Applied Numerical Mathematics 182 (2022) 87–99

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

An explicit stable finite difference method for the Allen–Cahn equation

Chaeyoung Lee [a], Yongho Choi [b], Junseok Kim [a,*]

[a] Department of Mathematics, Korea University, Seoul, 02841, Republic of Korea
[b] Department of Mathematics and Big Data, Daegu University, Gyeongsan-si, Gyeongsangbuk-do 38453, Republic of Korea

ARTICLE INFO

Article history:
Received 2 July 2021
Received in revised form 8 February 2022
Accepted 6 August 2022
Available online 12 August 2022

Keywords:
Stable numerical method
Operator splitting method
Allen–Cahn equation

ABSTRACT

We propose an explicit stable finite difference method (FDM) for the Allen–Cahn (AC) equation. The AC equation has been widely used for modeling various phenomena such as mean curvature flow, image processing, crystal growth, interfacial dynamics in material science, and so on. For practical use, an explicit method can be applied for the numerical approximation of the AC equation. However, there is a strict restriction on the time step size. To mitigate the disadvantage, we adopt the alternating direction explicit method for the diffusion term of the AC equation. As a result, we can use a relatively larger time step size than when the explicit method is used. Numerical experiments are performed to demonstrate that the proposed scheme preserves the intrinsic properties of the AC equation and it is stable compared to the explicit method.

1. Introduction

We present an explicit stable finite difference method (FDM) for the Allen–Cahn (AC) equation [2]:

$$\begin{aligned} &\frac{\partial \phi(\mathbf{x},t)}{\partial t} = -\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t), \quad \mathbf{x}\in\Omega,\ t>0, \\ &\mathbf{n}\cdot\nabla\phi(\mathbf{x},t) = 0 \text{ on } \partial\Omega, \end{aligned} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$, $\phi(\mathbf{x},t)$ is an order parameter, $F(\phi) = 0.25(\phi^2-1)^2$ is a double-well potential, and ϵ is the positive gradient energy coefficient. Here, $\mathbf{n}$ is the outer unit normal vector to $\partial\Omega$. The AC equation (1) is derived from the following total free energy functional [9]:

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

That is, the AC equation is the L^2-gradient flow of Eq. (2). The AC equation has been used for modeling a variety of phenomena such as phase separation [2,16,20,34,38], mean curvature flow [12,14,21–23], image processing [4,26,29], volume reconstruction [28,39], two-phase fluid flows [17,30,35], crystal growth [42], grain growth [1], and vesicle membranes [10,

* 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.apnum.2022.08.006
0168-9274/

11], interfacial dynamics in material science [7]. For the above applications, it is important to study numerical methods because the analytic solution may be found only under very limited conditions, i.e., traveling wave solutions [25].

Many researchers have studied FDM for solving nonlinear partial differential equations numerically. An explicit Euler method is one of the classical methods. It is simple and accurate, however, there is a strict restriction on the time step size [3,18]. To overcome the time step limitation for practical use of the AC equation, implicit or semi-implicit methods [8,15,25] have been proposed so that they allow the use of large time step sizes. In [8], Choi et al. presented an unconditionally gradient stable FDM for the AC equation applying Eyre's nonlinearly stabilized splitting method, which is a semi-implicit method in time and centered difference method in space. Hence, the method is stable when the time step size is large. Guan et al. [15] solved the AC type equation using a convex splitting method, which is unconditionally energy stable and second-order accurate in time. Li et al. [25] proposed a second-order accurate method both in time and space. Their method is also unconditionally stable and is based on an operator splitting method (OSM) dividing into the diffusion term and the free-energy term, which are solved by using the Crank–Nicolson method and a closed-form solution, respectively. The methods above have the common advantage of being unconditionally stable, however, a multigrid method is used to solve some implicit parts of the nonlinear equation in their papers.

The main purpose of this study is to present an explicit stable method for the AC equation using asymmetric approximations for the second derivatives developed by Saul'yev [33]. Explicit-type methods require less computation in comparison with implicit methods because of simplicity [31]. However, there is a serious constraint on the time step sizes to use the explicit methods. Therefore, the Saul'yev's method was proposed to overcome this limitation while maintaining the advantage of simplicity, and has been used in many studies. Tavakli and Davami [32] applied the Saul'yev's method to solve the diffusion equation more stably than when using an explicit method. The alternating direction explicit (ADE) method enables us to use larger time step sizes than the explicit method. In [6], Bučková et al. presented the ADE method for the Black–Scholes equations owing to its clarity and fine stability. Yang et al. [40] introduced the conservative ADE method for the heat equation because the ADE method does not guarantee the mass conservation property. They developed a simple weighted correction step, however, we exclude the step in this paper because one of the representative properties of the AC equation we study is that it does not conserve mass. In [41], the ADE method is applied for the Cahn–Hilliard equation which is the fourth-order parabolic equation. In spite of the highly nonlinear term and the biharmonic operator, it can be solved efficiently and numerical solutions can be obtained simply in complex domains by using the ADE method. Moreover, it is easy to use for parallel computing so that there have been several follow-up studies [13,37]. Using our proposed method, the AC equation can be solved with larger time step sizes than the time step restriction of an explicit scheme, therefore, the method is fast.

The rest of the paper is structured as follows. In Section 2, numerical solution algorithm is described, which use an OSM and the ADE method. In Section 3, the numerical results for several examples are presented to demonstrate that the numerical solutions of the proposed method follow the properties of the AC equation. In Section 4, conclusion is discussed.

2. Numerical method

For simplicity, the numerical scheme for the AC equation is described in a two-dimensional (2D) space $\Omega = (x_l, x_r) \times (y_l, y_r)$. Let $\Omega_h = \{x_i = x_l + (i-0.5)h,\ \ y_j = y_l + (j-0.5)h) |\ 1 \le i \le N_x,\ \ 1 \le j \le N_y\}$ be the discrete numerical domain, where $h = (x_r - x_l)/N_x = (y_r - y_l)/N_y$ is the uniform space step size; N_x and N_y are the positive integers. Let ϕ_{ij}^n be the numerical approximations of $\phi(x_i, y_j, n\Delta t)$, where Δt is the time step. The OSM is used to solve the AC equation (1). That is, first, we solve the diffusion equation

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \Delta \phi(\mathbf{x}, t) \tag{3}$$

and next, we solve the nonlinear equation

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2}. \tag{4}$$

We use the following ADE scheme [32] for the diffusion equation (3):

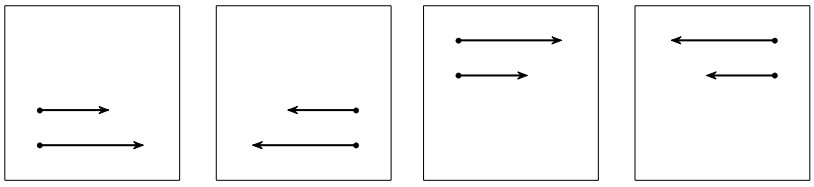
$$\text{For } j = 1, 2, \ldots, N_y, \quad \text{for } i = 1, 2, \ldots, N_x, \tag{5}$$

$$\frac{\phi_{ij}^* - \phi_{ij}^n}{\Delta t} = \frac{\phi_{i-1,j}^* + \phi_{i+1,j}^n - 2\phi_{ij}^n - 2\phi_{ij}^* + \phi_{i,j-1}^* + \phi_{i,j+1}^n}{h^2}. \tag{6}$$

Equation (6) can be rewritten as

$$\phi_{ij}^* = \frac{r\phi_{i-1,j}^* + r\phi_{i+1,j}^n + (1-2r)\phi_{ij}^n + r\phi_{i,j-1}^* + r\phi_{i,j+1}^n}{1+2r}, \tag{7}$$

where $r = \Delta t/h^2$. In 2D space, we have 4 cases of nested loops including Eq. (5); and the other cases are



(a) Eq. (5) (b) Eq. (8) (c) Eq. (9) (d) Eq. (10)

Fig. 1. Schematic diagram for 4 cases of nested loops, Eqs. (5), (8)–(10).

$$\text{For } j = 1, 2, \ldots, N_y, \quad \text{for } i = N_x, N_x - 1, \ldots, 1, \tag{8}$$

$$\text{For } j = N_y, N_y - 1, \ldots, 1, \quad \text{for } i = 1, 2, \ldots, N_x, \tag{9}$$

$$\text{For } j = N_y, N_y - 1, \ldots, 1, \quad \text{for } i = N_x, N_x - 1, \ldots, 1, \tag{10}$$

Fig. 1 illustrates the 4 cases of nested loops.

We use the zero Neumann boundary condition:

$$\begin{aligned} &\phi^*_{0j} = \phi^n_{1j}, \ \phi^*_{N_x+1,j} = \phi^n_{N_x,j}, \ \text{for } j = 1, \ldots, N_y, \\ &\phi^*_{i0} = \phi^n_{i1}, \ \phi^*_{i,N_y+1} = \phi^n_{i,N_y}, \ \text{for } i = 1, \ldots, N_x. \end{aligned} \tag{11}$$

Using the numerical solution ϕ^* from the first step, we analytically solve the nonlinear equation (4) using the separation of variables, as described in [19], and get the solution at the next time step:

$$\phi^{n+1}_{ij} = \frac{\phi^*_{ij}}{\sqrt{[1 - (\phi^*_{ij})^2]e^{-2\Delta t/\epsilon^2} + (\phi^*_{ij})^2}} \ \text{ for } \ 1 \le i \le N_x, \ 1 \le j \le N_y. \tag{12}$$

In a three-dimensional (3D) domain $\Omega = (x_l, x_r) \times (y_l, y_r) \times (z_l, z_r)$, similar to the description in a 2D domain, we define the uniform mesh grid size $h = (x_r - x_l)/N_x = (y_r - y_l)/N_y = (z_r - z_l)/N_z$ and the discrete domain $\Omega_h = \{(x_i = x_l + (i - 0.5)h, \ y_j = y_l + (j - 0.5)h, \ z_k = z_l + (k - 0.5)h) | \ 1 \le i \le N_x, \ 1 \le j \le N_y, \ 1 \le k \le N_z\}$, and N_z is the numbers grid point in the z-direction. Let ϕ^n_{ijk} be the numerical approximations of $\phi(x_i, y_j, z_k, n\Delta t)$. By using OSM, we solve Eqs. (3) and (4) in turn. In 3D, when solving Eq. (3) by applying ADE method, there are 8 cases of nested loops, which is an extension of 4 nested loops in 2D to 3D. For instance, one case is

$$\begin{aligned} &\text{For } k = 1, 2, \ldots, N_z, \quad \text{for } j = 1, 2, \ldots, N_y, \quad \text{for } i = 1, 2, \ldots, N_x, \\ &\phi^*_{ijk} = \frac{r\phi^*_{i-1,jk} + r\phi^n_{i+1,jk} + (1 - 3r)\phi^n_{ijk} + r\phi^*_{i,j-1,k} + r\phi^n_{i,j+1,k} + r\phi^*_{ij,k-1} + r\phi^n_{ij,k+1}}{1 + 3r}. \end{aligned}$$

The Neumann boundary condition is also applied similar to Eq. (11) for simplicity. Next, the analytic solution of Eq. (4) is, for $1 \le i \le N_x, \ 1 \le j \le N_y, \ 1 \le k \le N_z$,

$$\phi^{n+1}_{ijk} = \frac{\phi^*_{ijk}}{\sqrt{[1 - (\phi^*_{ijk})^2]e^{-2\Delta t/\epsilon^2} + (\phi^*_{ijk})^2}}.$$

3. Computational tests

Now, some computational experiments are performed such as linear stability, energy decay, convergence test, traveling wave solutions, motion by mean curvature, and temporal evolution of many shapes to show that the proposed method preserves the properties of the AC equation and to has the advantage that can use relatively large time steps than an explicit method. All tests are performed on an Intel Core i5-6400 CPU at 2.70 GHz with 4 GB of RAM.

3.1. Linear stability analysis

The linearization of the 2D AC equation around $\phi \equiv 0$ is as follows [27]:

$$\frac{\partial \phi}{\partial t} = \frac{\phi}{\epsilon^2} + \Delta \phi. \tag{13}$$

Let $\phi(x, y, t) = \alpha(t)\cos(k_1 \pi x)\cos(k_2 \pi y)$ with an amplitude $\alpha(t)$ at wave numbers k_1 and k_2. Substituting ϕ into Eq. (13), we obtain

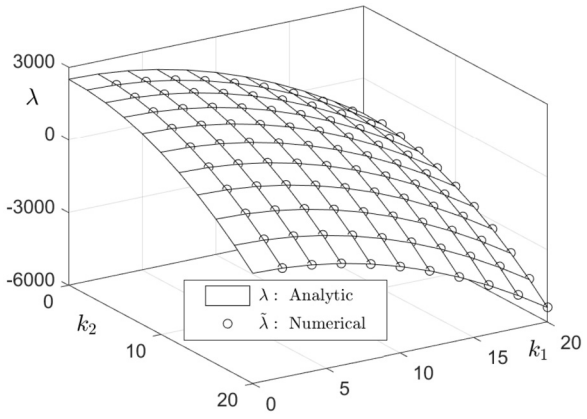


Fig. 2. Analytic and numerical growth rates with respect to wave numbers, k_1 and k_2.

$$\alpha'(t) = \left[\frac{1}{\epsilon^2} - (k_1\pi)^2 - (k_2\pi)^2\right]\alpha(t),$$

and the solution is $\alpha(t) = \alpha(0)\exp(\lambda t)$, where $\lambda = \frac{1}{\epsilon^2} - (k_1\pi)^2 - (k_2\pi)^2$ is the analytic growth rate of perturbation [27]. Let

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

be the numerical growth rate. Here, ϕ^{N_t} is the numerical solution at final time $T = N_t\Delta t$, N_t is a non-negative integer, and $||\cdot||_\infty$ is the discrete maximum norm. To investigate the linear stability, the initial condition is defined as

$$\phi = 0.01\cos(k_1\pi x)\cos(k_2\pi y),$$

in a 2D domain $\Omega = (0,1)\times(0,1)$. Here, k_1 and k_2 are 2, 4, ..., 20, and $\alpha(0) = 0.01$. $N_x = N_y = 200$, $h = 1/N_x$, $\Delta t = 0.01h^2$, $N_t = 1000$, and $\epsilon = 0.02$ are used. Fig. 2 illustrates the analytic and numerical growth rates with respect to different wave numbers, k_1 and k_2, and imply that the numerical results are in good agreement.

3.2. Decrease of the total energy

We can briefly prove the energy decreasing property [36] by differentiating Eq. (2) with respect to time t as follows:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{E}(\phi) &= \int_\Omega \left(\frac{F'(\phi)}{\epsilon^2}\frac{\partial\phi}{\partial t} + \nabla\phi\cdot\nabla\frac{\partial\phi}{\partial t}\right)\mathrm{d}\mathbf{x}\\ &= \int_{\partial\Omega}(\mathbf{n}\cdot\nabla\phi)\frac{\partial\phi}{\partial t}\mathrm{d}s + \int_\Omega\left(\frac{F'(\phi)}{\epsilon^2}\frac{\partial\phi}{\partial t} - \Delta\phi\frac{\partial\phi}{\partial t}\right)\mathrm{d}\mathbf{x}\\ &= -\int_\Omega\left(\frac{\partial\phi}{\partial t}\right)^2\mathrm{d}\mathbf{x} \le 0.\end{aligned}$$

Thus, the total energy decreases in time. Next, we demonstrate the discrete energy decay property. In this paper, the theoretical analysis of the energy stability of the proposed method still remains an open problem because the OSM is a method of dividing the governing equation based on the operators unlike the convex splitting method where the criterion for dividing the governing equation is the energy functional. Therefore, we numerically verify the energy decay property. The total energy functional (2) is discretized as

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

where the zero Neumann boundary condition is also used. On a 2D domain $\Omega = (-1,1)\times(-1,1)$ with $N_x = N_y = 100$, the initial condition is

$$\phi(x,y,0) = \begin{cases} 1 & \text{if } |x| < 0.8 \text{ and } |y| < 0.8,\\ -1 & \text{otherwise.}\end{cases}$$

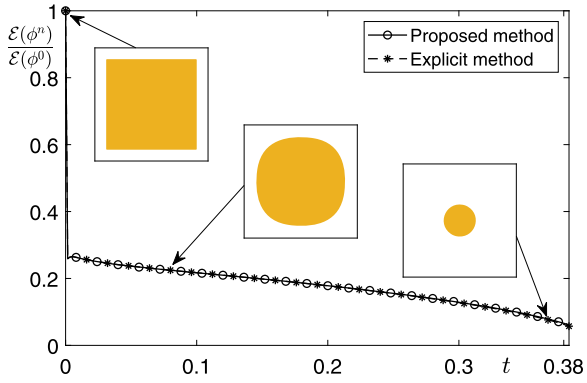


Fig. 3. Discrete total energy $\mathcal{E}(\phi^n)/\mathcal{E}(\phi^0)$.

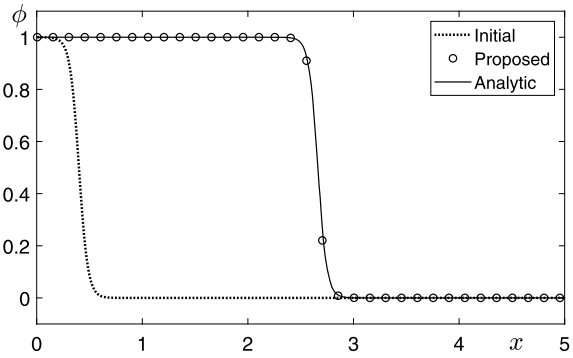


Fig. 4. Proposed and analytic traveling wave solutions at $t = 0$ and $t = 0.032$.

We use the following parameters: the space step size $h = 2/N_x$, the time step size $\Delta t = 0.1h^2$, and $\epsilon = 0.05$. Fig. 3 shows that the temporal evolution of normalized discrete total energy $\mathcal{E}(\phi^n)/\mathcal{E}(\phi^0)$ and the discrete energy is decreasing.

3.3. Traveling wave solutions

When the ADE method is applied, this is also an explicit method, but the time step size Δt can be used larger than when using the explicit hybrid method in [18], which differs from the ADE method only in solving Eq. (3). For one loop, Eq. (6) is replaced by

$$\frac{\phi_{ij}^* - \phi_{ij}^n}{\Delta t} = \frac{\phi_{i-1,j}^n + \phi_{i+1,j}^n - 4\phi_{ij}^n + \phi_{i,j-1}^n + \phi_{i,j+1}^n}{h^2} \tag{14}$$

or

$$\frac{\phi_{ijk}^* - \phi_{ijk}^n}{\Delta t} = \frac{\phi_{i-1,j,k}^n + \phi_{i+1,j,k}^n - 6\phi_{ij}^n + \phi_{i,j-1,k}^n + \phi_{i,j+1,k}^n + \phi_{ij,k-1}^n + \phi_{ij,k+1}^n}{h^2} \tag{15}$$

in 2D and 3D, respectively. We examine the range of possible Δt using traveling wave solutions. The AC equation (1) has a traveling wave solution:

$$\phi(x, y, t) = 0.5\left(1 - \tanh\left(\frac{x - 0.4 - st}{2\sqrt{2}\epsilon}\right)\right). \tag{16}$$

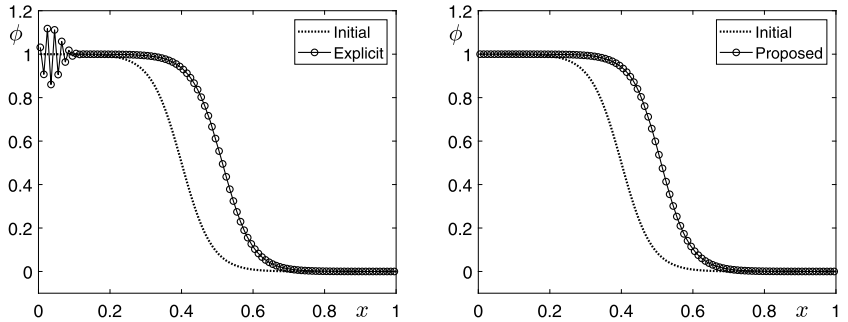
Here, the speed of the traveling wave is $s = 3/(\sqrt{2}\epsilon)$ [8]. The initial condition is given as

$$\phi(x, y, 0) = 0.5\left(1 - \tanh\left(\frac{x - 0.4}{2\sqrt{2}\epsilon}\right)\right), \tag{17}$$

on the domain $\Omega = (0, 5) \times (0, 0.1)$ with $N_x = 500$ and $N_y = 10$. $h = 0.01$, $\Delta t = 0.2h^2$, and $\epsilon = 0.03$ are used. Fig. 4 shows the proposed and analytic traveling wave solutions which are the slices at $y = 0.5h$ at $t = 0.032$.

Table 1
Discrete l_2-norm errors of the explicit hybrid method and the proposed method with respect to Δt.

Δt	Discrete l_2-norm error	
	Explicit	Proposed
$0.1h^2$	2.8145e-5	1.0073e-4
$0.2h^2$	9.7731e-5	2.9321e-4
$0.5h^2$	4.6118e-4	1.4176e-3
h^2	unstable	6.2023e-3
$2h^2$	unstable	2.3811e-2
$4h^2$	unstable	7.3777e-2
$8h^2$	unstable	2.0985e-1
$10h^2$	unstable	2.7316e-1



(a) Unstable (b) Stable

Fig. 5. With $\Delta t = h^2$, the numerical solutions of (a) the explicit hybrid method and (b) the proposed method at $t = 0.0016$.

Now, a numerical test was conducted to investigate the accuracy and stability of the proposed method, compared with the explicit hybrid method. The same initial condition and parameters in Section 3.3 are used and the final time T is taken as $40\Delta t$. The discrete l_2-norm error is defined as

$$||e||_2 = \sqrt{\sum_{i=1}^{N_x} (\phi_{i1}^{N_t} - \phi(x_i, y_1, N_t))/N_x}. \tag{18}$$

Here, we compare the discrete l_2-norm errors between the numerical and closed-form solutions with respect to Δt. Table 1 lists the discrete l_2-norm errors of the two numerical methods from $\Delta t = 0.1h^2$ to $10h^2$.

We compare the results of both methods using the time step size $\Delta t = h^2$. Other settings are the same as above. Fig. 5 shows the two numerical solutions at $t = 0.0016$. While oscillation is observed in Fig. 5(a) using the explicit method, it is numerically stable in Fig. 5(b) using our proposed method.

Next, we compare the proposed method with an implicit method. For a more accurate comparison, we use the implicit hybrid method, where the diffusion part is numerically solved by using an implicit method and a multigrid method, and the nonlinear part is analytically solved. It is the same as the explicit hybrid method, but all n on the right side of Eq. (14) and Eq. (15) are changed to $(n+1)$. On the computational domain $\Omega = (0, 4) \times (0, 1)$, the initial condition and its exact solution are also defined as Eq. (17) and Eq. (16), respectively. The parameters are used as $\epsilon = 0.03$, $N_x = 512$, $N_y = 128$, $h = 4/N_x = 1/N_y$, $\Delta t = h^2$, and $T = 400h^2$. Fig. 6 illustrates the initial condition and the numerical and analytic solutions at the final time.

Table 2 lists the errors between the numerical and analytic solutions computed by using the discrete l^2-norm error (18). With the fixed final time T, we compute the numerical solutions using the two methods as changing the time step size, and compare the errors and CPU time(s). It is observed that as Δt increases, the proposed method has low accuracy than the implicit hybrid method, however, for an accurate numerical solution, a small Δt should be used. In addition, the proposed method has less computational cost because its algorithm is simple.

As a final comparison, we consider the computational complexity of the two methods. Since the nonlinear part is solved using the analytic method in both methods, the comparison for that part can be omitted and only the diffusion part needs to be compared. The Gauss–Seidel type method, used in the proposed method, is widely adopted for the relaxation step in the multigrid method. The cost of performing one relaxation sweep is called a WU (work unit). Let each number of pre- and post-smoothing relaxation be the integer $\nu \geq 1$. In a 2D domain, the computational complexity of the proposed method

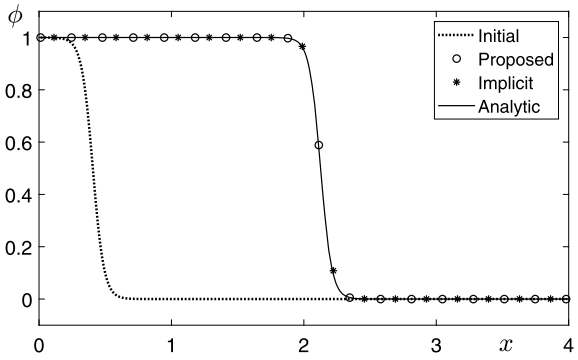


Fig. 6. Comparison between the proposed and implicit hybrid methods using the traveling wave solutions.

Table 2
Discrete l_2-norm errors and CPU time(s) of the proposed and implicit hybrid methods with respect to Δt.

Δt	l_2-norm error (CPU time)	
	Proposed	Implicit
$0.2h^2$	7.0986*e*-4 (4.2325*e*+0)	2.0623*e*-3 (5.3984*e*+1)
h^2	2.4606*e*-3 (0.8579*e*+0)	5.3359*e*-3 (1.0999*e*+1)
$5h^2$	2.2606*e*-1 (0.1939*e*+0)	2.2519*e*-2 (0.3282*e*+1)

Table 3
Discrete l^2-errors and rate of convergence.

$(N_x,\ N_t)$	(125, 60)	Rate	(250, 240)	Rate	(500, 960)
l_2-error	2.9689-4	2.00	7.4322*e*-5	1.98	1.8791*e*-5

is estimated to be almost WU, whereas the computation cost of a V-cycle in the multigrid method is greater than 8νWU/3. Hence, the proposed method is much faster than the multigrid method. Please refer to [5,40] for more details.

3.4. Convergence test

We numerically verify the proposed method is convergent with first-order accuracy in time and second-order accuracy in space. On $\Omega = (0, 5) \times (0, 0.2)$, the initial condition is defined as Eq. (17) with $\epsilon = 0.03$ because in this case the exact solution is known as a traveling wave solution (16). With the fixed final time $T = 1.0e\text{-}5$, the time step size is defined as $\Delta t = T/N_t$ and the space step size is defined as $h = 5/N_x = 0.2/N_y$. In this test, the discrete l^2-norm is defined as:

$$\|e^{\Delta t,h}\|_2 = \sqrt{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{1}{N_xN_y}(\phi_{ij}^{N_t} - \phi(x_i, y_j, N_t\Delta t))^2},$$

where $\phi_{ij}^{N_t}$ is the numerical solution and $\phi(x_i, y_j, N_t\Delta t)$ is the analytic solution when the time and space step sizes are used as Δt and h. The convergence rate is defined as $\log_2\left(\|e^{\Delta t,h}\|_2/\|e^{\Delta t/4,h/2}\|_2\right)$. Here, the time step size Δt is reduced by 4 times and the space step size h is reduced by 2 times. Table 3 shows that the proposed scheme is temporally first-order and spatially second-order accurate.

3.5. Motion by mean curvature

Another property of the AC equation is that the interface of the solution ϕ evolves in the direction of the normal vector $\mathbf{n}$ in proportion to its mean curvature [21,25]. Thus, the zero level set of ϕ moves according to the property in the limit of small ϵ, and the normal velocity V of the zero level set at each point $\mathbf{x}$ is

$$V = -\left(\frac{1}{R_1} + \frac{1}{R_2}\right), \tag{19}$$

where R_1 and R_2 are the principal radii of curvatures at $\mathbf{x}$. Refer to [21] to see the derivation of Eq. (19). Let us consider the same radii $R(t)$ for curvature. In a 2D space, because of $R_1 = R$ and $R_2 = 0$, $V = \mathrm{d}R(t)/\mathrm{d}t = -1/R$, therefore, the radius

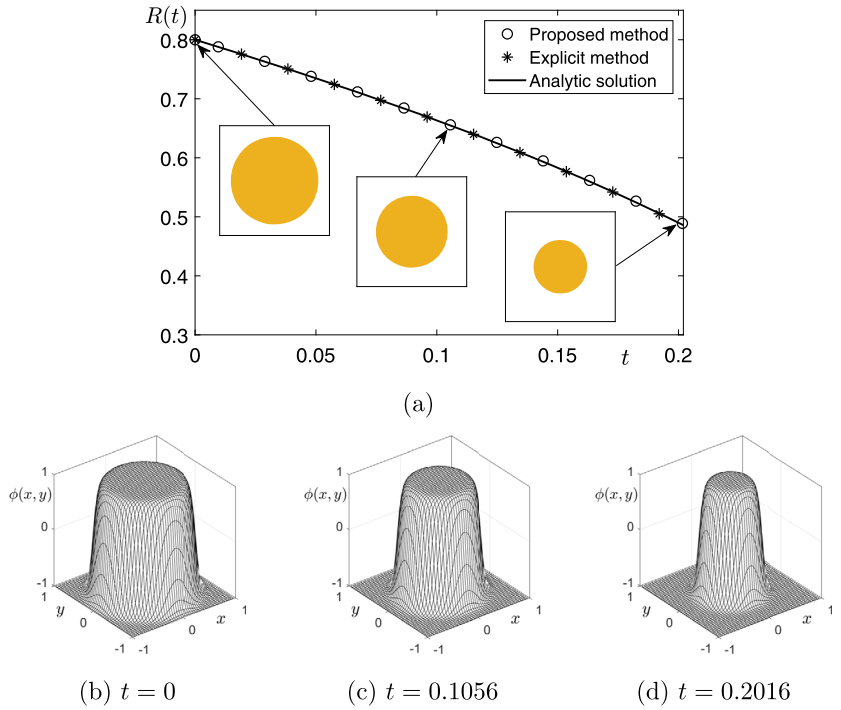


Fig. 7. (a) Radius $R(t)$ of circle in the 2D domain over time t. (b)–(d) Temporal evolutions of the numerical solution $\phi(x, y, t)$. The times are given below each figure.

$R(t) = \sqrt{R(0)^2 - 2t}$. In a 3D space, because of $R_1 = R_2 = R$, $V = \mathrm{d}R(t)/\mathrm{d}t = -2/R$, therefore, the radius $R(t) = \sqrt{R(0)^2 - 4t}$. In other words, $R(t) = \sqrt{R(0)^2 - 2t}$ in 2D and $R(t) = \sqrt{R(0)^2 - 4t}$ in 3D are called the asymptotic solutions, where the zero level set of the solution converges to the motion by mean curvature as ϵ approaches zero [24].

We compare the radii of the asymptotic solution, the proposed method, and the explicit hybrid method. First, we consider a 2D computational experiment with the initial condition

$$\phi(x, y, 0) = \tanh\left(\frac{0.8 - \sqrt{x^2 + y^2}}{\sqrt{2}\epsilon}\right),$$

on the domain $\Omega = (-1, 1) \times (-1, 1)$ with $N_x = N_y = 100$. That is a circle with center $(0, 0)$ and radius $R(0) = 0.8$. We take the parameters: $h = 2/N_x$, $\Delta t = 0.1h^2$, and $\epsilon = 0.05$. Fig. 7(a) shows the temporal evolution of the radius $R(t)$ of circle, which is the zero level contour of the numerical solution $\phi(x, y, t)$ in Fig. 7(b)–(d). When we use the small time step size such as $\Delta t = 0.1h^2$, the numerical results of both the proposed method and the explicit hybrid method are similar. Moreover, the numerical results are in good agreement to the asymptotic solution under the motion by mean curvature.

Now, we perform the same simulation on a 3D domain $\Omega = (-1, 1) \times (-1, 1) \times (-1, 1)$ with $N_x = N_y = N_z = 100$. The initial condition is a sphere with center $(0, 0, 0)$ and radius $R(0) = 0.8$:

$$\phi(x, y, z, 0) = \tanh\left(\frac{0.8 - \sqrt{x^2 + y^2 + z^2}}{\sqrt{2}\epsilon}\right).$$

We use the parameters: $h = 2/N_x$, $\Delta t = 0.1h^2$, and $\epsilon = 0.05$. The behavior of 3D numerical results as shown in Fig. 8 is similar to the previous 2D results.

Let us consider temporal evolutions of a maze on a 2D domain $\Omega = (0, 1) \times (0, 1)$ with $N_x = N_y = 100$. Here, we use the parameters, $h = 0.01$ and $\epsilon = 0.02$. The initial condition is $\phi(x, y, 0) = 1$ inside the maze and otherwise $\phi(x, y, 0) = -1$, see the first column in Fig. 9. Fig. 9(a) and (b) show the temporal evolutions using the explicit hybrid method and the proposed method, respectively. In this test, we set the time step size as $\Delta t = 0.2h^2$ for stability of the explicit hybrid method. In Fig. 9(c)–(e), using the proposed method, it is observed that we can use (c) twice, (d) four times, and (e) eight times large time step sizes than $\Delta t = 0.2h^2$.

A similar test is conducted in a 3D domain $\Omega = (0, 1) \times (0, 1) \times (0, 1)$ with $N_x = N_y = N_z = 100$. The parameters used are $h = 0.01$ and $\epsilon = 0.03$. The first column in Fig. 10 shows the initial states. Fig. 10(a) and (b) show the temporal evolutions using the explicit hybrid method and the proposed method, respectively. In this test, we set the time step size as $\Delta t = 0.1h^2$

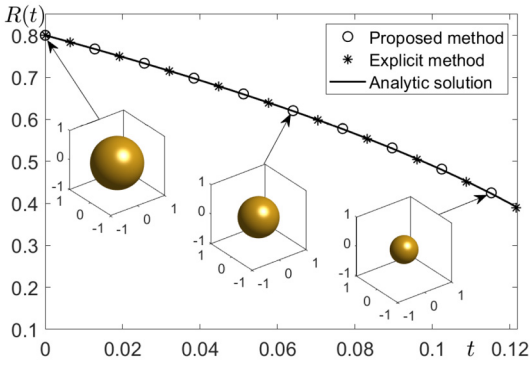


Fig. 8. Temporal evolutions of the radius $R(t)$ of sphere in the 3D domain over time t.

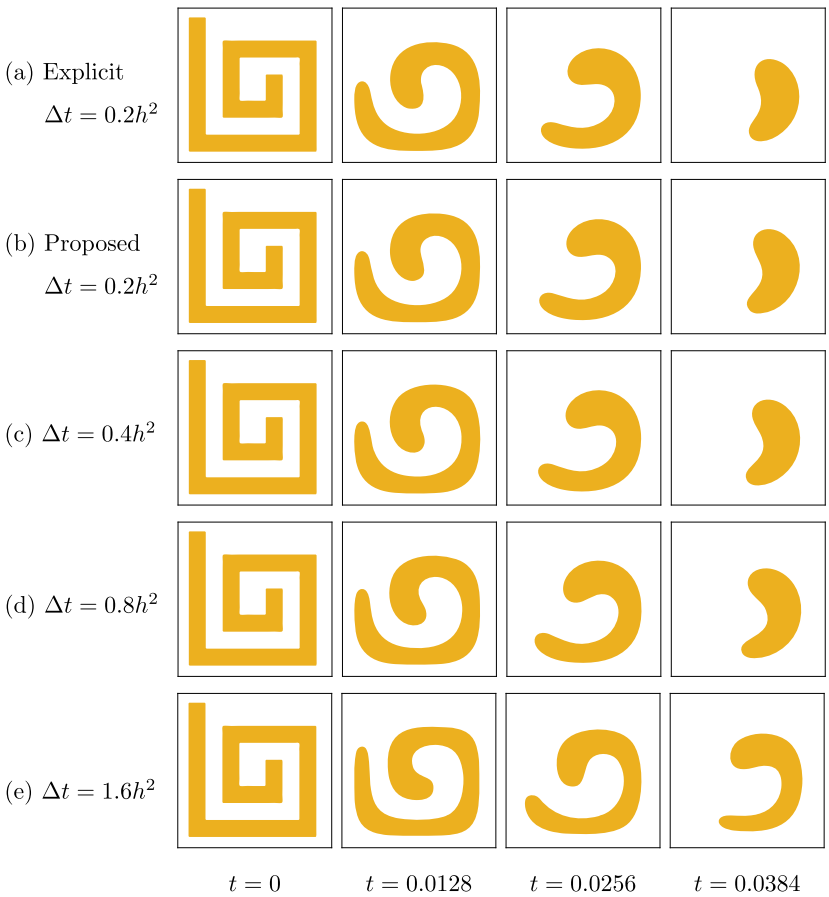


Fig. 9. Temporal evolutions from $t = 0$ to $t = 0.0384$ using (a) the explicit hybrid method and (b)–(e) the proposed method. (a) and (b) are the results with $\Delta t = 0.2h^2$. (c)–(e) When the proposed method is used, the time step sizes can be used as 2, 4, or 8 times larger than that when the explicit method is used, respectively. The times are given below the figures.

for stability of the explicit hybrid method. In Fig. 10(c), using the proposed method, it is observed that we can use eight times large time step sizes than $\Delta t = 0.1h^2$.

Table 4 lists the relative CPU times of the proposed method with several Δt to the explicit hybrid method with the fixed time step size Δt_e in Figs. 9 and 10. Here, the CPU times are averaged over 10 trials. Table 4 lists the relative CPU times and suggests that the proposed method is faster when we use larger time steps than the explicit hybrid method. Fig. 11 illustrates the results of Table 4.

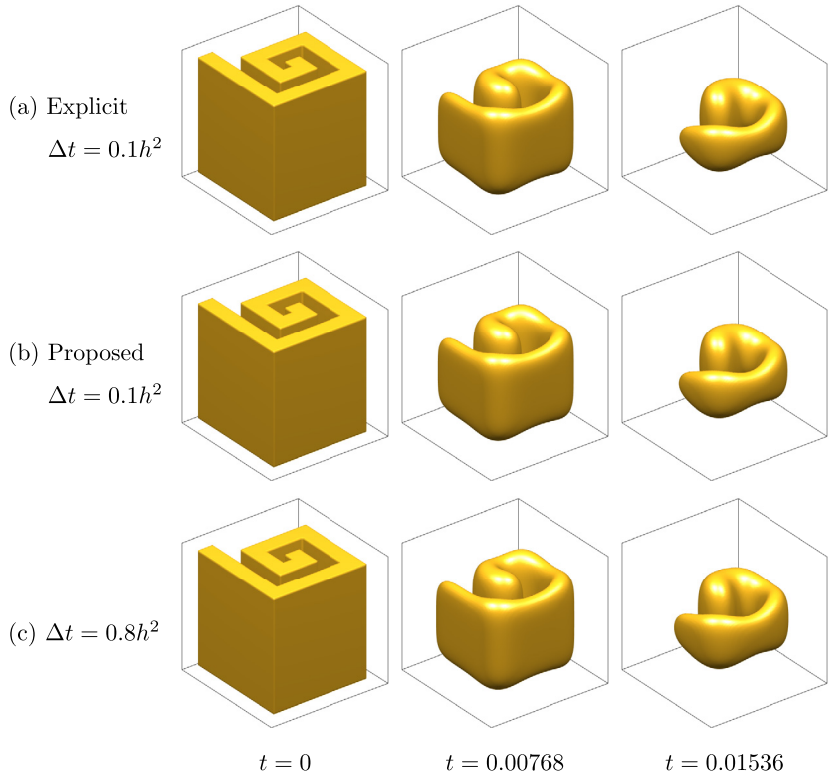


Fig. 10. Temporal evolutions from $t = 0$ to $t = 0.01536$ using (a) the explicit hybrid method and (b)–(c) the proposed method. (a) and (b) are the results with $\Delta t = 0.1h^2$. (c) When the proposed method is used, the time step sizes can be used as 8 times larger than that when the explicit method is used, respectively. The times are given below the figures.

Table 4
Relative CPU time of the proposed method (Δt) to the explicit hybrid method (Δt_e).

$\Delta t/\Delta t_e$		1	2	4	8
CPU times	Fig. 9	1.0251	0.4763	0.2418	0.1265
	Fig. 10	0.9603	0.4729	0.2436	0.1230

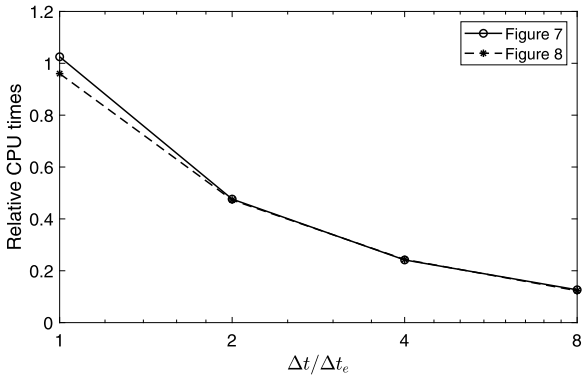
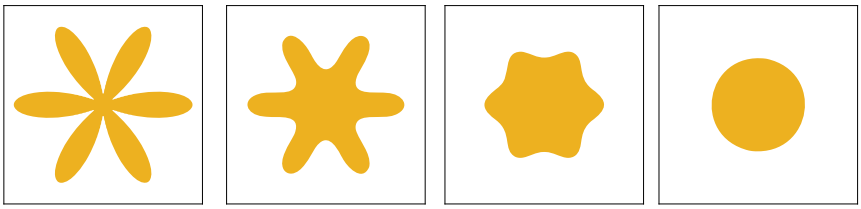


Fig. 11. Relative CPU time of the proposed method in Figs. 9–10.

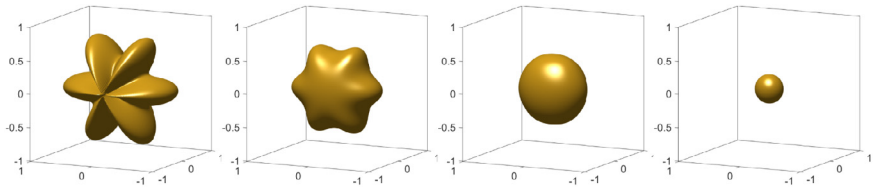
3.6. Temporal evolution of various initial conditions

We simulate with various initial conditions such as star shape in 2D and perturbed, torus shape in 3D. The initial condition of star shape in 2D is given as follows:



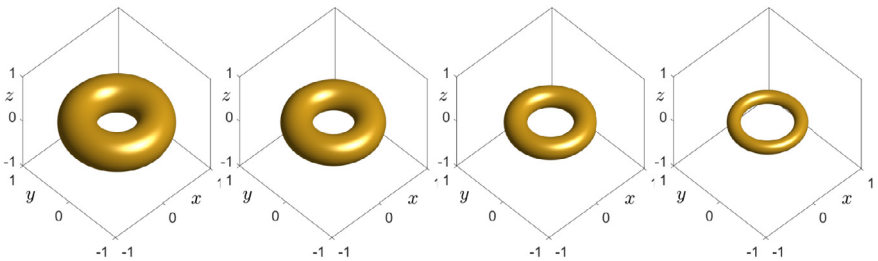
(a) $t = 0$ (b) $t = 0.008$ (c) $t = 0.024$ (d) $t = 0.056$

Fig. 12. Temporal evolution of star shape. The evolution time is presented on the bottom of each figure, and the evolutions are shown along with the motion by mean curvature.



(a) $t = 0$ (b) $t = 0.024$ (c) $t = 0.048$ (d) $t = 0.096$

Fig. 13. Temporal evolution of perturbed shape. The evolution time is presented on the bottom of each figure, and the evolutions are shown along with the motion by mean curvature.



(a) $t = 0$ (b) $t = 0.0192$ (c) $t = 0.0336$ (d) $t = 0.0432$

Fig. 14. Temporal evolution of torus shape. The evolution time is presented on the bottom of the figures.

$$\phi(x, y, 0) = \tanh\left(\frac{0.5 + 0.4\cos(6\theta) - \sqrt{x^2 + y^2}}{\sqrt{2}\epsilon}\right),$$

where $\theta = \tan^{-1}(y/x)$ for $(x, y) \in (-1, 1) \times (-1, 1)$, $\epsilon = 0.02$. Other parameters are used $N_x = N_y = 200$, $h = 0.01$, and $\Delta t = 0.1h^2$. Fig. 12 shows the evolution of star shape with each time. The shape is changed as following the motion by mean curvature.

We conduct the computational tests for the perturbed and torus shape in 3D. The initial conditions are

$$\phi(x, y, z, 0) = \tanh\left(\frac{0.7 + 0.2\cos(6\theta) - \sqrt{x^2 + 2y^2 + z^2}}{\sqrt{2}\epsilon}\right),$$

where $\theta = \tan^{-1}(z/x)$ for $(x, y, z) \in (-1, 1) \times (-1, 1) \times (-1, 1)$, $\epsilon = 0.03$. $N_x = N_y = N_z = 200$, $h = 0.01$, and $\Delta t = 0.1h^2$ are used. Fig. 13 displays the temporal evolution (b)–(d) of perturbed shape (a).

Fig. 14 displays the temporal evolutions of torus and which the initial condition is given as:

$$\phi(x, y, z, 0) = z^2 + (\sqrt{x^2 + y^2} - r_1)^2 - r_2,$$

where the major radius $r_1 = 0.6$ and the minor radius $r_2 = 0.3$. The computational domain $\Omega = (-1, 1)^3$ with $200 \times 200 \times 200$ mesh, $h = 0.01$, and $\Delta t = 0.1h^2$. The inner circle increases in size and outer circle shrinks because the motion by mean curvature in Fig. 14.

4. Conclusions

We proposed an explicit stable finite difference method for the AC equation by using the ADE method for the diffusion evolution and the closed-form solution for the free-energy evolution of the AC equation. It reduces the severe restriction on time step sizes when using an explicit method. Therefore, the proposed method is useful when applying the AC equation to various practical applications because the method is faster and more stable than an explicit method. Through the numerical experiments, we demonstrated that the proposed method can use larger time step sizes than an explicit method and it preserves the intrinsic properties of the AC equation.

Acknowledgement

The author (Y. Choi) was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF2020R1C1C1A0101153712). The corresponding author (J.S. Kim) was supported by Korea University Grant. The authors are grateful to the reviewers for the constructive and helpful comments on the revision of this article.

References

[1] K. Ahmed, T. Allen, A. El-Azab, Phase field modeling for grain growth in porous solids, J. Mater. Sci. 51 (3) (2016) 1261–1277.
[2] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metall. 27 (6) (1979) 1085–1095.
[3] S.P. Aly, N. Barth, B.W. Figgis, S. Ahzi, A fully transient novel thermal model for in-field photovoltaic modules using developed explicit and implicit finite difference schemes, J. Comput. Sci. 27 (2018) 357–369.
[4] M. Beneš, V. Chalupecký, K. Mikula, Geometrical image segmentation by the Allen–Cahn equation, Appl. Numer. Math. 51 (2–3) (2004) 187–205.
[5] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, Society for Industrial and Applied Mathematics, 2000.
[6] Z. Bučková, P. Pólvora, M. Ehrhardt, M. Günther, Implementation of alternating direction explicit methods for higher dimensional Black–Scholes equations, AIP Conf. Proc. 1773 (1) (2016) 030001.
[7] C. Chen, J. Zhang, X. Yang, Efficient numerical scheme for a new hydrodynamically-coupled conserved Allen–Cahn type Ohta–Kawaski phase-field model for diblock copolymer melt, Comput. Phys. Commun. (2020) 107418.
[8] J.W. Choi, H.G. Lee, D. Jeong, J. Kim, An unconditionally gradient stable numerical method for solving the Allen–Cahn equation, Physica A 338 (9) (2009) 1791–1803.
[9] J.M. Church, Z. Guo, P.K. Jimack, A. Madzvamuse, K. Promislow, B. Wetton, S.M. Wise, F. Yang, High accuracy benchmark problems for Allen–Cahn and Cahn–Hilliard dynamics, Commun. Comput. Phys. 26 (4) (2019) 947–972.
[10] Q. Du, Phase field calculus, curvature-dependent energies, and vesicle membranes, Philos. Mag. 91 (1) (2011) 165–181.
[11] Q. Du, C. Liu, X. Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, J. Comput. Phys. 198 (2) (2004) 450–468.
[12] L.C. Evans, H.M. Soner, P.E. Souganidis, Phase transitions and generalized motion by mean curvature, Commun. Pure Appl. Math. 45 (9) (1992) 1097–1123.
[13] Q. Feng, A class of parallel difference method for solving convection-diffusion equation with variable coefficient, in: Proceedings of the 8th WSEAS International Conference on Applied Computer and Applied Computational Sciences, 2009, pp. 379–382.
[14] X. Feng, A. Prohl, Numerical analysis of the Allen–Cahn equation and approximation for mean curvature flows, Numer. Math. 94 (1) (2003) 33–65.
[15] Z. Guan, J.S. Lowengrub, C. Wang, S.M. Wise, Second order convex splitting schemes for periodic nonlocal Cahn–Hilliard and Allen–Cahn equations, J. Comput. Phys. 277 (2014) 48–71.
[16] D. He, K. Pan, H. Hu, A spatial fourth-order maximum principle preserving operator splitting scheme for the multi-dimensional fractional Allen–Cahn equation, Appl. Numer. Math. 151 (2020) 44–63.
[17] D. Jeong, J. Kim, Conservative Allen–Cahn–Navier–Stokes system for incompressible two-phase fluid flows, Comput. Fluids 156 (2017) 239–246.
[18] D. Jeong, J. Kim, An explicit hybrid finite difference scheme for the Allen–Cahn equation, J. Comput. Appl. Math. 340 (2018) 247–255.
[19] C. Lee, S. Yoon, J. Park, J. Kim, An explicit hybrid method for the nonlocal Allen–Cahn equation, Symmetry 12 (8) (2020) 1218.
[20] D. Lee, The numerical solutions for the energy-dissipative and mass-conservative Allen–Cahn equation, Comput. Math. Appl. 80 (1) (2020) 263–284.
[21] D. Lee, J. Kim, Mean curvature flow by the Allen–Cahn equation, Eur. J. Appl. Math. 26 (4) (2015) 535–559.
[22] H.G. Lee, J. Yang, J. Park, J. Kim, Effect of space dimensions on equilibrium solutions of Cahn–Hilliard and conservative Allen–Cahn equations, Numer. Math. Theor. Methods Appl. 13 (3) (2020) 644–664.
[23] C. Li, Y. Huang, N. Yi, An unconditionally energy stable second order finite element method for solving the Allen–Cahn equation, J. Comput. Appl. Math. 353 (2019) 38–48.
[24] H. Li, Z. Song, J. Hu, Numerical analysis of a second-order IPDGFE method for the Allen–Cahn equation and the curvature-driven geometric flow, Comput. Math. Appl. 86 (2021) 49–62.
[25] Y. Li, H.G. Lee, D. Jeong, J. Kim, An unconditionally stable hybrid numerical method for solving the Allen–Cahn equation, Comput. Math. Appl. 60 (6) (2010) 1591–1606.
[26] Y. Li, D. Jeong, J.-I. Choi, S. Lee, J. Kim, Fast local image inpainting based on the Allen–Cahn model, Digit. Signal Process. 37 (2015) 65–74.
[27] Y. Li, D. Jeong, H. Kim, C. Lee, J. Kim, Comparison study on the different dynamics between the Allen–Cahn and the Cahn–Hilliard equations, Comput. Math. Appl. 77 (2) (2019) 311–322.
[28] Y. Li, S. Lan, X. Liu, B. Lu, L. Wang, An efficient volume repairing method by using a modified Allen–Cahn equation, Pattern Recognit. 107 (2020) 107478.
[29] X. Luo, A.L. Bertozzi, Convergence of the graph Allen–Cahn scheme, J. Stat. Phys. 167 (3–4) (2017) 934–958.
[30] L. Ma, R. Chen, X. Yang, H. Zhang, Numerical approximations for Allen–Cahn type phase field model of two-phase incompressible fluids with moving contact lines, Commun. Comput. Phys. 21 (3) (2017) 867–889.
[31] S. Pourghanbar, J. Manafian, M. Ranjbar, A. Aliyeva, Y.S. Gasimov, An efficient alternating direction explicit method for solving a nonlinear partial differential equation, Math. Probl. Eng. 2020 (2020) 9647416.
[32] S.H. Prassetyo, M. Gutierrez, High-order ADE scheme for solving the fluid diffusion equation in non-uniform grids and its application in coupled hydro-mechanical simulation, Int. J. Numer. Anal. Methods Geomech. 42 (16) (2018) 1976–2000.
[33] V.K. Saul'yev, Integration of Equations of Parabolic Type Equation by the Method of Net, Pergamon Press, New York, 1964.

[34] Y. Sun, X. Xiao, Z. Gao, X. Feng, An efficient space-time operator-splitting method for high-dimensional vector-valued Allen–Cahn equations, Int. J. Numer. Methods Heat Fluid Flow 29 (9) (2019) 3437–3453.
[35] N. Takada, J. Matsumoto, S. Matsumoto, K. Kurihara, Phase-field model-based simulation of two-phase fluid motion on partially wetted and textured solid surface, J. Comput. Sci. 17 (2016) 315–324.
[36] T. Tang, J. Yang, Implicit-explicit scheme for the Allen–Cahn equation preserves the maximum principle, J. Comput. Math. 34 (5) (2016) 451–461.
[37] R. Tavakoli, P. Davami, 2D parallel and stable group explicit finite difference method for solution of diffusion equation, Appl. Math. Comput. 188 (2) (2007) 1184–1192.
[38] D. Wang, Q. Du, L. Meng, H. Jia, Fast algorithm based on TT-M FE method for Allen–Cahn equation, East Asian J. Appl. Math. 10 (2) (2020) 316–337.
[39] J. Wang, Y. Li, Y. Choi, C. Lee, J. Kim, Fast and accurate smoothing method using a modified Allen–Cahn equation, Comput. Aided Des. 120 (2020) 102804.
[40] J. Yang, C. Lee, S. Kwak, Y. Choi, J. Kim, A conservative and stable explicit finite difference scheme for the diffusion equation, J. Comput. Sci. 56 (2021) 101491.
[41] J. Yang, Y. Li, C. Lee, H.G. Lee, S. Kwak, Y. Hwang, et al., An explicit conservative Saul'yev scheme for the Cahn–Hilliard equation, Int. J. Mech. Sci. 217 (2022) 106985.
[42] J. Zhang, C. Chen, X. Yang, A novel decoupled and stable scheme for an anisotropic phase-field dendritic crystal growth model, Appl. Math. Lett. 95 (2019) 122–129.