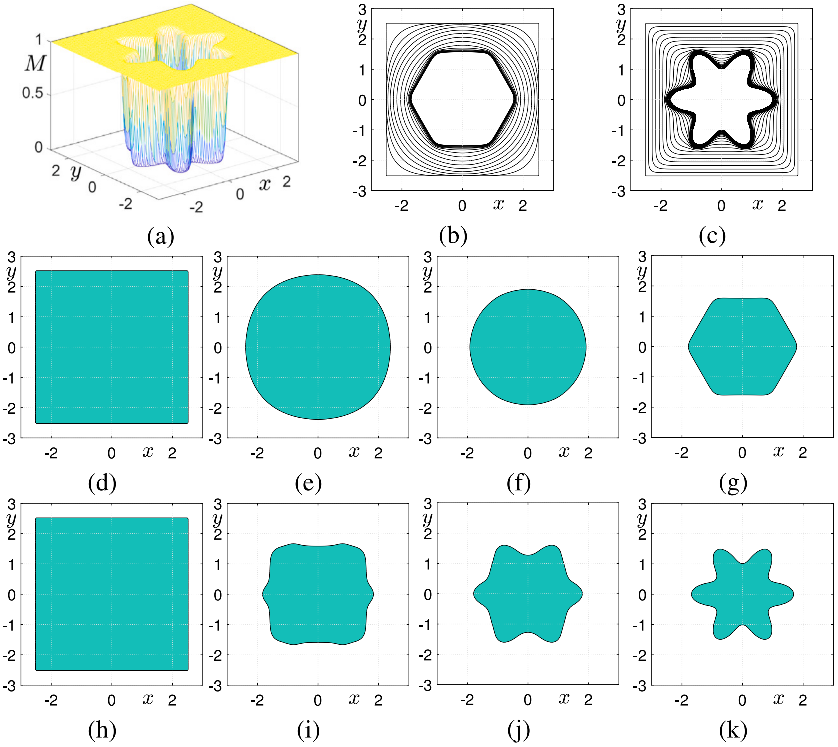

Fig. 3. (a) $M(x, y)$ is a space-dependent mobility. (b) is the temporal evolutions of contours with $\lambda = 0$. (d)–(g) are the temporal evolutions with $\lambda = 0$ at times $t = 0$, $2000\Delta t$, $4000\Delta t$, and $15000\Delta t$, respectively. (c) is the temporal evolutions of contours with $\lambda = 7$. (h)–(k) are the temporal evolutions with $\lambda = 7$ at times $t = 0$, $240\Delta t$, $330\Delta t$, and $1890\Delta t$, respectively.

Next, we explore the behavior of parameter λ through the following numerical test. While we have shrunk the square at Fig. 3, this time we start the evolution at the star-shaped morphology and demonstrate the expanding phenomena by applying a negative lambda, $\lambda = -7$. Using the identical space dependent mobility, the initial condition is given as $\phi(x, y, 0) = 1 - 2M(x, y)$. Fig. 4(a)–(d) illustrates the snapshots of the filled contours when $\lambda = 0$ at times $t = 0$, $600\Delta t$, $1200\Delta t$, and $2000\Delta t$, respectively. Then, Fig. 4(e)–(h) shows the temporal evolutions when $\lambda = -7$ at times $t = 0$, $600\Delta t$, $1200\Delta t$, and $2000\Delta t$, respectively. Similar to the results given at Fig. 3, the expansion goes further when $|\lambda|$ is larger.

3.5. Maximum principle test

Next, we consider the maximum principle of the proposed scheme. A random mobility and initial condition on the domain $\Omega = (-3, 3) \times (-3, 3)$ are defined as follows: $M(x, y) = \text{rand}(x, y)$ and $\phi(x, y, 0) = \text{rand}(x, y)$, where $\text{rand}(x, y)$ is a random number between -1 and 1.

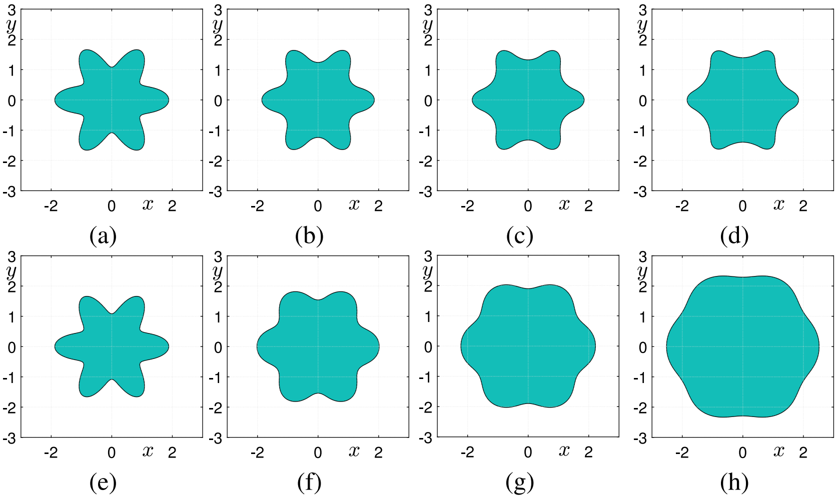

Fig. 4. (a)–(d) are the temporal evolutions with $\lambda = 0$ at times $t = 0,\ 600\Delta t,\ 1200\Delta t$, and $2000\Delta t$, respectively. (e)–(h) are the temporal evolutions with $\lambda = -7$ at times $t = 0,\ 600\Delta t,\ 1200\Delta t$, and $2000\Delta t$, respectively.

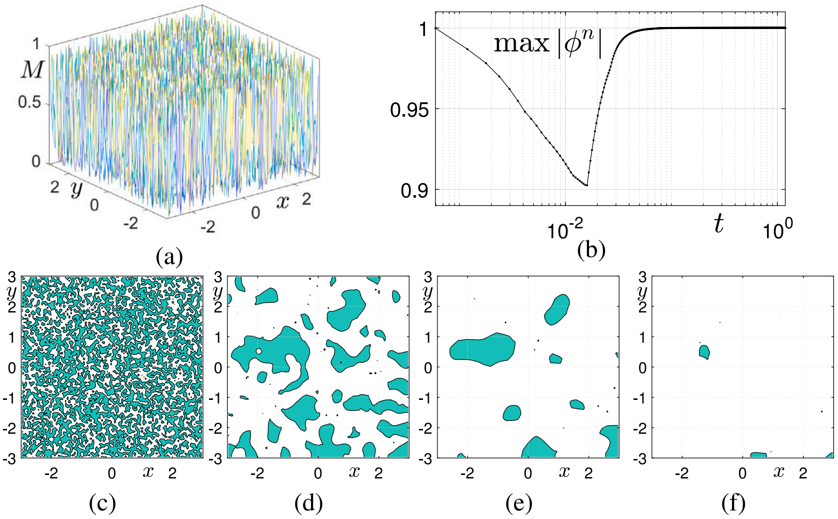

Fig. 5. (a) $M(x, y)$ is a space-dependent mobility. (b) is the temporal evolution of $\|\phi^n\|_\infty$. (c)–(f) are the temporal evolutions at times $t = 0,\ 90\Delta t,\ 480\Delta t$, and $1830\Delta t$.

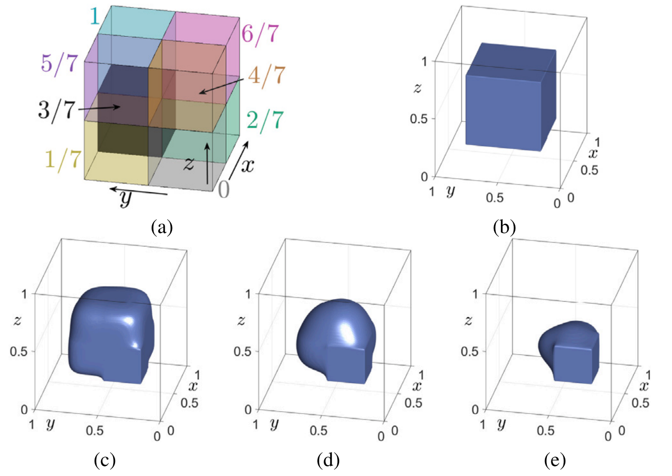

Fig. 6. (a) eight space-dependent mobility $M(x, y, z)$ is illustrated in different colors. (b)–(e) are the temporal evolutions at times $t = 0,\ 500\Delta t,\ 2000\Delta t$, and $5000\Delta t$, respectively.

Fig. 5(a) shows $M(x, y)$. Fig. 5(b) shows the temporal evolution of $\|\phi^n\|_\infty$ of the phase-field function. We can clearly observe that the numerical results satisfy the maximum principle. Fig. 5(c)–(f) shows the snapshots the filled contours at times $t = 0,\ 90\Delta t,\ 480\Delta t$, and $1830\Delta t$, respectively. Here, we use $N_x = N_y = 100$, $h = 6/N_x$, and $\lambda = 1$.

3.6. Eight different mobility values

The space-dependent mobility and initial condition on the domain $\Omega = (0, 1) \times (0, 1) \times (0, 1)$ are defined as follows:

$$M(x, y, z) = \begin{cases} 0, & \text{if } 0 < x,\ y,\ z < 0.5, \\ 1/7, & \text{if } 0 < x,\ z < 0.5,\ 0.5 \le y < 1, \\ 2/7, & \text{if } 0.5 \le x < 1,\ 0 < y,\ z < 0.5, \\ 3/7, & \text{if } 0.5 \le x,\ y < 1,\ 0 < z < 0.5, \\ 4/7, & \text{if } 0 < x,\ y < 0.5,\ 0.5 \le z < 1, \\ 5/7, & \text{if } 0 < x < 0.5,\ 0.5 \le y,\ z < 1, \\ 6/7, & \text{if } 0.5 \le x,\ z < 1,\ 0 < y < 0.5, \\ 1, & \text{otherwise}, \end{cases} \tag{31}$$

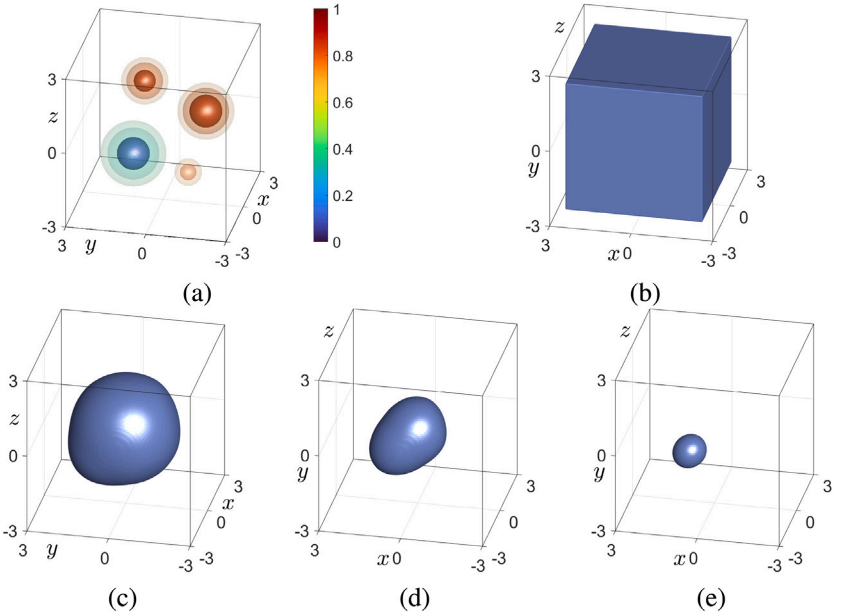

Fig. 7. (a) $M(x,y,z)$ is a space-dependent mobility with several spherical peaks. (b)–(e) are the temporal evolutions at times $t = 0$, $5000\Delta t$, $7500\Delta t$, and $10000\Delta t$, respectively.

$$\phi(x,y,z,0) = \begin{cases} +1, & \text{if } 0.2 < x,y,z < 0.8, \\ -1, & \text{otherwise,} \end{cases} \tag{32}$$

which are shown in Fig. 3(a) and(b), respectively. Fig. 3(b)–(e) show the snapshots of zero isosurface at times $t = 0$, $500\Delta t$, $2000\Delta t$, and $5000\Delta t$, respectively. Here, we use $N_x = N_y = N_z = 100$, $h = 1/N_x$, and $\lambda = 0$. From the computational results, we can clearly see that the temporal evolutions are dependent on the space-dependent mobility. In regions with high values of $M(x,y,z)$, evolution is fast, while in regions with low values of $M(x,y,z)$, evolution is slow. As shown in Fig. 6(e), the temporal evolution is frozen in the region of $M(x,y,z) = 0$.

3.7. Smooth transition of mobility with spherical peaks

We consider a space-dependent mobility on 3D space using spherical peaks on the domain $\Omega = (-3,3) \times (-3,3) \times (-3,3)$. Specifically, four spheres are centered at $C_1(1,1.5,1)$, $C_2(-1.5,0.5,0.5)$, $C_3(-1.2,-1.2,-1.2)$, and $C_4(1,-0.5,-1)$. Let

$$p(x,y,z) = \begin{cases} 5(1.2^2 - d_1^2), & \text{if } d_1 \le 1.2, \\ 4(1.4^2 - d_2^2), & \text{if } d_2 \le 1.4, \\ 9(0.8^2 - d_3^2), & \text{if } d_3 \le 0.8, \\ -4(1.3^2 - d_4^2), & \text{if } d_4 \le 1.3, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$d_1(x,y,z) = \sqrt{(x-1)^2 + (y-1.5)^2 + (z-1)^2},$$
$$d_2(x,y,z) = \sqrt{(x+1.5)^2 + (y-0.5)^2 + (z-0.5)^2},$$
$$d_3(x,y,z) = \sqrt{(x+1.2)^2 + (y+1.2)^2 + (z+1.2)^2},$$
$$d_4(x,y,z) = \sqrt{(x-1)^2 + (y+0.5)^2 + (z+1)^2}.$$

Note that each spheres have radius 1.2, 1.4, 0.8 and 1.3, respectively and have zero intersection. We define $M(x,y,z)$ as the normalized value of $p(x,y,z)$ in the following way:

$$M(x,y,z) = \frac{p(x,y,z) - \min_{(x,y,z)\in\Omega} p(x,y,z)}{\max_{(x,y,z)\in\Omega} p(x,y,z) - \min_{(x,y,z)\in\Omega} p(x,y,z)}, \tag{33}$$

which is shown in Fig. 7(a). The initial condition is defined as follows:

$$\phi(x,y,z,0) = \begin{cases} +1, & \text{if } -2.5 < x,y,z < 2.5, \\ -1, & \text{otherwise,} \end{cases} \tag{34}$$

which is shown in Fig. 7(b). Fig. 7(b)–(e) shows the snapshots of zero isosurface at times $t = 0$, $5000\Delta t$, $7500\Delta t$, and $10000\Delta t$, respectively. Here, we use $N_x = N_y = N_z = 100$, $h = 6/N_x$, and $\lambda = 0$. We can observe the effect of the space-dependent mobility $M(x,y,z)$ on the evolution dynamics. As shown in Fig. 7(e), the interface shrinks at the position of the small value of $M(x,y,z)$.

3.8. Star-shaped space-dependent mobility in 3D

We consider a similar test to a basic numerical experiment with a star-shaped initial condition referred from Li et al. [35]. A star-shaped space-dependent mobility and initial condition on the domain $\Omega = (-0.5,0.5) \times (-0.5,0.5) \times (-0.5,0.5)$ are defined as follows:

$$M(x,y,z) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{0.25 + 0.15\cos(6\theta) - \sqrt{x^2+y^2+z^2}}{2\sqrt{2}\epsilon}\right),$$

$$\text{where} \quad \theta = \begin{cases} \tan^{-1}(y/x), & \text{if } x < 0, \\ \tan^{-1}(\pi + y/x), & \text{otherwise ,} \end{cases}$$

$$\phi(x,y,z,0) = \begin{cases} +1, & \text{if } -0.4 < x,y,z < 0.4, \\ -1, & \text{otherwise.} \end{cases} \tag{35}$$

which are shown in Figs. 8(a) and (b), respectively. Here, we use $N_x = N_y = N_z = 100$, $h = 1/N_x$ and $\lambda = 0$ and 7. We first investigate the case when $\lambda = 0$. Figs. 8(c)–(e) show the snapshots of zero isosurface at zero level of the phase-field function at times $t = 500\Delta t$, $1000\Delta t$, and $4000\Delta t$, respectively. Because of the space-dependent mobility and motion by mean curvature, the evolution reaches a hexagon shape. However, in the case of non-zero λ value, the evolution is a combination of motion by mean curvature and shrinkage. Therefore, the evolution goes further to reach a star-shaped morphology which is guided by the space-dependent mobility. Here, we use $\lambda = 7$ and Figs. 8(f)–(h) show the snapshots of zero isosurface of the phase-field function at times $t = 500\Delta t$, $1000\Delta t$, and $4000\Delta t$, respectively.

3.9. Maximum principle test in 3D

Next, we consider the maximum principle of the proposed scheme. A random mobility and initial condition on the domain $\Omega = (-3,3) \times (-3,3) \times (-3,3)$ are defined as follows: $M(x,y,z) = (\text{rand}(x,y,z) + 1)/2$ and $\phi(x,y,z,0) = \text{rand}(x,y,z)$, where $\text{rand}(x,y,z)$ is a random number

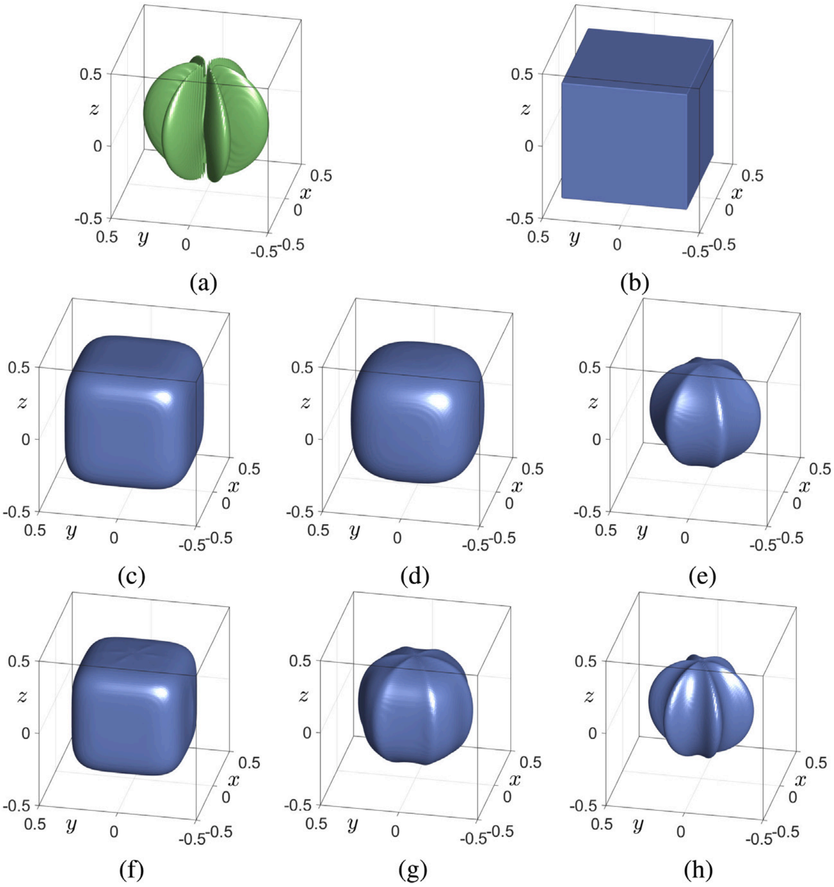

Fig. 8. (a) zero isosurface of space-dependent mobility $M(x, y, z)$. (b) initial ϕ_0. (c)–(e) are temporal evolution of zero isosurface when $\lambda = 0$ at times $t = 500\Delta t,\ 1000\Delta t$, and $4000\Delta t$, respectively. (f)–(h) are temporal evolution of zero isosurface when $\lambda = 7$ at times $t = 500\Delta t,\ 1000\Delta t$, and $4000\Delta t$, respectively.

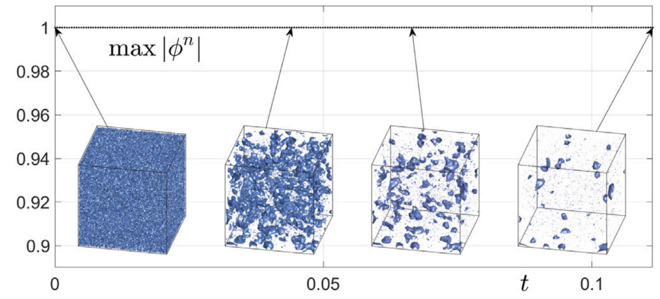

Fig. 9. $\|\phi^n\|_\infty$ of the phase-field function and snapshots of zero isosurface at times $t = 0,\ 100\Delta t,\ 150\Delta t$, and $250\Delta t$.

between −1 and 1. Fig. 9 illustrates $\|\phi^n\|_\infty$ of the phase-field function and snapshots of zero isosurface at times $t = 0,\ 100\Delta t,\ 150\Delta t$, and $250\Delta t$. We can clearly observe that the numerical results satisfy the maximum principle. Here, we use $N_x = N_y = N_z = 100$, $h = 6/N_x$ and $\lambda = 1$.

4. Conclusion

In conclusion, this paper has introduced a novel approach to model general motion by mean curvature using the AC equation with space-dependent mobility and a source term. The incorporation of a space-dependent mobility allows for precise control over the temporal evolution dynamics, while the addition of a source term permits manipulation of interface growth and shrinkage. We have used an operator splitting method to efficiently solve the governing equation and split it into the modified AC equation and the source term equation. The modified AC equation is solved with a fully explicit Euler method, and the source term equation is solved analytically. It is important to note that the overall numerical scheme preserves the maximum principle as long as the time step size satisfies a specific condition. To validate the effectiveness of the proposed mathematical model and its numerical algorithm, a series of computational experiments were conducted. The obtained numerical results demonstrated the efficiency and accuracy of the proposed model and its numerical approach. This research opens up new possibilities for the simulation and control of dynamic systems governed by mean curvature motion and offers a valuable tool for a wide range of scientific and engineering applications. As future research works, we will develop high-order numerical schemes such as a class of high-order maximum principle preserving methods for solving the AC equation [2,36]. The MATLAB source code for the proposed algorithm is provided in the Appendix for readers' reference.

CRediT authorship contribution statement

Junxiang Yang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Seungyoon Kang:** Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Data curation. **Soobin Kwak:** Data curation, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – review & editing. **Junseok Kim:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Use of AI tools declaration

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

Acknowledgments

J. Yang is supported by Macau University of Science and Technology Faculty Research Grants (FRG) (No. FRG-24-026-FIE). The corresponding author (J.S. Kim) was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A2C1003844). The authors are grateful to the reviewers whose comments significantly improved the initial manuscript.

Appendix. MATLAB source code

The main code for 2D star-shaped evolution is as follows:

```
clear; close all
Nx=100; Ny=100; Lx=-3; Rx=3; Ly=-3; Ry=3; h=(Rx-Lx)/Nx;
x=linspace(Lx-0.5*h,Rx+0.5*h,Nx+2); y=linspace(Ly-0.5*h,Ry+0.5*h,Ny+2)
      ;
ep=h; eps2=ep^2; dt=0.99*eps2*h^2/(2*h^2+4*eps2); lam=7; Maxiter=600;
      ns=30;
for i=1:Nx+2
    for j=1:Ny+2
theta=atan(y(j)/x(i));
M(i,j)=0.5*(1-tanh((1.5+0.4*cos(6*theta) ...
    -(sqrt(x(i)^2+y(j)^2)))/(sqrt(2)*ep)));
if abs(x(i))<2.5 && abs(y(j))<2.5
    p(i,j)=1;
else
    p(i,j)=-1;
end
    end
end
figure(1); clf; mesh(x(2:Nx+1),y(2:Ny+1),M(2:Nx+1,2:Ny+1)')
axis([Lx Rx Ly Ry 0 1]); set(gca,'fontsize',20); box on; grid on
text('Interpreter','latex','String','$$x$$', ...
    'Position',[0.6 -3.9 0],'fontsize',30);
text('Interpreter','latex','String','$$y$$', ...
    'Position',[-3.7 1 0],'fontsize',30);
text('Interpreter','latex','String','$$M$$', ...
    'Position',[-3.5 3.85 0.75],'fontsize',30);
figure(2); clf;
contour(x(2:Nx+1),y(2:Ny+1),p(2:Nx+1,2:Ny+1)',[0 0],'k'); hold on
figure(3); clf; contourf(x(2:Nx+1),y(2:Ny+1),p(2:Nx+1,2:Ny+1)',[0 0],'
      k');
axis image; axis([Lx Rx Ly Ry]); set(gca,'fontsize',22); box on; grid
      on
text('Interpreter','latex','String','$$x$$', ...
    'Position',[1. -3.45],'fontsize',30);
text('Interpreter','latex','String','$$y$$', ...
    'Position',[-3.5 2.5],'fontsize',30);
np=p;
for iter=1:Maxiter
p(1,:)=p(2,:);p(Nx+2,:)=p(Nx+1,:);p(:,1)=p(:,2);p(:,Ny+2)=p(:,Ny+1);
for i=2:Nx+1
    for j=2:Ny+1
     np(i,j)=p(i,j)+dt*M(i,j)*(-(p(i,j)^3-p(i,j))/eps2 ...
         +(p(i-1,j)+p(i+1,j)-4.0*p(i,j)+p(i,j-1)+p(i,j+1))/h^2);
    end
end
for i=2:Nx+1
    for j=2:Ny+1
     A=-sqrt(2)*lam*M(i,j)*dt/ep;
     p(i,j)=((1+np(i,j))*exp(A)-1+np(i,j))/((1+np(i,j))*exp(A)+1-np(i,
           j));
    end
end
if (mod(iter,ns)==0)
figure(2)
contour(x(2:Nx+1),y(2:Ny+1),p(2:Nx+1,2:Ny+1)',[0 0],'k'); pause(0.01)
figure(3); clf;
contourf(x(2:Nx+1),y(2:Ny+1),p(2:Nx+1,2:Ny+1)',[0 0],'k'); axis image;
axis([Lx Rx Ly Ry]); set(gca,'fontsize',22); box on; grid on
text('Interpreter','latex','String','$$x$$', ...
    'Position',[1. -3.45],'fontsize',30);
text('Interpreter','latex','String','$$y$$', ...
    'Position',[-3.5 2.5],'fontsize',30);
end
end
figure(2)
axis image; axis([Lx Rx Ly Ry]); set(gca,'fontsize',22); box on; grid
      on
text('Interpreter','latex','String','$$x$$', ...
    'Position',[1. -3.45],'fontsize',30);
text('Interpreter','latex','String','$$y$$', ...
    'Position',[-3.5 2.5],'fontsize',30);
```

The main code for 3D star-shaped evolution is as follows:

```
clear; clf; close all; Nx = 100; Ny = Nx; Nz = Nx;
Lx=-0.5; Ly=-0.5; Lz=-0.5; Rx=0.5; Ry=0.5; Rz=0.5; h=(Rx-Lx)/Nx;
x=linspace(Lx-0.5*h,Rx+0.5*h,Nx+2);y=linspace(Ly-0.5*h,Ry+0.5*h,Ny+2);
z=linspace(Lz-0.5*h,Rz+0.5*h,Nz+2); Maxiter=4000; ns=500; ep=h;ep2=ep
      ^2;
p = -ones(Nx+2,Ny+2,Nz+2); M=0*p+0.0; p=0*p-1;
for i=1:Nx+2
    for j=1:Ny+2
        for k=1:Nz+2
            if x(i)<0
                th = atan(y(j)/x(i));
            else
                th = pi+atan(y(j)/x(i));
            end
           M(i,j,k) = 0.5-0.5*tanh((0.25+0.15*cos(6*th) ...
                -sqrt(x(i)^2+y(j)^2+z(k)^2))/(2*sqrt(2)*ep));
            if abs(x(i))<0.4 && abs(y(j))<0.4 && abs(z(k))<0.4
                p(i,j,k)=1;
            end
        end
    end
end
np=p; np2=p; lmd=0; dt = 0.99*ep2*h^2/(2*h^2+6*ep2);
A = -sqrt(2)*lmd*M*dt/ep; p0 = p;
figure(1); hold on;
iso = isosurface(x(2:Nx+1),y(2:Ny+1),z(2:Nz+1), ...
    p(2:Nx+1,2:Ny+1,2:Nz+1), 0, 'k');
ISO = patch(iso); set(ISO, 'FaceColor',[0.5 0.6 0.9]);
set(ISO, 'EdgeColor','none'); axis image; axis([Lx Rx Ly Ry Lz Rz]);
view(-78, 26); set(gca,'fontsize',18); box on; grid on;
ax = gca; ax.BoxStyle = 'full';
text('Interpreter','latex','String','$$ x $$','Position', ...
    [0.32 -0.53 -0.61],'fontsize',25);
text('Interpreter','latex','String','$$ y $$','Position', ...
    [-0.57 0.32 -0.58],'fontsize',25);
text('Interpreter','latex','String','$$ z $$','Position', ...
    [-0.40 0.66 0.19],'fontsize',25);
lightangle(120, 150); drawnow;
for iter = 1:Maxiter
    p(1,:,:)=p(2,:,:); p(Nx+2,:,:)=p(Nx+1,:,:);
    p(:,1,:)=p(:,2,:); p(:,Ny+2,:)=p(:,Ny+1,:);
    p(:,:,1)=p(:,:,2); p(:,:,Nz+2)=p(:,:,Nz+1);
    for i=2:Nx+1
        for j=2:Ny+1
            for k=2:Nz+1
                np(i,j,k) = p(i,j,k) - dt*M(i,j,k)*(p(i,j,k)^3-p(i,j,k
                      ))...
                    /ep2+dt*M(i,j,k)*(p(i-1,j,k)+p(i+1,j,k)+p(i,j-1,k)
                          ...
                    +p(i,j+1,k)+p(i,j,k-1)+p(i,j,k+1)-6.0*p(i,j,k))/h
                          ^2;
                np2(i,j,k) = ((1+np(i,j,k))*exp(A(i,j,k))-1+np(i,j,k))
                       ...
                    /((1+np(i,j,k))*exp(A(i,j,k))+1-np(i,j,k));
            end
        end
    end
    p = np2;
    if (mod(iter,ns)==0)
        figure(iter/ns+1);
        iso6 = isosurface(x(2:Nx+1),y(2:Ny+1),z(2:Nz+1), ...
            p(2:Nx+1,2:Ny+1,2:Nz+1), 0, 'k');
        ISO6 = patch(iso6); set(ISO6, 'FaceColor',[0.5 0.6 0.9]);
        set(ISO6, 'EdgeColor','none'); axis image;
        axis([Lx Rx Ly Ry Lz Rz]); view(-78, 26); set(gca,'fontsize'
            ,18);
        ax = gca; ax.BoxStyle = 'full'; box on; grid on;
        text('Interpreter','latex','String','$$ x $$','Position', ...
            [0.32 -0.53 -0.61],'fontsize',25);
        text('Interpreter','latex','String','$$ y $$','Position', ...
            [-0.57 0.32 -0.58],'fontsize',25);
        text('Interpreter','latex','String','$$ z $$','Position', ...
            [-0.40 0.66 0.19],'fontsize',25);
        lightangle(120, 150); drawnow;
    end
end
```

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] H. Zhang, J. Yan, X. Qian, S. Song, Numerical analysis and applications of explicit high order maximum principle preserving integrating factor Runge–Kutta schemes for Allen–Cahn equation, Appl. Numer. Math. 161 (2021) 372–390.

[3] J. Wang, C. Lee, H.G. Lee, Q. Zhang, J. Yang, S. Yoon, J. Park J. Kim, Phase-field modeling and numerical simulation for ice melting, Numer. Math. Theor. Meth. Appl. 14 (2) (2021) 540–558.

[4] J. Shen, T. Tang, J. Yang, On the maximum principle preserving schemes for the generalized Allen–Cahn equation, Commun. Math. Sci. 14 (6) (2016) 1517–1534.

[5] L. Bertini, P. Butta, A. Pisante, Stochastic Allen–Cahn equation with mobility, NoDea-Nonlinear Differ. Equ. Appl. 24 (2017) 1–38.

[6] P.S. Morfe, Homogenization of the Allen–Cahn equation with periodic mobility, Calc. Var. Partial. Differ. Equ. 61 (3) (2022) 110.

[7] A.A. Lee, A. Munch, E. Suli, Sharp-Interface limits of the Cahn–Hilliard equation with degenerate mobility, SIAM J. Appl. Math. 76 (2) (2016) 433–456.

[8] X. Xiao, X. Feng, A second-order maximum bound principle preserving operator splitting method for the Allen–Cahn equation with applications in multi-phase systems, Math. Comput. Simulation 202 (2022) 36–58.

[9] Z. Qiao, Q. Zhang, Two-phase image segmentation by the Allen–Cahn equation and a nonlocal edge detection operator, Numer. Math. Theor. Meth. Appl. 15 (2022) 1147–1172.

[10] X. Wang, J. Kou, H. Gao, Linear energy stable and maximum principle preserving semi-implicit scheme for Allen–Cahn equation with double well potential, Commun. Nonlinear Sci. Numer. Simul. 98 (2021) 105766.

[11] J. Hötzer, A. Reiter, H. Hierl, P. Steinmetz, M. Selzer, B. Nestler, The parallel multi-physics phase-field framework Pace3D, J. Comput. Sci. 26 (2018) 1–12.

[12] J. Feng, Y. Zhou, T. Hou, A maximum-principle preserving and unconditionally energy-stable linear second-order finite difference scheme for Allen–Cahn equations, Appl. Math. Lett. 118 (2021) 107179.

[13] H. Zhang, X. Qian, S. Song, Third-order accurate, large time-stepping and maximum-principle-preserving schemes for the Allen–Cahn equation, Numer. Algor. (2023) http://dx.doi.org/10.1007/s11075-023-01606-w.

[14] Y.P. Sitompul, T. Aoki, S. Watanabe, T. Takaki, An ordered active parameter tracking method for efficient multiphase field simulations, J. Comput. Sci. 64 (2022) 101834.

[15] T.L. Hou, D.F. Xiu, W.Z. Jiang, A new second-order maximum-principle preserving finite difference scheme for Allen–Cahn equations with periodic boundary conditions, Appl. Math. Lett. 104 (2020) 106265.

[16] Z. Tan, C. Zhang, The discrete maximum principle and energy stability of a new second-order difference scheme for Allen–Cahn equations, Appl. Numer. Math. 166 (2021) 227–237.

[17] Y. Bo, D. Tian, X. Liu, Y. Jin, Discrete maximum principle and energy stability of the compact difference scheme for two-dimensional Allen–Cahn equation, J. Funct. Spaces 2022 (2022) 8522231.

[18] J. Wang, H. Xu, J. Yang, J. Kim, Fractal feature analysis based on phase transitions of the Allen–Cahn and Cahn–Hilliard equations, J. Comput. Sci. 72 (2023) 102114.

[19] Z. Gao, H. Zhang, X. Qian, S. Song, High-order unconditionally maximum-principle-preserving parametric integrating factor Runge–Kutta schemes for the nonlocal Allen–Cahn equation, Appl. Numer. Math. 194 (2023) 97–114.

[20] J. Yang, N. Yi, H. Zhang, High-order, unconditionally maximum-principle preserving finite element method for the Allen–Cahn equation, Appl. Numer. Math. 188 (2023) 42–61.

[21] J.W. Cahn, C.M. Elliott, A. Novick-Cohen, The Cahn–Hilliard equation with a concentration dependent mobility: motion by minus the Laplacian of the mean curvature, European J. Appl. Math. 7 (3) (1996) 287–301.

[22] C.M. Elliott, H. Garcke, On the Cahn–Hilliard equation with degenerate mobility, SIAM J. Numer. Anal. 27 (2) (1996) 404–423.

[23] J. Kim, A numerical method for the Cahn–Hilliard equation with a variable mobility, Commun. Nonlinear Sci. Numer. Simul. 12 (8) (2007) 1560–1571.

[24] S. Dai, Q. Du, Computational studies of coarsening rates for the Cahn–Hilliard equation with phase-dependent diffusion mobility, J. Comput. Phys. 310 (2016) 85–108.

[25] H.G. Lee, J.Y. Lee, A second order operator splitting method for Allen–Cahn type equations with nonlinear source terms, Phys. A 432 (2015) 24–34.

[26] Z. Weng, L. Tang, Analysis of the operator splitting scheme for the Allen–Cahn equation, Numer. Heat Transfer, Part B: Fundam. 70 (5) (2016) 472–483.

[27] Y. Huang, W. Yang, H. Wang, J. Cui, Adaptive operator splitting finite element method for Allen–Cahn equation, Numer. Methods Partial Differ. Eq. 35 (3) (2019) 1290–1300.

[28] C. Cui, J. Liu, Y. Mo, S. Zhai, An effective operator splitting scheme for two-dimensional conservative nonlocal Allen–Cahn equation, Appl. Math. Lett. 130 (2022) 108016.

[29] S. Ham, J.S. Kim, Stability analysis for a maximum principle preserving explicit scheme of the Allen–Cahn equation, Math. Comput. Simulation 207 (2023) 453–465.

[30] L.C. Evans, H.M. Soner, P.E. Souganidis, Phase transitions and generalized motion by mean curvature, Comm. Pure Appl. Math. 45 (9) (1992) 1097–1123.

[31] Q. Du, L. Ju, X. Li, Z. Qiao, Maximum bound principles for a class of semilinear parabolic equations and exponential time-differencing schemes, SIAM Rev. 63 (2) (2021) 317–359.

[32] H. Zhang, J. Yan, X. Qian, S. Song, Numerical analysis and applications of explicit high order maximum principle preserving integrating factor Runge–Kutta schemes for Allen–Cahn equation, Appl. Numer. Math. 161 (2021) 372–390.

[33] T. MathWorks, Matlab, Vol. 176, The MathWorks, Natick, MA, 2023.

[34] D. Lee, S. Lee, Image segmentation based on modified fractional Allen–Cahn equation, Math. Prob. Eng. 2019 (2019) 3980181.

[35] Y. Li, Q. Yu, W. Fang, B. Xia, J. Kim, A stable second-order BDF scheme for the three-dimensional Cahn–Hilliard–Hele–Shaw system, Adv. Comput. Math. 47 (2021) 1–18.

[36] H. Zhang, J. Yan, X. Qian, S. Song, Temporal high-order, unconditionally maximum-principle-preserving integrating factor multi-step methods for Allen–Cahn-type parabolic equations, Appl. Numer. Math. 186 (2023) 19–40.

Junxiang Yang is an assistant professor at Macau University of Science and Technology. He received the M.S. and Ph. D. degrees in Applied Mathematics from Korea University, Korea, in 2019 and 2021, respectively. And he received B.S. degree in Naval Architecture and Ocean Engineering from Chongqing Jiaotong University in 2017. His research interests include computational fluid dynamics, mathematical modeling, and scientific computing.

Seungyoon Kang received the B.S. degree from the Department of Mathematical Sciences, KAIST in 2019. Now, he is a Master student in Department of Mathematics, Korea University. His research interests include Additive Manufacturing, computational fluid dynamics, and scientific computing.

Soobin Kwak is a Ph.D. student at the Department of Mathematics, Korea University, Korea. And she received B.S. degree in Mathematics from Daegu University, Korea in 2020. Her research interests are mathematical modeling, computational fluid dynamics, and scientific computing.

Junseok Kim received his Ph.D. in Applied Mathematics from the University of Minnesota, U.S.A. in 2002. He also received his B.S. degree from the Department of Mathematics Education, Korea University, Korea in 1995. He joined the faculty of Korea University, Korea in 2008 where he is currently a full professor at the Department of Mathematics. His research interests are in computational finance and computational fluid dynamics.