

Article

An Efficient and Accurate Method for the Conservative Swift–Hohenberg Equation and Its Numerical Implementation

Hyun Geun Lee

Department of Mathematics, Kwangwoon University, Seoul 01897, Korea; leeh1@kw.ac.kr

Received: 21 August 2020; Accepted: 2 September 2020; Published: 4 September 2020



Abstract: The conservative Swift–Hohenberg equation was introduced to reformulate the phase-field crystal model. A challenge in solving the conservative Swift–Hohenberg equation numerically is how to treat the nonlinear term to preserve mass conservation without compromising efficiency and accuracy. To resolve this problem, we present a linear, high-order, and mass conservative method by placing the linear and nonlinear terms in the implicit and explicit parts, respectively, and employing the implicit-explicit Runge–Kutta method. We show analytically that the method inherits the mass conservation. Numerical experiments are presented demonstrating the efficiency and accuracy of the proposed method. In particular, long time simulation for pattern formation in 2D is carried out, where the phase diagram can be observed clearly. The MATLAB code for numerical implementation of the proposed method is provided in Appendix .

Keywords: conservative swift–hohenberg equation; linear method; high-order time accuracy; mass conservation; fourier spectral method

1. Introduction

The phase-field crystal (PFC) model describes the microstructure of two-phase systems on atomic length and diffusive time scales and has been used to study grain growth, dendritic and eutectic solidification, and epitaxial growth [1,2]. The PFC model is the H^{-1} -gradient flow for the Swift–Hohenberg (SH) energy functional [3]:

$$\mathcal{E}(\phi) := \int_{\Omega} \left(\Phi(\phi) - |\nabla \phi|^2 + \frac{1}{2} (\Delta \phi)^2 \right) d\mathbf{x},\tag{1}$$

where Ω is a domain in \mathbb{R}^d (d = 1, 2, 3), ϕ is the density field, $\Phi(\phi) = \frac{1}{4}\phi^4 - \frac{g}{3}\phi^3 + \frac{1-\epsilon}{2}\phi^2$, and $g \ge 0$ and $\epsilon > 0$ are positive constants with physical significance.

Recently, conservative SH equations were introduced to reformulate the PFC model [4,5]. In [4], Zhang and Yang derived the following equation:

$$\frac{\partial \phi}{\partial t} = -\left(\Phi'(\phi) + 2\Delta\phi + \Delta^2\phi\right) + \tilde{\beta}(t),\tag{2}$$

where $\tilde{\beta}(t)$ is a nonlocal Lagrange multiplier and $\tilde{\beta}(t) = \frac{1}{|\Omega|} \int_{\Omega} \Phi'(\phi(\mathbf{x}, t)) d\mathbf{x}$, and developed a second-order energy stable scheme by combining the invariant energy quadratization idea with the stabilization technique. However, the scheme involves solving a linear system with complicated variable coefficients. In [5], Lee introduced the following equation:

$$\frac{\partial \phi}{\partial t} = -\left(\Phi'(\phi) + 2\Delta\phi + \Delta^2\phi\right) + \left(\Phi(\phi)\right)^r \beta(t),\tag{3}$$



where $(\Phi(\phi))^r \beta(t)$ is a nonlocal and local Lagrange multiplier and $\beta(t) = \frac{\int_{\Omega} \Phi'(\phi(\mathbf{x},t)) d\mathbf{x}}{\int_{\Omega} (\Phi(\phi(\mathbf{x},t)))^r d\mathbf{x}'}$ and proposed mass conservative first- and second-order operator splitting methods. However, the methods lead to the necessity nonlinear equations to be solved at each time step thus require an iterative solver for solving the nonlinear equations.

Therefore, the aim of this paper is to present an efficient and accurate method that preserves mass conservation for solving the conservative SH Equation (3). We place the linear and nonlinear terms in the implicit and explicit parts, respectively, where an extra linear stabilizing term is added to improve the stability while preserving the simplicity. And we employ the implicit-explicit Runge–Kutta (RK) method [6]. As a result, our method is linear, high-order accurate in time, and mass conservative. We show analytically that the method inherits the mass conservation. In addition, the Fourier spectral method [5,7–10] is used for the spatial discretization. The MATLAB code for numerical implementation of the method in 2D is provided in Appendix A.

This paper is organized as follows. In Section 2, we construct the linear, high-order, and mass conservative method and show analytically that the method inherits the mass conservation. Numerical examples showing the efficiency and accuracy of the proposed method are presented in Section 3. Finally, conclusions are drawn in Section 4. In Appendix A, we provide the MATLAB code for numerical implementation of the proposed method in 2D.

2. Linear, High-Order, and Mass Conservative Method

For simplicity and clarity of exposition, we consider Equation (3) in one-dimensional space $\Omega = [0, L]$ with a periodic boundary condition:

$$\frac{\partial \phi(x,t)}{\partial t} = -\left(\Phi'(\phi(x,t)) + 2\frac{\partial^2 \phi(x,t)}{\partial x^2} + \frac{\partial^4 \phi(x,t)}{\partial x^4}\right) + \left(\Phi(\phi(x,t))\right)^r \beta(t),\tag{4}$$

where $\beta(t) = \frac{\int_{\Omega} \Phi'(\phi(x,t)) dx}{\int_{\Omega} (\Phi(\phi(x,t)))^r dx}$. Two- and three-dimensional cases are defined analogously. Let M be a positive integer, $\Delta x = \frac{L}{M}$ be the space step size, and Δt be the time step size. Let ϕ_m^n be an approximation of $\phi(x_m, t^n)$, where $x_m = m\Delta x$ for m = 0, 1, ..., M - 1 and $t^n = n\Delta t$. The discrete Fourier transform and its inverse transform are

$$\widehat{\phi}_k = \sum_{m=0}^{M-1} \phi_m e^{-ix_m \xi_k} \tag{5}$$

and

$$\phi_m = \frac{1}{M} \sum_{k=0}^{M-1} \widehat{\phi}_k e^{ix_m \xi_k},\tag{6}$$

where $\xi_k = \frac{2\pi k}{L}$.

To develop a linear, high-order (up to third-order), and mass conservative method for solving Equation (4), we treat $-\left(s\phi(x,t)+2\frac{\partial^2\phi(x,t)}{\partial x^2}+\frac{\partial^4\phi(x,t)}{\partial x^4}\right)$ implicitly and $-\left(\Phi'(\phi(x,t))-s\phi(x,t)\right)+\left(\Phi(\phi(x,t))\right)^r\beta(t)$ explicitly, where *s* is a non-negative number, and employ the implicit-explicit RK method. First- (**S1**), second- (**S2**), and third- (**S3**) order methods are as follows:

S1:
$$\phi_m^{n+1} = \phi_m^n + \Delta t \left(p(\phi_m^{n+1}) + q(\phi_m^n) \right),$$
 (7)

S2:
$$\phi_m^{(1)} = \phi_m^n + \Delta t \left(\gamma p(\phi_m^{(1)}) + \gamma q(\phi_m^n) \right),$$
 (8)

$$\phi_m^{n+1} = \phi_m^n + \Delta t \left(\gamma p(\phi_m^{n+1}) + (1-\gamma) p(\phi_m^{(1)}) + (1-\delta) q(\phi_m^{(1)}) + \delta q(\phi_m^n) \right), \tag{9}$$

S3:
$$\phi_m^{(1)} = \phi_m^n + \Delta t \left(\frac{1}{2} p(\phi_m^{(1)}) + \frac{1}{2} q(\phi_m^n) \right),$$
 (10)

$$\phi_m^{(2)} = \phi_m^n + \Delta t \left(\frac{1}{2} p(\phi_m^{(2)}) + \frac{1}{6} p(\phi_m^{(1)}) + \frac{1}{18} q(\phi_m^{(1)}) + \frac{11}{18} q(\phi_m^n) \right), \tag{11}$$

$$\phi_m^{(3)} = \phi_m^n + \Delta t \left(\frac{1}{2} p(\phi_m^{(3)}) + \frac{1}{2} p(\phi_m^{(2)}) - \frac{1}{2} p(\phi_m^{(1)}) + \frac{1}{2} q(\phi_m^{(2)}) - \frac{5}{6} q(\phi_m^{(1)}) + \frac{5}{6} q(\phi_m^n) \right),$$
(12)

$$\phi_m^{n+1} = \phi_m^n + \Delta t \left(\frac{1}{2} p(\phi_m^{n+1}) + \frac{1}{2} p(\phi_m^{(3)}) - \frac{3}{2} p(\phi_m^{(2)}) + \frac{3}{2} p(\phi_m^{(1)}) - \frac{7}{4} q(\phi_m^{(3)}) + \frac{3}{4} q(\phi_m^{(2)}) + \frac{7}{4} q(\phi_m^{(1)}) + \frac{1}{4} q(\phi_m^n) \right),$$
(13)

where $p(\phi_m^{(\cdot)}) = -\left(s\phi_m^{(\cdot)} + 2\frac{\partial^2 \phi_m^{(\cdot)}}{\partial x^2} + \frac{\partial^4 \phi_m^{(\cdot)}}{\partial x^4}\right), \quad q(\phi_m^{(\cdot)}) = -\left(\Phi'(\phi_m^{(\cdot)}) - s\phi_m^{(\cdot)}\right) + \left(\Phi(\phi_m^{(\cdot)})\right)^r \beta^{(\cdot)},$ $\beta^{(\cdot)} = \frac{\sum_{m=0}^{M-1} \Phi'(\phi_m^{(\cdot)})}{\sum_{m=0}^{M-1} \left(\Phi(\phi_m^{(\cdot)})\right)^r}, \quad \gamma = \frac{2-\sqrt{2}}{2}, \text{ and } \delta = 1 - \frac{1}{2\gamma}.$

For the method S1, Equation (7) can be transformed into the discrete Fourier space using (6):

$$\widehat{\phi}_k^{n+1} = \frac{\widehat{\phi}_k^n + \Delta t \widehat{q}_k^n}{1 + \Delta t \left(s - 2\xi_k^2 + \xi_k^4\right)},\tag{14}$$

where $\hat{q}_k^{(\cdot)} = \mathcal{F}[q(\phi_m^{(\cdot)})]$ and \mathcal{F} denotes the discrete Fourier transform. After updating $\hat{\phi}_k^{n+1}$ with $\hat{\phi}_k^n$, we recover ϕ_m^{n+1} from $\hat{\phi}_k^{n+1}$ using (6). To satisfy the mass conservation property, we should have $\sum_{m=0}^{M-1} \phi_m^{n+1} = \sum_{m=0}^{M-1} \phi_m^n$ for $n = 0, 1, \ldots$ From Equation (14), we get

$$\sum_{m=0}^{M-1} \phi_m^{n+1} = \widehat{\phi}_0^{n+1} = \frac{\widehat{\phi}_0^n + \Delta t \widehat{q}_0^n}{1 + s\Delta t} = \frac{\widehat{\phi}_0^n + s\Delta t \widehat{\phi}_0^n}{1 + s\Delta t} = \widehat{\phi}_0^n = \sum_{m=0}^{M-1} \phi_m^n$$
(15)

since

$$\widehat{q}_{0}^{(\cdot)} = \sum_{m=0}^{M-1} q(\phi_{m}^{(\cdot)}) = -\sum_{m=0}^{M-1} (\Phi'(\phi_{m}^{(\cdot)}) - s\phi_{m}^{(\cdot)}) + \beta^{(\cdot)} \sum_{m=0}^{M-1} \left(\Phi(\phi_{m}^{(\cdot)}) \right)^{r}$$

$$= -\sum_{m=0}^{M-1} (\Phi'(\phi_{m}^{(\cdot)}) - s\phi_{m}^{(\cdot)}) + \frac{\sum_{m=0}^{M-1} \Phi'(\phi_{m}^{(\cdot)})}{\sum_{m=0}^{M-1} \left(\Phi(\phi_{m}^{(\cdot)}) \right)^{r}} \sum_{m=0}^{M-1} \left(\Phi(\phi_{m}^{(\cdot)}) \right)^{r} = s \sum_{m=0}^{M-1} \phi_{m}^{(\cdot)} = s \widehat{\phi}_{0}^{(\cdot)}.$$
(16)

Thus, the method **S1** inherits the mass conservation. Next, for the method **S2**, we have

$$\widehat{\phi}_{0}^{(1)} = \frac{\widehat{\phi}_{0}^{n} + \gamma \Delta t \widehat{q}_{0}^{n}}{1 + \gamma s \Delta t} = \frac{\widehat{\phi}_{0}^{n} + \gamma s \Delta t \widehat{\phi}_{0}^{n}}{1 + \gamma s \Delta t} = \widehat{\phi}_{0}^{n}$$
(17)

from Equation (8) and

r

$$\sum_{m=0}^{M-1} \phi_m^{n+1} = \widehat{\phi}_0^{n+1} = \frac{\widehat{\phi}_0^n + \Delta t \left(-(1-\gamma)s\widehat{\phi}_0^{(1)} + (1-\delta)\widehat{q}_0^{(1)} + \delta\widehat{q}_0^n \right)}{1+\gamma s \Delta t}$$
$$= \frac{\widehat{\phi}_0^n + \Delta t \left(-(1-\gamma)s\widehat{\phi}_0^n + (1-\delta)s\widehat{\phi}_0^n + \delta s\widehat{\phi}_0^n \right)}{1+\gamma s \Delta t} = \widehat{\phi}_0^n = \sum_{m=0}^{M-1} \phi_m^n$$
(18)

from Equation (9). For the method **S3**, we have $\hat{\phi}_0^{(3)} = \hat{\phi}_0^{(2)} = \hat{\phi}_0^{(1)} = \hat{\phi}_0^n$ from Equations (10)–(12) and

$$\sum_{m=0}^{M-1} \phi_m^{n+1} = \widehat{\phi}_0^{n+1} = \frac{\widehat{\phi}_0^n + \Delta t \left(-\frac{s}{2} \widehat{\phi}_0^{(3)} + \frac{3s}{2} \widehat{\phi}_0^{(2)} - \frac{3s}{2} \widehat{\phi}_0^{(1)} - \frac{7}{4} \widehat{q}_0^{(3)} + \frac{3}{4} \widehat{q}_0^{(2)} + \frac{7}{4} \widehat{q}_0^{(1)} + \frac{1}{4} \widehat{q}_0^n \right)}{1 + \frac{s\Delta t}{2}} = \widehat{\phi}_0^n = \sum_{m=0}^{M-1} \phi_m^n$$
(19)

from Equation (13). Thus, the methods S2 and S3 also inherit the mass conservation.

3. Numerical Experiments

3.1. Convergence Test

We demonstrate the convergence of the proposed methods with the initial condition [11,12]

$$\phi(x,0) = 0.07 - 0.02 \cos\left(\frac{2\pi(x-12)}{32}\right) + 0.02 \cos^2\left(\frac{\pi(x+10)}{32}\right) -0.01 \sin^2\left(\frac{4\pi x}{32}\right)$$
(20)

on $\Omega = [0, 32]$. We set $\epsilon = 0.25$, g = 0, r = 0, and s = 0, and compute $\phi(x, t)$ for $0 < t \le 96$. The grid size is fixed to $\Delta x = \frac{1}{3}$ which provides enough spatial accuracy. To estimate the convergence rate with respect to Δt , simulations are performed by varying $\Delta t = 2^{-10}, 2^{-9}, \ldots, 2^{-4}$. We take the quadruply over-resolved numerical solution using the method **S3** as the reference solution. Figure 1a shows the relative l_2 -errors of $\phi(x, 40)$ for various time steps. Here, the errors are computed by comparison with the reference solution. In addition, Figure 1b–d show the evolution of $\int_{\Omega} (\phi(x, t) - \phi(x, 0)) dx$ using the methods **S1–S3**, respectively. Here, $\int_{\Omega} (\phi(x, t) - \phi(x, 0)) dx$ is approximated by $\sum_{m=0}^{M-1} (\phi_m^n - \phi_m^0) \Delta x$. It is observed that the methods give desired order of accuracy in time and conserve the total mass.

3.2. Efficiency of the Proposed Method

To show the efficiency of the proposed method, we take the initial condition (20) and parameter values used to create Figure 1. Figure 2 presents the CPU time (in seconds, averaged over 10 trials performed on Intel Core i5-7500 CPU at 3.40 GHz with 8 GB RAM) consumed using the methods **S1–S3** for various time steps. The results suggest that the CPU time is almost linear with respect to the number of steps and the methods **S2** and **S3** are about two and four times more expensive than the method **S1**, respectively.



Figure 1. (a) Relative l_2 -errors of $\phi(x, 40)$ for various time steps with $\epsilon = 0.25$ and $\Delta x = \frac{1}{3}$. (b–d) Evolution of $\int_{\Omega} (\phi(x, t) - \phi(x, 0)) dx$ for various time steps using the methods S1–S3.



Figure 2. CPU time versus time step. Each line segment is obtained by least squares fitting of all corresponding points.

3.3. Phase Diagram in 2D

In 2D, the phase diagram contains striped, hexagonal, and constant states depending on the values of $\bar{\phi}$ and ϵ [1] (see Figure 3a). To verify that the proposed method does lead to the expected states, we take an initial condition as $\phi(x, y, 0) = \bar{\phi} + \text{rand}$ on $\Omega = [0, 32] \times [0, 32]$. Here, rand is a random number between -0.1 and 0.1 at the grid points, and we use g = 0, r = 0, $\Delta x = \Delta y = \frac{1}{3}$, s = 2, and the method **S3**. For saving computational time, we choose different time steps as the solution evolves from random noisy stage to smooth stage: $\Delta t = \frac{1}{4}$ for $0 < t \le 128$ and $\Delta t = 4$ for

 $128 < t \le 2176$. To estimate the phase diagram numerically, we calculate the indicator function defined similarly in [13]:

$$\Lambda(t) = \begin{cases} \int_{\Omega} |\phi(\mathbf{x},t) - \bar{\phi}| \, d\mathbf{x} & \text{if } \int_{\Omega} |\nabla(\phi(\mathbf{x},t) - \bar{\phi})| \, d\mathbf{x} \approx 0\\ \frac{\int_{\Omega} |\phi(\mathbf{x},t) - \bar{\phi}| \, d\mathbf{x}}{\int_{\Omega} |\nabla(\phi(\mathbf{x},t) - \bar{\phi})| \, d\mathbf{x}} & \text{otherwise} \end{cases}$$
(21)

Here, we set $\int_{\Omega} |\nabla(\phi(\mathbf{x},t)-\bar{\phi})| d\mathbf{x} \approx 0$ if $\int_{\Omega} |\nabla(\phi(\mathbf{x},t)-\bar{\phi})| d\mathbf{x}$ is less than 10^{-10} . Figure 3b shows $\Lambda(t)$ at t = 2176 with various $\bar{\phi} = 0.02, 0.04, \ldots, 0.3$ and $\epsilon = 0.02, 0.04, \ldots, 0.3$. Results in Figure 3b are consistent with the phase diagram in Figure 3a. Sample time evolutions of $\phi(x, y, t)$ with $(\bar{\phi}, \epsilon) = (0.02, 0.1), (0.14, 0.1), \text{ and } (0.26, 0.1)$ are shown in Figures 4–6, respectively. Figure 7 shows evolutions of $\mathcal{E}(t)$ and $\Lambda(t)$ with $(\bar{\phi}, \epsilon)$ used in Figures 4–6. We remark that the solution $\phi(x, y, t)$ with $(\bar{\phi}, \epsilon) = (0.26, 0.1)$ has a small but measurable perturbation from a constant state until t = 128 (see Figure 6). In this case, $\int_{\Omega} |\nabla(\phi(\mathbf{x}, t) - \bar{\phi})| d\mathbf{x} \not\approx 0$ and $\Lambda(t)$ is calculated using $\int_{\Omega} |\phi(\mathbf{x}, t) - \bar{\phi}| d\mathbf{x} / \int_{\Omega} |\nabla(\phi(\mathbf{x}, t) - \bar{\phi})| d\mathbf{x} \approx 0$. Thus, $\Lambda(t)$ is calculated using $\int_{\Omega} |\phi(\mathbf{x}, t) - \bar{\phi}| d\mathbf{x} \approx 0$ and $\int_{\Omega} |\nabla(\phi(\mathbf{x}, t) - \bar{\phi})| d\mathbf{x} \approx 0$.



Figure 3. (a) Phase diagram (Reprinted with permission from [1]). Here, $\bar{\psi}$ denotes the averaged field $(=\bar{\phi})$. (b) Values of $\Lambda(t)$ at t = 2176 with various $\bar{\phi}$ and ϵ .



Figure 4. Evolution of $\phi(x, y, t)$ using the method **S3** with $(\bar{\phi}, \epsilon) = (0.02, 0.1)$. In each snapshot, the yellow, green, and blue regions indicate $\phi = 0.3810, 0.0195$, and -0.3420, respectively.



Figure 5. Evolution of $\phi(x, y, t)$ using the method **S3** with $(\bar{\phi}, \epsilon) = (0.14, 0.1)$. In each snapshot, the yellow, green, and blue regions indicate $\phi = 0.3926$, 0.0108, and -0.3710, respectively.



Figure 6. Evolution of $\phi(x, y, t)$ using the method **S3** with $(\bar{\phi}, \epsilon) = (0.26, 0.1)$. In each snapshot, the yellow, green, and blue regions indicate $\phi = 0.2610, 0.2600$, and 0.2590, respectively.



Figure 7. Evolutions of (a) $\mathcal{E}(t)$ and (b) $\Lambda(t)$ with $(\bar{\phi}, \epsilon)$ used in Figures 4–6.

3.4. Comparison with Other Method

To compare the proposed method with other method, we solve the conservative SH Equation (3) using the proposed method **S2** and the second-order operator splitting (OS2) method in [5] with the initial condition and parameter values used to create Figure 4 except for Δt . Figures 8 and 9 show evolutions of $\phi(x, y, t)$ using the method OS2 with $\Delta t = \frac{1}{4}$ and 2, respectively. The method OS2 with a smaller time step $\Delta t = \frac{1}{4}$ leads to the expected striped state, whereas a constant state is observed for $\Delta t = 2$. Figures 10 and 11 show evolutions of $\phi(x, y, t)$ using the method **S2** gives the striped state even for a large time step. Evolutions of $\mathcal{E}(t)$ for Figures 8–11 are shown in Figure 12.



Figure 8. Evolution of $\phi(x, y, t)$ using the method OS2 with $\Delta t = \frac{1}{4}$. In each snapshot, the yellow, green, and blue regions indicate $\phi = 0.3810, 0.0195$, and -0.3420, respectively.



Figure 9. Evolution of $\phi(x, y, t)$ using the method OS2 with $\Delta t = 2$. In each snapshot, the yellow, green, and blue regions indicate $\phi = 0.3810, 0.0195$, and -0.3420, respectively.



Figure 10. Evolution of $\phi(x, y, t)$ using the method **S2** with $\Delta t = \frac{1}{4}$. In each snapshot, the yellow, green, and blue regions indicate $\phi = 0.3810, 0.0195$, and -0.3420, respectively.



Figure 11. Evolution of $\phi(x, y, t)$ using the method **S2** with $\Delta t = 2$. In each snapshot, the yellow, green, and blue regions indicate $\phi = 0.3810, 0.0195$, and -0.3420, respectively.



Figure 12. Evolutions of $\mathcal{E}(t)$ for Figures 8–11.

4. Conclusions

In this paper, we developed linear, first-, second-, and third-order, and mass conservative methods for the conservative SH equation by placing the linear and nonlinear terms in the implicit and explicit parts, respectively, and employing the implicit-explicit RK method. We confirmed that the proposed methods give desired order of accuracy in time, inherit the mass conservation, and are efficient (the CPU time was almost linear with respect to the number of steps and of stages). And we performed long time simulation for pattern formation in 2D, where the phase diagram can be observed clearly.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1C1C1011112).

Acknowledgments: The corresponding author thanks the reviewers for the constructive and helpful comments on the revision of this article

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Matlab Code

MATLAB code for numerical implementation of the methods S1-S3 (7)-(13) in 2D.

```
1 clear; clc; clf;
2
3 Lx = 32; xleft = 0; xright = xleft+Lx;
4 Ly = 32; yleft = 0; yright = yleft+Ly;
5 Mx = 96; dx = Lx/Mx; x = xleft+(0:Mx-1)*dx;
6 My = 96; dy = Ly/My; y = yleft+(0:My-1)*dy;
7 xix = 2*pi*[0:Mx/2-1 -Mx/2:-1]/Lx;
s xiy = 2*pi*[0:My/2-1 -My/2:-1]/Ly; [kx,ky] = ndgrid(xix,xiy); dt = 1/4;
10 R = 0.1*(2*rand(Mx, My)-1); ophi = 0.02+(R-mean(mean(R)));
11 epsilon = 0.1; g = 0; r = 0; s = 2; T = 128; order = 3;
12 kp = -(s-2*(kx.^{2}+ky.^{2})+(kx.^{2}+ky.^{2}).^{2});
13
14 switch order
       case 1
15
           MI = 1; ME = MI;
16
17
       case 2
           gamma = (2-sqrt(2))/2; \Delta = 1-1/(2*gamma);
18
           MI = [gamma 0; 1-gamma gamma]; ME = [gamma 0; \land 1-\land];
19
20
       case 3
           MI = [1/2 \ 0 \ 0 \ 0; \ 1/6 \ 1/2 \ 0 \ 0; \ -1/2 \ 1/2 \ 1/2 \ 0; \ 3/2 \ -3/2 \ 1/2 \ 1/2];
21
           ME = [1/2 0 0 0; 11/18 1/18 0 0; 5/6 -5/6 1/2 0; 1/4 7/4 3/4 -7/4];
22
23 end
24 ns = length(MI);
25
26 for n=1:round(T/dt)
27
       ophi_hat = fft2(ophi);
28
       phis = zeros(Mx, My, ns); q = zeros(Mx, My, ns);
29
```

```
30
       for i = 1:ns
           Phip = ophi.^3-g*ophi.^2+(1-epsilon)*ophi;
31
           Phir = (1/4*ophi.^4-g/3*ophi.^3+(1-epsilon)/2*ophi.^2).^r;
32
           beta = sum(sum(Phip))/sum(sum(Phir));
33
           q(:,:,i) = -(Phip-s*ophi)+Phir*beta;
34
           IM = 0; EX = ME(i,i)*q(:,:,i);
35
36
           for j=1:i-1
37
               IM = IM + MI(i,j)*phis(:,:,j);
               EX = EX + ME(i,j)*q(:,:,j);
38
39
           end
           phis(:,:,i) = (ophi_hat+dt*(kp.*IM+fft2(EX)))./(1-MI(i,i)*dt*kp);
40
41
           ophi = ifft2(phis(:,:,i));
42
       end
43
44
  end
45
   [xx,yy] = ndgrid(x,y); pcolor(xx,yy,ophi); shading interp; axis image off;
```

References

- Elder, K.R.; Katakowski, M.; Haataja, M.; Grant, M. Modeling elasticity in crystal growth. *Phys. Rev. Lett.* 2002, *88*, 245701. [CrossRef]
- 2. Elder, K.R.; Grant, M. Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals. *Phys. Rev. E* 2004, *70*, 051605. [CrossRef] [PubMed]
- 3. Swift, J.; Hohenberg, P.C. Hydrodynamic fluctuations at the convective instability. *Phys. Rev. A* 1977, 15, 319–328. [CrossRef]
- 4. Zhang, J.; Yang, X. Numerical approximations for a new *L*²-gradient flow based Phase field crystal model with precise nonlocal mass conservation. *Comput. Phys. Commun.* **2019**, 243, 51–67. [CrossRef]
- 5. Lee, H.G. A new conservative Swift-Hohenberg equation and its mass conservative method. *J. Comput. Appl. Math.* **2020**, *375*, 112815. [CrossRef]
- 6. Ascher, U.M.; Ruuth, S.J.; Spiteri, R.J. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.* **1997**, *25*, 151–167. [CrossRef]
- Lee, H.G. A semi-analytical Fourier spectral method for the Swift-Hohenberg equation. *Comput. Math. Appl.* 2017, 74, 1885–1896. [CrossRef]
- 8. Lee, H.G. An energy stable method for the Swift-Hohenberg equation with quadratic-cubic nonlinearity. *Comput. Methods Appl. Mech. Eng.* **2019**, 343, 40–51. [CrossRef]
- 9. Chen, X.; Song, M.; Song, S. A fourth order energy dissipative scheme for a traffic flow model. *Mathematics* **2020**, *8*, 1238. [CrossRef]
- 10. Yoon, S.; Jeong, D.; Lee, C.; Kim, H.; Kim, S.; Lee, H.G.; Kim, J. Fourier-spectral method for the phase-field equations. *Mathematics* **2020**, *8*, 1385. [CrossRef]
- 11. Hu, Z.; Wise, S.M.; Wang, C.; Lowengrub, J.S. Stable and efficient finite-difference nonlinear-multigrid schemes for the phase field crystal equation. *J. Comput. Phys.* **2009**, *228*, 5323–5339. [CrossRef]
- Baskaran, A.; Hu, Z.; Lowengrub, J.S.; Wang, C.; Wise, S.M.; Zhou, P. Energy stable and efficient finite-difference nonlinear multigrid schemes for the modified phase field crystal equation. *J. Comput. Phys.* 2013, 250, 270–292. [CrossRef]
- 13. Shin, J.; Lee, H.G.; Lee, J.-Y. Long-time simulation of the phase-field crystal equation using high-order energy-stable CSRK methods. *Comput. Methods Appl. Mech. Eng.* **2020**, *364*, 112981. [CrossRef]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).