Contents lists available at ScienceDirect

Physica A

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

A fourth-order spatial accurate and practically stable compact scheme for the Cahn–Hilliard equation

Chaeyoung Lee [a], Darae Jeong [a], Jaemin Shin [a], Yibao Li [b], Junseok Kim [a,*]

[a] Department of Mathematics, Korea University, Seoul, 136-713, Republic of Korea

[b] Department of Computational Science and Engineering, Yonsei University, Seoul, 120-749, Republic of Korea

HIGHLIGHTS

- We propose a fourth-order accurate compact scheme for the Cahn–Hilliard equation.
- The scheme is mass conserving and practically stable.
- The scheme is good for parallel and adaptive mesh refinement implementation.

ARTICLE INFO

Article history:
Received 2 August 2013
Received in revised form 27 January 2014
Available online 30 April 2014

Keywords:
Fourth-order compact scheme
Cahn–Hilliard equation
Multigrid
Practically stable scheme
Parallel computing
Adaptive mesh refinement

ABSTRACT

We present a fourth-order spatial accurate and practically stable compact difference scheme for the Cahn–Hilliard equation. The compact scheme is derived by combining a compact nine-point formula and linearly stabilized splitting scheme. The resulting system of discrete equations is solved by a multigrid method. Numerical experiments are conducted to verify the practical stability and fourth-order accuracy of the proposed scheme. We also demonstrate that the compact scheme is more robust and efficient than the non-compact fourth-order scheme by applying to parallel computing and adaptive mesh refinement.

1. Introduction

The Cahn–Hilliard (CH) equation was originally introduced as a phenomenological model of phase separation in a binary alloy [1,2] and has been widely applied in many areas such as image processing [3,4], microstructure [5], multi-phase fluid flows [6], planet formation [7], and tumor growth [8,9]. We consider the following CH equation

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = M\Delta\mu(\phi(\mathbf{x},t)), \quad \mathbf{x}\in\Omega, t>0, \tag{1}$$

$$\mu(\phi(\mathbf{x},t)) = F'(\phi(\mathbf{x},t)) - \epsilon^2\Delta\phi(\mathbf{x},t), \tag{2}$$

where the order parameter $\phi(\mathbf{x},t)$ is the difference of the local concentrations of the two components in a domain Ω. M is the mobility, μ is the chemical potential, $F(\phi) = 0.25(\phi^2-1)^2$ is the free energy density, and ϵ is a positive constant related to the interfacial thickness. The total free energy functional of the CH equation is defined as $\mathcal{E}(\phi) = \int_\Omega \left(F(\phi) + 0.5\epsilon^2|\nabla\phi|^2\right) \mathrm{d}\mathbf{x}$. For the sake of convenience, we consider the constant mobility, $M = 1$.

[*] Corresponding author. Tel.: +82 2 3290 3077; fax: +82 2 929 8562.
E-mail addresses: cfdkim@korea.ac.kr, junseok_kim@yahoo.com (J. Kim).
URL: http://math.korea.ac.kr/~cfdkim (J. Kim).

http://dx.doi.org/10.1016/j.physa.2014.04.038

The CH equation has fourth-order spatial derivatives and a Laplacian acting on the nonlinear term $F'(\phi)$. In general, explicit time discretizations require small time step sizes for stability. To overcome constraints of the time step sizes, several implicit time discretizations have been proposed [10–16]. However, most of these numerical solutions have the second-order accuracy in spatial discretizations. Note that the spectral [17–19] and finite element [20] methods for the CH equation can be approximated to arbitrary accuracy. However, there are advantages and disadvantages to each numerical method. Among these methods, the finite difference method is suitable in the framework of adaptive mesh refinement (AMR) [21].

In recent years, high-order compact difference methods have been developed for simulating computational fluid dynamics [22], acoustics [23], electromagnetic [24,25], and option pricing in stochastic volatility models [26]. Moreover, there are various studies about fourth-order compact schemes for the Poisson [27–29], heat [30], Navier–Stokes [31–33], biharmonic [34,35], reaction–diffusion [36], and convection–diffusion [37,38] equations. Also, sixth-order compact schemes for the Poisson [29,39], Helmholtz [24], and convection–diffusion equations [22] have been developed.

Up to now, there are many works on the second-order finite difference method but few works on the high-order difference method for the CH equation. Li et al. [40] established a three-level linearized compact difference scheme for the CH equation. In this paper, we propose the compact scheme by combining a compact nine-point formula and linearly stabilized splitting scheme [41]. The proposed numerical scheme has a merit that it can be straightforwardly applied to AMR framework.

This paper is organized as follows. In Section 2, we derive the fourth-order compact finite difference scheme and describe the multigrid, parallel computing, and AMR methods for the CH equation. In Section 3, numerical results are presented to confirm the superiority of our scheme. Finally, conclusions are drawn in Section 4.

2. Numerical solution

2.1. Discretization

We discretize the CH equation in two-dimensional domain $\Omega = (a, b) \times (c, d)$. Let N_x and N_y be positive even integers, $h = (b - a)/N_x = (d - c)/N_y$ be the uniform mesh size. We denote a discrete computational domain by $\Omega_h = \{(x_i, y_j) : x_i = a + (i - 0.5)h,\ y_j = c + (j - 0.5)h,\ 1 \le i \le N_x, 1 \le j \le N_y\}$, which is the set of cell-centers. Let ϕ_{ij}^n be the approximation of $\phi(x_i, y_j, n\Delta t)$, where $\Delta t = T/N_t$ is the time step, T is the final time, and N_t is the total number of time steps. We use periodic boundary conditions for ϕ and μ as follows:

$$\phi_{i0} = \phi_{i,Nx}, \qquad \phi_{i,Nx+1} = \phi_{i1}, \qquad \phi_{0j} = \phi_{Ny,j}, \qquad \phi_{Ny+1,j} = \phi_{1j}.$$

The discrete differentiation operators are

$$D_x\phi_{i+\frac{1}{2},j} = \frac{1}{12}\frac{\phi_{i+1,j+1} - \phi_{i,j+1}}{h} + \frac{5}{6}\frac{\phi_{i+1,j} - \phi_{ij}}{h} + \frac{1}{12}\frac{\phi_{i+1,j-1} - \phi_{i,j-1}}{h},$$
$$D_y\phi_{i,j+\frac{1}{2}} = \frac{1}{12}\frac{\phi_{i+1,j+1} - \phi_{i+1,j}}{h} + \frac{5}{6}\frac{\phi_{i,j+1} - \phi_{ij}}{h} + \frac{1}{12}\frac{\phi_{i-1,j+1} - \phi_{i-1,j}}{h},$$

and we use the notation $\nabla_c\phi_{ij} = \left(D_x\phi_{i+\frac{1}{2},j}, D_y\phi_{i,j+\frac{1}{2}}\right)$ to represent the discrete gradient of ϕ. The discrete divergence operator is defined at cell-center point as

$$\nabla_d \cdot (u, v)_{ij} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{h} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{h}.$$

We then define the discrete l_2-inner products as

$$(\phi, \psi)_h := h^2 \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \phi_{ij}\psi_{ij},$$

$$(\nabla_c\phi, \nabla_c\psi)_e := h^2 \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \left(D_x\phi_{i+\frac{1}{2},j}D_x\psi_{i+\frac{1}{2},j} + D_y\phi_{i,j+\frac{1}{2}}D_y\psi_{i,j+\frac{1}{2}}\right),$$

and the discrete norms as $\|\phi\|^2 = (\phi, \phi)_h$ and $\|\nabla\phi\|_e^2 = (\nabla_c\phi, \nabla_c\phi)_e$. We define the discrete total energy functional by

$$\mathcal{E}^h(\phi^n) = (F(\phi^n), 1)_h + \frac{\epsilon^2}{2}\|\nabla\phi^n\|_e^2.$$

2.2. Compact finite difference scheme

The nine-point Laplacian operator Δ_c [42] is defined as

$$\Delta_c\phi_{ij} = \nabla_d \cdot \nabla_c\phi_{ij} = \frac{1}{6h^2}\left(\phi_{i-1,j+1} + 4\phi_{i,j+1} + \phi_{i+1,j+1} + 4\phi_{i-1,j} - 20\phi_{ij} + 4\phi_{i+1,j} + \phi_{i-1,j-1} + 4\phi_{i,j-1} + \phi_{i+1,j-1}\right).$$

By the Taylor series in two variables, we can obtain

$$\phi(x+\Delta x, y+\Delta y) = \sum_{k=0}^{5} \frac{1}{k!}\left(\Delta x\frac{\partial}{\partial x} + \Delta y\frac{\partial}{\partial y}\right)^k \phi(x,y) + O((\Delta x)^6 + (\Delta y)^6).$$

By replacing Δx and Δy with different values $\pm h$, we get

$$\phi(x+h,y)+\phi(x-h,y)+\phi(x,y-h)+\phi(x,y+h) = 4\phi + h^2\phi_{xx} + h^2\phi_{yy} + \frac{h^4}{12}\phi_{xxxx} + \frac{h^4}{12}\phi_{yyyy} + O(h^6), \quad (3)$$

$$\phi(x-h,y-h)+\phi(x-h,y+h)+\phi(x+h,y-h)+\phi(x+h,y+h)$$
$$= 4\phi + 2h^2\phi_{xx} + 2h^2\phi_{yy} + \frac{h^4}{6}\phi_{xxxx} + h^4\phi_{xxyy} + \frac{h^4}{6}\phi_{yyyy} + O(h^6). \quad (4)$$

From Eqs. (3) and (4), we have

$$\phi(x-h,y-h)+\phi(x-h,y+h)+\phi(x+h,y-h)+\phi(x+h,y+h)$$
$$+4\left[\phi(x+h,y)+\phi(x-h,y)+\phi(x,y-h)+\phi(x,y+h)\right] - 20\phi(x,y)$$
$$= 6h^2\left(\phi_{xx}+\phi_{yy}\right)(x,y) + \frac{h^4}{2}\left(\phi_{xxxx}+2\phi_{xxyy}+\phi_{yyyy}\right)(x,y) + O(h^6).$$

Finally, we have

$$\Delta_c\phi_{ij} = \Delta\phi(x_i,y_j) + \frac{h^2}{12}\Delta^2\phi(x_i,y_j) + O(h^4), \quad (5)$$

where $\Delta^2\phi = \Delta(\Delta\phi)$ is the biharmonic operator. Note that another standard fourth-order nine-point Laplacian operator Δ_s is defined as

$$\Delta_s\phi_{ij} = \frac{1}{12h^2}\left(-\phi_{i-2,j} + 16\phi_{i-1,j} - 30\phi_{ij} + 16\phi_{i+1,j} - \phi_{i+2,j}\right)$$
$$+ \frac{1}{12h^2}\left(-\phi_{i,j-2} + 16\phi_{i,j-1} - 30\phi_{ij} + 16\phi_{i,j+1} - \phi_{i,j+2}\right).$$

In a similar manner, we can derive $\Delta_s\phi_{ij} = \Delta\phi(x_i,y_j) + O(h^4)$. Now, we derive the fourth-order spatial accurate and practically stable compact finite difference scheme for the CH equation. Beginning with the nine-point Laplacian operator for μ_{ij}, we have

$$\Delta_c\mu_{ij} = \Delta\mu(x_i,y_j) + \frac{h^2}{12}\Delta^2\mu(x_i,y_j) + O(h^4)$$
$$= \phi_t(x_i,y_j) + \frac{h^2}{12}\Delta\phi_t(x_i,y_j) + O(h^4) = (\phi_t)_{ij} + \frac{h^2}{12}\Delta_c(\phi_t)_{ij} + O(h^4). \quad (6)$$

Note that Eq. (1) is used for the second equality in Eq. (6). We approximate the temporal operator ϕ_t to first-order accuracy by treating it implicitly:

$$\Delta_c\mu_{ij}^{n+1} = \frac{\phi_{ij}^{n+1}-\phi_{ij}^{n}}{\Delta t} + \frac{h^2}{12}\frac{\Delta_c\phi_{ij}^{n+1}-\Delta_c\phi_{ij}^{n}}{\Delta t} + O(\Delta t) + O(h^4). \quad (7)$$

We apply the linearly stabilized splitting scheme [41] to Eq. (2):

$$\mu^{n+1} = (\phi^3-3\phi)^n + 2\phi^{n+1} - \epsilon^2\Delta\phi^{n+1}. \quad (8)$$

By substituting Eq. (8) into Eq. (5), we get

$$\Delta_c\phi_{ij}^{n+1} = \frac{(\phi^3-3\phi)_{ij}^n + 2\phi_{ij}^{n+1} - \mu_{ij}^{n+1}}{\epsilon^2} + \frac{h^2}{12}\Delta\left(\frac{(\phi^3-3\phi)^n + 2\phi^{n+1} - \mu^{n+1}}{\epsilon^2}\right)_{ij} + O(h^4)$$
$$= \frac{(\phi^3-3\phi)_{ij}^n + 2\phi_{ij}^{n+1} - \mu_{ij}^{n+1}}{\epsilon^2} + \frac{h^2}{12\epsilon^2}\left(\Delta_c(\phi^3-3\phi)_{ij}^n + 2\Delta_c\phi_{ij}^{n+1} - \Delta_c\mu_{ij}^{n+1}\right) + O(h^4). \quad (9)$$

Finally, from Eqs. (7) and (9), we have the fourth-order spatial accurate and practically stable compact finite difference scheme for the CH equation:

$$\frac{\phi_{ij}^{n+1}}{\Delta t} + \frac{h^2}{12\Delta t}\Delta_c\phi_{ij}^{n+1} - \Delta_c\mu_{ij}^{n+1} = \frac{\phi_{ij}^{n}}{\Delta t} + \frac{h^2}{12\Delta t}\Delta_c\phi_{ij}^{n}, \quad (10)$$

$$-\frac{2}{\epsilon^2}\phi_{ij}^{n+1} + \left(1-\frac{h^2}{6\epsilon^2}\right)\Delta_c\phi_{ij}^{n+1} + \frac{1}{\epsilon^2}\mu_{ij}^{n+1} + \frac{h^2}{12\epsilon^2}\Delta_c\mu_{ij}^{n+1} = \frac{1}{\epsilon^2}(\phi^3-3\phi)_{ij}^n + \frac{h^2}{12\epsilon^2}\Delta_c(\phi^3-3\phi)_{ij}^n. \quad (11)$$

2.3. Mass conservation

We verify that the compact scheme inherits the total mass conservation. Taking the inner product to Eq. (10) with a constant grid function **1**, we get

$$\left(\phi^{n+1},\mathbf{1}\right)_h+\frac{h^2}{12}\left(\Delta_c\phi^{n+1},\mathbf{1}\right)_h-\Delta t\left(\Delta_c\mu^{n+1},\mathbf{1}\right)_h=\left(\phi^{n+1},\mathbf{1}\right)_h+\frac{h^2}{12}\left(\Delta_c\phi^{n},\mathbf{1}\right)_h.$$

For $(\Delta_c\phi^n,\mathbf{1})_h$, we have

$$\begin{aligned}\left(\Delta_c\phi^n,\mathbf{1}\right)_h&=h^2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\Delta_c\phi^n_{ij}\\&=h^2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left(\frac{D_x\phi^n_{i+\frac{1}{2},j}-D_x\phi^n_{i-\frac{1}{2},j}}{h}+\frac{D_x\phi^n_{i,j+\frac{1}{2}}-D_x\phi^n_{i,j-\frac{1}{2}}}{h}\right)\\&=h\sum_{j=1}^{N_y}\left(D_x\phi^n_{N_x+\frac{1}{2},j}-D_x\phi^n_{\frac{1}{2},j}\right)+h\sum_{i=1}^{N_x}\left(D_x\phi^n_{i,N_y+\frac{1}{2}}-D_x\phi^n_{i,\frac{1}{2}}\right)=0.\end{aligned}$$

Here, we have used the periodic boundary condition for ϕ, and $\left(\Delta_c\phi^{n+1},\mathbf{1}\right)_h=(\Delta_c\mu^n,\mathbf{1})_h=0$ can be proved in a similar manner. Thus, we have the mass conserving property, i.e., $(\phi^n,\mathbf{1})_h=\left(\phi^{n+1},\mathbf{1}\right)_h$.

2.4. Multigrid algorithm

We briefly describe the multigrid method and implementation to solve the resulting system. We represent the discrete CH system as

$$L_h(\phi^{n+1},\mu^{n+1})=(\xi^n,\psi^n),$$

where the linear operator L_h is defined as

$$\begin{aligned}L_h(\phi^{n+1},\mu^{n+1})=\Bigg(&\frac{\phi^{n+1}_{ij}}{\Delta t}+\frac{h^2}{12\Delta t}\Delta_c\phi^{n+1}_{ij}-\Delta_c\mu^{n+1}_{ij},\\&-\frac{2}{\epsilon^2}\phi^{n+1}_{ij}+\left(1-\frac{h^2}{6\epsilon^2}\right)\Delta_c\phi^{n+1}_{ij}+\frac{1}{\epsilon^2}\mu^{n+1}_{ij}+\frac{h^2}{12\epsilon^2}\Delta_c\mu^{n+1}_{ij}\Bigg),\end{aligned}$$

and the source term is

$$\left(\xi^n,\psi^n\right)=\left(\frac{\phi^n_{ij}}{\Delta t}+\frac{h^2}{12\Delta t}\Delta_c\phi^n_{ij},\ \frac{1}{\epsilon^2}(\phi^3-3\phi)^n_{ij}+\frac{h^2}{12\epsilon^2}\Delta_c(\phi^3-3\phi)^n_{ij}\right).$$

2.4.1. Smoothing

Compute $\left(\bar\phi_k,\bar\mu_k\right)$ by applying ν smoothing procedures to (ϕ_k,μ_k).

$$\left(\bar\phi_k,\bar\mu_k\right)=\text{SMOOTH}^{\nu}\left(\phi_k,\mu_k,L_h,\xi_k,\psi_k\right)$$

on a mesh grid Ω_k. The SMOOTH$^{\nu}$ function means that it performs a SMOOTH relaxation operator with approximations ϕ_k and μ_k, and source terms ξ_k and ψ_k. The superscript ν denotes how many times the given relaxation operator is applied to obtain the updated approximations $\left(\bar\phi_k,\bar\mu_k\right)$. This relaxation step is evaluated using pointwise Gauss–Seidel iterative methods. One SMOOTH relaxation operator step is completed by solving the system by a 2 × 2 matrix inversion for each i and j.

2.4.2. V-cycle

One V-cycle step comprises the presmoothing, coarse grid correction, and postsmoothing steps. Please refer to the reference text for additional details and background [13].

$$\left(\phi_k^{n+1,m+1},\mu_k^{n+1,m+1}\right)=\text{V-cycle}(k,\phi_k^{n+1,m},\mu_k^{n+1,m},L_h,\xi_k^n,\psi_k^n,\nu_1,\nu_2)$$

where $\phi_k^{n+1,m+1}$ and $\phi_k^{n+1,m}$ are the approximations of ϕ_k^{n+1} before and after the V-cycle. Next, we define the V-cycle.

Presmoothing

$$(\bar\phi_k^{n+1,m},\bar\mu_k^{n+1,m})=\text{SMOOTH}^{\nu_1}(\phi_k^{n+1,m},\mu_k^{n+1,m},L_h,\xi_k^n,\psi_k^n).$$

Coarse grid correction

(1) Find the defect: $\left(\bar d^m_{1,k},\bar d^m_{2,k}\right)=\left(\xi^n_k,\psi^n_k\right)-L_h\left(\bar\phi_k^{n+1,m},\bar\mu_k^{n+1,m}\right)$.

(2) Restrict the defect: $\bar d^m_{1,k-1}=I_k^{k-1}\bar d^m_{1,k}$, $\bar d^m_{2,k-1}=I_k^{k-1}\bar d^m_{2,k}$.

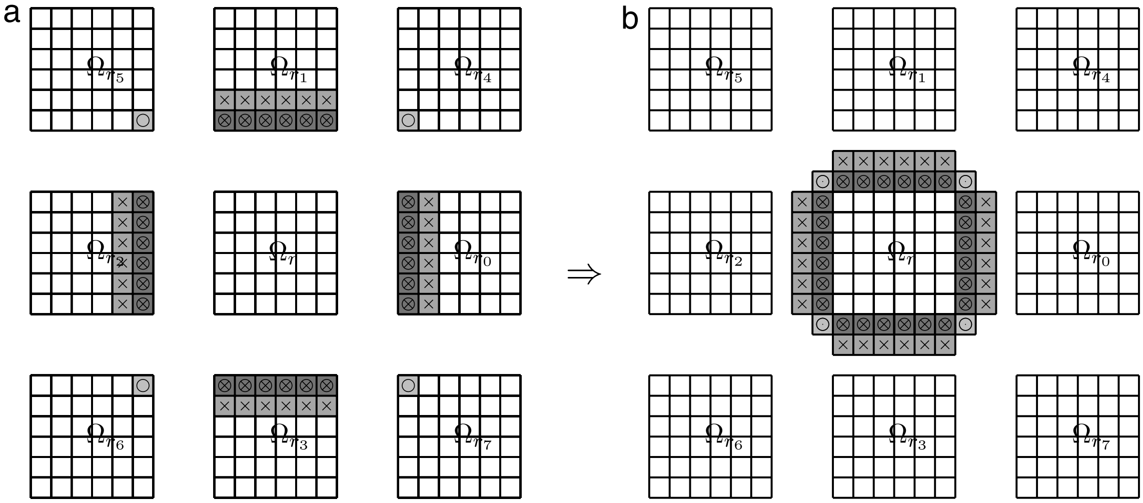


Fig. 1. Data communication between adjacent sub-domains. (a) Ω_{r_i} sends data to Ω_r, and (b) Ω_r receives the data from Ω_{r_i}. Circles and crosses are corresponding to the data for communication of CNPF and NPF, respectively.

(3) Evaluate approximations $\left(\hat{v}^{n+1,m}_{1,k-1}, \hat{v}^{n+1,m}_{2,k-1}\right)$ of the following coarse grid system on Ω_{k-1}: $L_{2h}\left(\hat{v}^{n+1,m}_{1,k-1}, \hat{v}^{n+1,m}_{2,k-1}\right) = \left(\bar{d}^{m}_{1,k-1}, \bar{d}^{m}_{2,k-1}\right)$. If $k > 1$, then we can solve the coarse grid system using the zero grid functions as initial approximations and the defect functions as source terms

$$\left(\hat{v}^{n+1,m}_{1,k-1}, \hat{v}^{n+1,m}_{2,k-1}\right) = \text{V-cycle}\left(k-1, 0, 0, L_{2h}, \bar{d}^{m}_{1,k-1}, \bar{d}^{m}_{2,k-1}, \nu_1, \nu_2\right).$$

Otherwise, we apply the smoothing procedure to obtain the approximations.

(4) Interpolate the correction: $\hat{v}^{n+1}_{1,k} = I^{k}_{k-1}\hat{v}^{n+1,m}_{1,k-1}$, $\hat{v}^{n+1,m}_{2,k} = I^{k}_{k-1}\hat{v}^{n+1,m}_{2,k-1}$.

(5) Compute the corrected approximation on Ω_k:

$$\left(\widetilde{\phi}^{n+1,m}_{k}, \widetilde{\mu}^{n+1,m}_{k}\right) = \left(\bar{\phi}^{n+1,m}_{k}, \bar{\mu}^{n+1,m}_{k}\right) + \left(\hat{v}^{n+1,m}_{1,k}, \hat{v}^{n+1,m}_{2,k}\right).$$

Postsmoothing

$$\left(\phi^{n+1,m+1}_{k}, \mu^{n+1,m+1}_{k}\right) = \text{SMOOTH}^{\nu_2}\left(\widetilde{\phi}^{n+1,m}_{k}, \widetilde{\mu}^{n+1,m}_{k}, L_h, \xi^{n}_{k}, \psi^{n}_{k}\right).$$

This completes the description of the V-cycle.

2.5. Parallel computation

In industrial sciences, parallel algorithms have been developed for large-data applications. Parallel computation can divide a large domain into smaller sub-domains, and carry out many calculations simultaneously. We apply CNPF and NPF to the parallel multigrid method [43], which is based on the grid partitioning strategy and data communication. Fig. 1 illustrates the data communication. Let us consider the rth sub-domain, Ω_r, and we define its adjacent sub-domains $\Omega_{r_0}, \Omega_{r_1}, \ldots, \Omega_{r_7}$. Shaded layers are sub-boundaries which need the data communication because of the Laplacian operator. Communications are performed before the smoothing and defect procedures. On the other hand, no data communication is needed before restriction and prolongation. The Message Passing Interface (MPI) [44] is used for the interprocessor communication.

2.6. Adaptive mesh refinement

The AMR method [45,46] is more efficient than a method based on the uniform mesh because it allows a multi-resolution in interest regions without requiring a fine grid resolution of the whole domain. With its advantage, they are widely used [21,47–50]. In the adaptive approach, we introduce a hierarchy of grids, $\Omega_0, \Omega_1, \Omega_2, \ldots, \Omega_{l+0}, \Omega_{l+1}, \ldots, \Omega_{l+l^*}$. Here, $\Omega_0, \Omega_1, \ldots, \Omega_l$ denote a hierarchy of global grids in the non-adaptive multigrid on uniform grids and $\Omega_{l+0}, \Omega_{l+1}, \ldots, \Omega_{l+l^*}$ denote a hierarchy of increasingly finer grids which are restricted to smaller and smaller sub-domains around the phase-field interfacial transition zone. Fig. 2 shows the grid structure used in AMR [45] with $l^* = 3$. Here, contours represent the interface of phase field, i.e., $\phi \in [-0.99, 0.99]$.

When the new level is changed, we have to fill the data in the new level from the previous level. In this time, the ghost-layer values are obtained by the interpolation process from the previous level data. The main idea of the interpolation process is to use the quadratic function. First, we can get the intermediate values (solid circles) by using the quadratic interpolation

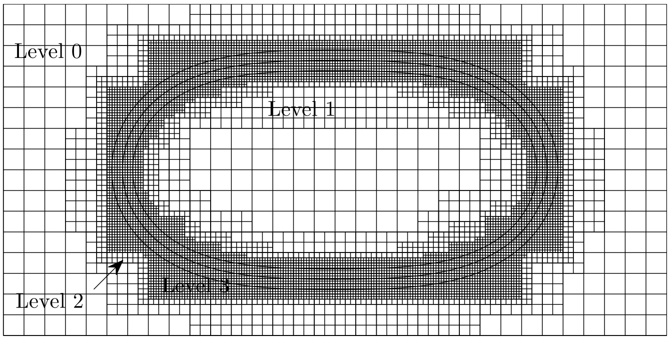


Fig. 2. Block-structured local refinement with the level $l^* = 3$. Contours represent the interface of phase-field, i.e., $\phi \in [-0.99, 0.99]$.

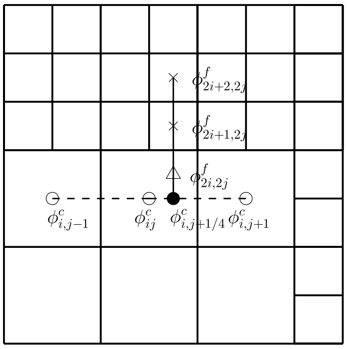


Fig. 3. Coarse–fine boundary interpolation. First, the intermediate value (solid circle) is obtained by the coarse grid points (open circles). Then, the ghost cell value (open triangle) is obtained by the intermediate value (solid circle) and the fine grid points ($\times$'s).

with the data in coarse grid (open circles). Then, the ghost cell values (open triangles) are obtained in the same manner with the fine cells ($\times$'s) and the intermediate values (solid circles), see Fig. 3.

3. Numerical results

In this section, various numerical experiments are given to demonstrate the fourth-order convergence, non-increase of total energy, mass conservation, linear stability analysis, and robustness of the scheme. In addition, we present the evolution up to the steady state, compare with three numerical methods, and show the applicability to the parallel computation and the AMR method.

3.1. Convergence test

A numerical convergence test for the three schemes is performed with increasingly finer grids $h = 1/2^n$, for $n = 3, 4, 5$, and 6. The initial state is defined as $\phi(x, y, 0) = 0.1\cos(2\pi x)\cos(2\pi y)$ in $\Omega = (0, 1) \times (0, 1)$ (Fig. 4(a)), and Fig. 4(b) illustrates the numerical solution at $T = 24\Delta t$, where we use $\epsilon = 0.0075$ and $\Delta t = 6 \times 10^{-4}$. We consider a reference solution, because it is generally hard to find the exact solution of the CH equation. We define the reference solution ϕ^{ref}_{ij} by the local average of the numerical solution on a much finer grid, and then denote the error by $e_{ij} := \phi_{ij} - \phi^{ref}_{ij}$. We use a 1024×1024 mesh grid and FPF for the reference solution ϕ^{ref}_{ij}. The convergence rate is defined as the ratio of successive errors, $\log_2(\|e_h\|/\|e_{h/2}\|)$, where $\|e_h\|$ is the discrete l_2-norm of error function e_h.

To compare the accuracy, we use the five-point formula (FPF), nine-point formula (NPF), and compact nine-point formula (CNPF) for the calculation of the Laplacian operator. Table 1 lists the discrete l_2-norm of errors and convergence rates with different three formulas. Using NPF and CNPF, we have the fourth-order accuracy in space as we expect from the discretization.

3.2. Non-increase of total energy and conservation of mass

Fig. 5 demonstrates that the discrete total energy is monotonically decreasing and the mass is conserved. The inscribed small figures show the phase separation at the indicated times. For the numerical test, in $\Omega = (0, 1) \times (0, 1)$, the initial state is taken as a random perturbation $\phi(x, y, 0) = 0.5\,\text{rand}(x, y)$, where $\text{rand}(x, y)$ is a random value, which is uniformly distributed between -1 and 1. For other parameters, $h = 1/256$, $\Delta t = 0.002$, $T = 3$, and $\epsilon = 0.0038$ are used.

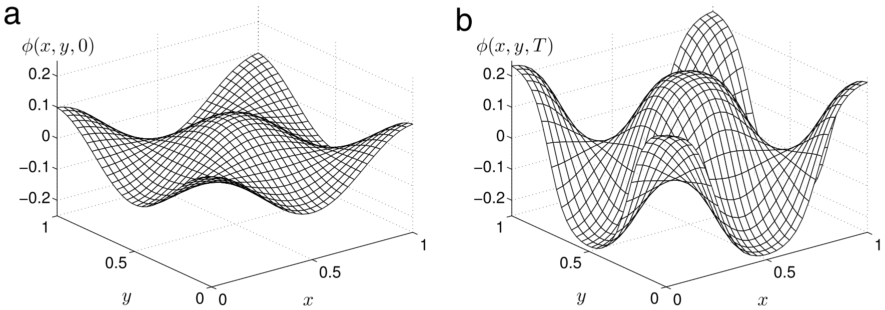


Fig. 4. (a) Initial condition $\phi(x, y, 0)$ and (b) numerical solution $\phi(x, y, T)$ at time $T = 0.0144$.

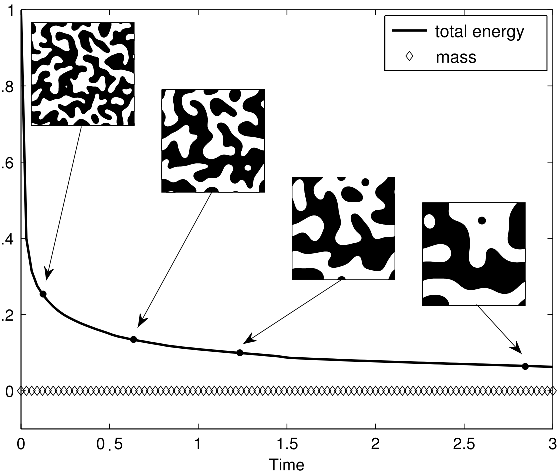


Fig. 5. Non-dimensional discrete total energy $\mathcal{E}^h(\phi^n)/\mathcal{E}^h(\phi^0)$ and mass concentration with the initial condition $\phi(x, y, 0) = 0.5\,\text{rand}\,(x, y)$.

Table 1
l_2-norm errors and convergence rates for FPF, NPF, and CNPF.

h	FPF		NPF		CNPF	
	Error	Order	Error	Order	Error	Order
1/8	1.03×10^{-2}		8.14×10^{-3}		5.02×10^{-3}	
1/16	2.76×10^{-3}	1.90	6.59×10^{-4}	3.63	3.36×10^{-4}	3.90
1/32	7.26×10^{-4}	1.93	4.54×10^{-5}	3.86	2.01×10^{-5}	4.06
1/64	1.83×10^{-4}	1.99	2.31×10^{-6}	4.30	1.27×10^{-6}	3.98

3.3. Linear stability analysis

Let us consider the linear stability analysis for CH equations (1) and (2)

$$\phi_t = \Delta\left(\phi^3 - \phi - \epsilon^2\Delta\phi\right), \quad \mathbf{x} \in \Omega, t > 0, \tag{12}$$

where $\Omega = (0, 2\pi) \times (0, 2\pi)$. We assume that the solution can be expressed by

$$\phi(x, y, t) = \bar{\phi} + \sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty} \beta_{k_1k_2}(t)\cos(k_1x)\cos(k_2y) + \sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty} \gamma_{k_1k_2}(t)\sin(k_1x)\sin(k_2y), \tag{13}$$

where $\bar{\phi}$ is the average of ϕ, and $\beta_{k_1k_2}(t)$ and $\gamma_{k_1k_2}(t)$ are amplification factors at wave numbers k_1 and k_2. After linearizing Eq. (12) and substituting Eq. (13) into the linearized equation, we have

$$\frac{\mathrm{d}\beta_{k_1k_2}(t)}{\mathrm{d}t} = k^2\left(1 - 3\bar{\phi}^2 - \epsilon^2k^2\right)\beta_{k_1k_2}(t), \tag{14}$$

where we denote $k^2 = k_1^2 + k_2^2$. We only consider $\beta_{k_1k_2}(t)$ because the same ordinary differential equation holds for $\gamma_{k_1k_2}(t)$. The solution of Eq. (14) is $\beta_{k_1k_2}(t) = \beta_{k_1k_2}(0)\exp(\eta_{k_1k_2}t)$, where $\eta_{k_1k_2} = k^2(1 - 3\bar{\phi}^2 - \epsilon^2k^2)$ is the growth rate.

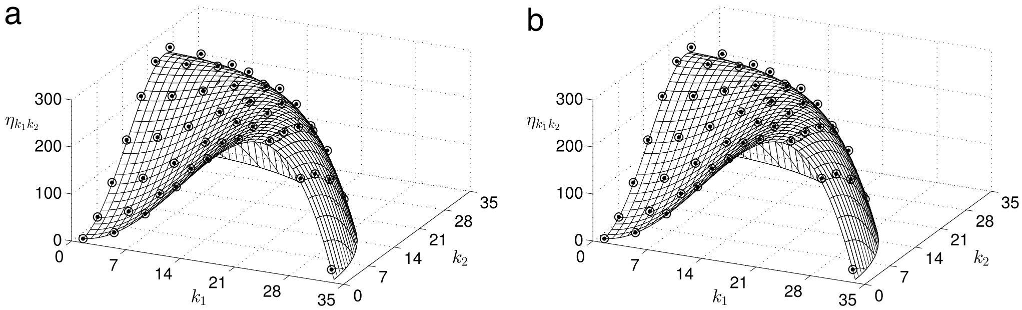
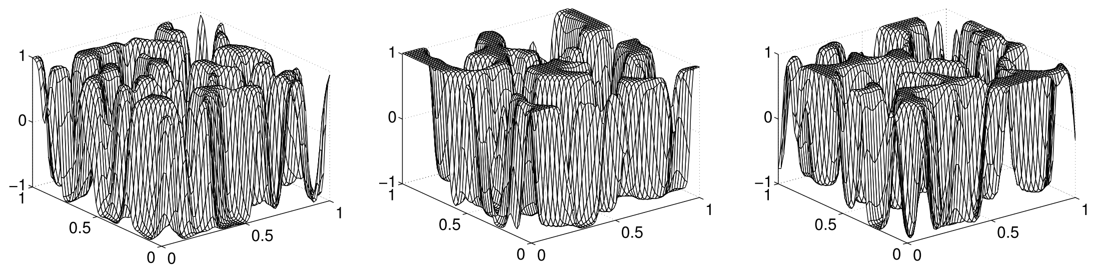


Fig. 6. Growth rate versus the wave numbers k_1 and k_2 for the (a) CH and (b) linearized CH equations.

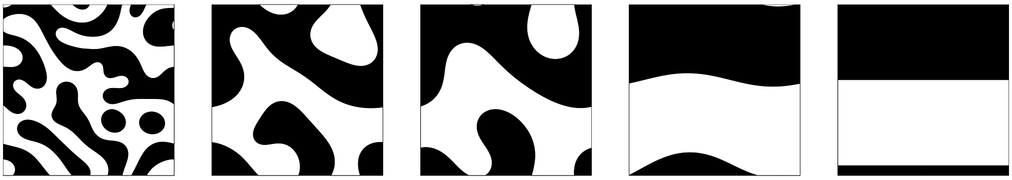


(a) $\Delta t = 0.01$.

(b) $\Delta t = 10$.

(c) $\Delta t = 10000$.

Fig. 7. Evolutions with different time steps $\Delta t = 0.01$, 10, and 10000.



$t = 0.076$ $t = 0.382$ $t = 0.763$ $t = 3.052$ $t = 152.6$

Fig. 8. Evolution of ϕ up to the steady state time $t = 152.6$. The times are shown below in each figure.

The numerical growth rate is defined as $\widetilde{\eta}_{k_1k_2} = \log(\|\phi^m\|_\infty/\|\phi^0\|_\infty)/(m\Delta t)$. We take the initial condition $\phi(x, y, 0) = 0.01\cos(k_1x)\cos(k_2y)$ with $m = 100$, $\Delta t = 10^{-8}$, $h = \pi/256$, and $\epsilon = 0.03$. Fig. 6(a) and (b) show the numerical growth rate $\widetilde{\eta}_{k_1k_2}$ versus the wave numbers k_1 and k_2 for the CH and linearized CH equations, respectively. Open and closed circles correspond to the solutions from the linear stability analysis and CNPF, respectively. The numerical results are in good agreement with the analytic solutions from the linear stability analysis.

3.4. Stability of the proposed scheme

We demonstrate a practical stability of the scheme through a numerical experiment with spinodal decomposition of a binary mixture. In the simulation, the initial condition is taken as $\phi(x, y, 0) = 0.5\,\mathrm{rand}(x, y)$ in $\Omega = (0, 1) \times (0, 1)$. Note that the maximum amplitude is 0.5 at the initial time. For numerical parameters, $h = 1/128$ and $\epsilon = 0.0113$ are used and different time steps $\Delta t = 0.01$, 10, and 10000 are employed. In Fig. 7, we illustrate the evolutions after fifteen time iterations. As the numerical results, the maximum amplitudes are bounded, and the numerical solutions do not blow up. Therefore, our proposed scheme is stable regardless of the time step size.

3.5. Steady state

We examine the evolution of a random perturbation up to the steady state. The initial condition is taken to be $\phi(x, y, 0) = 0.01\mathrm{rand}(x, y)$ in $\Omega = (0, 1) \times (0, 1)$. We then take the simulation parameters as $\epsilon = 0.0075$, $h = 1/256$, and $\Delta t = 10h^2$. We stop the numerical computations when the discrete l_2-norm of the difference between $(n + 1)$th and nth step solutions becomes less than 10^{-9}, i.e., $\|\phi^{n+1} - \phi^n\| \le 10^{-9}$. Fig. 8 shows the snapshots of filled contour of the concentration ϕ. We observe that the randomly perturbed concentration ϕ evolves to a complex interconnected pattern. After a long time evolution, a numerical equilibrium state is reached.

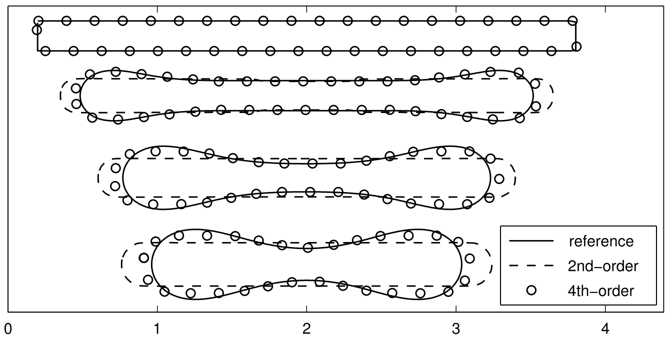


Fig. 9. Evolutions for FPF and CNPF with a mesh grid 256 × 64. Note that the reference solution is defined by using much more finer mesh 1024 × 256. From top to bottom, times are $t = 0, 2, 6$, and 10.

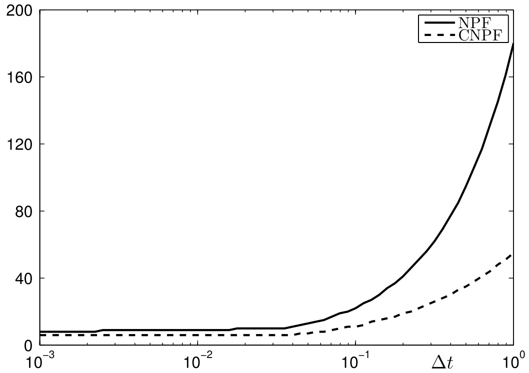


Fig. 10. V-cycle number with different time steps Δt.

Table 2
CPU time (s) for FPF, NPF, and CNPF.

FPF	NPF	CNPF
1819	3095	2218

3.6. Comparison of three methods

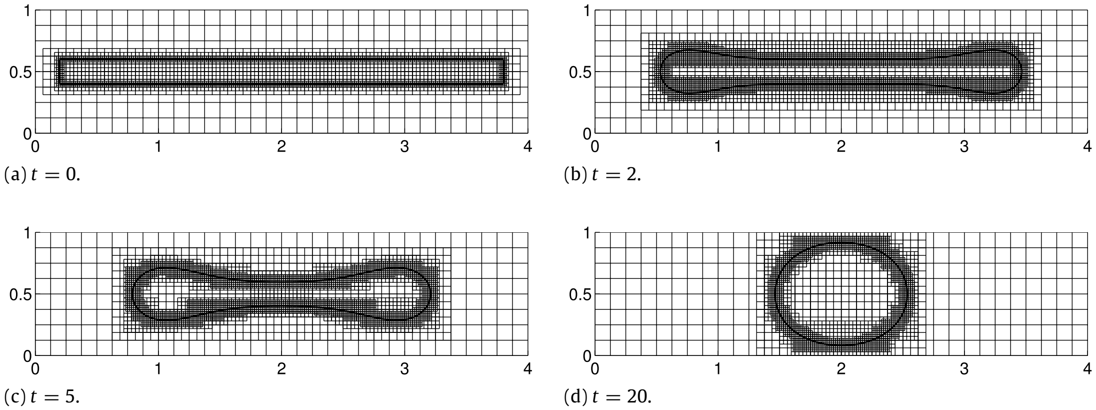
To show the superiority of CNPF, we compare the evolution results from CNPF and FPF, and list the computational times for the three methods. The initial condition is given as

$$\phi(x, y, 0) = \begin{cases} 1 & \text{if } 0.2 < x < 3.8 \text{ and } 0.4 < y < 0.6, \\ -1 & \text{otherwise} \end{cases} \tag{15}$$

in $\Omega = (0, 4) \times (0, 1)$. Here, a mesh grid 256 × 64, $\epsilon = 0.015$, $\Delta t = 0.0005$, and $T = 10$ are used. We define the reference solution by numerical solutions, using FPF, on a finer mesh 1024 × 256. In Fig. 9, we illustrate evolutions by CNPF (circles) and FPF (dashed line) with the reference solutions (solid line) at $t = 0, 2, 6$, and 10. Table 2 lists CPU times of the three schemes to $T = 10$. Although CNPF has a slight difference of CPU time than FPF, it has higher accuracy compared to FPF. Furthermore, NPF needs more V-cycle iterations than CNPF to reach the V-cycle tolerance, and it results in the increment of CPU time.

3.7. Comparison between CNPF and NPF

We compare the numerical convergence of two formulas, CNPF and NPF. For the numerical test, we take the initial condition as $\phi(x, y, 0) = 0.1\cos(2\pi x)\cos(2\pi y)$ in $\Omega = (0, 1) \times (0, 1)$. For other parameters, we use $h = 1/64$ and $\epsilon = 0.015$. We count the number of V-cycle until the maximum value of residual error is less than 10^{-10}. Fig. 10 plots the numbers of V-cycle for one time iteration versus Δt. From the results, we can observe that CNPF requires less V-cycle iterations than NPF for all time step sizes.



(a) $t = 0$.

(b) $t = 2$.

(c) $t = 5$.

(d) $t = 20$.

Fig. 11. Contours and adaptive meshes of ϕ. The times are shown below in each figure. The zero-level grid Ω_{l+0} has 128×32 points and three levels of mesh refinement are used as $l^* = 3$ in $\Omega = (0, 4) \times (0, 1)$.

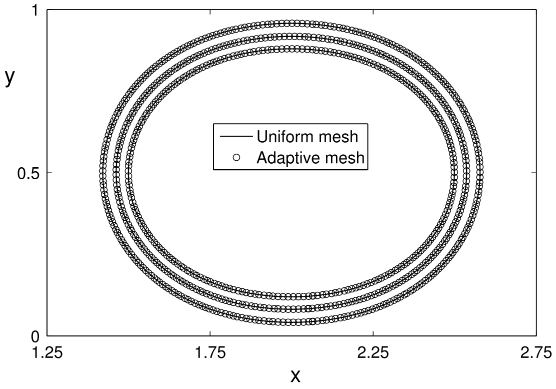


Fig. 12. Contours at $\phi = -0.95, 0, 0.95$ at $t = 20$. Solid lines and circles represent the results on the uniform and adaptive meshes, respectively.

3.8. Parallel computation

The parallel performance is usually evaluated based on speed-up. The value of the speed-up is used to measure the ratio of the time spent in the serial mode to that spent in the parallel mode. Let $T(P)$ be the execution time, i.e., the computational time, using P processors. The speed-up is defined as $S(P) = T(1)/T(P)$. Ideally, we expected $S(P) \approx P$ for the parallel algorithm.

The execution time is the CPU time taken to reach $T = 100\Delta t$. The calculations are performed with the initial data $\phi(x, y, 0) = 0.1 \cos(2\pi x) \cos(2\pi y)$ in $\Omega = (0, 1) \times (0, 1)$. For other parameters, $h = 1/1024$, $\Delta t = h^2$, and $\epsilon = 0.0019$ are used. Table 3 lists the CPU time and speed-up for two formulas. Because the NPF has a disadvantage for the data communication, the speed-up of CNPF shows the better performance than the one of NPF.

3.9. Adaptive mesh refinement

In this section, we compare the numerical results obtained on uniform and adaptive meshes with CNPF. For the numerical test, we use the initial condition (15) on the computational domain $\Omega = (0, 4) \times (0, 1)$. The other parameters are taken as $\Delta t = 0.001$ and $\epsilon = 0.015$. And we use 1024×256 grid points on uniform mesh. On adaptive mesh, we use the zero-level grid Ω_{l+0} which has 128×32 grid points and the three levels for mesh refinement as $l^* = 3$. Fig. 11 shows the evolution using the AMR method and dynamical adjustment of grid hierarchy structure around interface transition region.

In Fig. 12, contours of results by uniform and adaptive meshes are drawn and they are in a good agreement. The numerical test on the uniform mesh requires more than 10.48 h. The test on adaptive mesh needs only 1.15 h, which is about 9 times faster than the uniform mesh.

4. Conclusions

We proposed the fourth-order spatial accurate and practically stable compact scheme, by using the compact nine-point formula (CNPF) and linearly stabilized splitting scheme, to solve the Cahn–Hilliard equation. We numerically demonstrated

Table 3
CPU time (s) and speed-up with different number of processor.

P	NPF		CNPF	
	$T(P)$	$S(P)$	$T(P)$	$S(P)$
1	275.2	1.00	227.9	1.00
2	138.8	1.98	114.9	1.98
4	70.4	3.91	58.0	3.93
8	37.2	7.39	29.8	7.66
16	18.6	14.8	14.8	15.4
32	9.8	28.2	7.77	29.3

the practical stability, fourth-order accuracy, decrease of total energy, and mass conservation. Also, to show the superiority of CNPF, we compared the standard nine-point formula (NPF) with CNPF. The computational time of CNPF was less than that of NPF because NPF needs more V-cycle iterations than CNPF to reach the tolerance. We then applied NPF and CNPF to the parallel multigrid method. It was found that NPF has a disadvantage on the data communication due to the wide stencil. Furthermore, we demonstrated the applicability of CNPF to the adaptive mesh refinement method.

Acknowledgment

The corresponding author (J.S. Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2011-0023794). The authors thank the reviewers for the constructive and helpful comments on the revision of this article.

References

[1] J.W. Cahn, J.E. Hilliard, Free energy of a non-uniform system I. Interfacial free energy, J. Chem. Phys. 28 (1958) 258–267.
[2] J.W. Cahn, Free energy of a nonuniform system II: thermodynamic basis, J. Chem. Phys. 30 (1959) 1121–1124.
[3] I.C. Dolcetta, S.F. Vita, R. March, Area-preserving curve-shortening flows: from phase separation to image processing, Interfaces Free Bound. 4 (2002) 325–343.
[4] A.L. Bertozzi, S. Esedoglu, A. Gillette, Inpainting of binary imagesusing the Cahn–Hilliard equation, IEEE Trans. Image Process. 16 (2007) 285–291.
[5] L.Q. Chen, Phase-field models for microstructure evolution, Ann. Rev. Mater. Res. 32 (2002) 113–140.
[6] J. Kim, A continuous surface tension force formulation for diffuse-interface models, J. Comput. Phys. 204 (2005) 784–804.
[7] S. Tremaine, On the origin of irregular structure in Saturn's rings, Astron. J. 125 (2003) 894–901.
[8] S. Wise, J. Lowengrub, H. Frieboes, V. Cristini, Three-dimensional multispecies nonlinear tumor growth: I. Model and numerical method, J. Theoret. Biol. 253 (2008) 524–543.
[9] V. Cristini, X. Li, J.S. Lowengrub, S.M. Wise, Nonlinear simulations of solid tumor growth using a mixture model: invasion and branching, J. Math. Biol. 58 (2009) 723–763.
[10] C.M. Elliott, The Cahn–Hilliard model for the kinetics of phase separation, in: J.F. Rodrigues (Ed.), Mathematical Models for Phase Change Problems, in: International Series of Numerical Mathematics, vol. 88, Birkhauser-Verlag, Basel, 1989, pp. 35–73.
[11] M. Copetti, C.M. Elliott, Kinetics of phase decomposition processes: numerical solutions to the Cahn–Hilliard equation, Labour Hist. Rev. 6 (1990) 273–283.
[12] D. Furihata, A stable and conservative finite difference scheme for the Cahn–Hilliard equation, Numer. Math. 87 (2001) 675–699.
[13] J. Kim, K. Kang, J. Lowengrub, Conservative multigrid methods for Cahn–Hilliard fluids, J. Comput. Phys. 193 (2004) 511–543.
[14] E.V.L. de Mello, O.T. da, S. Filho, Numerical study of the Cahn–Hilliard equation in one, two and three dimensions, Physica A 347 (2005) 429–443.
[15] H.G. Lee, J. Kim, A second-order accurate non-linear difference scheme for the N-component Cahn–Hilliard system, Physica A 387 (2008) 4787–4799.
[16] H.G. Lee, J.-W. Choi, J. Kim, A practically unconditionally gradient stable scheme for the N-component Cahn–Hilliard system, Physica A 391 (2012) 1009–1019.
[17] L. Chen, J. Shen, Applications of semi-implicit Fourier-spectral method to phase field equations, Comput. Phys. Comm. 108 (1998) 147–158.
[18] J. Zhu, L.Q. Chen, J. Shen, V. Tikare, Coarsening kinetics from a variable-mobility Cahn–Hilliard equation: application of a semi-implicit Fourier spectral method, Phys. Rev. E 60 (1999) 3564–3572.
[19] V.E. Badalassi, H.D. Ceniceros, S. Banerjee, Computation of multiphase systems with phase field models, J. Comput. Phys. 190 (2003) 371–397.
[20] D. Kay, R. Welford, A multigrid finite element solver for the Cahn–Hilliard equation, J. Comput. Phys. 212 (2006) 288–304.
[21] D.F. Martin, P. Colella, M. Anghel, F.L. Alexander, Adaptive mesh refinement for multiscale nonequilibrium physics, Comput. Sci. Eng. 7 (2005) 24–31.
[22] Y. Wang, J. Zhang, Fast and robust sixth-order multigrid computation for the three-dimensional convection–diffusion equation, J. Comput. Appl. Math. 234 (2010) 3496–3506.
[23] S. Kim, Compact schemes for acoustics in the frequency domain, Math. Comput. Modelling 37 (2003) 1335–1341.
[24] E. Turkel, D. Gordon, R. Gordon, S. Tsynkov, Compact 2D and 3D sixth order schemes for the Helmholtz equation with variable wave number, J. Comput. Phys. 232 (2013) 272–287.
[25] J. Nordstrom, R. Gustafsson, High order finite difference approximations of electromagnetic wave propagation close to material discontinuities, J. Sci. Comput. 18 (2003) 215–234.
[26] B. Düring, M. Fournié, High-order compact finite difference scheme for option pricing in stochastic volatility models, J. Comput. Appl. Math. 236 (2012) 4462–4473.
[27] M.M. Gupta, J. Kouatchou, J. Zhang, Comparison of second and fourth order discretizations for multigrid Poisson solver, J. Comput. Phys. 132 (1997) 226–232.
[28] J. Zhang, Multigrid method and fourth order compact difference scheme for 2D Poisson equation with unequal meshsize discretization, J. Comput. Phys. 179 (2002) 170–179.
[29] S. Zhai, X. Feng, Y. He, A family of fourth-order and sixth-order compact difference schemes for the three-dimensional Poisson equation, J. Sci. Comput. 54 (2013) 97–120.
[30] H. Sun, J. Zhang, A high-order compact boundary value method for solving one-dimensional heat equations, Numer. Methods Partial Differential Equations 19 (2003) 846–857.
[31] M. Li, T. Tang, A compact fourth-order finite difference scheme for unsteady viscous incompressible flows, J. Sci. Comput. 16 (2001) 29–45.
[32] W.F. Spotz, G.F. Carey, High-order compact scheme for the steady stream-function vorticity equations, Int. J. Numer. Methods Eng. 38 (1995) 3497–3512.

[33] M. Li, T. Tang, B. Fornberg, A compact fourth-order finite difference scheme for the steady incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 20 (1995) 1137–1151.
[34] J.W. Stephenson, Single cell discretizations of order two and four for biharmonic problems, J. Comput. Phys. 55 (1984) 65–80.
[35] I. Altas, J. Dym, M.M. Gupta, R. Manohar, Multigrid solution of automatically generated high-order discretizations for the biharmonic equation, SIAM J. Sci. Comput. 19 (1998) 1575–1585.
[36] C. Clavero, J.L. Gracia, A higher order uniformly convergent method with Richardson extrapolation in time for singularly perturbed reaction–diffusion parabolic problems, J. Comput. Appl. Math. 252 (2013) 75–85.
[37] A.C. Radhakrishna Pillai, Fourth-order exponential finite difference methods for boundary value problems of convective diffusion type, Int. J. Numer. Methods Fluids 37 (2001) 87–106.
[38] Z.F. Tian, S.Q. Dai, High-order compact exponential finite difference methods for convection–diffusion type problems, J. Comput. Phys. 220 (2007) 952–974.
[39] Y. Wang, J. Zhang, Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation, J. Comput. Phys. 228 (2009) 137–146.
[40] J. Li, Z. Sun, X. Zhao, A three level linearized compact difference scheme for the Cahn–Hilliard equation, Sci. China Math. 55 (2012) 805–826.
[41] D.J. Eyre, An unconditionally stable one-step scheme for gradient systems, Preprint. Available from: http://www.math.utah.edu/~eyre/research/methods/stable.ps.
[42] R.J. LeVeque, Finite Difference Methods for Differential Equations, in: Lecture Notes, University of Washington, 1998.
[43] J. Shin, D. Lee, S. Kim, J. Kim, A parallel multigrid method for the Cahn–Hilliard equation, Comput. Mater. Sci. 71 (2013) 89–96.
[44] Message Passing Interface Forum, MPI, A Message Passing Interface Standard, 1994.
[45] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1998) 1–46.
[46] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, An adaptive level set approach for incompressible two-phase flows, J. Comput. Phys. 148 (1999) 81–124.
[47] Y. Li, J. Kim, Phase-field simulations of crystal growth with adaptive mesh refinement, Int. J. Heat Mass Transfer 55 (2012) 7926–7932.
[48] J. Kim, H. Bae, An unconditionally gradient stable adaptive mesh refinement for Cahn–Hilliard equation, J. Korean Phys. Soc. 53 (2008) 672–679.
[49] J. Kim, Phase-field models for multi-component fluid flows, Commun. Comput. Phys. 12 (2012) 613–661.
[50] S. Wise, J. Kim, J. Lowengrub, Solving the regularized, strongly anisotropic Cahn–Hilliard equation by an adaptive nonlinear multigrid method, J. Comput. Phys. 226 (2007) 414–446.