

A space-adaptive, unconditionally maximum bound preserving scheme for the 3D Allen–Cahn equation

Soobin Kwak , Junseok Kim *

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

ARTICLE INFO

Keywords:

Space-adaptive method
3D Allen–Cahn equation
Operator splitting method

ABSTRACT

We present a space-adaptive, unconditionally maximum bound preserving algorithm for the three-dimensional (3D) Allen–Cahn (AC) equation. The proposed method uses an operator splitting scheme with direction-wise implicit solvers and locally refined nonuniform meshes. This strategy guarantees numerical stability even for large time steps and preserves the maximum bound throughout the simulation. Spatial adaptivity is used to concentrate computational resources in interfacial regions, where the phase field exhibits sharp transitions. This method maintains high resolution of interface dynamics while reducing the computational cost. The nonlinear reaction and linear diffusion components of the AC equation are treated separately; the nonlinear reaction term is solved using an analytic solution, while the linear diffusion term is handled via a directionally split finite difference method (FDM) with tridiagonal solvers for computational efficiency. To further improve computational performance, the mesh is adaptively updated at fixed time intervals based on the phase field gradient. In this method, we use fine grids near the diffuse interface and coarse grids in the other regions. Numerical tests, such as curvature-driven motion and droplet deformation, show that the scheme can accurately describe complex interface behavior in 3D space. The numerical results agree well with theoretical predictions and confirm that the method is stable, efficient, and suitable for large-scale phase-field simulations. This study provides a reliable algorithm for 3D simulations of interfacial dynamics based on the AC model and suggests directions for further development of adaptive numerical methods.

1. Introduction

Interfacial dynamics, such as phase separation and droplet evolution, are important in materials science [1], chemical engineering, and biology [2]. The Allen–Cahn (AC) equation is a widely used phase-field model for simulating interface motion [3,4]. This nonlinear parabolic equation describes the time evolution of a phase-field variable that smoothly connects two phases without explicit interface tracking [5,6]. Many numerical methods exist for two-dimensional (2D) problems; however, three-dimensional (3D) simulations remain computationally expensive because the number of grid points increases greatly [7]. Stable and accurate 3D simulations must also control computational cost [8]. Conventional numerical methods often face difficulties when large time steps are used [9,10]. Recent studies have shown that unconditional stability, energy dissipation, and maximum bound preservation are important for phase-field simulations. Yang et al. [11] proposed high-order schemes with a cut-off step to enforce the maximum bound. Ju et al.

* Corresponding author.

E-mail addresses: soobin23@korea.ac.kr (S. Kwak), cfdkim@korea.ac.kr (J. Kim).

URL: <https://mathematicians.korea.ac.kr/cfdkim> (J. Kim)

<https://doi.org/10.1016/j.cnsns.2026.110842>

Received 22 May 2026; Received in revised form 12 August 2026; Accepted 14 September 2026

Available online 16 September 2026

1007-5704/© 2026 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

[12] developed stabilized SAV schemes that preserve energy dissipation and preserve the maximum bound principle. Li et al. [13] introduced a second-order semi-Lagrangian exponential time differencing scheme.

Mesh adaptation is important for the AC model because it resolves sharp interfaces and reduces computational cost [14–16]. A uniform mesh requires very fine grids over the whole domain, while an adaptive mesh refines only near interfaces and uses coarse grids elsewhere. This method improves accuracy and efficiency, especially in large-scale or 3D simulations. Wang et al. [15] applied adaptive mesh refinement with a lattice Boltzmann method and achieved accurate interface tracking with lower cost. Li et al. [16] developed a polygonal adaptive method that captures thin interfaces and preserves energy dissipation. Shah et al. [17] showed that adaptive finite element methods improve accuracy for steady and time-dependent problems. Zhang et al. [18] proposed a structure-preserving BDF2 scheme that ensures energy stability, second-order accuracy, and the maximum bound principle. These results demonstrate the effectiveness of adaptive methods for the AC equation. In addition, mesh refinement studies [19,20] have been extensively studied in a broader range of contexts beyond the computational solution of the AC model.

The primary contributions and possible applications of this study are summarized as follows. We develop a 3D adaptive finite difference method (FDM) [21] that guarantees unconditional stability and preserves physical bounds of the solution. The proposed method efficiently resolves interfacial dynamics through local mesh refinement, which reduces computational cost while maintaining accuracy. It demonstrates stable and reliable performance for complex morphological changes, including curvature-driven motion and droplet breakup. The scheme is scalable and suitable for large-scale 3D simulations. Furthermore, the method can be applied to interfacial problems in materials science, fluid dynamics, chemical engineering, and biology. In addition, the proposed method can be extended to mass-conserving models [22].

The novelty of this work lies partly in dealing with a fundamental challenge that is specific to 3D AC dynamics. Unlike the two-dimensional case, where embedded closed interfaces become convex and eventually contract to a point, the three-dimensional evolution of general closed surfaces can involve singularities and topological transitions. In particular, a nonconvex 3D interface, such as a dumbbell-shaped interface, can develop a thin neck and eventually break into separate parts. Therefore, a 3D numerical algorithm is needed to accurately capture such complex interface dynamics. As a result, the numerical strategy used in two dimensions cannot be transferred directly to three dimensions without modification. This observation justifies the introduction of a space-adaptive algorithm, which enables efficient and accurate resolution of thin interfaces, localized curvature concentration, and complex geometric transitions while preserving the unconditional maximum bound preserving property. In addition, the mesh size is more important in 3D computations because a small decrease in the mesh size greatly increases the total number of grid points. Therefore, an efficient adaptive mesh strategy is especially important for 3D simulations. Furthermore, we extend the proposed algorithm to the AC model with a time-dependent interfacial parameter. The interfacial parameter and the finest mesh size are adjusted according to the current size of the interface. This extension allows the proposed method to be applied to a more general AC model and to maintain sufficient spatial resolution as the interface becomes smaller.

The rest of this paper is organized as follows. Section 2 describes the numerical algorithm based on operator splitting and local mesh refinement for the 3D AC equation. Section 3 presents computational experiments that verify the accuracy and efficiency of the proposed method through simulations of curvature-driven motion. Section 4 explains possible extensions of the adaptive FDM to domains with complex geometries. Section 5 provides a summary of the main results.

2. Numerical solution algorithm

We now formulate the 3D AC equation, which serves as the basis for the proposed adaptive computational algorithm. The governing equation is defined on $\Omega = (L_x, R_x) \times (L_y, R_y) \times (L_z, R_z)$ as

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} + \phi_{xx}(\mathbf{x}, t) + \phi_{yy}(\mathbf{x}, t) + \phi_{zz}(\mathbf{x}, t), \quad (1)$$

$$\mathbf{x} = (x, y, z) \in \Omega, \quad t > 0,$$

where $\phi(\mathbf{x}, t)$ is the order parameter, ϵ is positive, and $F(\phi) = 0.25(\phi^2 - 1)^2$. We use the zero Neumann boundary condition on $\partial\Omega$: $\mathbf{n} \cdot \nabla \phi(\mathbf{x}, t) = 0$, where $\mathbf{n}$ is the normal vector.

The detailed computational solution procedure is developed within an unconditionally maximum bound preserving and adaptive FDM that incorporates operator splitting and spatial mesh refinement. For a detailed account of the two-dimensional formulation and corresponding numerical implementation of the method discussed in this study, the reader is referred to [23], where the method is developed and analyzed in depth. To implement the operator splitting strategy, the 3D AC equation is decomposed into four sub-equations: a nonlinear reaction term (2), and linear diffusion terms in the x -, y -, and z -directions given by Eqs. (3)–(5), respectively:

$$\frac{\partial \phi(x, y, z, t)}{\partial t} = -\frac{F'(\phi(x, y, z, t))}{\epsilon^2}, \quad (2)$$

$$\frac{\partial \phi(x, y, z, t)}{\partial t} = \phi_{xx}(x, y, z, t), \quad (3)$$

$$\frac{\partial \phi(x, y, z, t)}{\partial t} = \phi_{yy}(x, y, z, t), \quad (4)$$

$$\frac{\partial \phi(x, y, z, t)}{\partial t} = \phi_{zz}(x, y, z, t). \quad (5)$$

Let $\Omega_d^n = \{(x_i^n, y_j^n, z_k^n) | 1 \leq i \leq N_x^n, 1 \leq j \leq N_y^n, 1 \leq k \leq N_z^n\}$ denote the time-dependent nonuniform discrete domain at time $t = n\Delta t$, where Δt represents the temporal step size. At each grid point (x_i^n, y_j^n, z_k^n) and time level n , the value of the phase-field variable is

denoted by $\phi_{ijk}^n = \phi(x_i^n, y_j^n, z_k^n, n\Delta t)$. Then, the closed-form solution after time Δt of Eq. (2) is given by

$$\phi_{ijk}^{n+1,1} = \frac{\phi_{ijk}^n}{\sqrt{(\phi_{ijk}^n)^2 + \left[1 - (\phi_{ijk}^n)^2\right] e^{\frac{-2\Delta t}{\epsilon^2}}}}, \quad \text{for all } i, j, k. \quad (6)$$

Here, $\phi^{n+1,1}$ represents the intermediate solution obtained by solving the nonlinear reaction Eq. (2) with the initial condition ϕ^n . The spatial mesh sizes are given by $\Delta x_i^n = x_{i+1}^n - x_i^n$ for $i = 1, \dots, N_x^n$, $\Delta y_j^n = y_{j+1}^n - y_j^n$ for $j = 1, \dots, N_y^n$, $\Delta z_k^n = z_{k+1}^n - z_k^n$ for $k = 1, \dots, N_z^n$. To impose the zero Neumann boundary conditions on the discrete cell-centered domain, we use symmetric boundary mesh intervals given by $\Delta x_0^n = \Delta x_1^n$, $\Delta x_{N_x^n}^n = \Delta x_{N_x^n-1}^n$, $\Delta y_0^n = \Delta y_1^n$, $\Delta y_{N_y^n}^n = \Delta y_{N_y^n-1}^n$, $\Delta z_0^n = \Delta z_1^n$, and $\Delta z_{N_z^n}^n = \Delta z_{N_z^n-1}^n$ together with the corresponding ghost point values. Here, x_0^n , $x_{N_x^n+1}^n$, y_0^n , $y_{N_y^n+1}^n$, z_0^n , and $z_{N_z^n+1}^n$ denote ghost points. In the subsequent steps, the operator splitting technique is applied sequentially along each spatial axis to update the numerical solution using an FDM. We denote by $\phi^{n+1,2}$, $\phi^{n+1,3}$, and ϕ^{n+1} the intermediate solutions obtained by solving Eqs. (3)–(5), respectively. For fixed indices j and k , Eq. (3) is solved implicitly for $i = 1, \dots, N_x^n$:

$$\frac{\phi_{ijk}^{n+1,2} - \phi_{ijk}^{n+1,1}}{\Delta t} = \frac{1}{\Delta x_{i-\frac{1}{2}}^n} \left(\frac{\phi_{i+1,jk}^{n+1,2} - \phi_{ijk}^{n+1,2}}{\Delta x_i^n} - \frac{\phi_{ijk}^{n+1,2} - \phi_{i-1,jk}^{n+1,2}}{\Delta x_{i-1}^n} \right), \quad (7)$$

where $\Delta x_{i-\frac{1}{2}}^n = (\Delta x_{i-1}^n + \Delta x_i^n)/2$. Eq. (7) can be rewritten as follows:

$$-\frac{1}{\Delta x_{i-1}^n \Delta x_{i-\frac{1}{2}}^n} \phi_{i-1,jk}^{n+1,2} + \left(\frac{1}{\Delta t} + \frac{2}{\Delta x_i^n \Delta x_{i-\frac{1}{2}}^n} \right) \phi_{ijk}^{n+1,2} - \frac{1}{\Delta x_i^n \Delta x_{i-\frac{1}{2}}^n} \phi_{i+1,jk}^{n+1,2} = \frac{\phi_{ijk}^{n+1,1}}{\Delta t}, \quad (8)$$

for $i = 1, \dots, N_x^n$, which can be solved using an efficient Thomas algorithm [24]. Here, the homogeneous Neumann boundary condition is applied:

$$\phi_{0jk}^{n+1,2} = \phi_{1jk}^{n+1,2}, \quad \phi_{N_x^n+1,jk}^{n+1,2} = \phi_{N_x^n,jk}^{n+1,2}, \quad \text{for all } j = 1, \dots, N_y^n \text{ and } k = 1, \dots, N_z^n. \quad (9)$$

After completing the procedure along the x -axis, the computation proceeds independently in the y -direction. The update is based on Eq. (4). In this step, the indices i and k are fixed, and the update is performed sequentially for each j ranging from 1 to N_y^n .

$$\frac{\phi_{ijk}^{n+1,3} - \phi_{ijk}^{n+1,2}}{\Delta t} = \frac{1}{\Delta y_{j-\frac{1}{2}}^n} \left(\frac{\phi_{i,j+1,k}^{n+1,3} - \phi_{ijk}^{n+1,3}}{\Delta y_j^n} - \frac{\phi_{ijk}^{n+1,3} - \phi_{i,j-1,k}^{n+1,3}}{\Delta y_{j-1}^n} \right), \quad (10)$$

where $\Delta y_{j-\frac{1}{2}}^n = (\Delta y_{j-1}^n + \Delta y_j^n)/2$. Eq. (10) admits the following equivalent formulation:

$$-\frac{1}{\Delta y_{j-1}^n \Delta y_{j-\frac{1}{2}}^n} \phi_{i,j-1,k}^{n+1,3} + \left(\frac{1}{\Delta t} + \frac{2}{\Delta y_j^n \Delta y_{j-\frac{1}{2}}^n} \right) \phi_{ijk}^{n+1,3} - \frac{1}{\Delta y_j^n \Delta y_{j-\frac{1}{2}}^n} \phi_{i,j+1,k}^{n+1,3} = \frac{\phi_{ijk}^{n+1,2}}{\Delta t}, \quad (11)$$

for $j = 1, \dots, N_y^n$, which can be solved using an efficient Thomas algorithm. Here, the zero Neumann boundary condition is imposed:

$$\phi_{i0k}^{n+1,3} = \phi_{i1k}^{n+1,3}, \quad \phi_{i,N_y^n+1,k}^{n+1,3} = \phi_{i,N_y^n,k}^{n+1,3}, \quad \text{for all } i = 1, \dots, N_x^n \text{ and } k = 1, \dots, N_z^n. \quad (12)$$

Finally, using Eq. (5), ϕ^{n+1} is obtained by updating along the z -direction, while holding i and j constant. For $k = 1, \dots, N_z^n$,

$$\frac{\phi_{ijk}^{n+1} - \phi_{ijk}^{n+1,3}}{\Delta t} = \frac{1}{\Delta z_{k-\frac{1}{2}}^n} \left(\frac{\phi_{ij,k+1}^{n+1} - \phi_{ijk}^{n+1}}{\Delta z_k^n} - \frac{\phi_{ijk}^{n+1} - \phi_{ij,k-1}^{n+1}}{\Delta z_{k-1}^n} \right), \quad (13)$$

where $\Delta z_{k-\frac{1}{2}}^n = (\Delta z_{k-1}^n + \Delta z_k^n)/2$. Eq. (13) can be rewritten as follows:

$$-\frac{1}{\Delta z_{k-1}^n \Delta z_{k-\frac{1}{2}}^n} \phi_{ij,k-1}^{n+1} + \left(\frac{1}{\Delta t} + \frac{2}{\Delta z_k^n \Delta z_{k-\frac{1}{2}}^n} \right) \phi_{ijk}^{n+1} - \frac{1}{\Delta z_k^n \Delta z_{k-\frac{1}{2}}^n} \phi_{ij,k+1}^{n+1} = \frac{\phi_{ijk}^{n+1,3}}{\Delta t}, \quad (14)$$

$k = 1, \dots, N_z^n$, which can be solved using an efficient Thomas algorithm. Here, the zero Neumann boundary condition is imposed:

$$\phi_{ij0}^{n+1} = \phi_{ij1}^{n+1}, \quad \phi_{ij,N_z^n+1}^{n+1} = \phi_{ij,N_z^n}^{n+1}, \quad \text{for all } i = 1, \dots, N_x^n \text{ and } j = 1, \dots, N_y^n. \quad (15)$$

To summarize, the numerical scheme effectively solves the following equation on the domain Ω_h^n :

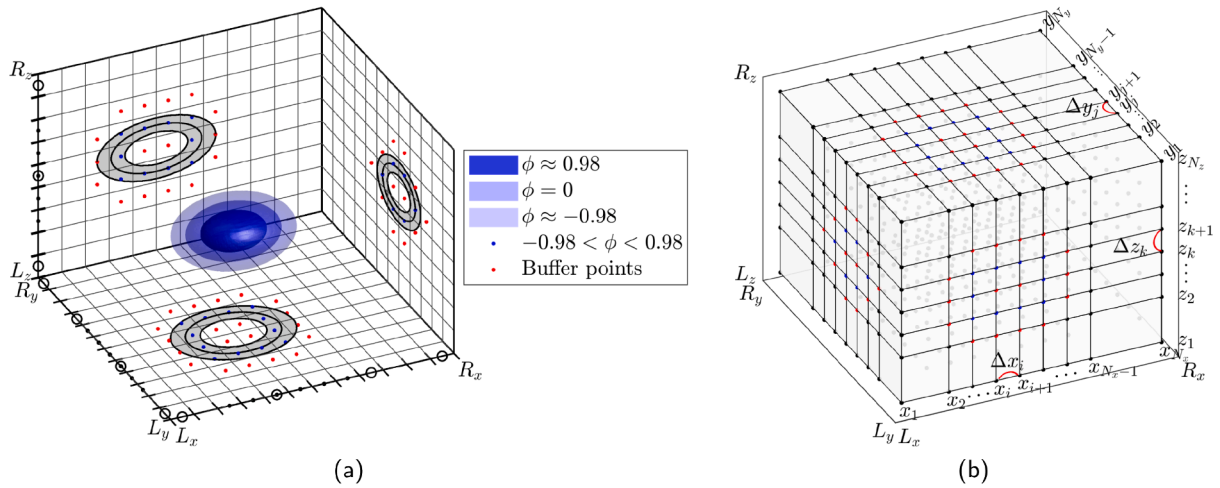


Fig. 1. Diagrammatic representation of the AMR strategy. (a) Identification of the diffuse interface region $|\phi| < 0.98$ and buffer points in a fine uniform mesh. (b) Refined mesh.

$$\begin{aligned} \frac{\phi_{ijk}^{n+1} - \phi_{ijk}^{n+1,1}}{\Delta t} &= \frac{1}{\Delta x_{i-\frac{1}{2}}^n} \left(\frac{\phi_{i+1,j,k}^{n+1,2} - \phi_{ijk}^{n+1,2}}{\Delta x_i^n} - \frac{\phi_{ijk}^{n+1,2} - \phi_{i-1,j,k}^{n+1,2}}{\Delta x_{i-1}^n} \right) + \frac{1}{\Delta y_{j-\frac{1}{2}}^n} \left(\frac{\phi_{i,j+1,k}^{n+1,3} - \phi_{ijk}^{n+1,3}}{\Delta y_j^n} - \frac{\phi_{ijk}^{n+1,3} - \phi_{i,j-1,k}^{n+1,3}}{\Delta y_{j-1}^n} \right) \\ &\quad + \frac{1}{\Delta z_{k-\frac{1}{2}}^n} \left(\frac{\phi_{i,j,k+1}^{n+1} - \phi_{ijk}^{n+1}}{\Delta z_k^n} - \frac{\phi_{ijk}^{n+1} - \phi_{i,j,k-1}^{n+1}}{\Delta z_{k-1}^n} \right), \end{aligned}$$

where $\phi_{ijk}^{n+1,1} = \phi_{ijk}^n / \sqrt{(\phi_{ijk}^n)^2 + [1 - (\phi_{ijk}^n)^2]e^{-2\Delta t/\epsilon^2}}$.

We present a strategy for constructing an adaptive mesh tailored to efficiently resolve interfacial dynamics. The method begins with the specification of a uniform reference grid with spatial resolution h . Fig. 1(a) illustrates the spatial discretization, where the uniform fine grid spacing h is indicated along each of the x -, y -, and z -axes. The potential mesh nodes are located at the center points of the corresponding grid intervals. Mesh refinement is guided by the diffuse interface region, shown in gray, which is defined by the condition $|\phi(x, y, z, t)| < 0.98$ for the phase-field variable. As the phase-field variable shows minimal variation outside the transition region, computational resources are concentrated within the interfacial zone. Blue dots correspond to nodes located in the interfacial zone, while red dots denote buffer nodes surrounding it. The identified nodes are subsequently mapped onto each of the coordinate directions. The adaptive mesh is ultimately generated by merging the projected points with a coarse base grid, marked by black circles, which includes the endpoints of the computational domain as essential nodes. The values of the phase-field variable on the newly generated mesh are computed using linear interpolation based on the previous grid configuration. Fig. 1(b) illustrates the resulting spatial mesh used for solving the AC equation.

For clarity, the complete space-adaptive solution procedure is summarized in Algorithm 1.

The proposed adaptive numerical method is constructed to unconditionally preserve the maximum bound. Consequently, the numerical solution remains bounded within the physically admissible range for any choice of time step size. Specifically, if $\|\phi^n\|_\infty \leq 1$, then the inequality $\|\phi^{n+1}\|_\infty \leq 1$ also holds. We now present the proof of the unconditional maximum bound preserving property. The discrete ℓ^∞ -norm is defined as $\|\phi^n\|_\infty = \max_{i,j,k} |\phi_{ijk}^n|$.

Lemma 1. Let $\mathcal{I}^{n \rightarrow \tilde{n}}$ denote the trilinear interpolation operator that transfers a grid function from the old mesh to the reconstructed mesh. Then,

$$\|\mathcal{I}^{n \rightarrow \tilde{n}} v\|_\infty \leq \|v\|_\infty.$$

In particular, if $-1 \leq v_{ijk} \leq 1$ at all nodes of the old mesh, then the interpolated values on the reconstructed mesh also belong to $[-1, 1]$.

Proof. Consider a new grid point $(\tilde{x}_i, \tilde{y}_j, \tilde{z}_k) \in [x_p, x_{p+1}] \times [y_q, y_{q+1}] \times [z_r, z_{r+1}]$. Define

$$\xi = \frac{\tilde{x}_i - x_p}{x_{p+1} - x_p}, \quad \eta = \frac{\tilde{y}_j - y_q}{y_{q+1} - y_q}, \quad \zeta = \frac{\tilde{z}_k - z_r}{z_{r+1} - z_r},$$

where $0 \leq \xi, \eta, \zeta \leq 1$. Set $w_0^x = 1 - \xi$, $w_1^x = \xi$, $w_0^y = 1 - \eta$, $w_1^y = \eta$, $w_0^z = 1 - \zeta$, and $w_1^z = \zeta$. The trilinearly interpolated value is

$$\tilde{v}_{ijk} = \sum_{\alpha, \beta, \gamma=0}^1 w_\alpha^x w_\beta^y w_\gamma^z v_{p+\alpha, q+\beta, r+\gamma}.$$

Algorithm 1 Space-adaptive algorithm for the 3D Allen–Cahn equation.

Require: Fine mesh size h , time step Δt , remeshing interval N_r , interface threshold η , buffer width N_b , and coarse grid stride s .

- 1: Construct the uniform fine reference grids in the x -, y -, and z -directions with mesh size h .
- 2: Construct the coarse base grids by selecting every s th point from each fine reference grid.
- 3: Set the initial phase field ϕ^0 on the fine reference mesh.
- 4: Identify the interfacial points that satisfy $|\phi_{ijk}^0| < \eta$.
- 5: Project the interfacial points onto the three coordinate directions and add N_b buffer points on both sides.
- 6: Construct the initial adaptive mesh by combining the buffered points with the coarse base grid points.
- 7: Transfer ϕ^0 to the initial adaptive mesh and construct the tridiagonal systems for the three directional diffusion steps.
- 8: **for** $n = 0, 1, \dots, N_t - 1$ **do**
- 9: **if** $n > 0$ and $\text{mod}(n + 1, N_r) = 0$ **then**
- 10: Identify the current interfacial points that satisfy $|\phi_{ijk}^n| < \eta$.
- 11: Map the identified points to the fine reference grids, project them onto the three coordinate directions, and add N_b buffer points on both sides.
- 12: Construct the new adaptive mesh by combining the buffered points with the coarse base-grid points.
- 13: Transfer ϕ^n to the new mesh by trilinear interpolation.
- 14: Recompute the mesh widths and tridiagonal coefficients.
- 15: **end if**
- 16: Compute the nonlinear reaction step using Eq. (6).
- 17: Solve the x -direction diffusion step using Eq. (8).
- 18: Solve the y -direction diffusion step using Eq. (11).
- 19: Solve the z -direction diffusion step using Eq. (14).
- 20: **end for**

All interpolation weights are nonnegative, and they satisfy $\sum_{\alpha, \beta, \gamma=0}^1 w_\alpha^x w_\beta^y w_\gamma^z = 1$. Therefore,

$$|\bar{v}_{ijk}| \leq \sum_{\alpha, \beta, \gamma=0}^1 w_\alpha^x w_\beta^y w_\gamma^z |v_{p+\alpha, q+\beta, r+\gamma}| \leq \|v\|_\infty \sum_{\alpha, \beta, \gamma=0}^1 w_\alpha^x w_\beta^y w_\gamma^z = \|v\|_\infty.$$

The maximum over all new grid points gives

$$\|\bar{v}_{ijk}\|_\infty \leq \|v\|_\infty.$$

Hence, trilinear interpolation preserves the interval $[-1, 1]$. $\square$

Theorem 1. Assume that the numerical solution at time level n satisfies the boundedness condition $\|\phi^n\|_\infty \leq 1$. Then, the numerical solution at the subsequent time level also preserves this property, namely, $\|\phi^{n+1}\|_\infty \leq 1$.

Proof. We first note that the nonlinear reaction Eq. (2) admits the closed-form solution (6), which satisfies

$$|\phi_{ijk}^{n+1,1}| = \frac{|\phi_{ijk}^n|}{\sqrt{(\phi_{ijk}^n)^2 + \left[1 - (\phi_{ijk}^n)^2\right] e^{\frac{-2\Delta t}{\epsilon^2}}}} \leq \frac{|\phi_{ijk}^n|}{\sqrt{(\phi_{ijk}^n)^2}} = 1, \quad (16)$$

where the assumption $\|\phi^n\|_\infty \leq 1$ has been used. If $\phi_{ijk}^n = 0$, then Eq. (6) directly gives $\phi_{ijk}^{n+1,1} = 0$, and the bound holds trivially. Therefore, it follows that $\|\phi^{n+1,1}\|_\infty \leq 1$.

Next, we consider the diffusion Eqs. (3)–(5). Each of these subproblems is discretized implicitly, as shown in Eqs. (8), (11) and (14). Eq. (8) can be defined in matrix form as

$$M_x \phi_{jk}^{n+1,2} = \phi_{jk}^{n+1,1},$$

where $\phi_{jk}^n = (\phi_{1jk}^n, \phi_{2jk}^n, \dots, \phi_{N_x^n-1,jk}^n, \phi_{N_x^n,jk}^n)^T$, and

$$M_x = \begin{pmatrix} \frac{1}{\Delta t} + \frac{1}{\Delta x_0^n \Delta x_{\frac{1}{2}}^n} & -\frac{1}{\Delta x_{\frac{1}{2}}^n \Delta x_1^n} & 0 & \dots & 0 \\ -\frac{1}{\Delta x_1^n \Delta x_{\frac{1}{2}}^n} & \frac{1}{\Delta t} + \frac{2}{\Delta x_1^n \Delta x_2^n} & -\frac{1}{\Delta x_{\frac{1}{2}}^n \Delta x_2^n} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\frac{1}{\Delta x_{N_x^n-2}^n \Delta x_{N_x^n-\frac{3}{2}}^n} & \frac{1}{\Delta t} + \frac{2}{\Delta x_{N_x^n-2}^n \Delta x_{N_x^n-1}^n} & -\frac{1}{\Delta x_{N_x^n-\frac{3}{2}}^n \Delta x_{N_x^n-1}^n} \\ 0 & \dots & 0 & -\frac{1}{\Delta x_{N_x^n-1}^n \Delta x_{N_x^n-\frac{1}{2}}^n} & \frac{1}{\Delta t} + \frac{1}{\Delta x_{N_x^n-1}^n \Delta x_{N_x^n-\frac{1}{2}}^n} \end{pmatrix}.$$

The matrix M_x is strictly diagonally dominant because, in every row, the absolute value of the diagonal entry exceeds the sum of the absolute values of the off-diagonal entries. According to the Lévy–Desplanques theorem [25], any strictly diagonally dominant matrix is nonsingular. Therefore, M_x is invertible for any choice of positive mesh spacings Δx_i^n and time step size $\Delta t > 0$, which guarantees the existence and uniqueness of the solution to the linear system. Eq. (8) can be rewritten as

$$\left(\frac{1}{\Delta t} + \frac{2}{\Delta x_{i-1}^n \Delta x_i^n}\right) \phi_{ijk}^{n+1,2} = \frac{1}{\Delta t} \phi_{ijk}^{n+1,1} + \frac{1}{\Delta x_{i-1}^n \Delta x_{i-\frac{1}{2}}^n} \phi_{i-1,jk}^{n+1,2} + \frac{1}{\Delta x_{i-\frac{1}{2}}^n \Delta x_i^n} \phi_{i+1,jk}^{n+1,2}.$$

If we take absolute values and use the triangle inequality, we obtain

$$\begin{aligned} \left(\frac{1}{\Delta t} + \frac{2}{\Delta x_{i-1}^n \Delta x_i^n}\right) |\phi_{ijk}^{n+1,2}| &= \left| \frac{1}{\Delta t} \phi_{ijk}^{n+1,1} + \frac{1}{\Delta x_{i-1}^n \Delta x_{i-\frac{1}{2}}^n} \phi_{i-1,jk}^{n+1,2} + \frac{1}{\Delta x_{i-\frac{1}{2}}^n \Delta x_i^n} \phi_{i+1,jk}^{n+1,2} \right| \\ &\leq \frac{1}{\Delta t} |\phi_{ijk}^{n+1,1}| + \frac{1}{\Delta x_{i-1}^n \Delta x_{i-\frac{1}{2}}^n} |\phi_{i-1,jk}^{n+1,2}| + \frac{1}{\Delta x_{i-\frac{1}{2}}^n \Delta x_i^n} |\phi_{i+1,jk}^{n+1,2}| \\ &\leq \frac{1}{\Delta t} \|\phi^{n+1,1}\|_\infty + \frac{\Delta x_{i-1}^n + \Delta x_i^n}{\Delta x_{i-1}^n \Delta x_{i-\frac{1}{2}}^n \Delta x_i^n} \|\phi^{n+1,2}\|_\infty. \end{aligned}$$

By taking the maximum over all indices, we obtain

$$\|\phi^{n+1,2}\|_\infty \leq \|\phi^{n+1,1}\|_\infty.$$

By applying the same argument to the y -direction and z -direction diffusion steps given in Eqs. (11) and (14), we find that

$$\|\phi^{n+1,3}\|_\infty \leq \|\phi^{n+1,2}\|_\infty, \text{ and } \|\phi^{n+1}\|_\infty \leq \|\phi^{n+1,3}\|_\infty.$$

Consequently, each implicit diffusion step in any spatial direction does not increase the discrete ℓ^∞ -norm of the numerical solution.

At a mesh-update time, let $\tilde{\phi}^{n+1}$ denote the numerical solution transferred from the current mesh to the reconstructed mesh by trilinear interpolation. Lemma 1 gives

$$\|\tilde{\phi}^{n+1}\|_\infty \leq \|\phi^{n+1}\|_\infty \leq 1.$$

If no mesh update is performed, we simply set $\tilde{\phi}^{n+1} = \phi^{n+1}$. Therefore, the complete adaptive procedure, which consists of the reaction step, the three direction-wise diffusion steps, and the mesh interpolation step, preserves the maximum bound without any restriction on Δt . After the mesh redistribution, we relabel $\tilde{\phi}^{n+1}$ as ϕ^{n+1} . $\square$

Comprehensive details in the context of a two-dimensional case can be found in [23].

Although the present scheme is first-order accurate in time, the algorithm can be extended to have second-order temporal accuracy. One possible method is to use a symmetric second-order splitting for the reaction and diffusion operators and apply the Douglas–Gunn ADI method [26] to the diffusion step. The Douglas–Gunn method has been successfully applied to the 3D AC equation with second-order temporal accuracy [27]. Such an extension would retain the direction-wise computational structure of the present method, although a separate analysis would be needed to verify the maximum bound preserving property.

3. Computational tests

We perform numerical tests on the 3D AC equation to test the fidelity and efficiency of the developed adaptive method. In the simulations, the interfacial thickness parameter is chosen as $\epsilon_m = mh/[2\sqrt{2}\tanh^{-1}(0.9)]$, where m is a positive integer. Here, h represents the minimum of the spatial step sizes Δx , Δy , and Δz in the initial computational mesh Ω_h^0 . Unless otherwise stated, the zero Neumann boundary condition (Eqs. (9), (12), and (15)) is used as the boundary condition.

3.1. Numerical convergence

Numerical experiments are performed using a traveling wave solution of the AC equation in order to quantitatively evaluate the convergence rate of the proposed algorithm. This test allows for quantitative evaluation of the spatial and temporal discretization errors by comparing the numerical and exact solutions. Temporal and spatial convergence tests are conducted separately so that the accuracy of each discretization can be examined. The following traveling wave solution is considered

$$\Phi(x, y, z, t) = \frac{1}{2} \left(1 + \tanh \left(\frac{1-x-st}{2\sqrt{2}\epsilon} \right) \right),$$

where $s = 3/(\sqrt{2}\epsilon)$ denotes the speed of the traveling wave. The computational domain is defined as $\Omega = (0, 1.5) \times (0, 0.25) \times (0, 0.25)$. The initial condition is obtained from the exact solution at $t = 0$:

$$\phi(x, y, z, 0) = \frac{1}{2} \left(1 + \tanh \left(\frac{1-x}{2\sqrt{2}\epsilon} \right) \right),$$

Table 1Temporal convergence errors and orders at final time T .

Δt	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}
$\ e^n\ _\infty$	1.245703e-2	6.270336e-3	3.150358e-3	1.583393e-3	7.981230e-4
order		0.99	0.99	0.99	0.99
$\ e^n\ _2$	1.142986e-3	5.718999e-4	2.864005e-4	1.436533e-4	7.228644e-5
order		1.00	1.00	1.00	0.99

Table 2Spatial convergence errors and orders at final time T .

h	1/8	1/16	1/32	1/64	1/128
$\ e^n\ _\infty$	3.070938e-2	8.128632e-3	2.496995e-3	6.628985e-4	1.703634e-4
order		1.92	1.70	1.91	1.96
$\ e^n\ _2$	3.029900e-3	7.792141e-4	2.003330e-4	5.096973e-5	1.319676e-5
order		1.96	1.96	1.97	1.95

and the solution is evolved to a final time T . We fix the parameter $\epsilon = 0.1875/(2\sqrt{2}\tanh^{-1}(0.9))$. The convergence test uses the nonuniform spatial discretization. The mesh contains two distinct spatial resolutions, namely the coarse spacing $2h$ and the refined spacing h . The refined mesh is used in the interfacial region, whereas the coarse mesh is used away from the interface. The mesh configuration may change as the interface moves. The same refinement, remeshing, and interpolation procedures are applied at every resolution.

For the temporal convergence test, we fix the fine spatial mesh size as $h = 1/256$ and set the final time $T = 0.00390625$. The time step sizes are selected as $\Delta t = 2^{-11}$, 2^{-12} , 2^{-13} , 2^{-14} , and 2^{-15} . The sufficiently fine spatial mesh is selected so that the temporal discretization error dominates the spatial discretization error over the considered range of Δt .

For the spatial convergence test, the final time is set to $T = 0.003125$. The fine mesh size h is systematically reduced by a factor of two. We use $h = 1/8$, $1/16$, $1/32$, $1/64$, and $1/128$. The time step size is chosen as $\Delta t = 0.1 \times 2^{-20}$. Because the time integration is first-order accurate, this choice yields a temporal error of order $O(h^2)$. Therefore, the temporal discretization error does not reduce the expected second-order spatial convergence rate. Let $e_{ijk}^n = \phi_{ijk}^n - \Phi(x_i^n, y_j^n, z_k^n, n\Delta t)$. The numerical errors are measured in the discrete ℓ^∞ -norm and ℓ^2 -norm as follows:

$$\|e^n\|_\infty = \max_{i,j,k} |e_{ijk}^n|, \quad \|e^n\|_2 = \sqrt{\sum_{i=1}^{N_x^n} \sum_{j=1}^{N_y^n} \sum_{k=1}^{N_z^n} |e_{ijk}^n|^2 \Delta x_{i-\frac{1}{2}}^n \Delta y_{j-\frac{1}{2}}^n \Delta z_{k-\frac{1}{2}}^n}.$$

The temporal errors and their corresponding convergence orders are summarized in Table 1. Both the discrete ℓ^∞ - and ℓ^2 -errors decrease by approximately a factor of two when the time-step size is halved. The measured convergence orders are close to one for all time-step sizes. These results confirm that the proposed algorithm is first-order accurate in time, which is consistent with the operator splitting procedure and the first-order time integration used in the diffusion solver.

Table 2 presents the spatial convergence results. The errors decrease by approximately a factor of four when the mesh size is reduced by a factor of two. The corresponding convergence orders remain close to two. Therefore, the proposed nonuniform spatial discretization provides second-order spatial accuracy.

Fig. 2 illustrates the simulation results obtained for $h = 1/64$, $\Delta t = 0.25h^2$ and $T = 0.003125$. The numerical solution is observed to be in close visual agreement with the analytical solution.

3.2. Motion by mean curvature

We consider the motion by mean curvature using a 3D AMR strategy. The numerical domain is defined as $\Omega = (-0.5, 0.5)^3$. The initial condition is given by

$$\phi(x, y, z, 0) = \tanh\left(\frac{r_0 - \sqrt{x^2 + y^2 + z^2}}{\sqrt{2}\epsilon}\right),$$

where r_0 denotes the radius of the initial spherical interface, with $r_0 = 0.3$. The interfacial thickness parameter is set to $\epsilon = \epsilon_{10}$, the smallest spatial step size is set to $h = 1/100$, and the time step size is set to $\Delta t = 0.25h^2$. The final time is chosen so that the sphere vanishes completely under the mean curvature flow. To improve computational efficiency, the mesh is adaptively updated every ten time steps by refining near the diffuse interface region where $|\phi| < 0.98$ and coarsening elsewhere. To examine the energy behavior of the numerical solution, we define the discrete energy on the nonuniform mesh as follows:

$$\mathcal{E}_d(\phi^n) = \frac{1}{2} \sum_{i,j,k} |\nabla_h \phi_{ijk}^n|^2 \Delta x_{i-\frac{1}{2}}^n \Delta y_{j-\frac{1}{2}}^n \Delta z_{k-\frac{1}{2}}^n + \frac{1}{\epsilon^2} \sum_{i,j,k} F(\phi_{ijk}^n) \Delta x_{i-\frac{1}{2}}^n \Delta y_{j-\frac{1}{2}}^n \Delta z_{k-\frac{1}{2}}^n.$$

In the following numerical experiments, we use the normalized discrete energy $\mathcal{E}_d(\phi^n)/\mathcal{E}_d(\phi^0)$ to show its change over time.

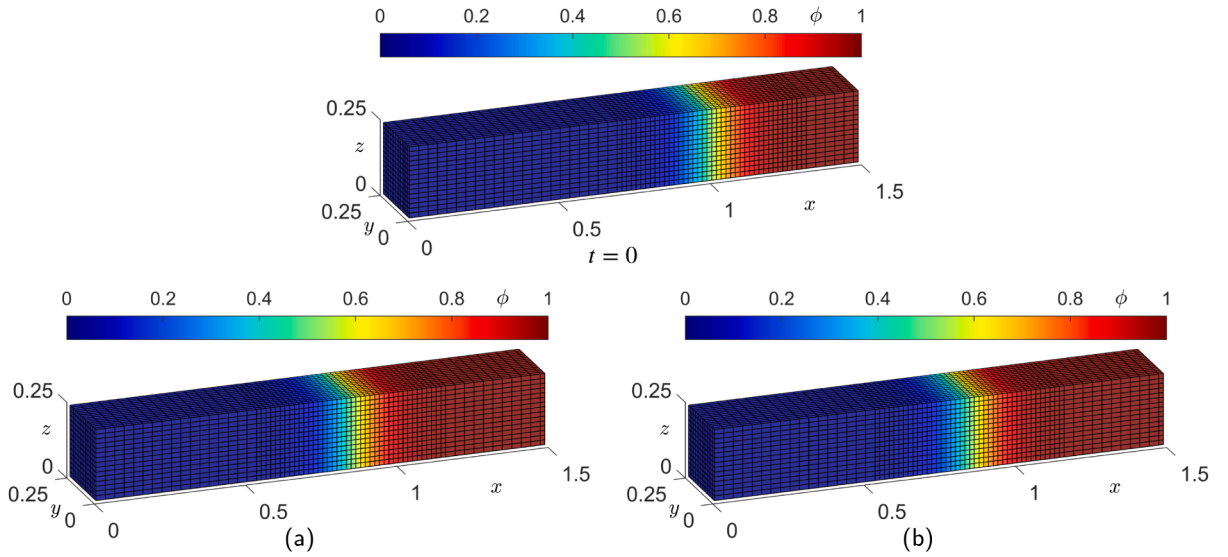


Fig. 2. The time evolution of the traveling wave solution. (a) Numerical solution. (b) Analytical solution.

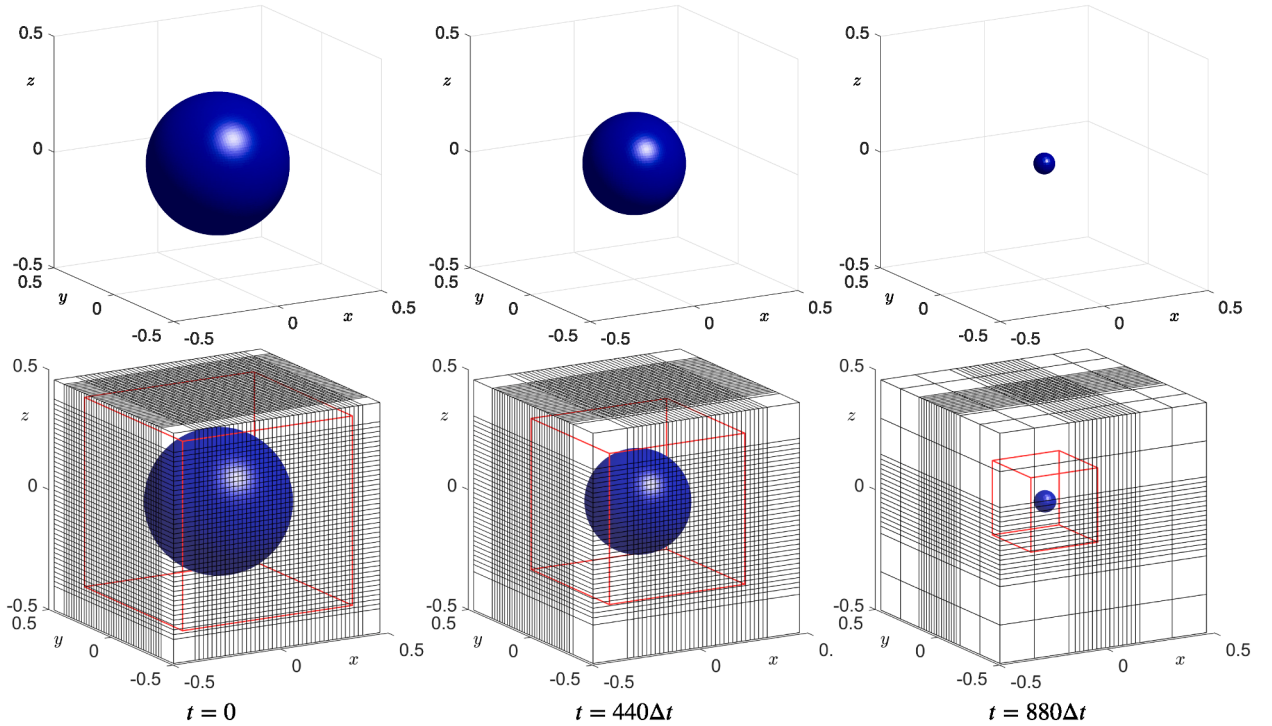


Fig. 3. The first row presents 3D snapshots of the temporal evolution of the sphere. The second row shows the corresponding adaptive meshes used at the times shown in the first-row results. For visualization purposes, a resolution that is coarser by a factor of two than that of the actual computational mesh is used.

Fig. 3 and **4** present the results of this numerical experiment. At $t = 0$, $t = 440\Delta t$, and $t = 880\Delta t$, **Fig. 3(a)** presents 3D visualizations of the interface defined by the zero isosurface of the phase-field variable ϕ . The sphere becomes smaller over time, and its round shape is well preserved during the computation. In **Fig. 3(b)**, the corresponding adaptive meshes are shown. As expected, the mesh is refined around the interface and coarsened elsewhere. This adaptivity allows for efficient resolution of the interfacial transition layer without excessive computational cost.

Fig. 4(a) compares the numerical radius with the theoretical radius. The numerical result agrees well with the theoretical prediction throughout the simulation. **Fig. 4(b)** shows the normalized discrete energy $\mathcal{E}_d(\phi^n)/\mathcal{E}_d(\phi^0)$. The discrete energy decreases monotonically as the sphere shrinks, and no noticeable increase is observed at the remeshing times. This result numerically indicates that the mesh redistribution and interpolation do not disturb the energy decay in this simulation.

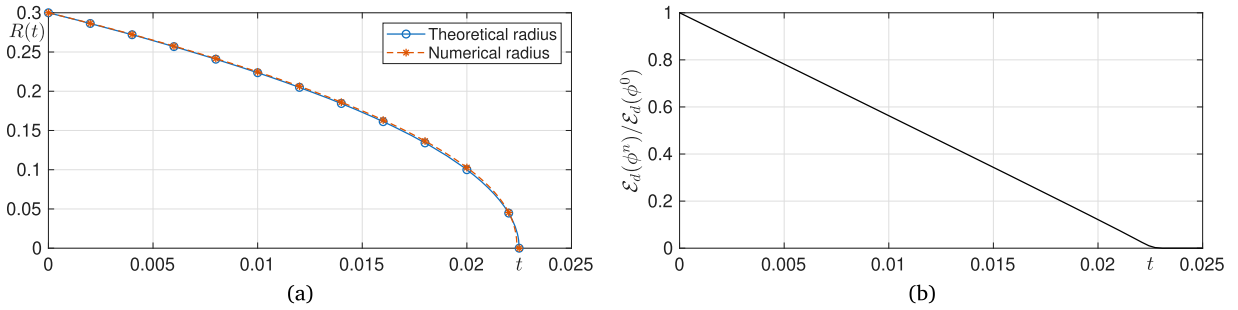


Fig. 4. (a) Changes in theoretical and numerical radii over time. (b) Monotonic decay of the normalized discrete energy $\mathcal{E}_d(\phi^n)/\mathcal{E}_d(\phi^0)$.

3.3. Breakup dynamics of a dumbbell-shaped droplet

To investigate the effectiveness of the proposed adaptive refinement strategy in resolving topological changes, we consider the evolution of a dumbbell-shaped interface. The initial condition is designed as a dumbbell-shaped interface, which consists of two spheres smoothly connected by a cylindrical neck. The initial condition is specified by

$$\phi(x, y, z, 0) = \tanh\left(\frac{\max\{d_1, d_2, d_{\text{neck}}\}}{\sqrt{2}\epsilon}\right),$$

where the geometric quantities d_1 , d_2 , and d_{neck} are defined as

$$\begin{aligned} d_1 &= r_0 - \sqrt{(x-0.5)^2 + y^2 + z^2}, & d_2 &= r_0 - \sqrt{(x+0.5)^2 + y^2 + z^2}, \\ d_{\text{neck}} &= \begin{cases} w - \sqrt{y^2 + z^2} & \text{if } -0.5 < x < 0.5 \\ -1 & \text{otherwise} \end{cases}. \end{aligned}$$

Here, $r_0 = 0.3$ denotes the radius of the spheres, and $w = 0.15$ is the neck width. The domain is defined as $\Omega = (-1, 1) \times (-0.5, 0.5) \times (-0.5, 0.5)$ with $\epsilon = \epsilon_4$. The finest mesh has a grid step size of $h = 1/128$, and mesh refinement is performed every 10 iterations with a time step size of $\Delta t = 0.0001$. Fig. 5 displays the time evolution of the interface (Fig. 5(a)) and the corresponding adaptive mesh structure (Fig. 5(b)) at $t = 0$, $t = 150\Delta t$, and $t = 300\Delta t$. As the simulation progresses, the neck region becomes progressively thinner and eventually undergoes a topological separation.

This numerical experiment highlights the capability of our scheme to resolve complex morphological transitions, such as interface breakup, with minimal computational cost and without sacrificing accuracy.

Fig. 6 displays the normalized discrete energy and the maximum and minimum values of ϕ^n . The discrete energy decreases during the simulation, while the numerical solution remains within $[-1, 1]$. These results numerically support the energy decay and maximum bound preserving property during the topological change.

3.4. Evolution of multiple spheres with different initial radii

To evaluate the robustness of the adaptive refinement strategy for interfaces of heterogeneous scale, we consider the simultaneous shrinkage of three isolated spheres. The phase field is prescribed as the maximum of three signed-distance functions,

$$\phi(x, y, z, 0) = \tanh\left(\frac{\max_{i=1,2,3}\{r_i - \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}\}}{\sqrt{2}\epsilon}\right),$$

with radii $r_1 = 0.6$, $r_2 = 0.4$, $r_3 = 0.2$, and centres $(x_i, y_i, z_i) \in \{(0.7, 0.7, 0.7), (1.2, 1.5, 1.5), (1.7, 0.8, 1.5)\}$. The computational domain $\Omega = (0, 2)^3$ is discretised by the finest grid spacing $h = 1/64$. The interfacial thickness parameter is $\epsilon = \epsilon_4$. A time step size $\Delta t = 10^{-4}$ is used, and mesh refinement is triggered every 20 time steps.

Table 3 reports the total CPU times for both the adaptive and uniform mesh simulations. The simulations are performed in MATLAB R2024b using a desktop system with an Intel Core i9-13900K processor (3.00 GHz) and 32 GB of RAM. The uniform mesh uses a spatial step size $h = 1/45$, which is deliberately selected so that its total CPU time is comparable to that of the adaptive simulation with the finest mesh size $h = 1/64$. Except for the spatial resolution, all other computational parameters are kept identical. For the adaptive computation, the number of grid points changes dynamically as the interfaces evolve; therefore, the grid size reported in Table 3 corresponds to the final time $t = T$. Although the computational costs are nearly identical, the accuracy of the solutions differs significantly, as shown in Fig. 7.

Fig. 7 compares the temporal evolution of three different spheres with different initial radii under curvature-driven motion. In Fig. 7, the first row shows the initial configuration together with both the adaptive mesh and the uniform mesh. The second row presents the adaptive mesh results at $t = 200\Delta t$, $t = 500\Delta t$, and $t = 1000\Delta t$. The third row shows simulation results on a uniform grid at the same times as the second-row results. For visualization purposes, all meshes shown in Fig. 7 are rendered with a resolution

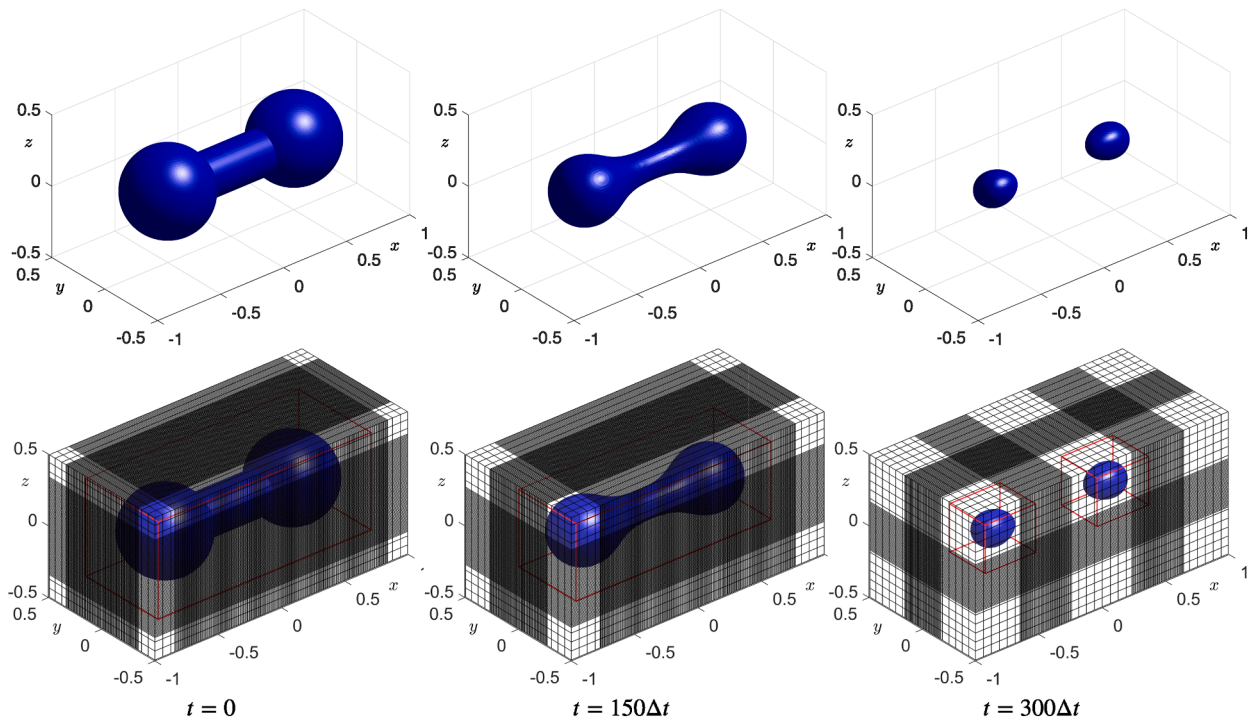


Fig. 5. The first row shows the time evolution of a dumbbell-shaped interface. The second row shows the corresponding adaptive mesh refinements.

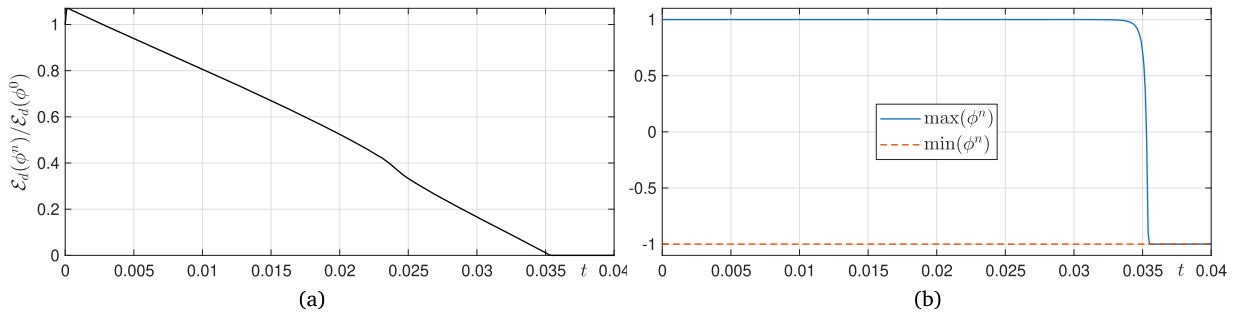


Fig. 6. Temporal evolution of (a) the normalized discrete energy $\mathcal{E}_d(\phi^n)/\mathcal{E}_d(\phi^0)$ and (b) the maximum and minimum values of ϕ^n for the dumbbell-shaped interface.

Table 3

CPU time comparison between adaptive refinement and uniform mesh. The uniform mesh size ($h = 1/45$) was deliberately selected to yield a CPU time comparable to that of the adaptive computation, whose finest mesh size is $h = 1/64$.

Method	Adaptive mesh	Uniform mesh
CPU time (s)	51.269874	50.146194
Grid size ($N_x^{N_t}, N_y^{N_t}, N_z^{N_t}$)	(39, 39, 39)	(90, 90, 90)

coarser by a factor of two than that of the actual computational mesh. The uniform mesh results show notable differences despite the similar computational cost. The spheres disappear faster than in the adaptive case, which implies a loss of accuracy in capturing mean curvature flow when the interface is under-resolved. This experiment confirms that, even with equivalent CPU times, the adaptive refinement method provides a higher spatial resolution near the moving interfaces. As a result, it produces more accurate interface geometries and improved preservation of the fundamental physical dynamics compared to those obtained with a uniform mesh.

Fig. 8 shows the results for the shrinking spherical interface. As shown in Fig. 8(a), the normalized discrete energy decreases during the simulation. No increase in the energy is observed after remeshing. Fig. 8(b) shows that the maximum and minimum values of ϕ^n remain within $[-1, 1]$. These results numerically confirm the maximum bound preserving property of the proposed method.

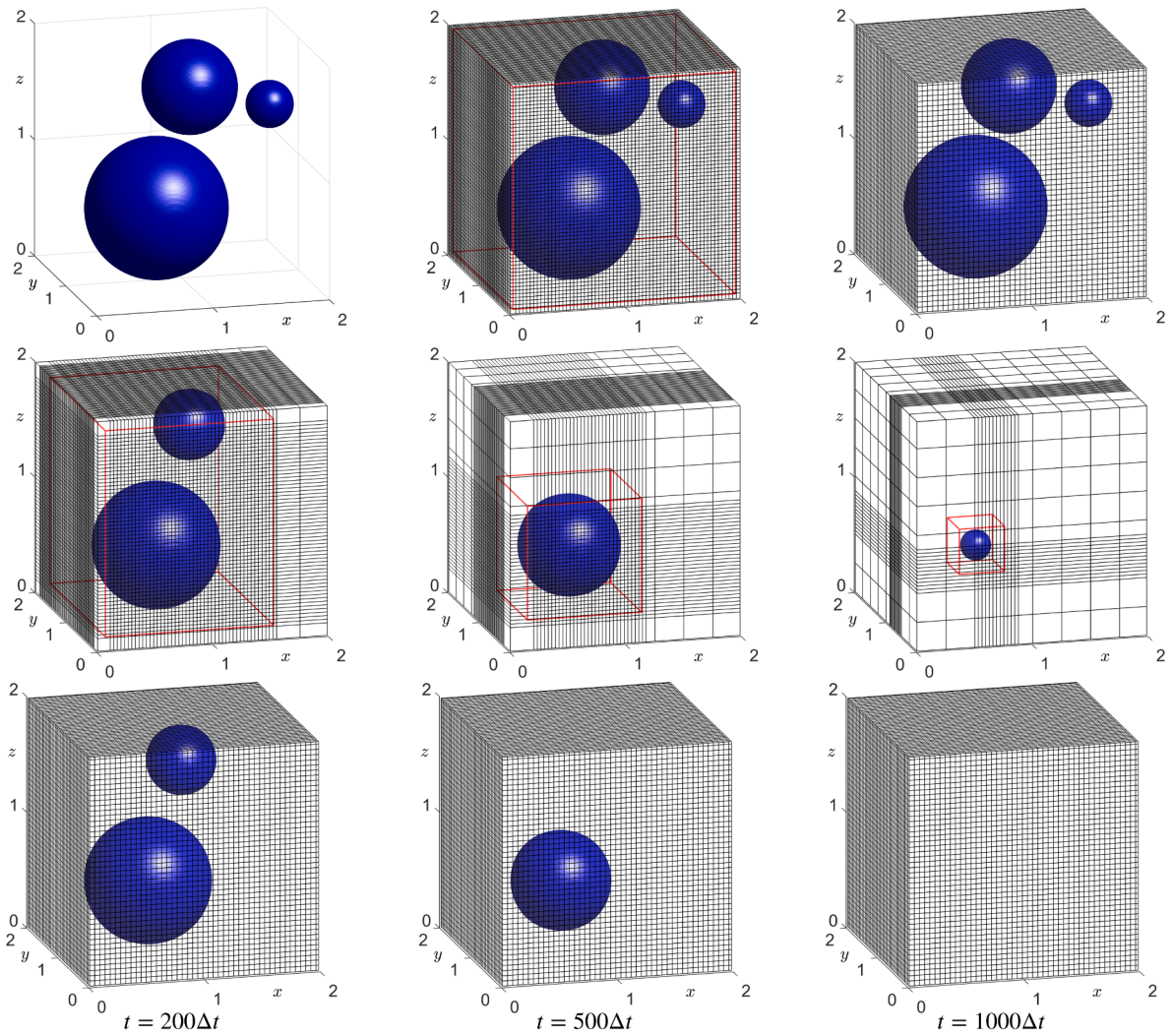


Fig. 7. Temporal evolution of spheres with various initial radii. The first row shows the initial configurations. The second row illustrates the corresponding adaptive refinement meshes with the interface at each specific time. The third row shows the numerical solution computed on a uniform mesh.

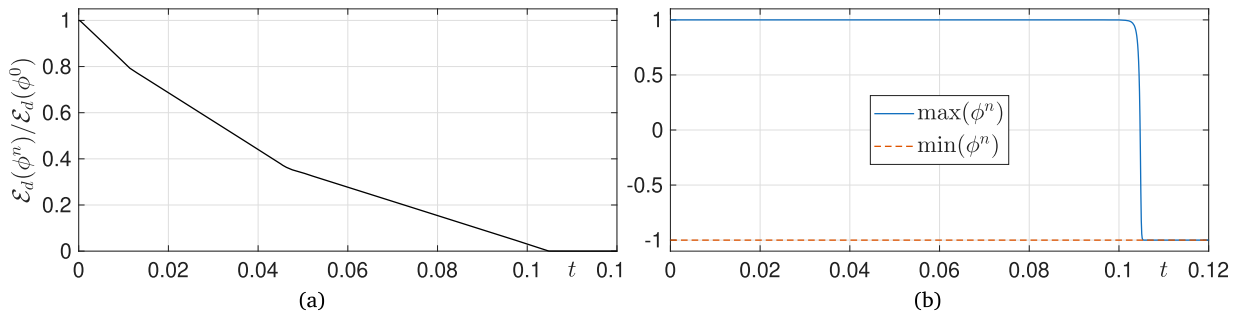


Fig. 8. Temporal evolution of (a) the normalized discrete energy $\mathcal{E}_d(\phi^n)/\mathcal{E}_d(\phi^0)$ and (b) the maximum and minimum values of ϕ^n for the shrinking spherical interface.

Table 4
CPU time comparison with other existing methods.

Method	Proposed method	Non-uniform (fixed)	Explicit method
Δt	$0.4h^2$	$0.4h^2$	$0.1h^2$
CPU time (s)	198.7374	227.0356	600.7676
Grid size ($N_x^{N_t}, N_y^{N_t}, N_z^{N_t}$)	(101, 126, 88)	(133, 169, 97)	(180, 200, 160)

3.5. Comparison with other methods

We compared the proposed algorithm with two representative methods. The first method, referred to as the non-uniform fixed-mesh method, uses the same non-uniform mesh constructed from the initial phase-field configuration as the proposed method, but keeps the mesh fixed throughout the simulation without subsequent remeshing or interpolation. The second method, referred to as the explicit adaptive method, uses the same mesh-refinement and remeshing procedures as the proposed method, while the diffusion subproblems are discretized explicitly rather than solved with the direction-wise implicit Thomas algorithm. Consequently, a smaller time step is required for the explicit method to satisfy its stability restriction. The computational efficiency is assessed in terms of CPU time, and the numerical results are summarized in Table 4. For the numerical tests, we consider the domain $\Omega = (-0.9, 0.9) \times (-1, 1) \times (-0.8, 0.8)$. The mesh size is chosen as $h = 0.01$. The interface parameter was set to $\epsilon = \epsilon_{10}$. The initial condition is given by a smoothed rectangular interface,

$$\phi(x, y, z, 0) = \tanh\left(\frac{\min\{\min\{0.5 - |x|, 0.7 - |y|\}, 0.3 - |z|\}}{\sqrt{2}\epsilon}\right).$$

The computational results in Table 4 show that the proposed algorithm achieved the lowest CPU time while maintaining stability with a relatively large time step size. The non-uniform fixed mesh method requires more CPU time, and the explicit method showed a significant increase in computational cost because of its stricter stability constraint on the time step size.

Throughout the computation, the refinement algorithm confines the fine grid to a narrow band around each interface; hence, the total number of active cells declines as the spheres shrink. This behaviour demonstrates that the adaptive procedure controls computational cost while it preserves accuracy. Therefore, the numerical test shows that the proposed scheme works well for multi-scale geometries and maintains its efficiency even when individual features disappear.

3.6. Extension to a time-dependent interfacial parameter

The proposed algorithm can be extended to the AC equation with a time-dependent interfacial parameter. The parameter ϵ controls the thickness of the diffuse interface. As the enclosed volume changes, a fixed value of ϵ may not be suitable for the size of the interface. The spatial resolution should also change with the interface size. We consider the following AC model with a time-dependent interfacial parameter:

$$\frac{\partial \phi}{\partial t} = -\frac{F'(\phi)}{\epsilon^2(t)} + \phi_{xx} + \phi_{yy} + \phi_{zz}.$$

A time-dependent interfacial parameter has also been considered with adaptive mesh refinement in previous work [28]. In that method, ϵ is determined from the current mesh size. The mesh size and ϵ are reduced stepwise when a prescribed refinement condition is satisfied. The present method uses a different criterion. We determine both the fine mesh size and ϵ directly from the current enclosed volume. No prescribed reduction factor is used. Let V^0 be the initial volume enclosed by the zero-level isosurface of ϕ , and let V^n be the corresponding volume at time $t = n\Delta t$. At each remeshing step, we define the fine mesh size as

$$h^n = h^0 \left(\frac{V^n}{V^0} \right)^{1/3},$$

where h^0 is the initial fine mesh size. This definition is based on the three-dimensional volume. The effective number of fine-grid cells inside the enclosed region is approximately proportional to $V^n/(h^n)^3$. Using Eq. (17), we obtain

$$\frac{V^n}{(h^n)^3} = \frac{V^0}{(h^0)^3}.$$

Thus, this quantity remains constant even when the volume changes. The enclosed region is therefore represented by approximately the same effective number of fine-grid cells. This idea does not depend on a specific interface shape. We update the interfacial parameter using the same volume ratio:

$$\epsilon^n = \epsilon^0 \left(\frac{V^n}{V^0} \right)^{1/3},$$

where ϵ^0 is the initial value. It follows that

$$\frac{\epsilon^n}{h^n} = \frac{\epsilon^0}{h^0}.$$

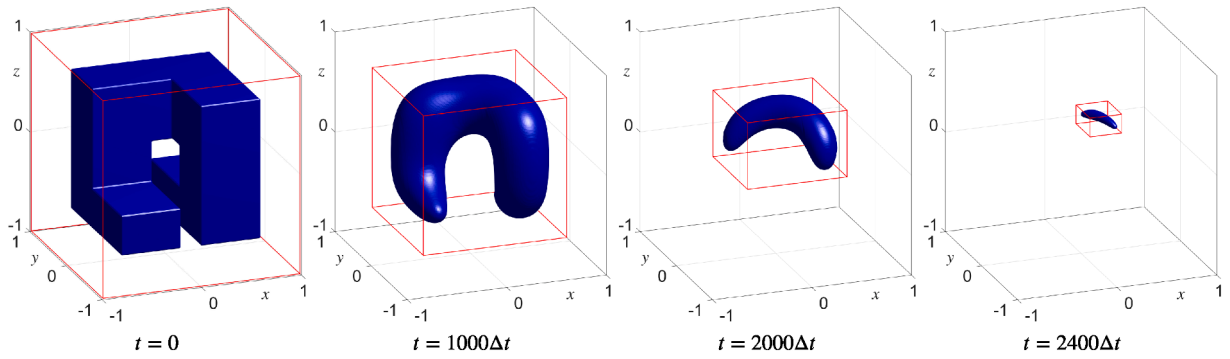


Fig. 9. Time evolution of a complex 3D interface with the volume-based adaptive parameters at $t = 0, 0.04, 0.08$, and 0.096 . The red boxes show the regions with the finest mesh resolution.

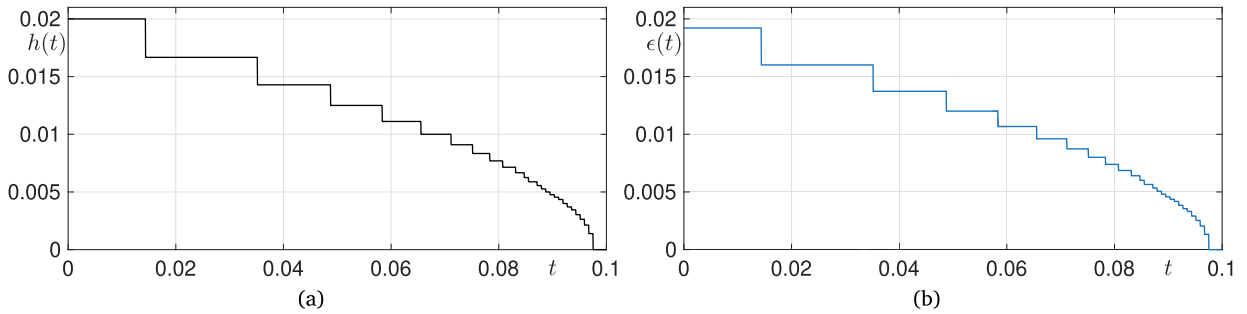


Fig. 10. Time evolution of (a) the finest mesh size $h(t)$ and (b) the interfacial parameter $\epsilon(t)$.

Therefore, the relative resolution across the diffuse interface is also maintained. The values of h^n and ϵ^n are updated only at the remeshing steps. They remain fixed until the next remeshing step. After h^n is updated, a new fine reference mesh is constructed. The adaptive mesh is then generated using the same procedure described in Section 2. Trilinear interpolation is used to transfer the numerical solution to the new mesh.

The main difference from the previous method is the rule used to update h and ϵ . The previous method uses a stepwise refinement rule and changes the parameters by a prescribed factor. In contrast, the present method calculates both parameters directly from the current volume. Thus, their values follow the geometric change of the interface at every remeshing step. This volume-based rule also keeps the effective spatial resolution nearly constant. Moreover, it can be applied to general three-dimensional interfaces without assuming a particular shape.

To examine the volume-based method for a complex geometry, we consider the three-dimensional initial shape shown in Fig. 9. The shape consists of several connected parts and has no simple characteristic radius. Thus, its size cannot be represented by the radius of a sphere. This example is suitable for testing the proposed volume-based rule. The numerical domain is $\Omega = (-1, 1)^3$. The initial finest mesh size is set to $h^0 = 0.02$, and the time step is $\Delta t = 0.1(h^0)^2$. The mesh is updated every 20 time steps. Fig. 9 displays the evolution of the interface. The initially complex shape becomes smoother as time increases. Its overall size also decreases. At the same time, the region with the finest mesh resolution follows the moving interface. This allows a fine spatial resolution to be maintained without refining the entire computational domain. As the enclosed volume decreases, a smaller target mesh size is required. The actual mesh size is selected by dividing each coarse mesh interval into an integer number of subintervals. Therefore, $h(t)$ changes in a stepwise manner rather than continuously. The interfacial parameter is updated using the same change in the mesh size.

Fig. 10 shows the changes in $h(t)$ and $\epsilon(t)$. Both values decrease as the volume becomes smaller. Their changes are stepwise because the parameters are updated only at the remeshing steps. In addition,

$$\frac{\epsilon^n}{h^n} = \frac{\epsilon^0}{h^0}$$

is maintained throughout the computation. Hence, the relative resolution across the diffuse interface remains unchanged. These numerical results display that the proposed scaling can adjust the spatial resolution according to the size of a general 3D interface.

4. Discussion

As shown in Fig. 11, for complex domains such as topological optimization [29], the numerical methodology requires both mesh adaptation and stable time integration. By imposing boundary conditions using a Dirichlet boundary condition method [30], the proposed scheme can be extended to complex geometries.

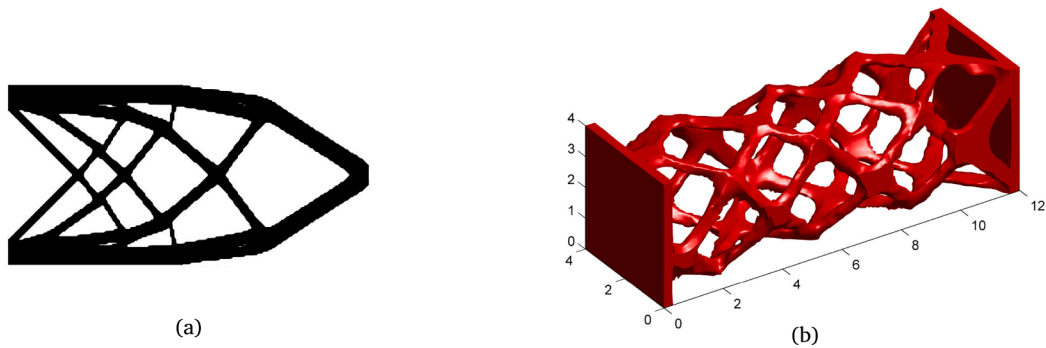


Fig. 11. Topological optimization results: (a) optimized 2D structure (reproduced with permission from Xie et al. [31]), and (b) optimized 3D structure (reproduced with permission from Zhang et al. [32]).

Although the proposed adaptive FDM shows robust numerical stability and accuracy in 3D simulations, a theoretical convergence analysis for AMR schemes is still highly challenging. The main difficulty comes from the dynamic nature of mesh redistribution, which introduces complex error propagation mechanisms that do not appear in uniform meshes. Prior studies, such as [14,16] for adaptive finite element and discontinuous Galerkin methods, provided a priori error estimates under restrictive assumptions, but the extension to fully discrete FDMs in 3D is still unresolved. Mackenzie & Mekwi [33] also noted the complexity of convergence theory in moving mesh settings, even in one dimension. Thus, while our numerical results give strong evidence of convergence, the proof of a complete theoretical method for adaptive FDM in 3D remains an important topic for future work.

5. Conclusions

In this study, we developed an unconditionally maximum bound preserving and adaptive FDM for solving the 3D AC equation. The proposed scheme uses operator splitting and an adaptive mesh to improve efficiency and accuracy. It separates the reaction and diffusion terms and preserves the maximum bound without time step restrictions. Numerical results show that the proposed method efficiently captures complex interface changes in three dimensions. The proposed method is robust and suitable for large-scale simulations, and it can be extended to more complex phase-field models in future work.

CRediT authorship contribution statement

Soobin Kwak: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Junseok Kim:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Formal analysis.

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.

Acknowledgement

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 sincerely thank the reviewers for their valuable comments and helpful suggestions.

Data availability

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

References

- [1] Youssef H, Elmelouky A, Louzazni M, Belhora F, Monkade M. A numerical study of interface dynamics in fluid materials. *Mater Tech* 2024;112(4):401. <https://doi.org/10.1051/mattech/2024018>
- [2] Shakeel M, Abbas N, Rehman M JU, Alshammari FS, Al-Yaari A. Lie symmetry analysis and solitary wave solution of biofilm model allen-cahn. *Sci Rep* 2024;14(1):12844. <https://doi.org/10.1038/s41598-024-62315-5>
- [3] Li Y, Qin K, Xia Q, Kim J. A second-order unconditionally stable method for the anisotropic dendritic crystal growth model with an orientation-field. *Appl Numer Math* 2023;184:512–26. <https://doi.org/10.1016/j.apnum.2022.11.006>

- [4] Zu Y, Li A, Wei H. Phase-field lattice boltzmann model for interface tracking of a binary fluid system based on the Allen–Cahn equation. *Phys Rev E* 2020;102(5):053307. <https://doi.org/10.1103/PhysRevE.102.053307>
- [5] Allen SM, Cahn JW. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metall* 1979;27(6):1085–95. [https://doi.org/10.1016/0001-6160\(79\)90196-2](https://doi.org/10.1016/0001-6160(79)90196-2)
- [6] Song X, Xia Q, Kim J, Li Y. An unconditional energy stable data assimilation scheme for Navier–Stokes–Cahn–Hilliard equations with local discretized observed data; vol. 164. Elsevier; 2024. <https://doi.org/10.1016/j.camwa.2024.03.018>
- [7] Song X, Xia B, Li Y. An efficient data assimilation based unconditionally stable scheme for cahn–hilliard equation. *Comput Appl Math* 2024;43(3):121. <https://doi.org/10.1007/s40314-024-02632-7>
- [8] Xia Q, Liu Y, Kim J, Li Y. Binary thermal fluids computation over arbitrary surfaces with second-order accuracy and unconditional energy stability based on phase-field model. *J Comput Appl Math* 2023;433:115319. <https://doi.org/10.1016/j.cam.2023.115319>
- [9] Xia Q, Lai S, Kim J, Li Y. Phase field modeling of melting and solidification dynamics in metallic powders during the bed fusion process. *Commun Nonlinear Sci Numer Simul* 2025;146:108762. <https://doi.org/10.1016/j.cnsns.2025.108762>
- [10] Lv Z, Song X, Feng J, Xia Q, Xia B, Li Y. Reduced-order prediction model for the cahn–hilliard equation based on deep learning. *Eng Anal Bound Elem* 2025;172:106118. <https://doi.org/10.1016/j.enganabound.2025.106118>
- [11] Yang J, Yuan Z, Zhou Z. Arbitrarily high-order maximum bound preserving schemes with cut-off postprocessing for allen–cahn equations. *J Sci Comput* 2022;90(2):76. <https://doi.org/10.1007/s10915-021-01746-y>
- [12] Ju L, Li X, Qiao Z. Stabilized exponential-sav schemes preserving energy dissipation law and maximum bound principle for the allen–cahn type equations. *J Sci Comput* 2022;92(2):66. <https://doi.org/10.1007/s10915-022-01921-9>
- [13] Li J, Lan R, Cai Y, Ju L, Wang X. Second-order semi-lagrangian exponential time differencing method with enhanced error estimate for the convective allen–cahn equation. *J Sci Comput* 2023;97(1):7. <https://doi.org/10.1007/s10915-023-02316-0>
- [14] Li J, Zeng J, Li R. An adaptive discontinuous finite volume element method for the Allen–Cahn equation. *Adv Comput Math* 2023;49(4):55. <https://doi.org/10.1007/s10444-023-10031-5>
- [15] Wang K, Xia Y, Li Z. A phase-field lattice boltzmann method for liquid-vapor phase change problems based on conservative Allen–Cahn equation and adaptive treegrid. *Comput Fluids* 2023;264:105973. <https://doi.org/10.1016/j.compfluid.2023.105973>
- [16] Li R, Gao Y, Chen Z. Adaptive discontinuous galerkin finite element methods for the Allen–Cahn equation on polygonal meshes. *Numer Algor* 2024;95(4):1981–2014. <https://doi.org/10.1007/s11075-023-01635-5>
- [17] Shah A, Sabir M, Qasim M, Bastian P. Efficient numerical scheme for solving the Allen–Cahn equation; *Numer meth part differ equ vol. 34*. Wiley; 2018. <https://doi.org/10.1002/num.22255>
- [18] Zhang B, Fu H, Lan R, Xie S. Energy dissipation law and maximum bound principle-preserving linear BDF2 schemes with variable steps for the Allen–Cahn equation. *J Sci Comput* 2025;105(2):51. <https://doi.org/10.1007/s10915-025-03077-8>
- [19] Loubère R, Turpault R, Bourriaud A. A MOOD-like compact high order finite volume scheme with adaptive mesh refinement. *Appl Math Comput* 2023;443:127792. <https://doi.org/10.1016/j.amc.2022.127792>
- [20] Hernández-López E, Ullah M, Wences G, Wang J. Adaptive parameter-space refinement and Markov state models for eco-epidemiological dynamics. *Appl Math Comput* 2026;521:129974. <https://doi.org/10.1016/j.amc.2026.129974>
- [21] Neena A, Clemence-Mkhope D, Awasthi A. Nonstandard finite difference schemes for linear and non-linear Fokker–Planck equations. *J Eng Math* 2024;145(1):11. <https://doi.org/10.1007/s10665-024-10346-2>
- [22] Lee D, Lee C. Assessment of morphological similarities for the conservative Allen–Cahn and Cahn–Hilliard equations; vol. 76. Elsevier; 2024. <https://doi.org/10.1016/j.jocs.2024.102240>
- [23] Kim H, Ham S, Kwak S, Kim J. An unconditionally stable adaptive finite difference scheme for the Allen–Cahn equation. *Comput Phys Commun* 2025;315:109712. <https://doi.org/10.1016/j.cpc.2025.109712>
- [24] Kwak S. Practical implementation of boundary conditions in the Thomas algorithm. *J Korean Soc Ind Appl Math* 2025;29:171–83. <https://doi.org/10.12941/jksiam.2025.29.171>
- [25] Varga R. Matrix iterative analysis (2nd ed.). Springer, Berlin Heidelberg; 2000. <https://doi.org/10.1007/978-3-642-05156-2>
- [26] Douglas J, Gunn JE. A general formulation of alternating direction methods. *Numer Math* 1964;6:428–53. <https://doi.org/10.1007/BF01386093>
- [27] Zhai S, Feng X, He Y. Numerical simulation of the three dimensional Allen–Cahn equation by the high-order compact ADI method. *Comput Phys Commun* 2014;185(10):2449–55. <https://doi.org/10.1016/j.cpc.2014.05.017>
- [28] Yang J, Lee D, Kwak S, Ham S, Kim J. The Allen–Cahn model with a time-dependent parameter for motion by mean curvature up to the singularity. *Chaos Solit Fractals* 2024;182:114803. <https://doi.org/10.1016/j.chaos.2024.114803>
- [29] Sugimoto S, Sakane S, Takaki T. Multi-phase-field framework for multi-material topology optimization. *Comput Mater Sci* 2024;244:113201. <https://doi.org/10.1016/j.commatsci.2024.113201>
- [30] Kwak S, Ham S, Wang J, Kim H, Kim J. Effective perpendicular boundary conditions in phase-field models using Dirichlet boundary conditions. *Eng Comput* 2025;41(4):2377–92. <https://doi.org/10.1007/s00366-025-02109-z>
- [31] Xie W, Xia Q, Yu Q, Li Y. An effective phase field method for topology optimization without the curvature effects; vol. 146. Elsevier; 2023. <https://doi.org/10.1016/j.camwa.2023.06.037>
- [32] Zhang W, Chen J, Zhu X, Zhou J, Xue D, Lei X, et al. Explicit three dimensional topology optimization via moving morphable void (MMV) approach. *Comput Meth Appl Mech Eng* 2017;322:590–614. <https://doi.org/10.1016/j.cma.2017.05.002>
- [33] Mackenzie JA, Mekwi WR. An analysis of stability and convergence of a finite-difference discretization of a model parabolic pde in 1D using a moving mesh. *IMA J Numer Anal* 2007;27(3):507–28. <https://doi.org/10.1093/imanum/drl034>