Thesis for the Degree
of Doctor

Moving overset grid method to model dendritic growth with convection

by

Lee, Seunggyu

Department of Mathematics

Graduate School
Korea University

December , 2015

김준석 教授指導

博士學位論文

Moving overset grid method to model dendritic growth with convection

이 論文을 理學 博士學位 論文으로 提出함.

2015 年 12 月 29 日

高麗大學校大學院

數學科

李 丞 揆

李丞揆의 理學 博士學位 論文
審査를 完了함.

2015 年 12 月 11 日

委員長 김 준 석 (印)

委員 안 인 경 (印)

委員 황 운 재 (印)

委員 김 양 진 (印)

委員 박 춘 재 (印)

Contents

Abstract

The primary purpose of this thesis is to study the modeling dendritic growth with convection in phase-field simulations. To solve the equations for a crystal growth in a flow, we first discuss the phase-field method, dendritic growth without an external force, and the incompressible viscous fluid flows. In a phase-field method, preceding researchers have developed several numerical schemes. We compare performances of the widely used ones and give proofs of some analytical properties; solvability, stability and boundedness of numerical solutions. A dendritic crystal growth is one of the practical problems which can be solved by using a phase-field method. Here, we review the operator splitting method for solving a governing equation of a morphological change in a crystal growth and give some numerical results to show the robustness of the chosen method. The Navier–Stokes equation was originated to estimate dynamics of an incompressible viscous fluid flow, which contains a metallic solution of a crystal. I present the fast, efficient, and robust numerical solver based on the Chorin's projection method. By synthesizing the treated methods in a former part of this dissertation, we finally consider a numerical solution of a dendritic growth with convection. Applying convection to the crystal equation, there are problems such as deformation of crystal shape and ambiguity of the crystal orientation for the anisotropy. To resolve these difficulties, we present a phase-field method by using a moving overset grid for the dendritic growth under a flow. The numerical results are presented to indicate a usefulness of our proposed method without depending a condition of a given underlying fluid.

Acknowledgments

I first would like to express sincere thanks and gratitude to my advisor Professor Junseok Kim. His encouragement and unfailing support have been valuable during a period of graduate work, and I feel deeply privileged for the opportunity to work in his research group. I also gratefully acknowledge Professor Inkyung Ahn, Professor Woonjae Hwang, Professor Chunjae Park, and Professor Yangjin Kim, of my dissertation committee members.

Next, I have very much enjoyed working in friendly atmosphere of colleagues: Dr. Hyun Geun Lee, Dr. Ana Yun, Dr. Darae Jeong, Dr. Jaemin Shin, Dr. Yibao Li, Dr. Haobo Hua, Dr. Dongsun Lee, Jong-Hee Lee, Sungki Kim, Jaehyun Jo, Youngki Choo, Jinsu Shin, Jiseon Kim, Chaeyong Lee, Ha-kyu Song, Yongho Choi, Minhyun Yoo, Minhwa Bae, and Hyeju Son. Their passion always inspire me with the whole process of my research. Thanks you for understanding that my research with good intentions.

Sincere thanks to all my acquaintances and friends who always encourage me: Byunggwon Kang, Bogyung Kim, Seora Kim, Junsu Kim, Kyungrip Min, Sungwoo Park, Sinki Park, Gyungmin Baek, Yangsub Byun, Yuneung Song, Moonsub Won, Jina Yoon, Jongseung Lee, Byunggi Im, Jaeyoung Bae, Jihoon Jeon, Minwook Chung, Sangmin Ji, Junsang Han, Junggyu Kang, Youngwoo Kwon, Ganghyun Kim, Guyoon Kim, Kitak Kim, Jieun Kim, Sunju Namgung, Minhye Noh, Junghoon Park, Jinho Park, Hansaem Bak, Yunsub Sung, Sinae Song, Kyoyoung Shin, Heesang Ann, Hanju Yoo, Dasom Lee, Dongu Lee, Dr. Wanho Lee, Yun-e Lee, Chanyoung Lee, Cholong Lee, Seunghwan Jung, Cathy Ha, and all members of the 2nd squad, 1st platoon, 4th company, 25th regiment in Korea Army Training Center. I cannot tell all you how many times looking at you make me continue working on my projects to completion.

My deepest gratitude goes to my close relatives including late grandparents Wonjae Lee, Jungsoon Lee, uncles Hanju Lee and Hanchang Lee, aunts Junghyun Lee and Hyunju Jin, cousins Jieun Lee, Donghee Lee, Seunghyun Lee, Taegyu Lee, and Somin Lee, for supporting and for encouraging and believing in me.

Above all things, dear my mother, Indeog Park, you have been an outstanding inspiration to me, and dear my father, Hanchul Lee, your soul may rest in peace.

Chapter 1

Introduction

A dendrite is the growth structure of the solid crystal during solidification, which has a morphological unstable interface with its melt under typical solidification conditions of an alloy [107]. Here, the name "dendrite" is originated from a Greek word "dendron", which means a tree, because the structure of the crystal is tree-like. To observe a dendrite, the undercooled (or supercooled), below the freezing point of the solid, liquid and a spherical solid nucleus are required. As the nucleus grows, the spherical morphology becomes unstable and the growth directions prefers anisotropic way. The solid generally attempts to minimize the their surface with the highest surface energy in metallic system; thus the tip of dendrite grows shaper and sharper [34]. In the practical point of view, the understanding dendritic growth is crucial in metal casting. By quoting a sentence of an article in Nature, "Worldwide, as many as 10 billion metallic dendrites are produced in industry every second" [111]. This number has been increasing since last two decades with settling their good theoretical models and developing numerical tools which make practical engineering problems be resolved [120].

Moreover, convection of the crystal in the melt is of great interest for the practical processes to understand the dendritic solidification. The formation of the directional solidification such as the arm spacing, growth rate, and morphology is largely changed by natural convection due to gravity [13, 40, 105] and the forced convection due to the melt flow [104, 108]. Especially, the melt convection effect is never neglected in a dendrite growth whose rate is relatively slow [54]. The convective effects on free dendritic crystal growth have been investigated in many researches experimentally [59, 68, 105] and numerically [9, 35, 113, 127, 128].

The melt convection has been included as an incompressible fluid flow in numerical approaches. Therefore, it is required to solve the phase transformation, describe the fluid flow, and incorporate of solid boundary into the solution of fluid flow at the same time; and solving the system is a quite challenging problem.

There are two major approaches for simulating multi-phase or multi-component flows to characterize moving interfaces: the interface tracking and interface capture methods. First, the interface tracking method uses Lagrangian particles to track interfaces and a velocity field is generated by advection of the particles. Examples of the interface tracking method include the volume of fluid [56], front tracking [41, 130], and immersed boundary method [91]. In contrast, interface capturing methods implicitly capture an interface by using the contours of particular scalar functions. Examples include the level-set [116] and the phase-field methods [75].

Among such various numerical approaches, the phase-field method is considered as the most powerful and accurate one to model dendritic growth, and there are many review papers regarding phase-field models and simulations [5, 14, 120, 124].

However, it is difficult to precisely capture an interface of a crystal applying the advection equation. To resolve the difficulty in a convection problem in dendrite growth, I propose a moving overset grid method in a phase-field simulation to convect the crystal feature.

In this dissertation, the contents of the following published or working papers are contained:

1. Seunggyu Lee, Chaeyoung Lee, Hyun Geun Lee and Junseok Kim, Comparison of different numerical schemes for the Cahn–Hilliard equation, Journal of Korean Society for Industrial and Applied Mathematics 17(3) (2013) 197–207.
2. Seunggyu Lee, Darae Jeong, Wanho Lee and Junseok Kim, An immersed boundary method for a contractile elastic ring in a three-dimensional Newtonian fluid, Journal of Scientific Computing, in press, DOI: 10.1007/s10915-015-0110-8.
3. Seunggyu Lee, Dongsun Lee, Yongho Choi, Jaemin Shin and Junseok Kim, Stabilized numerical method for the Cahn–Hilliard equation; unique solvability and gradient stability, working paper.
4. Seunggyu Lee, Yibao Li, Jaemin shin and Junseok Kim, Phase-field simulations of crystal growth in a two-dimensional cavity flow, submitted paper.

The outline of this dissertation is as follow: I give an introduction of a phase-field method which is a useful technique for modeling a dendritic growth in Chapter 2. The crystal growth, or dendritic solidification process is discussed focusing on a numerical analysis point of view in Chapter 3. The content of Chapter 4 is the Navier–Stokes (NS) equation, describing the motion of viscous and incompressible fluid substances. The numerical solver of the convection of the crystal under a flow, the main achievements of the thesis paper, is presented in Chapter 5. Finally, the conclusion is drawn in Chapter 6.

Chapter 2

Phase-field method

2.1. Introduction

A phase-field method is a mathematical tool for solving an interfacial dynamics problem. It is known that the method was first introduced by Fix [51] and Langer [86] and it has also been applied to many industrial problems such as solidification dynamics [133], image inpainting [11, 94], volume reconstruction [90], tumor growth [27], block copolymer [63, 64], and etc. Of interest to researchers at present is the coupling of the phase-field method to the NS equations of fluid flow for solving multi-phase fluid flows [118, 136].

A formulation of the phase-field model are based on a free energy functional depending on an order parameter (or the phase-field) and a diffusive field (variational formulations). The order parameter can be a scalar function such as a fraction of a mixture in two-phase problem (or a solid in solid-liquid phase change problems), that varies from -1 (or 0) in one phase (or liquid) to 1 in the other phase (or solid), or a vector function for anisotropic surfaces [57]. By minimizing a given free energy functional, the governing differential equations for the evolution of the multi-phase system are derived [121].

The description of some phenomena involves the definition of a precisely located interface on which boundary conditions are imposed, especially yields a normal velocity at which the interface is moving. This is so-called a sharp-interface approach. Nevertheless, a small scale of the interface width is prohibitive in a computational point of view. To solve this problem, a finite width ϵ between distinct phases is considered as a limit of a sharp-interface model. This limit is usually taken by asymptotic expansions in powers of the interface width. Such a model is called a diffuse-interface model. The concept of a transition zone between two co-existing phases was already introduced by Gibbs; however, this notion has been employed in phase transition phenomena by Landau and Khalatnikov, who firstly introduced an additional parameter to label the different phases on the absorption of liquid helium. Essentially, diffuse-interface modeling appeared subsequently in the literature in the context of phase transition phenomena and such models have advanced numerical treatment as well as understanding of interfacial growth phenomena [47].

The changes of a structure interface are described implicitly by the time evolution of an order parameter instead of tracking the interface explicitly. Although interface tracking approaches can be successful in lower-dimensional systems, they becomes impractical for complicated three-dimensional structures [25]. The one of other strong points of a phase-field method is its simplicity in treating morphological changes with preserving the properties of a governing equation.

In this dissertation, we introduce two governing equations of the phase-field method: the Allen–Cahn (AC) equation and the Cahn–Hilliard (CH) equation, deriving from the Ginzbug–Landau free energy functional.

2.2. Allen–Cahn equation

2.2.1. Governing equations. The AC equation is a reaction-diffusion equation describing the process of phase separation in iron alloys, including order-disorder transitions [2]. The equilibrium configuration of the Ginzburg–Landau free energy functional has been applied to a wide range of problems such as image inpainting [94], multi-phase problem [118, 136], crystal growth [133], pattern dynamics [45, 109], and etc.

The equation is given as follow [2]:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = -\frac{F'(\phi(\mathbf{x},t))}{\epsilon^2} + \Delta\phi(\mathbf{x},t),\ \mathbf{x} \in \Omega, 0 < t \leq T, \tag{2.1}$$

where $\mathbf{x}$ is the spatial variable in a domain $\Omega \in \mathbb{R}^d$ $(d = 1, 2, 3)$, t is the temporal variable in $[0, T]$, ϕ is the quantity defined as a difference between concentrations of two mixture components (for example, $(m_1 - m_2)/(m_1 + m_2)$ where m_1 and m_2 are masses of components 1 and 2),

$$F(\phi) = \frac{1}{4}\left(1 - \phi^2\right)^2$$

is the double-well potential free energy potential approximated by a polynomial of degree four (see Fig. 2.1), and $\epsilon > 0$ is the gradient energy coefficient related to an interfacial energy. The boundary condition is

$$\frac{\partial \phi}{\partial \mathbf{n}} = \mathbf{n} \cdot \nabla \phi = 0 \text{ on } \partial\Omega, \tag{2.2}$$

where $\mathbf{n}$ is the outgoing unit normal vector to the domain boundary $\partial\Omega$.

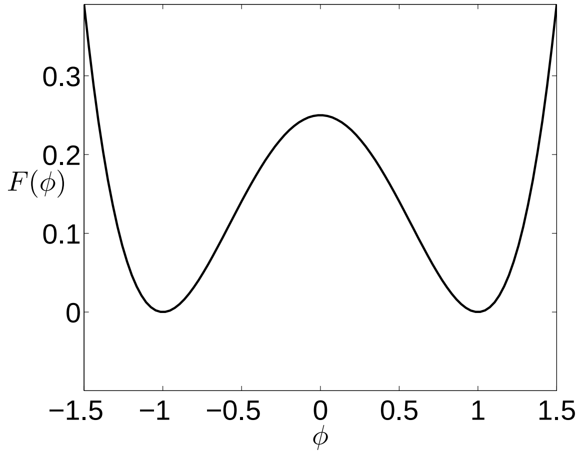


FIGURE 2.1. A double well potential $F(\phi) = 0.25(1 - \phi^2)^2$.

The AC equation is the L_2-gradient flow of the Ginzburg–Landau free energy functional, which has an important role in nonlinear evolution equation [102], and the functional has a form as follow:

$$\mathcal{E}(\phi) = \int_\Omega \left[\frac{F(\phi)}{\epsilon^2} + \frac{1}{2} \left| \nabla \phi \right|^2 \right] d\mathbf{x}. \tag{2.3}$$

Differentiating (2.3) with respect to t gives

$$\begin{aligned}
\frac{d}{dt}\mathcal{E}(\phi) &= \frac{d}{dt}\int_\Omega \left[\frac{F(\phi)}{\epsilon^2} + \frac{1}{2}\left|\nabla\phi\right|^2\right] d\mathbf{x} \\
&= \int_\Omega \left[\frac{F'(\phi)}{\epsilon^2}\phi_t + \nabla\phi\cdot\nabla\phi_t\right] d\mathbf{x}, \\
&= \int_\Omega \left[\frac{F'(\phi)}{\epsilon^2} - \Delta\phi\right] \phi_t d\mathbf{x} + \int_{\partial\Omega} (\nabla\phi\cdot\mathbf{n})\phi_t d\mathbf{x} \\
&= \int_\Omega \left(-\phi_t\right)\phi_t d\mathbf{x} \\
&= -\int_\Omega \left(\phi_t\right)^2 d\mathbf{x} \le 0, \qquad (2.4)
\end{aligned}$$

using the integration by parts and the boundary condition (2.2) and it implies that the total energy is non-increasing in time.

Allen and Cahn also showed that the normal velocity V on a single closed interface Γ is governed by its mean curvature [2]:

$$V(\mathbf{x}, t) = \kappa(\mathbf{x}, t) \text{ for } \mathbf{x} \in \Gamma, \qquad (2.5)$$

where $\kappa(\mathbf{x}, t)$ is the mean curvature of the interface Γ. This dynamical property has been studied in [16, 103, 112, 119]. Figure 2.2 shows the temporal evolutions of curves with the AC equation in two dimension. The dashed line is the initial curve and the solid lines are the evolutions of interfaces. The directions of evolutions are indicated by arrows.

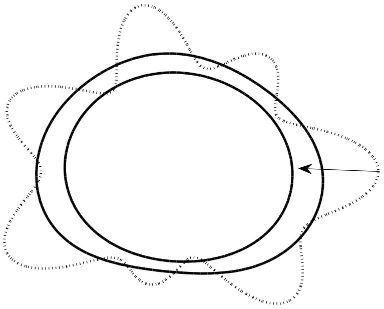


FIGURE 2.2. Temporal evolutions of arbitrary curves with the AC equation. The dashed line is the initial curves and directions of evolutions are indicated by arrows.

We observe that the AC equation does not conserve its initial mass. We can check that the AC type dynamics does not preserve the volume fractions as follows:

$$\begin{aligned}\frac{d}{dt}\int_\Omega \phi d\mathbf{x} &= \int_\Omega \phi_t d\mathbf{x}\\ &= \int_\Omega \left[-\frac{F'(\phi)}{\epsilon^2}+\Delta\phi\right] d\mathbf{x}\\ &= -\int_\Omega \frac{F'(\phi)}{\epsilon^2} d\mathbf{x} + \int_{\partial\Omega} \mathbf{n}\cdot\nabla\phi ds\\ &= -\int_\Omega \frac{F'(\phi)}{\epsilon^2} d\mathbf{x} \le 0,\end{aligned}$$

which is not always zero.

2.2.1.1. *Mass-conservative form.* As seen in above, we check that the classical AC equation does not preserve the mass in both theoretically and numerically. To preserve the volume, Rubinstein and Sternberg introduced a Lagrange multiplier $\beta(t)$ into the AC model [115]:

$$\frac{\partial c(\mathbf{x},t)}{\partial t} = -\frac{F'(c(\mathbf{x},t))}{\epsilon^2} + \Delta c(\mathbf{x},t) + \beta(t),\ \mathbf{x}\in\Omega, 0<t\le T, \tag{2.6}$$

where notations are followed the classical AC equation except the phase-field function $c \in [0,1]$ instead of $\phi \in [-1,1]$ and the double well energy potential

$$F(c) = \frac{1}{2}c^2(1-c^2).$$

Here, $\beta(t)$ must satisfy

$$\beta(t) = \frac{\int_\Omega F'(c(\mathbf{x},t)d\mathbf{x}}{\epsilon^2 \int_\Omega d\mathbf{x}},$$

to keep the mass conservation, and this formulation has been widely used [8, 137, 141]. the normal velocity V on a single closed interface Γ is changed to

$$V(\mathbf{x},t) = \kappa(\mathbf{x},t) - \frac{1}{|\Gamma|}\int_\Gamma \kappa ds \text{ for } \mathbf{x} \in \Gamma,$$

by volume-preserving mean curvature flow, where $|\Gamma|$ is the total curve length in a two-dimensional space and the total area in a three-dimensional space.

Figure 2.3 shows the temporal evolutions of curves with the mass-conservative AC equation in two dimension. The dashed line is the initial curve and the solid lines are the evolutions of interfaces. The directions of evolutions are indicated by arrows.

The Rubinstein and Sternberg's model has been studied analytically and numerically [16, 10, 19, 132, 138, 95]. However, it has a drawback on preserving small features since the Lagrange multiplier is only a function of time variable. For example, there is a critical radius of drop

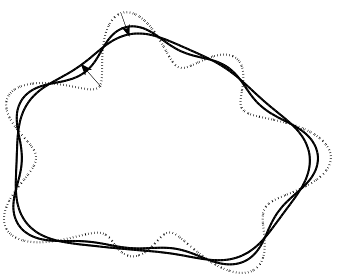


Figure 2.3. Temporal evolutions of arbitrary curves with the conservative AC equation. The dashed line is the initial curve and directions of evolutions are indicated by arrows.

which eventually disappears below the radius. This phenomenon is observed in the frame of the Cahn–Hilliard model [138].

Brassel and Bretin proposed the following conservative AC equation to preserve small geometric features [16]:

$$\frac{\partial c(\mathbf{x},t)}{\partial t} = -\frac{F'(c(\mathbf{x},t))}{\epsilon^2} + \Delta c(\mathbf{x},t) + \beta(t)\sqrt{2F(c(\mathbf{x},t))},\ \mathbf{x}\in\Omega, 0<t\le T \tag{2.7}$$

where

$$\beta(t) = \frac{\int_\Omega F'(c(\mathbf{x},t))d\mathbf{x}}{\epsilon^2\int_\Omega\sqrt{2F(c(\mathbf{x},t))}d\mathbf{x}}. \tag{2.8}$$

Then, the solution $c(\mathbf{x},t)$ of the conservative AC equation (2.7) satisfies the total mass conservation property:

$$\frac{d}{dt}\int_\Omega c d\mathbf{x} = \int_\Omega c_t d\mathbf{x}$$

$$
\begin{aligned}
&=\int_\Omega \left[-\frac{F'(c)}{\epsilon^2}+\Delta c+\beta(t)\sqrt{2F(c)}\right]d\mathbf{x}\\
&=-\frac{1}{\epsilon^2}\int_\Omega F'(c)d\mathbf{x}+\int_{\partial\Omega}\mathbf{n}\cdot\nabla c ds+\beta(t)\int_\Omega\sqrt{2F(c)}d\mathbf{x}\\
&=-\frac{1}{\epsilon^2}\int_\Omega F'(c)d\mathbf{x}+\beta(t)\int_\Omega\sqrt{2F(c)}d\mathbf{x}\\
&=-\frac{1}{\epsilon^2}\int_\Omega F'(c)d\mathbf{x}+\frac{\int_\Omega F'(c(\mathbf{x},t))d\mathbf{x}}{\epsilon^2\int_\Omega\sqrt{2F(c(\mathbf{x},t))}d\mathbf{x}}\int_\Omega\sqrt{2F(\phi)}d\mathbf{x}\\
&=0,
\end{aligned}
$$

using the divergence theorem, the homogeneous Neumann boundary condition (2.2), and the definition of the Lagrange multiplier (2.8). Note that the Cahn–Hilliard equation, discussed in the later section, is an also mass conservative model.

2.2.2. Numerical solution. We consider discretization of the AC equation to implement a finite-difference method. For simplicity, an one-dimensional case is introduced where $\Omega = (a, b)$ and higher dimensional case can be extended straightforwardly. Let N be a positive even integer, $h = (b - a)/N$ be a uniform spatial step size,

$$\Omega_h = \{x_i = (i - 0.5)h, 1 \leq i \leq N\}$$

be a set of cell-centers of a computational domain. Denote the approximate value of the order parameter ϕ as

$$\phi_i^n \approx \phi(x_i, n\Delta t)$$

where Δt is the temporal step size defined as T/N_t, T is the final time and N_t is the total number of temporal steps and the discrete order parameter at time $n\Delta t$ is defined as

$$\boldsymbol{\phi}^n = (\phi_1^n, \phi_2^n, \cdots, \phi_N^n).$$

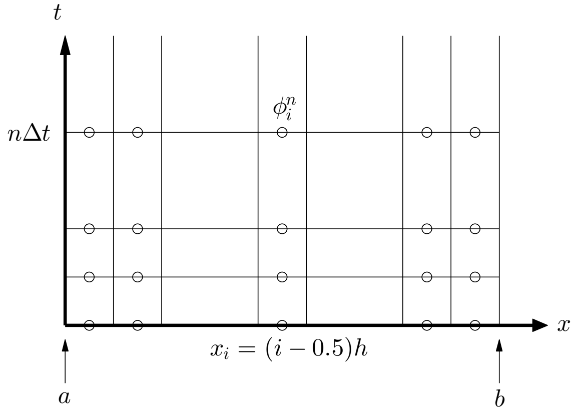


FIGURE 2.4. Approximation value ϕ_i^n in $\Omega_h \times [0, T]$.

Let ∇_h and Δ_h be discrete differential and Laplacian operators as

$$\nabla_h \phi_{i+\frac{1}{2}}^n = \frac{\phi_{i+1}^n - \phi_i^n}{h},$$

$$\Delta_h \phi_i^n = \frac{\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n}{h^2},$$

respectively. The operators can be applied to not only pointwise value, but also vector-valued function as

$$\nabla_h \boldsymbol{\phi} = \left(\nabla_h \phi_{\frac{1}{2}}^n, \cdots, \nabla_h \phi_{N+\frac{1}{2}}^n \right).$$

Here, the discrete Laplacian operator uses the central differentiation and it can be expressed by ∇_h as

$$\Delta_h \phi_i^n = \nabla_h \left(\nabla_h \phi_{i+\frac{1}{2}}^n - \nabla_h \phi_{i-\frac{1}{2}}^n \right).$$

Note that the boundary condition (2.2) can be written in a discrete sense as

$$\nabla_h \phi_{\frac{1}{2}}^n = \nabla_h \phi_{N+\frac{1}{2}}^n = 0. \tag{2.9}$$

Moreover, we define the discrete l_2-inner products as

$$\begin{aligned} \langle \boldsymbol{\phi}^n, \boldsymbol{\psi}^n \rangle_h =& h \sum_{i=1}^{N} \phi_i^n \psi_i^n, \\ (\nabla_h \boldsymbol{\phi}^n, \nabla_h \boldsymbol{\psi}^n)_h =& h \sum_{i=0}^{N} \nabla_h \phi_{i+\frac{1}{2}}^n \nabla_h \psi_{i+\frac{1}{2}}^n, \end{aligned} \tag{2.10}$$

and the discrete discrete l_2-norm as

$$\|\boldsymbol{\phi}^n\|_h^2 = \langle \boldsymbol{\phi}^n, \boldsymbol{\phi}^n \rangle_h .$$

Using the summation by parts, a discrete version of the integration by parts, the following property holds with the discrete boundary condition (2.9):

$$\langle \Delta_h \boldsymbol{\phi}^n, \boldsymbol{\psi}^n \rangle_h = - \left(\boldsymbol{\phi}^n, \boldsymbol{\psi}^n \right)_h = \langle \boldsymbol{\phi}^n, \Delta_h \boldsymbol{\psi}^n \rangle_h.$$

Now, we consider the following discrete AC equation:

$$\frac{\phi_i^{n+1}-\phi_i^n}{\Delta t} =\frac{-\alpha\left(\phi_i^{n+1}\right)^3-(1-\alpha)\left(\phi_i^n\right)^3+\beta\phi_i^{n+1}+(1-\beta)\phi_i^n}{\epsilon^2}$$

$$+\Delta_h\left(\gamma\phi_i^{n+1}+(1-\gamma)\,\phi_i^n\right), \tag{2.11}$$

where α, β, and γ are constants for the weighted average between explicit and implicit schemes. We analyze numerical schemes such as explicit, implicit, Crank–Nicolson, nonlinearly stabilized splitting (NLSS), and linearly stabilized splitting (LSS) ones in solvability, stability, and boundedness points of views. The basic strategies of proofs are based on [65]. For each cases, α, β, and γ have the following values:

1. Explicit scheme ($\alpha=\beta=\gamma=0$)

$$\frac{\phi_i^{n+1}-\phi_i^n}{\Delta t}=\frac{-\left(\phi_i^n\right)^3+\phi_i^n}{\epsilon^2}+\Delta_h\phi_i^n, \tag{2.12}$$

2. Implicit scheme ($\alpha=\beta=\gamma=1$)

$$\frac{\phi_i^{n+1}-\phi_i^n}{\Delta t}=\frac{-\left(\phi_i^{n+1}\right)^3+\phi_i^{n+1}}{\epsilon^2}+\Delta_h\phi_i^{n+1}, \tag{2.13}$$

3. Crank–Nicolson scheme ($\alpha=\beta=\gamma=\frac{1}{2}$)

$$\frac{\phi_i^{n+1}-\phi_i^n}{\Delta t} =\frac{-\left(\phi_i^{n+1}\right)^3-\left(\phi_i^n\right)^3+\phi_i^{n+1}+\phi_i^n}{2\epsilon^2}+\frac{1}{2}\Delta_h\left(\phi_i^{n+1}+\phi_i^n\right), \tag{2.14}$$

4. NLSS scheme ($\alpha = 1,\ \beta = 0,\ \gamma = 1$)

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{-\left(\phi_i^{n+1}\right)^3 + \phi_i^n}{\epsilon^2} + \Delta_h \phi_i^{n+1}, \tag{2.15}$$

5. LSS scheme ($\alpha = 0,\ \beta = -2,\ \gamma = 1$)

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{-\left(\phi_i^{n}\right)^3 - 2\phi_i^{n+1} + 3\phi_i^n}{\epsilon^2} + \Delta_h \phi_i^{n+1}. \tag{2.16}$$

2.2.2.1. *Analysis of the schemes.* We first check the unique solvability of the schemes. We need not to consider the case (2.12) since solvability of the explicit scheme clearly holds. Bearing in mind that the discrete Eq. (2.11) as the Euler equation of a functional, we can consider the following functional:

$$\begin{aligned} G(\boldsymbol{\phi}) =& \frac{1}{2\Delta t} \left\| \boldsymbol{\phi} - \boldsymbol{\phi}^n \right\|_h^2 + \left\langle \frac{\alpha \boldsymbol{\phi}^3}{4\epsilon^2} + \frac{(1-\alpha)\left(\boldsymbol{\phi}^n\right)^3}{\epsilon^2} - \frac{\beta \boldsymbol{\phi}}{2\epsilon^2} - \frac{(1-\beta)\,\boldsymbol{\phi}^n}{\epsilon^2}, \boldsymbol{\phi} \right\rangle_h \\ &+ \frac{\gamma}{2} \left\| \nabla_h \boldsymbol{\phi} \right\|_h^2 + (1-\gamma)\left(\nabla_h \boldsymbol{\phi}^n, \nabla_h \boldsymbol{\phi}\right)_h, \end{aligned} \tag{2.17}$$

Here, $\boldsymbol{\phi}\boldsymbol{\psi}$ is the element-wise multiplication $(\phi_1\psi_1, \cdots, \phi_N\psi_N)$. Let $\boldsymbol{\phi}$ and $\boldsymbol{\psi}\,(\neq \mathbf{0})$ be a fixed vector and s be a real number. Then, the following polynomial $H(s)$ is a quadratic:

$$\begin{aligned} H(s) =& G\left(\boldsymbol{\phi} + s\boldsymbol{\psi}\right) \\ =& \frac{1}{2\Delta t} \left\| (\boldsymbol{\phi} + s\boldsymbol{\psi}) - \boldsymbol{\phi}^n \right\|_h^2 + \left\langle \frac{\alpha\left(\boldsymbol{\phi} + s\boldsymbol{\psi}\right)^3}{4\epsilon^2} + + \frac{(1-\alpha)\,\boldsymbol{\phi}^3}{\epsilon^2} - \frac{\beta\left(\boldsymbol{\phi} + s\boldsymbol{\psi}\right)}{2\epsilon^2} \right. \\ & \left. - \frac{(1-\beta)\,\boldsymbol{\phi}^n}{\epsilon^2}, \boldsymbol{\phi} + s\boldsymbol{\psi} \right\rangle_h + \frac{\gamma}{2} \left\| \nabla_h \left(\boldsymbol{\phi} + s\boldsymbol{\psi}\right) \right\|_h^2 \end{aligned}$$

$$
\begin{aligned}
&+(1-\gamma)\left(\nabla_h\boldsymbol{\phi}^n,\nabla_h\left(\boldsymbol{\phi}+s\boldsymbol{\psi}\right)\right)_h,\\
=&G(\boldsymbol{\phi})+s\left\langle\frac{\boldsymbol{\phi}-\boldsymbol{\phi}^n}{\Delta t}+\frac{\alpha\boldsymbol{\phi}^3}{\epsilon^2}+\frac{(1-\alpha)\left(\boldsymbol{\phi}^n\right)^3}{\epsilon^2}-\frac{\beta\boldsymbol{\phi}}{\epsilon^2}-\frac{(1-\beta)\,\boldsymbol{\phi}^n}{\epsilon^2}\right.\\
&-\left.\ \Delta_h\left(\gamma\boldsymbol{\phi}+(1-\gamma)\,\boldsymbol{\phi}^n\right),\boldsymbol{\psi}\right\rangle_h+s^2\left\langle\frac{\boldsymbol{\psi}}{2\Delta t}+\frac{\left(3\alpha\boldsymbol{\phi}^2-\beta\right)\boldsymbol{\psi}}{2\epsilon^2}-\frac{\gamma\Delta_h\boldsymbol{\psi}}{2},\boldsymbol{\psi}\right\rangle_h\\
&+s^3\left\langle\frac{\alpha\boldsymbol{\phi}\boldsymbol{\psi}^2}{\epsilon^2},\boldsymbol{\psi}\right\rangle_h+s^4\left\langle\frac{\alpha\boldsymbol{\psi}^3}{4\epsilon^2},\boldsymbol{\psi}\right\rangle_h.
\end{aligned}
\tag{2.18}
$$

To check the convexity of the functional, we calculate the first and second derivatives of (2.18) as following:

$$
\begin{aligned}
H'(s)=&\left\langle\frac{\boldsymbol{\phi}-\boldsymbol{\phi}^n}{\Delta t}+\frac{\alpha\boldsymbol{\phi}^3}{\epsilon^2}+\frac{(1-\alpha)\left(\boldsymbol{\phi}^n\right)^3}{\epsilon^2}-\frac{\beta\boldsymbol{\phi}}{\epsilon^2}-\frac{(1-\beta)\,\boldsymbol{\phi}^n}{\epsilon^2}\right.\\
&-\left.\ \Delta_h\left(\gamma\boldsymbol{\phi}+(1-\gamma)\,\boldsymbol{\phi}^n\right),\boldsymbol{\psi}\right\rangle_h+2s\left\langle\frac{\boldsymbol{\psi}}{2\Delta t}+\frac{\left(3\alpha\boldsymbol{\phi}^2-\beta\right)\boldsymbol{\psi}}{2\epsilon^2}-\frac{\gamma\Delta_h\boldsymbol{\psi}}{2},\boldsymbol{\psi}\right\rangle_h\\
&+3s^2\left\langle\frac{\alpha\boldsymbol{\phi}\boldsymbol{\psi}^2}{\epsilon^2},\boldsymbol{\psi}\right\rangle_h+4s^3\left\langle\frac{\alpha\boldsymbol{\psi}^3}{4\epsilon^2},\boldsymbol{\psi}\right\rangle_h,
\end{aligned}
\tag{2.19}
$$

and

$$
\begin{aligned}
H''(s)=&2\left\langle\frac{\boldsymbol{\psi}}{2\Delta t}+\frac{\left(3\alpha\boldsymbol{\phi}^2-\beta\right)\boldsymbol{\psi}}{2\epsilon^2}-\frac{\gamma\Delta_h\boldsymbol{\psi}}{2},\boldsymbol{\psi}\right\rangle_h+6s\left\langle\frac{\alpha\boldsymbol{\phi}\boldsymbol{\psi}^2}{\epsilon^2},\boldsymbol{\psi}\right\rangle_h\\
&+12s^2\left\langle\frac{\alpha\boldsymbol{\psi}^3}{4\epsilon^2},\boldsymbol{\psi}\right\rangle_h,\\
=&\left(\frac{1}{\Delta t}-\frac{\beta}{\epsilon^2}\right)\|\boldsymbol{\psi}\|_h^2+\frac{3\alpha}{\epsilon^2}\left\|(\boldsymbol{\phi}+s\boldsymbol{\psi})^2\right\|_h^2+\gamma\left\|\nabla_h\boldsymbol{\psi}\right\|_h^2.
\end{aligned}
\tag{2.20}
$$

If $\alpha>0$, $\beta<\epsilon^2/\Delta t$, and $\gamma>0$, the derivative $H''(s)$ is strictly positive from (2.20); hence, the polynomial $H(s)$ is strictly convex and the functional $G(\boldsymbol{\phi})$ is bounded below. Therefore, there is the minimizer $\boldsymbol{\phi}^*$ such that $H(s) = G\left(\boldsymbol{\phi}^* + s\boldsymbol{\psi}\right)$ and $G\left(\boldsymbol{\phi}^*\right) \leq G(\boldsymbol{\phi})$ for all $\boldsymbol{\phi}$. Moreover, taking $s = 0$ to (2.19), we get

$$\begin{aligned} H'(0) = \Bigg\langle & \frac{\boldsymbol{\phi}^* - \boldsymbol{\phi}^n}{\Delta t} + \frac{\alpha\left(\boldsymbol{\phi}^*\right)^3}{\epsilon^2} + \frac{(1-\alpha)\left(\boldsymbol{\phi}^n\right)^3}{\epsilon^2} - \frac{\beta\boldsymbol{\phi}^*}{\epsilon^2} - \frac{(1-\beta)\,\boldsymbol{\phi}^n}{\epsilon^2} \\ & - \Delta_h\left(\gamma\boldsymbol{\phi}^* + (1-\gamma)\,\boldsymbol{\phi}^n\right), \boldsymbol{\psi}\Bigg\rangle_h = 0, \end{aligned} \tag{2.21}$$

Here, the equation becomes zero because of the property of the critical point of the convex functional. Since Eq. (2.21) holds regardless of $\boldsymbol{\psi}$, we have

$$\begin{aligned} \frac{\boldsymbol{\phi}^* - \boldsymbol{\phi}^n}{\Delta t} = & \frac{-\alpha\left(\boldsymbol{\phi}^*\right)^3 - (1-\alpha)\left(\boldsymbol{\phi}^n\right)^3 + \beta\boldsymbol{\phi}^* + (1-\beta)\boldsymbol{\phi}^n}{\epsilon^2} \\ & + \Delta_h\left(\gamma\boldsymbol{\phi}^* + (1-\gamma)\boldsymbol{\phi}^n\right). \end{aligned} \tag{2.22}$$

To show the uniqueness of the minimizer $\boldsymbol{\phi}^*$, let us assume that $\hat{\boldsymbol{\phi}}$ is another minimizer of the functional G; i.e., $G(\hat{\boldsymbol{\phi}}) = G(\boldsymbol{\phi}^*)$ and $\boldsymbol{\psi} = \hat{\boldsymbol{\phi}} - \boldsymbol{\phi}^* \neq \mathbf{0}$. Using strictly convexity of H, proven in above, the following inequality holds:

$$G(\boldsymbol{\phi}^* + 0.5\boldsymbol{\psi}) = H(0.5) < \frac{H(0) + H(1)}{2} = \frac{G(\boldsymbol{\phi}^*) + G(\hat{\boldsymbol{\phi}})}{2} = G(\boldsymbol{\phi}^*),$$

which leads to a contradiction. Let us denote $\boldsymbol{\phi}^*$ as $\boldsymbol{\phi}^{n+1}$, then the Eq. (2.22) has the form of the Eq. (2.11).

Inequality (2.21) is satisfied with any temporal step size for LSS and NLSS schemes. On the other hand, the Crank–Nicolson and implicit schemes holds if $\Delta t \leq 2\epsilon^2$ and $\Delta t \leq \epsilon^2$, respectively.

Next, we check that the stability of NLSS and LSS schemes, which have no restriction in solvability as discussed above. A numerical scheme is called unconditionally gradient stable when the discrete total free energy is non-increasing for any temporal step size Δt, i.e., the purpose of the part is to show the NLSS and LSS schemes inherit a decreasing the total energy properties. It is known that a numerical scheme for the AC equation is unconditionally gradient stable if we split the free energy appropriately into contractive and expansive parts,

$$\mathcal{E}(\phi) = \int_a^b \left[\frac{F(\phi)}{\epsilon^2} + \frac{1}{2}\phi_x^2 \right] dx = \mathcal{E}_c(\phi) - \mathcal{E}_e(\phi),$$

and treat the former part $\mathcal{E}_c(\phi)$ implicitly and the latter part $\mathcal{E}_e(\phi)$ explicitly [49]. The discrete energy functional $\mathcal{E}^h(\boldsymbol{\phi}^n)$ of the AC equation is defined as

$$\mathcal{E}^h(\boldsymbol{\phi}^n) = \frac{h}{4\epsilon^2} \sum_{i=1}^{N} ((\phi_i^n)^2 - 1)^2 + \frac{h}{2} \sum_{i=1}^{N-1} |\nabla_h \phi_{i+\frac{1}{2}}^n|^2$$

for each n. It is convenient to consider the discrete energy functional by decomposing into three parts:

$$\begin{aligned}
\mathcal{E}^{(1)}(\boldsymbol{\phi}^n) =& \frac{h}{2\epsilon^2} \sum_{i=1}^{N} (\phi_i^n)^2, \\
\mathcal{E}^{(2)}(\boldsymbol{\phi}^n) =& \frac{h}{2} \sum_{i=1}^{N-1} \left| \nabla_h \phi_{i+\frac{1}{2}}^n \right|^2, \\
\mathcal{E}^{(3)}(\boldsymbol{\phi}^n) =& \frac{h}{4\epsilon^2} \sum_{i=1}^{N} \left((\phi_i^n)^4 + 1 \right).
\end{aligned}$$

Using the constants α, β, and γ used in (2.11), we can rewrite the decomposed discrete energy functional as

$$\mathcal{E}^h(\boldsymbol{\phi}^n) = \mathcal{E}^h_c(\boldsymbol{\phi}^n) - \mathcal{E}^h_e(\boldsymbol{\phi}^n), \tag{2.23}$$

where

$$\begin{aligned}
\mathcal{E}^h_c(\boldsymbol{\phi}^n) =& - \beta\mathcal{E}^{(1)}(\boldsymbol{\phi}^n) + \gamma\mathcal{E}^{(2)}(\boldsymbol{\phi}^n) + \alpha\mathcal{E}^{(3)}(\boldsymbol{\phi}^n),\\
\mathcal{E}^h_e(\boldsymbol{\phi}^n) =& (1-\beta)\mathcal{E}^{(1)}(\boldsymbol{\phi}^n) - (1-\gamma)\mathcal{E}^{(2)} - (1-\alpha)\mathcal{E}^{(3)},
\end{aligned}$$

Using a discrete total energy, we can derive the numerical scheme in Eq. (2.15) from a gradient of the discrete total energy as

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = -\frac{1}{h}\nabla\mathcal{E}^h_c(\boldsymbol{\phi}^{n+1})_i + \frac{1}{h}\nabla\mathcal{E}^h_e(\boldsymbol{\phi}^n)_i, \quad \text{for } i = 1, \cdots, N. \tag{2.24}$$

Given the discrete energy functional $\mathcal{E}^{(i)}(\boldsymbol{\phi})$, one defines the Hessian $\mathbf{H}^{(i)}$ to be the Jacobian of the $\nabla\mathcal{E}^{(i)}(\boldsymbol{\phi})$ and hence the Hessian for $i = 1, 2, 3$ is represented by

$$\begin{aligned}
\mathbf{H}^{(1)} =& \nabla^2\mathcal{E}^{(1)}(\boldsymbol{\phi})\\
=& \frac{h}{\epsilon^2}\begin{pmatrix} 1 & & & & 0\\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ 0 & & & & 1 \end{pmatrix},
\end{aligned}$$

$$
\begin{aligned}
\mathbf{H}^{(2)} =& \nabla^2 \mathcal{E}^{(2)}(\boldsymbol{\phi}) \\
=& h \begin{pmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{pmatrix}, \\
\mathbf{H}^{(3)} =& \nabla^2 \mathcal{E}^{(3)}(\boldsymbol{\phi}) \\
=& \frac{3h}{\epsilon^2} \begin{pmatrix} \phi_1^2 & & & & 0 \\ & \phi_2^2 & & & \\ & & \ddots & & \\ & & & \phi_{N-1}^2 & \\ 0 & & & & \phi_N^2 \end{pmatrix},
\end{aligned}
$$

where we have used the boundary condition in Eq. (2.9). The eigenvalues of $\mathbf{H}^{(1)}$, $\mathbf{H}^{(2)}$, and $\mathbf{H}^{(3)}$ are

$$
\begin{aligned}
\lambda_k^{(1)} =& \frac{h}{\epsilon^2}, \\
\lambda_k^{(2)} =& \frac{4}{h} \sin^2 \frac{(k-1)\pi}{2N}, \\
\lambda_k^{(3)} =& \frac{3h}{\epsilon^2} \phi_k^2,
\end{aligned}
\tag{2.25}
$$

where $k = 1, 2, \cdots, N$, respectively. Note that all eigenvalues in (2.25) are non-negative. Let

$$
\mathbf{v}_k = \frac{\mathbf{w}_k}{|\mathbf{w}_k|},
$$

be the orthonormal eigenvector corresponding to the eigenvalues $\lambda_k^{(2)}$ where

$$\mathbf{w}_k = \left(\cos\frac{(k-1)\pi}{2N}, \cos\frac{3(k-1)\pi}{2N}, \cdots, \cos\frac{(2N-1)(k-1)\pi}{2N} \right),$$

then $\boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^n$ can be expressed in terms of a linear combination of $\mathbf{v}_k$ as

$$\boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^n = \sum_{k=1}^{N} \alpha_k \mathbf{v}_k. \tag{2.26}$$

We establish the decrease of the discrete energy functional. If $\boldsymbol{\phi}^{n+1}$ is the solution of Eq. (2.15) with a given $\boldsymbol{\phi}^n$, then

$$\mathcal{E}^h(\phi^{n+1}) \leq \mathcal{E}^h(\boldsymbol{\phi}^n). \tag{2.27}$$

Next, we prove Eq. (2.27). This inequality has been shown for the nonlinear gradient stabilized scheme in [28] and here we consider all five finite difference schemes. Using an exact Taylor expansion of $\mathcal{E}^h(\boldsymbol{\phi}^n)$ about $\boldsymbol{\phi}^{n+1}$ up to the second order, we have

$$\begin{aligned} \mathcal{E}^h(\boldsymbol{\phi}^{n+1}) - \mathcal{E}^h(\boldsymbol{\phi}^n) = & \left\langle \frac{1}{h} \nabla \mathcal{E}^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^n \right\rangle_h \\ & - \left\langle \frac{1}{2h} \nabla^2 \mathcal{E}^h(\boldsymbol{\xi})(\boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^n \right\rangle_h, \end{aligned} \tag{2.28}$$

where

$$\boldsymbol{\xi} = \theta \boldsymbol{\phi}^n + (1-\theta)\boldsymbol{\phi}^{n+1} \text{ and } 0 \leq \theta \leq 1.$$

For the first term of the right-hand side of Eq. (2.28), using Eq. (2.24), and the mean value theorem, we have

$$
\begin{aligned}
&\left\langle \frac{1}{h}\nabla\mathcal{E}^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \qquad (2.29)\\
&= \left\langle \frac{1}{h}\nabla\mathcal{E}_c^h(\boldsymbol{\phi}^{n+1}) - \frac{1}{h}\nabla\mathcal{E}_e^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&\quad - \left\langle \frac{\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n}{\Delta t} + \frac{1}{h}\nabla\mathcal{E}_c^h(\boldsymbol{\phi}^{n+1}) - \frac{1}{h}\nabla\mathcal{E}_e^h(\boldsymbol{\phi}^{n}), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&= -\frac{1}{h}\left\langle \nabla\mathcal{E}_e^h(\boldsymbol{\phi}^{n+1}) - \nabla\mathcal{E}_e^h(\boldsymbol{\phi}^{n}), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h - \frac{1}{\Delta t}\|\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n\|_h \\
&\leq -\frac{1}{h}\left\langle \nabla\mathcal{E}_e^h(\boldsymbol{\phi}^{n+1}) - \nabla\mathcal{E}_e^h(\boldsymbol{\phi}^{n}), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&= -\frac{1}{h}\left\langle \nabla\mathcal{E}_e^h(\boldsymbol{\eta})(\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&= -\frac{1}{h}\left\langle \left[(1-\beta)\mathbf{H}^{(1)} - (1-\gamma)\mathbf{H}^{(2)} - (1-\alpha)\mathbf{H}^{(3)}\right](\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h,
\end{aligned}
$$

where

$$\boldsymbol{\eta} = \theta\boldsymbol{\phi}^n + (1-\theta)\boldsymbol{\phi}^{n+1}$$

and $0 \leq \theta \leq 1$. Also, for the second term of the right-hand side of Eq. (2.28), using

$$\mathcal{E}^h = -\mathcal{E}^{(1)} + \mathcal{E}^{(2)} + \mathcal{E}^{(3)},$$

we have

$$
\begin{aligned}
&-\left\langle \frac{1}{2h}\nabla^2\mathcal{E}^h(\boldsymbol{\xi})(\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&= \frac{1}{2h}\left\langle \left(\mathbf{H}^{(1)}-\mathbf{H}^{(2)}-\mathbf{H}^{(3)}\right)(\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&\le \frac{1}{2h}\left\langle \left(\mathbf{H}^{(1)}-\mathbf{H}^{(2)}\right)(\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h . \qquad (2.30)
\end{aligned}
$$

From the equality (2.26), and inequalities (2.29) and (2.30), we have

$$
\begin{aligned}
&\mathcal{E}^h(\boldsymbol{\phi}^{n+1})-\mathcal{E}^h(\boldsymbol{\phi})^n \\
&\le -\frac{1}{h}\left\langle \left[(1-\beta)\mathbf{H}^{(1)}-(1-\gamma)\mathbf{H}^{(2)}-(1-\alpha)\mathbf{H}^{(3)}\right](\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&\quad + \frac{1}{2h}\left\langle \left(\mathbf{H}^{(1)}-\mathbf{H}^{(2)}\right)(\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&= \left\langle \left[\frac{2\beta-1}{2h}\mathbf{H}^{(1)}+\frac{1-2\gamma}{2h}\mathbf{H}^{(2)}+\frac{1-\alpha}{h}\mathbf{H}^{(3)}\right](\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n), \boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^n \right\rangle_h \\
&= \sum_{k,l=1}^{N}\left\langle \left[\frac{2\beta-1}{2h}\lambda_k^{(1)}+\frac{1-2\gamma}{2h}\lambda_k^{(2)}+\frac{1-\alpha}{h}\lambda_k^{(3)}\right]\alpha_k\mathbf{v}_k, \alpha_l\mathbf{v}_l \right\rangle_h \\
&= \sum_{k,l=1}^{N}\left\langle \left[\frac{2\beta-1}{2\epsilon^2}+\frac{2-4\gamma}{h^2}\sin^2\frac{(k-1)\pi}{2N}+\frac{3-3\alpha}{\epsilon^2}\eta_k^2\right]\alpha_k\mathbf{v}_k, \alpha_l\mathbf{v}_l \right\rangle_h \\
&= \sum_{k=1}^{N}\left[\frac{2\beta-1}{2\epsilon^2}+\frac{2-4\gamma}{h^2}\sin^2\frac{(k-1)\pi}{2N}+\frac{3-3\alpha}{\epsilon^2}\eta_k^2\right]\alpha_k^2. \qquad (2.31)
\end{aligned}
$$

If the right hand side of Eq. (2.31) is negative, it is guaranteed that the energy functional is non-increasing. Therefore, the nonlinearly stabilized splitting scheme ($\alpha = 1,\ \beta = 0,\ \gamma = 1$) inherits the energy non-increasing property. For the linearly stabilized splitting scheme ($\alpha = 0,\ \beta =$

$-2,\ \gamma = 1$), a sufficient condition for having negative value of Eq. (2.31) is

$$\eta_k \leq \sqrt{5/6}.$$

Finally, we show that the decrease of the discrete total energy functional implies the pointwise boundedness of the numerical solution for the AC equation [28]. Let $\boldsymbol{\phi}^n$ be a numerical solution for the discrete AC equation and

$$\mathcal{E}^h(\boldsymbol{\phi}^n) \leq \mathcal{E}^h(\boldsymbol{\phi}^{n-1}).$$

Then,

$$\begin{aligned}
\mathcal{E}^h(\boldsymbol{\phi}^0) \geq & \mathcal{E}^h(\boldsymbol{\phi}^n) \\
= & \frac{h}{4\epsilon^2}\sum_{i=1}^{N}((\phi_i^n)^2-1)^2 + \frac{h}{2}\sum_{i=1}^{N-1}|\nabla_h \phi_{i+\frac{1}{2}}^n|^2 \\
\geq & \frac{h}{4\epsilon^2}\sum_{i=1}^{N}((\phi_i^n)^2-1)^2 \\
\geq & \frac{h}{4\epsilon^2}((\phi_i^n)^2-1)^2,
\end{aligned}$$

for any $1 \leq i \leq N$. Therefore, we have

$$\|\boldsymbol{\phi}^n\|_\infty \leq \sqrt{1 + 2\epsilon\sqrt{\mathcal{E}^h(\boldsymbol{\phi}^0)/h}},$$

and it implies that the largest solution at $t = n\Delta t$ is always pointwisely bounded depending on the initial discrete total energy functional value.

2.2.3. Mass-conservative form. In [78], we propose a practically unconditionally stable numerical scheme for the conservative AC equation with a space-time dependent Lagrange multiplier. The scheme is based on the recently developed hybrid scheme for the AC equation [95] with an exact mass-conserving update at each time step.

Here, we present a discrization of a two-dimensional case. Let a computational domain Ω_h is the set of points $(x_i, y_j) = (a+(i-0.5)h, c+(j-0.5)h)$ for $i = 1, \cdots, N_x$ and $j = 1, \cdots, N_y$ where h is a uniform spatial step size $(b-a)/N_x = (d-c)/N_y$ for $\Omega = [a,b] \times [c,d]$, N_x and N_y are the numbers of cells in x- and y-directions, respectively. Denote the approximate value or the order parameter c as

$$c_{ij}^n \approx c(x_i, y_j, n\Delta t),$$

and the discrete order parameter at time $n\Delta t$ is defined as

$$\mathbf{c}^n = \begin{pmatrix} c_{11}^n & c_{12}^n & \cdots & c_{1N_y}^n \\ c_{21}^n & c_{22}^n & \cdots & c_{2N_y}^n \\ \vdots & \vdots & \ddots & \vdots \\ c_{N_x1}^n & c_{N_x2}^n & \cdots & c_{N_xN_y}^n \end{pmatrix}.$$

The zero Neumann boundary condition is implemented by

$$\nabla_h^x c_{\frac{1}{2},j}^n = \nabla_h^x c_{N_x+\frac{1}{2},j}^n = \nabla_h^y c_{i,\frac{1}{2}}^n = \nabla_h^y c_{i,N_y+\frac{1}{2}}^n = 0,$$

where

$$
\begin{aligned}
\nabla_h^x c_{i+\frac{1}{2},j}^n =& \frac{c_{i+1,j}^n - c_{ij}^n}{h}, \\
\nabla_h^y c_{i,j+\frac{1}{2}}^n =& \frac{c_{i,j+1}^n - c_{ij}^n}{h},
\end{aligned}
$$

is the discrete gradient operator. Further, the discrete Laplace operator is defined as

$$
\begin{aligned}
\Delta_h c_{ij}^n =& \frac{\nabla_h^x c_{i+\frac{1}{2},j}^n - \nabla_h^x c_{i-\frac{1}{2},j}^n + \nabla_h^y c_{i,j+\frac{1}{2}}^n - \nabla_h^y c_{i,j+\frac{1}{2}}^n}{h} \\
=& \frac{c_{i+1,j}^n + c_{i-1,j}^n - 4c_{ij}^n + c_{i,j+1} + c_{i,j-1}}{h^2}.
\end{aligned}
$$

Applying an operator splitting method to Eq. (2.7), we can rewrite the equation as a sequence of simpler problems in a numerical analytic point of view:

$$
c_t = \Delta c, \tag{2.32}
$$

$$
c_t = -\frac{F'(c)}{\epsilon^2}, \tag{2.33}
$$

$$
c_t = \beta\sqrt{2F(c)}. \tag{2.34}
$$

For the first step, we calculate the temporary solution $\mathbf{c}^{n+1,1}$ numerically from $\mathbf{c}^n$, which is the solution at the n-th time step. Here, the IE method is used as a numerical scheme for solving the heat equation (2.32):

$$
\frac{c_{ij}^{n+1,1} - c_{ij}^n}{\Delta t} = \Delta_h c_{ij}^{n+1,1}.
$$

The multigrid method, known as one of the fastest method to solve common discretization problems [4, 17, 129], is applied to solve the above discrete equation. The detailed of the multigrid method can be shown in appendix. Note that the CN scheme can be also applied as in [95]; however, it is well-known that the scheme suffers from oscillatory behavior with large time step although it is also unconditionally stable and has higher order accuracy.

Next, the second temporary solution $\mathbf{c}^{n+1,2}$ is derived from Eq. (2.33) analytically with an initial condition $\mathbf{c}^{n+1,1}$ by using the separation of the variable [123]. Since there are no spatial derivatives in the governing equation of this step, we assume that c depends on only the time variable t:

$$\frac{dc(t)}{dt} = -\frac{c(t) - 3\left(c(t)\right)^2 + 2\left(c(t)\right)^3}{\epsilon^2}.$$

Here, we can assume that $c \in (0, 1)$ since $c = 0$ or $c = 1$ implies that the time derivation becomes zero. By using the separation of the variable,

$$\begin{aligned} & -\frac{dc}{c - 3c^2 + 2c^3} = \frac{dt}{\epsilon^2}, \\ \Leftrightarrow & -\frac{dc}{c(1-c)(2-c)} = \frac{dt}{\epsilon^2}, \\ \Leftrightarrow & \left(-\frac{1}{c} + \frac{1}{1-c} - \frac{4}{1-2c}\right) dc = \frac{dt}{\epsilon^2}, \end{aligned}$$

and integrating both sides,

$$\begin{aligned}
&\int_{c_{ij}^{n+1,1}}^{c_{ij}^{n+1,2}} \left(-\frac{1}{c} + \frac{1}{1-c} - \frac{4}{1-2c}\right) dc = \int_{n\Delta t}^{(n+1)\Delta t} \frac{dt}{\epsilon^2}, \\
\Leftrightarrow &\ln\left|\frac{(1-2c_{ij}^{n+1,2})^2}{c_{ij}^{n+1,2}(1-c_{ij}^{n+1,2})}\right| - \ln\left|\frac{(1-2c_{ij}^{n+1,1})^2}{c_{ij}^{n+1,1}(1-c_{ij}^{n+1,1})}\right| = \frac{\Delta t}{\epsilon^2}, \\
\Leftrightarrow &\left|\frac{(1-2c_{ij}^{n+1,2})^2}{c_{ij}^{n+1,2}(1-c_{ij}^{n+1,2})}\right| = \left|\frac{(1-2c_{ij}^{n+1,1})^2}{c_{ij}^{n+1,1}(1-c_{ij}^{n+1,1})}\right| e^{\frac{\Delta t}{\epsilon^2}}. \qquad (2.35)
\end{aligned}$$

We can directly get rid of the absolute value functions since both terms in the functions are positive with assumption $c \in (0,1)$. By expanding and simplifying, we can rewrite Eq. (2.35) as a quadratic polynomial of $c_{ij}^{n+1,2}$:

$$(A+4)\left(c_{ij}^{n+1,2}\right)^2 - (A+4)c_{ij}^{n+1,2} + 1 = 0,$$

where

$$A = \frac{\left(1-2c_{ij}^{n+1,1}\right)^2}{c_{ij}^{n+1,1}\left(1-c_{ij}^{n+1,1}\right)} e^{\frac{\Delta t}{\epsilon^2}}.$$

Solving the polynomial, we get

$$\begin{aligned}
c_{ij}^{n+1,2} =& \frac{1}{2} \pm \frac{\sqrt{A}}{2\sqrt{A+4}} \\
=& \frac{1}{2} \pm \frac{\left|1-2c_{ij}^{n+1,1}\right|}{2\sqrt{\left(1-2c_{ij}^{n+1,2}\right)^2 + 4c_{ij}^{n+1,1}\left(1-c_{ij}^{n+1,1}\right)e^{-\frac{\Delta t}{\epsilon^2}}}}.
\end{aligned}$$

From the boundedness of $c_{ij}^{n+1,2}$ in $[0,1]$, we can cancel out the plus-minus sign and the absolute value function. Therefore, the solution can be simplified as follow:

$$c_{ij}^{n+1,2} = \frac{1}{2} - \frac{1-2c_{ij}^{n+1,1}}{2\sqrt{\left(1-2c_{ij}^{n+1,1}\right)^2 + 4c_{ij}^{n+1,1}\left(1-c_{ij}^{n+1,1}\right)e^{-\frac{\Delta t}{\epsilon^2}}}}.$$

Finally, we discretize Eq. (2.34) as

$$\frac{c_{ij}^{n+1} - c_{ij}^{n+1,2}}{\Delta t} = \beta^{n+1,2}\sqrt{2F(c_{ij}^{n+1,2})},$$

or

$$c_{ij}^{n+1} = c_{ij}^{n+1,2} + \Delta t \beta^{n+1,2}\sqrt{2F(c_{ij}^{n+1,2})}. \tag{2.36}$$

To satisfy the mass conservation property in a discrete sense, the following condition should holds from Eq. (2.36):

$$\begin{aligned}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} c_{ij}^0 = & \sum_{i=1}^{N_X}\sum_{j=1}^{N_y} c_{ij}^{n+1} \\ = & \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left(c_{ij}^{n+1,2} + \Delta t \beta^{n+1,2}\sqrt{2F(c_{ij}^{n+1,2})}\right),\end{aligned}$$

and it implies

$$\beta^{n+1,2} = \frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left(c_{ij}^0 - c_{ij}^{n+1,2}\right)}{\Delta t \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sqrt{2F(c_{ij}^{n+1,2})}}.$$

Here, the denominator of β can has a zero value pointwisely, i.e., $c_{ij}^{n+1,2} = 0$ or 1 (or $F(c_{ij}^{n+1,2}) = 0$ for some specific i and j. However, the summation cannot be zero since we does not consider the constant case: $\mathbf{c} \equiv 0$ or $\mathbf{c} \equiv 1$. Note that if $\mathbf{c}$ is not a constant, there should be at least one c_{ij} in $(0, 1)$ because of a diffused-interface.

In summery, the proposed algorithm for solving the conservative AC equation numerically in [78] is

$$\frac{c_{ij}^{n+1,1} - c_{ij}^{n}}{\Delta t} = \Delta_h c_{ij}^{n+1,1}, \tag{2.37}$$

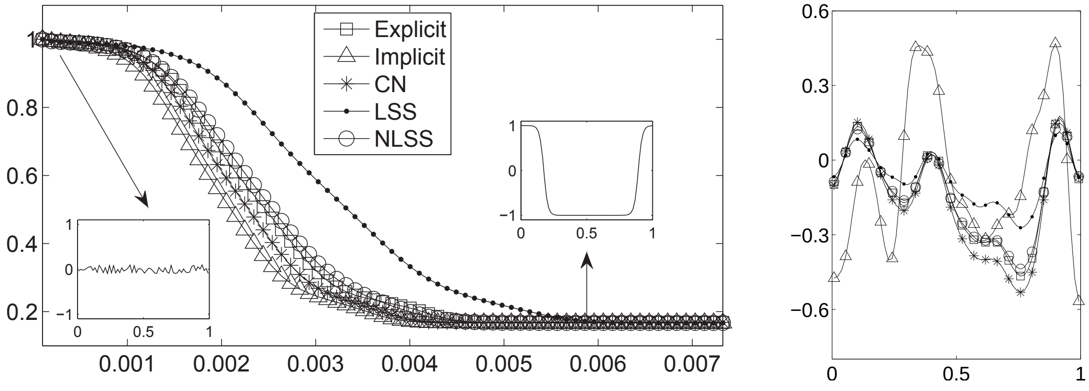
$$c_{ij}^{n+1,2} = \frac{1}{2} - \frac{1 - 2c_{ij}^{n+1,1}}{2\sqrt{(1 - 2c_{ij}^{n+1,1})^2 + 4c_{ij}^{n+1,1}(1 - c_{ij}^{n+1,1})e^{-\frac{\Delta t}{\epsilon^2}}}}, \tag{2.38}$$

$$c_{ij}^{n+1} = c_{ij}^{n+1,2} + \Delta t \beta^{n+1,2} \sqrt{2F(c_{ij}^{n+1,2})}. \tag{2.39}$$

2.2.4. Numerical experiments.

2.2.4.1. *Decrease of the total energy.* To check the robustness of the numerical schemes, we consider the evolution of the discrete total energy. The random perturbation $0.1\text{rand}(x)$ is given as an initial condition $\phi(x, 0)$ on $\Omega = (0, 1)$ with 64 grid points, where $\text{rand}(x)$ is a random number between -1 and 1.

We use the simulation parameters ϵ_6 and $\Delta t = 0.4h^2$. In Fig. 2.5(a), the temporal evolution of the non-dimensional discrete total energy $\mathcal{E}^h(\phi^n)/\mathcal{E}^h(\phi^0)$ is shown. The total discrete energy is non-increasing. Also, the inscribed small figures are the concentration fields at the indicated times. Figure 2.5(b) is a snapshot of $\phi(x,t)$ at $t = 0.0015$.



(a) (b)

FIGURE 2.5. (a) Temporal evolution of non-dimensional discrete total energy $\mathcal{E}^h(\boldsymbol{\phi}^n)/\mathcal{E}^h(\boldsymbol{\phi}^0)$ with an initial data, $\phi(x,0) = 0.2\text{rand}(x)$. (b) Snapshot of $\phi(x,t)$ at $t = 0.0015$.

2.2.4.2. *Motion by mean curvature.* It was formally proved that, as $\epsilon \to 0$, the zero level set of ϕ evolves according to the geometric law (2.5)

$$\begin{aligned} V &= -\kappa \\ &= -\left(\frac{1}{R_1} + \frac{1}{R_2}\right), \end{aligned} \tag{2.40}$$

where R_1 and R_2 are the principal radii of curvatures at the points of the surface [2]. In a two-dimensional space, Eq. (2.40) becomes

$$V = -\frac{1}{R}.$$

An initial condition is given as a circle with center (0.5, 0.5) and radius $R_0 = 0.35$ on the computational domain $\Omega = (0,1) \times (0,1)$:

$$\phi(x,y,0) = \tanh\left(\frac{R_0 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{\sqrt{2}\epsilon}\right).$$

Let R_0 and $R(t)$ be the initial radius and the radius at time t of the circle, respectively. Then, Eq. (2.40) becomes

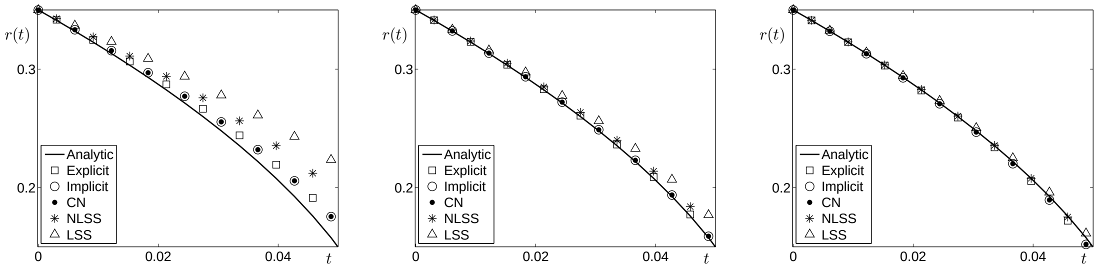
$$\frac{dR(t)}{dt} = -\frac{1}{R(t)}.$$

Therefore, analytic solution is given as

$$R(t) = \sqrt{R_0^2 - 2t}.$$

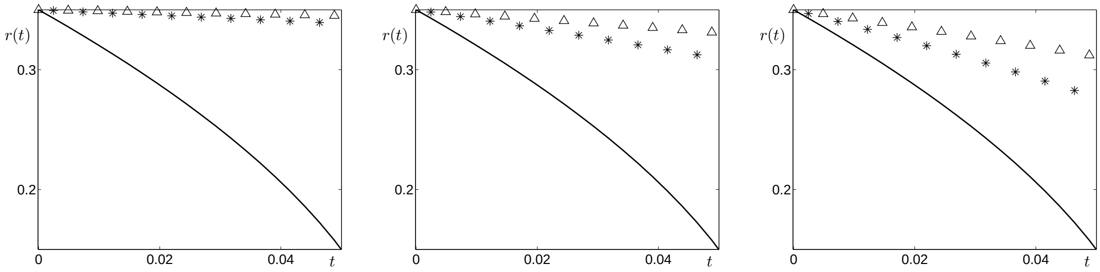
In order to compare the motion by mean curvature flow with several numerical schemes, we implement numerical simulations with various ϵ values (ϵ_4, ϵ_8, and ϵ_{12}), $h = 1/64$, and $T = 250h^2$. Figures 2.6 and 2.7 show the results with $\Delta t = 0.1h^2$ and $\Delta t = 10h^2$, respectively.

With different ϵ values, the numerical solutions have different errors comparing with the analytic solution as shown in Fig. 2.6. The most sharped-interface case, ϵ_4 has the biggest errors among our choices. On the other hand, ϵ_8 case has the most accurate solutions for the explicit, implicit, and the Crank–Nicolson schemes and the speed of shrinking is little faster than other cases in ϵ_{12} case.



(a) ϵ_4 (b) ϵ_8 (c) ϵ_{12}

FIGURE 2.6. Comparison of temporal evolutions of the radius with various ϵ values and $\Delta t = 0.1h^2$ from $t = 0$ to $t = 250h^2$ in two-dimensional space.



(a) ϵ_4 (b) ϵ_8 (c) ϵ_{12}

FIGURE 2.7. Comparison of temporal evolutions of the radius with various ϵ values and $\Delta t = 10h^2$ from $t = 0$ to $t = 250h^2$ in two-dimensional space.

In Fig. 2.7, we could check the effect of ϵ values as in Fig. 2.6. Similar to previous case, the speed of shrinking is proportional to the magnitude of ϵ. Comparing the effects in time step size, the numerical results with various schemes are close to analytic solution when $\Delta t = 0.1h^2$. With a large time step $\Delta t = 10h^2$, we have only two results with LSS and NLSS as shown in Fig. 2.7 and we can see that the gaps between analytic and numerical results are large. The other schemes have unstable results with large time step size.

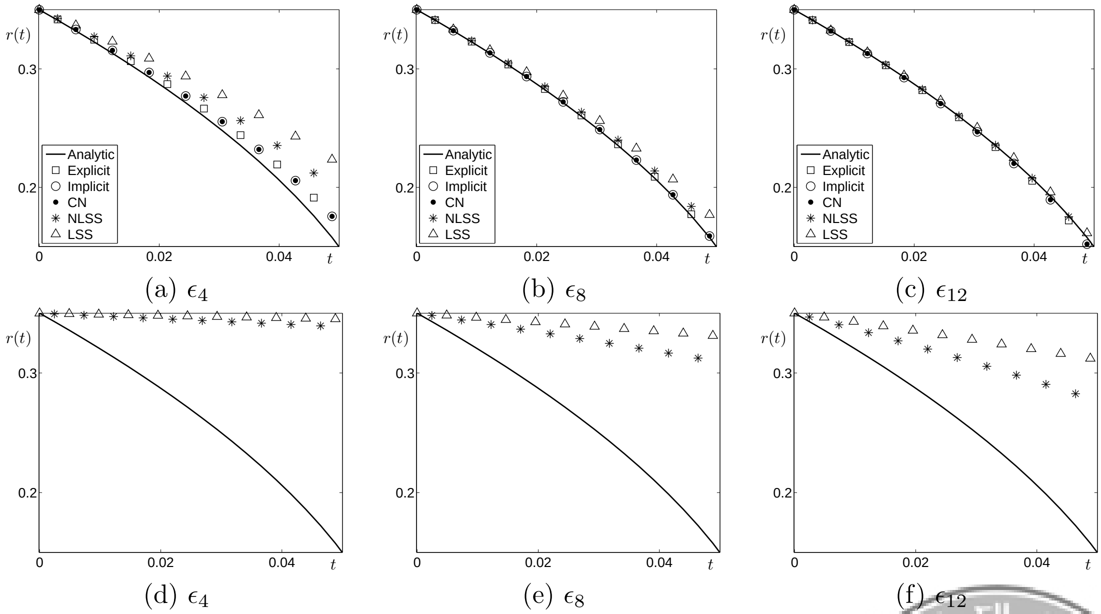
Moreover, Eq. (2.5) becomes

$$\frac{dR(t)}{dt} = -\frac{1}{2R(t)}.$$

in three-dimensional space and the radius of an sphere evolves with

$$R(t) = \sqrt{R_0^2 - 4t}.$$

The numerical results comparing with analytic solution are shown in Fig. 2.8. The behavior of numerical solutions is similar to results in the previous two-dimensional test. We used all same parameter values as the two-dimensional case except $T = 120h^2$.



(a) ϵ_4 (b) ϵ_8 (c) ϵ_{12}

(d) ϵ_4 (e) ϵ_8 (f) ϵ_{12}

FIGURE 2.8. Comparison of temporal evolutions of the radius with various ϵ values when $\Delta t = 0.1h^2$ (upper row) and (b) $\Delta t = 10h^2$ (lower row) from $t = 0$ to $t = 250h^2$ in three-dimensional space.

2.2.4.3. *Mass-conservative form.* In this section, we perform numerical experiments such as the basic mechanism of the model, a comparison with previous model, and the temporal evolution of drops in a two-dimensional space. Note that the equilibrium state of the order parameter is given by

$$c = \frac{1}{2}\left(1 + \tanh\left(\frac{x}{2\epsilon}\right)\right),$$

varies from 0.05 to 0.95 over a distance of approximately $4\epsilon\tanh^{-1}(0.9)$ across the interfacial regions (See Fig. 2.9). Therefore, if we want this value to be approximately m grid points, then ϵ value is given as $\epsilon_m = hm/[4\tanh^{-1}(0.9)]$ [75]. Throughout the rest of this section, we shall use ϵ_8 if not otherwise specified.

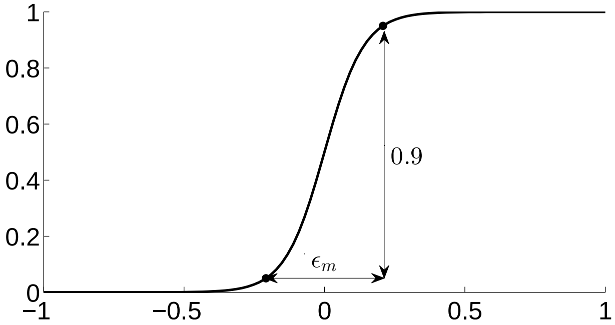


FIGURE 2.9. Phase transition of the equilibrium profile $c(x) = \left(1 + \tanh(x/(\sqrt{2}\epsilon))\right)/2$.

We start with an example which illustrates the basic mechanism of the algorithm Eqs. (2.37)–(2.39). Let us consider an elliptical initial shape (see dotted line in Fig. 2.10). If we take only the AC step Eqs. (2.37) and (2.38), then the initial shape shrinks under the motion by mean curvature (see dashed line) [60]. The position with a higher curvature moves faster than those with lower curvatures on the curve. However, with the mass correction step Eq. (2.39), the curve uniformly moves to the outward normal direction (see solid line). By continuing this process, the initial ellipse relaxes to the circular shape with the same mass.

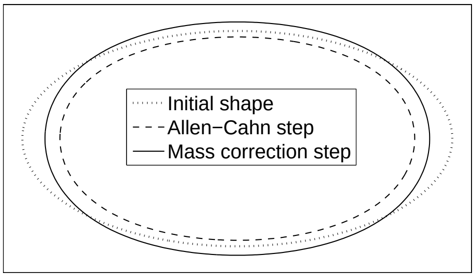


FIGURE 2.10. Basic mechanism of the proposed numerical scheme.

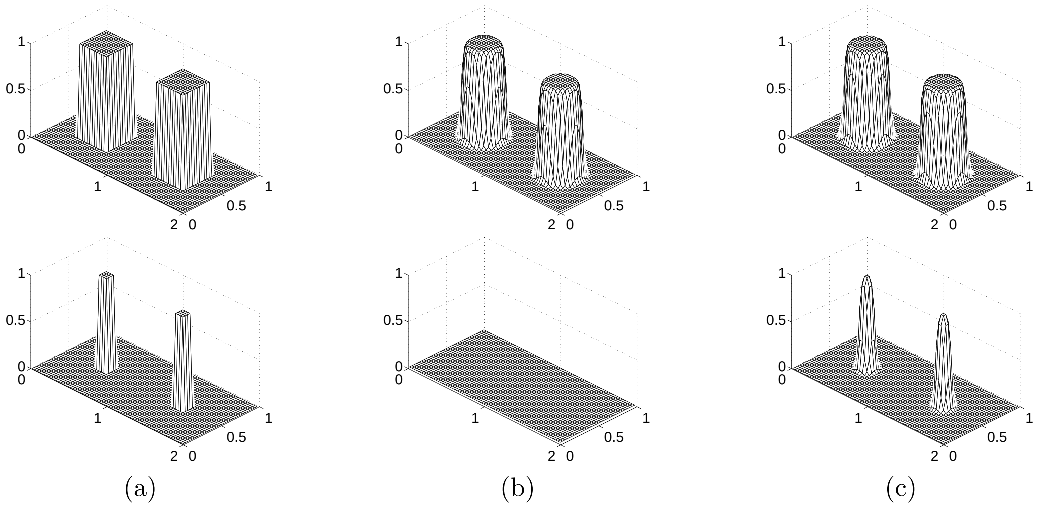
Moreover, to see the difference between two models Eqs. (2.6) and (2.7), we consider the following numerical experiments. On a computational domain $\Omega = (0,1) \times (0,2)$ with a mesh grid of 128×256, the initial conditions are given as

$$\text{(i) } \phi_{ij} = \begin{cases} 1 & \text{if } 40 \le i \le 88,\ 168 \le i \le 216, \text{ and } 40 \le i \le 88, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\text{(ii) } \phi_{ij} = \begin{cases} 1 & \text{if } 56 \le i \le 72,\ 184 \le i \le 190, \text{ and } 56 \le i \le 72, \\ 0 & \text{otherwise,} \end{cases}$$

(see Fig. 2.11(a)). The temporal step size is chosen as $\Delta t = 10^{-5}$.



(a) (b) (c)

FIGURE 2.11. (a) Initial conditions with two different shapes. (b) and (c) are numerical results from Eqs. (2.6) and (2.7), respectively.

Figures 2.11(b) and (c) show the numerical results of Eqs. (2.6) and (2.7) at a steady state with two different initial conditions, respectively. Here, we define the numerical steady state as the state when the discrete l_2 norm of the difference between ϕ^{n+1} and ϕ^n becomes less than a given tolerance, $tol = 10^{-6}$. Observing the numerical results in the top row of Fig. 2.11, we can see that both models work well when the initial feature is large enough. It should be noted that the order parameter in the outside phase is 0.009 for Eq. (2.6), on the other hand, the value is 0.0 for Eq. (2.7) with our proposed numerical scheme. The reason why the order parameters have different values is that our scheme corrects mass loss in the interfacial region. If the geometry is small, then the geometry disappears with Eq. (2.6) (see the second row of Fig. 2.11(b)). On the other hand, with our scheme, the drop stays as shown in the second row of Fig. 2.11(c).

In [19], the authors gave the evolution law for radii of spheres in n-dimensional geometric flows. For the m interfaces of radii r_i for $i = 1, 2, \ldots, m$ with $r_j < r_{j+1}$ for $j = 1, 2, \ldots, m-1$, the equations of evolution in n-dimensional case are given by

$$\frac{dr_i}{dt} = (n-1)\left(\frac{\sum_{k=1}^{m} r_k^{n-2}}{\sum_{k=1}^{m} r_k^{n-1}} - \frac{1}{r_i}\right),\ i = 1, 2, \cdots, m.$$

We consider two disjoint circular interfaces in two-dimensional space. Assume that the two interfaces have radii r and R with $r < R$, then the equations of evolution become

$$\begin{cases} \dfrac{dr}{dt} = \dfrac{2}{r+R} - \dfrac{1}{r}, \\[2ex] \dfrac{dR}{dt} = \dfrac{2}{r+R} - \dfrac{1}{R}. \end{cases} \tag{2.41}$$

From the above equations, we can get the time t_f at which smaller circle disappears (i.e., $r(t_f) = 0$) by solving a system of ordinary differential equations [16]. To solve the system (2.41), we consider the change of variable as follow:

$$\begin{aligned} \xi =& rR, \\ \eta =& r^2 + R^2, \end{aligned}$$

and the system (2.41) can be rewritten as

$$\begin{aligned} \frac{d\xi}{dt} =& \frac{d}{dt}(rR) \\ =& \frac{dr}{dt}R + r\frac{dR}{dt} \end{aligned}$$

$$
\begin{aligned}
&=\left(\frac{2}{r+R}-\frac{1}{r}\right)R+r\left(\frac{2}{r+R}-\frac{1}{R}\right)\\
&=\left(\frac{2R}{r+R}-\frac{R}{r}\right)+\left(\frac{2r}{r+R}-\frac{r}{R}\right)\\
&=\frac{2(r+R)}{r+R}-\frac{R}{r}-\frac{r}{R}\\
&=2-\frac{r^2+R^2}{rR}\\
&=2-\frac{\eta}{\xi},\\
\frac{d\eta}{dt}&=\frac{d}{dt}(r^2+R^2)\\
&=2r\frac{dr}{dt}+2R\frac{dR}{dt}\\
&=2r\left(\frac{2}{r+R}-\frac{1}{r}\right)+2R\left(\frac{2}{r+R}-\frac{1}{R}\right)\\
&=\left(\frac{4r}{r+R}-\frac{2r}{r}\right)+\left(\frac{4R}{r+R}-\frac{2R}{R}\right)\\
&=\frac{4(r+R)}{r+R}-2-2\\
&=4-2-2\\
&=0. \qquad (2.42)
\end{aligned}
$$

Note that the area of two circles $\pi r^2 + \pi R^2$ is a constant since the governing equation has a conservative property and it has a good agreement with Eq. (2.42). Now, let η be a constant up to the time variable. Using the separation of variable, we can solve the differential equation as follow:

$$
\begin{aligned}
&\frac{1}{2-\dfrac{\eta}{\xi}}d\xi = dt\\
\Leftrightarrow &\frac{\xi}{2\xi-\eta}d\xi = dt\\
\Leftrightarrow &\frac{\frac{1}{2}(2\xi-\eta)+\frac{1}{2}\eta}{2\xi-\eta}d\xi = dt\\
\Leftrightarrow &\left(\frac{1}{2}-\frac{\eta}{2(\eta-2\xi)}\right)d\xi = dt\\
\Leftrightarrow &\int_{\xi_0}^{\xi(t_f)}\left(\frac{1}{2}-\frac{\eta}{2(\eta-2\xi)}\right)d\xi = \int_0^{t_f}dt\\
\Leftrightarrow &\int_{\xi_0}^{0}\left(\frac{1}{2}-\frac{1}{2}\frac{\eta}{(\eta-2\xi)}\right)d\xi = \int_0^{t_f}dt\\
&(\because \xi(t_f) = r(t_f)R(t_f) = 0)\\
\Leftrightarrow &\left[\frac{\xi}{2}+\frac{1}{2}\left(-\frac{\eta}{2}\right)\ln\left(\frac{\eta}{\eta-2\xi}\right)\right]_{\eta_0}^{0} = t_f\\
\Leftrightarrow &\left[\frac{\xi}{2}+\frac{1}{2}\left(-\frac{\eta}{2}\right)\ln\left(\frac{\eta-2\xi+2\xi}{\eta-2\xi}\right)\right]_{\eta_0}^{0} = t_f\\
\Leftrightarrow &\left[\frac{\xi}{2}+\frac{1}{2}\left(-\frac{\eta}{2}\right)\ln\left(1+\frac{2\xi}{\eta-2\xi}\right)\right]_{\eta_0}^{0} = t_f\\
\Leftrightarrow &-\frac{\xi_0}{2}+\frac{\eta_0}{4}\ln\left(1+\frac{2\xi_0}{\eta_0-2\xi_0}\right) = t_f
\end{aligned}
$$

Therefore, by replacing ξ_0 and η_0 to r_0R_0 and $r_0^2+R_0^2$, respectively, we get

$$t_f = -0.5r_0R_0 + 0.25\left(r_0^2+R_0^2\right)\ln\left(1+\frac{2r_0R_0}{(R_0-r_0)^2}\right), \tag{2.43}$$

where r_0 and R_0 are the initial radii. We present results for $r_0 = 0.05$ and $R_0 = 0.2$ using a temporal step size $\Delta t = 0.1 * h^2$ on $\Omega = (0,1) \times (0,1)$ with a mesh grid 128×128. Then $t_f = 1.7574 \times 10^{-3}$ by Eq. (2.43). For the reference solutions of r and R, we numerically solve the ordinary differential equations by using the fourth order Runge–Kutta method [12, 142, 20].

In Fig. 2.12, the solid lines represent the result from the Runge–Kutta method, dot and star represent the radius evolutions of R and r with Eq. (2.7), respectively, and circle and diamond also represent the radius evolutions of R_p and r_p with Eq. (2.6), respectively. As shown in Fig. 2.12, R grows monotonically with our numerical scheme and r disappears at the similar time as predicted from the analytic calculation. Compared to Eq. (2.7), the results from Eq. (2.6) do not predict the theoretical prediction because most mass diffuse into the bulk phase from a global mass conservative Lagrange multiplier.

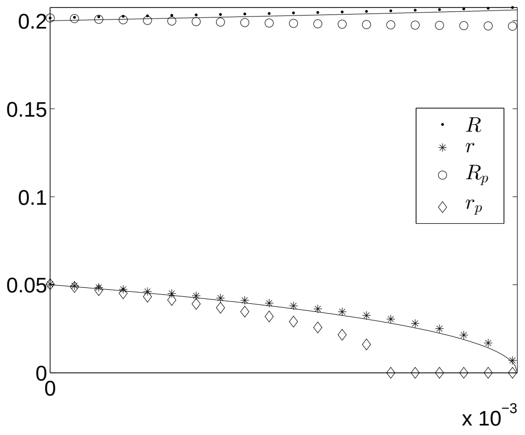


FIGURE 2.12. Evolution of the radii of two distinct circles against time. R and r are radii from Eq. (2.6) and R_p and r_p are radii from Eq. (2.7). The solid lines are the corresponding reference solutions.

2.3. Cahn–Hilliard equation

2.3.1. Governing equations. The CH equations were originated from a model of the phase separation, called the spinodal decomposition, in a binary alloy at a fixed temperature [21]. Phase separation occurs when a single phase homogeneous system composed of two mixtures in thermal equilibrium is rapidly cooled to a temperature below a critical temperature where the system is unstable with respect to infinitesimal concentration fluctuations. Since the spinodal decomposition is one of few phase transformation models in solids, the equations have been applied to various problems in theoretical and experimental material science fields as a governing equation of a phase-field method in image inpainting [11], volume reconstruction [90], block copolymer [63, 64], elastic non-equilibrium [82], multiphase fluid flow [72, 73, 77, 79], phase separation [46], flow visualization [52], quantum dot [134] and etc. It is closely related to the AC equation described in section 2.2.

One of the general forms of the CH equation can be written as follows:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = \nabla \cdot \left[M(\phi(\mathbf{x},t)) \nabla \mu(\phi(\mathbf{x},t)) \right], \ \mathbf{x} \in \Omega, 0 < t \leq T, \tag{2.44}$$

$$\mu(\phi(\mathbf{x},t)) = F'(\phi(\mathbf{x},t)) - \epsilon^2 \Delta \phi(\mathbf{x},t), \tag{2.45}$$

where $\mathbf{x}$ is the spatial variable in a domain $\Omega \in \mathbb{R}^d$ $(d = 1, 2, 3)$, t is the temporal variable in $[0, T]$, $\phi(\mathbf{x}, t)$ is the difference between concentrations of two mixtures, $M(\phi) > 0$ is a diffusional mobility, $\mu(\phi)$ is a chemical potential, T is the final time, $F(\phi) = 0.25(1 - \phi^2)^2$ is the Helmholtz free energy which has a double well potential, and $\epsilon > 0$ is a gradient energy coefficient related to an interfacial energy. The natural boundary for the CH equation is the homogenous Neumann condition (or the no flux boundary condition) as follow:

$$\mathbf{n} \cdot \nabla \phi = \mathbf{n} \cdot \nabla \mu = 0 \text{ on } \partial \Omega, \tag{2.46}$$

where $\mathbf{n}$ is the outgoing unit normal vector to the domain boundary $\partial\Omega$.

The CH equation is deduced from the Ginzburg–Landau free energy functional in a H^{-1} space and the functional has a form as follow [87]:

$$\mathcal{E}(\phi) = \int_\Omega \left[F(\phi) + \frac{\epsilon^2}{2} |\nabla \phi|^2 \right] d\mathbf{x}. \tag{2.47}$$

If we assume that $M(\phi)$ is a constant, i.e., $\nabla \cdot (M \nabla \mu) = M \Delta \mu$, differentiating (2.47) with respect to t gives

$$\begin{aligned}
\frac{d}{dt}\mathcal{E}(\phi) &= \frac{d}{dt} \int_\Omega \left[F(\phi) + \frac{\epsilon^2}{2} |\nabla \phi|^2 \right] d\mathbf{x} \\
&= \int_\Omega \left[F'(\phi)\phi_t + \epsilon^2 \left(\nabla \phi \cdot \nabla \phi_t \right) \right] d\mathbf{x}, \\
&= \int_\Omega \left[F'(\phi) - \epsilon^2 \Delta \phi \right] \phi_t d\mathbf{x} + \int_{\partial\Omega} (\mathbf{n} \cdot \nabla \phi) \phi_t d\mathbf{x} \\
&= \int_\Omega \mu \phi_t d\mathbf{x} \\
&= \int_\Omega \mu (M \Delta \mu) d\mathbf{x}
\end{aligned}$$

$$
\begin{aligned}
&= M\left[-\int_\Omega |\nabla\mu|^2\, d\mathbf{x} + \int_{\partial\Omega} \mu\,(\mathbf{n}\cdot\nabla\mu)\, d\mathbf{x}\right] \\
&= -M\int_\Omega |\nabla\mu|^2\, d\mathbf{x} \leq 0, \qquad (2.48)
\end{aligned}
$$

using integration by parts and the boundary condition (2.46) and it implies that the total energy is non-increasing in time. Besides, the derivative of the total mass with respect to t gives

$$
\begin{aligned}
\frac{d}{dt}\int_\Omega \phi d\mathbf{x} &= \int_\Omega \phi_t d\mathbf{x} \\
&= \int_\Omega M\Delta\mu d\mathbf{x} \\
&= M\int_{\partial\Omega} (\mathbf{n}\cdot\nabla\mu)\, d\mathbf{x} \\
&= 0. \qquad (2.49)
\end{aligned}
$$

Hence, the total mass conserves in time.

There have been development in many numerical algorithms to solving the CH equation such as phase-field [44, 79, 140], immersed boundary [42, 81], volume of fluid [56], front tracking [41, 130], boundary integral [36, 58], immersed interface [93, 117], and level set [110, 116] methods.

As shown in Eqs. (2.44) and (2.45), the system of equation is the fourth-order differential equation in space and it implies that there are some difficulties in numerical analysis for the CH equation. Many spatial stencils are needed and the time step restriction is stringent, for example, $\Delta t = O(h^4)$ for using the explicit methods. Moreover,the nonlinear terms does not guarantee numerical stability at the lower order spatial derivatives. The explicit scheme is simple but less efficient duo to severe time step restriction, whereas the implicit scheme is efficient but needs large linear systems of equations to solve. Likewise, each numerical methods has its own unique advantages and disadvantages for specific needs. Therefore, comparison of different schemes have been discussed to use adequate schemes for specific problems. In this dissertation, we focus on six widely used schemes in numerical analysis such as the explicit Euler' s, the implicit Euler' s, the Crank-Nicolson, the semi-implicit Euler' s, the linearly stabilized splitting and the non-linearly stabilized splitting schemes.

2.3.2. Discretization. In this section, we present fully-discrete finite-difference methods for the CH equation with six different schemes and discuss some analytical properties of the schemes. We shall discretize the CH equation in two-dimensional domain $\Omega = (a, b) \times (c, d)$. one- or three-dimensional discretizations are defined analogously.

Let positive even integers N_x and N_y be a numbers of spatial step sizes in x- and y-directions, respectively, $h = (b-a)/N_x = (d-c)/N_y$ be the uniform mesh size and the computational domain $\Omega_h = \{(x_i, y_j) : x_i = (i-0.5)h,\ y_j = (j-0.5)h,\ 1 \le i \le N_x,\ 1 \le j \le N_y\}$ be the set of cell-centered points. Let ϕ_{ij}^n and μ_{ij}^n be approximations of $\phi(x_i, y_j, n\Delta t)$ and $\mu(x_i, y_j, n\Delta t)$, respectively, where $\Delta t = T/N_t$ is the temporal step, T is the final time, N_t is the total number of time steps, and a matrix-valued phase-field $\boldsymbol{\phi}^n$ is defined as

$$\boldsymbol{\phi}^n = \begin{pmatrix} \phi^n_{11} & \phi^n_{12} & \cdots & \phi^n_{1N_y} \\ \phi^n_{21} & \phi^n_{22} & \cdots & \phi^n_{2N_y} \\ \vdots & \vdots & \ddots & \vdots \\ \phi^n_{N_x1} & \phi^n_{N_x2} & \cdots & \phi^n_{N_xN_y} \end{pmatrix}.$$

To implement the no flux boundary condition (2.46), we define the ghost points ϕ^n_{i0}, ϕ^n_{i,N_y+1}, ϕ^n_{0j}, and $\phi^n_{N_x+1,j}$ for each n as

$$\phi^n_{i0} = \phi^n_{i1},\ \phi^n_{i,N_y+1} = \phi^n_{i,N_y} \text{ for } 1 \leq i \leq N_x,$$

$$\phi^n_{0j} = \phi^n_{1j},\ \phi^n_{N_x+1,j} = \phi^n_{N_x,j} \text{ for } 1 \leq j \leq N_y.$$

Next, we define the discrete energy functional $\mathcal{E}^h$ by

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

and the discrete Laplacian Δ_h by the standard five-point stencil

$$\Delta_h \phi_{ij} = \frac{\phi_{i-1,j} + \phi_{i+1,j} - 4\phi_{ij} + \phi_{i,j-1} + \phi_{i,j+1}}{h^2}.$$

We also define the discrete l_∞ norm as

$$\|\boldsymbol{\phi}\|_\infty = \max_{\substack{1\leq i\leq N_x \\ 1\leq j\leq N_y}} |\phi_{ij}|.$$

For simplicity, we assume $M(\phi) \equiv 1$; i.e., the mobility is independent of the quantity ϕ and denote $F'(\phi)$ as $f(\phi)$. Now, we consider the following six numerical schemes as mentioned in above of Eqs. (2.44) and (2.45):

1. Explicit Euler's (EE) scheme [100]

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^{n}}{\Delta t} = \Delta_h \mu_{ij}^{n},$$

$$\mu_{ij}^{n} = f(\phi_{ij}^{n}) - \epsilon^2 \Delta_h \phi_{ij}^{n}.$$

2. Implicit Euler's (IE) scheme [48]

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^{n}}{\Delta t} = \Delta_h \mu_{ij}^{n+1},$$

$$\mu_{ij}^{n+1} = f(\phi_{ij}^{n+1}) - \epsilon^2 \Delta_h \phi_{ij}^{n+1}.$$

3. Crank–Nicolson (CN) scheme [70, 72, 74]

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^{n}}{\Delta t} = \frac{1}{2} \Delta_h (\mu_{ij}^{n+1} + \mu_{ij}^{n}),$$

$$\mu_{ij}^{n+1} = f(\phi_{ij}^{n+1}) - \epsilon^2 \Delta_h \phi_{ij}^{n+1}.$$

4. Semi-implicit Euler's (SIE) scheme [22, 32, 43, 139]

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^{n}}{\Delta t} = \Delta_h \mu_{ij}^{n+1},$$

$$\mu_{ij}^{n+1} = f(\phi_{ij}^{n}) - \epsilon^2 \Delta_h \phi_{ij}^{n+1}.$$

5. Linearly stabilized splitting (LSS) scheme [1, 11]

$$\frac{\phi_{ij}^{n+1}-\phi_{ij}^{n}}{\Delta t}=\Delta_h\mu_{ij}^{n+1},$$

$$\mu_{ij}^{n+1}=f(\phi_{ij}^{n})-2\phi_{ij}^{n}+2\phi_{ij}^{n+1}-\epsilon^2\Delta_h\phi_{ij}^{n+1}.$$

6. Non-linearly stabilized splitting (NLSS) scheme [49, 50, 76]

$$\frac{\phi_{ij}^{n+1}-\phi_{ij}^{n}}{\Delta t}=\Delta_h\mu_{ij}^{n+1}, \tag{2.50}$$

$$\mu_{ij}^{n+1}=f(\phi_{ij}^{n+1})-\epsilon^2\Delta_h\phi_{ij}^{n+1}+\phi_{ij}^{n+1}-\phi_{ij}^{n}. \tag{2.51}$$

Compared with other general schemes, LSS and NLSS are known as having larger time step sizes [49]. We prove this property roughly here and precisely in the latter part of this section.

2.3.3. Analysis of the schemes. Next, we prove the unique solvability precisely in a functional analysis' point of view and stability using a similar strategy introduced in section 2.2. To reduce the complexity of a proof, an one-dimensional case is considered here. A functional has the unique minimizer if it is not only strictly convex, but also coercive and lower semi-continuous [61, 84]. Here, we prove the unique solvability more precise way than the way. First, we define the discrete inner product for one-dimensional vector-valued functions $\boldsymbol{\phi}=(\phi_1,\cdots,\phi_N)$ and

$\boldsymbol{\psi} = (\psi_1, \cdots, \psi_N)$ as

$$\begin{aligned}\langle \boldsymbol{\phi}, \psi \rangle_h =& h\sum_{i=1}^{N} \boldsymbol{\phi}_i \psi_i, \\ (\nabla_h \boldsymbol{\phi}, \nabla_h \psi)_h =& h\sum_{i=0}^{N} \nabla_h \boldsymbol{\phi}_{i+\frac{1}{2}} \psi_{i+\frac{1}{2}}.\end{aligned}$$

Applying the nonlinear stabilized splitting scheme [49, 50], the one-dimensional CH equation can be viewd as the following discrete scheme:

$$\begin{aligned}\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} &= \Delta_h \mu_i^{n+1} \\ \mu_i^{n+1} &= \left(\phi_i^{n+1}\right)^3 - \phi_i^n - \epsilon^2 \Delta_h \phi_i^{n+1},\end{aligned} \tag{2.52}$$

for $i = 1, \cdots, N$. Using the discrete boundary condition, we have a discrete summation by parts

$$\langle \Delta_h \boldsymbol{\phi}, \boldsymbol{\psi} \rangle_h = \langle \boldsymbol{\phi}, \Delta_h \boldsymbol{\psi} \rangle_h = -(\nabla_h \boldsymbol{\phi}, \nabla_h \boldsymbol{\psi})_h,$$

and the scheme (2.52) inherits the mass conservation in the sense

$$h\sum_{i=1}^{N} \phi_i^0 = h\sum_{i=1}^{N} \phi_i^n.$$

This is readily as follows:

$$\begin{aligned}\left\langle \boldsymbol{\phi}^{n+1}, \mathbf{1} \right\rangle_h =& \langle \boldsymbol{\phi}^n, \mathbf{1} \rangle_h + \Delta t \left\langle \Delta_h \boldsymbol{\mu}^{n+1}, \mathbf{1} \right\rangle_h \\ =& \langle \boldsymbol{\phi}^n, \mathbf{1} \rangle_h - \Delta t \left(\nabla_h \boldsymbol{\mu}^{n+1}, \nabla_h \mathbf{1}\right)_h = \langle \boldsymbol{\phi}^n, \mathbf{1} \rangle_h,\end{aligned} \tag{2.53}$$

where $\mathbf{1} = (1, 1, \cdots, 1)$. Next, let us construct a discrete energy functional

$$\mathcal{E}^h(\boldsymbol{\phi}^n) = \frac{h}{4}\sum_{i=1}^{N}\left((\phi_i^n)^2 - 1\right)^2 + \frac{\epsilon^2 h}{2}\sum_{i=0}^{N}\left|\nabla_h \phi_{i+\frac{1}{2}}^n\right|^2. \tag{2.54}$$

For our convenience, we separate the discrete energy functional $\mathcal{E}^h(\boldsymbol{\phi}^n)$ into three energy terms:

$$\mathcal{E}^{(1)}(\boldsymbol{\phi}^n) = \frac{h}{2}\sum_{i=1}^{N}(\phi_i^n)^2, \tag{2.55}$$

$$\mathcal{E}^{(2)}(\boldsymbol{\phi}^n) = \frac{\epsilon^2 h}{2}\sum_{i=0}^{N}\left|\nabla_h \phi_{i+\frac{1}{2}}^n\right|^2, \tag{2.56}$$

$$\mathcal{E}^{(3)}(\boldsymbol{\phi}^n) = h\sum_{i=1}^{N}\frac{(\phi_i^n)^4 + 1}{4}. \tag{2.57}$$

Let

$$\mathcal{E}_c^h(\boldsymbol{\phi}^n) = \mathcal{E}^{(2)}(\boldsymbol{\phi}^n) + \mathcal{E}^{(3)}(\boldsymbol{\phi}^n)$$

and

$$\mathcal{E}_e^h(\boldsymbol{\phi}^n) = \mathcal{E}^{(1)}(\boldsymbol{\phi}^n)$$

so that

$$\mathcal{E}^h(\phi^n) = \mathcal{E}_c^h(\boldsymbol{\phi}^n) - \mathcal{E}_e^h(\boldsymbol{\phi}^n).$$

Evolving $\boldsymbol{\phi}^n$ as the gradient flow, we define the operator "grad$_h$" as follows:

$$\begin{aligned}\text{grad}_h \mathcal{E}^h(\boldsymbol{\phi}^n)_i =& -\frac{\Delta_h}{h}\nabla\mathcal{E}^h(\boldsymbol{\phi}^n)_i \\ =& -\Delta_h(\phi_i^n)^3 + \Delta_h\phi_i^n + \epsilon^2\Delta_h^2\phi_i^n,\end{aligned}$$

where $\Delta_h^2\phi_i = \Delta_h(\Delta_h\phi_i)$ is the discrete biharmonic operator and

$$\nabla\mathcal{E}^h(\boldsymbol{\phi}) = \left(\frac{\partial\mathcal{E}^h(\boldsymbol{\phi})}{\partial\phi_1}, \cdots, \frac{\partial\mathcal{E}^h(\boldsymbol{\phi})}{\partial\phi_N}\right)$$

is the gradient in $\mathbb{R}^N$, that is,

$$\nabla\mathcal{E}^h(\boldsymbol{\phi})_i = h\left[(\phi_i)^3 - \phi_i - \epsilon^2\Delta_h\phi_i\right].$$

Then, we can rewrite the scheme (2.52) in terms of a gradient of the discrete functional $\mathcal{E}^h(\boldsymbol{\phi})$ as

$$\frac{\phi_i^{n+1}-\phi_i^n}{\Delta t} = -\text{grad}_h\mathcal{E}_c^h(\boldsymbol{\phi}^{n+1})_i + \text{grad}_h\mathcal{E}_e^h(\boldsymbol{\phi}^n)_i, \tag{2.58}$$

for $i = 1, \ldots, N$. Let Δ_d be the matrix version of Δ_h:

$$\Delta_d = \frac{1}{h^2}\begin{pmatrix} -1 & 1 & & & 0 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & 1 & -1 \end{pmatrix}.$$

The matrix $-\Delta_d$ is the semi-positive definite with eigenvalues

$$\lambda_i = \frac{4}{h^2}\sin^2\frac{(i-1)\pi}{2N},$$

for $i = 1, \cdots, N$. Let

$$\mathbf{v}_i = \mathbf{w}_i / |\mathbf{w}_i|$$

be the orthonormal eigenvector of $-\Delta_d$ corresponding to the eigenvalue λ_i, where

$$\mathbf{w}_i = \left(\cos\frac{(i-1)\pi}{2N}, \cos\frac{3(i-1)\pi}{2N}, \ldots, \cos\frac{(2N-1)(i-1)\pi}{2N}\right).$$

For simplicity, we define by

$$\mathbf{v}_1 = \mathbf{1},$$

and denote

$$H_0 = \{\boldsymbol{\psi} \in \mathbb{R}^N \mid \langle\boldsymbol{\psi}, \mathbf{1}\rangle_h = 0\}.$$

For

$$\mathbf{X} = \sum_{i=1}^{N} x_i \mathbf{v}_i,$$
$$\mathbf{Y} = \sum_{i=1}^{N} y_i \mathbf{v}_i,$$

in $\mathbb{R}^N$ such that $\mathbf{X} \in H_0$ or $\mathbf{Y} \in H_0$, we also define an inner product by:

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{-1,h} := h \sum_{i=2}^{N} \lambda_i^{-1} x_i y_i.$$

Note that we have the identity

$$\begin{aligned} \langle \mathbf{X}, \mathbf{Y} \rangle_h =& h \sum_{i=2}^{N} x_i y_i \\ =& h \sum_{i=2}^{N} \lambda_i^{-1} \lambda_i x_i y_i \\ =& \langle -\Delta_d \mathbf{X}, \mathbf{Y} \rangle_{-1,h} . \end{aligned} \tag{2.59}$$

Here, the matrix $-\Delta_d$ is the positive definite with eigenvalues

$$\lambda_i = \frac{4}{h^2} \sin^2 \frac{(i-1)\pi}{2N},$$

for $i = 1, \cdots, N$.

The Hessian of $\mathcal{E}^{(1)}(\boldsymbol{\phi})$, denoted by $\mathbf{H}^{(1)}$, is the Jacobian of $\nabla \mathcal{E}^{(1)}(\boldsymbol{\phi})$ and is thus given by

$$\begin{aligned} \mathbf{H}^{(1)} =& \nabla^2 \mathcal{E}^{(1)}(\boldsymbol{\phi}) \\ =& h I_N, \end{aligned}$$

where I_N is the identity matrix of order N, and the Hessian matrices of $\mathcal{E}^{(2)}(\boldsymbol{\phi})$ and $\mathcal{E}^{(3)}(\boldsymbol{\phi})$ are

$$\begin{aligned} \mathbf{H}^{(2)} =& \nabla^2 \mathcal{E}^{(2)}(\boldsymbol{\phi}) \\ =& -h\epsilon^2 \Delta_d, \end{aligned}$$

$$\begin{aligned}\mathbf{H}^{(3)} =& \nabla^2 \mathcal{E}^{(3)}(\boldsymbol{\phi}) \\ =& 3h\mathbf{D},\end{aligned}$$

where $\mathbf{D} = \operatorname{diag}\left(\phi_1^2, \phi_2^2, \ldots, \phi_N^2\right)$. Moreover, the eigenvalues of $\mathbf{H}^{(1)}$, $\mathbf{H}^{(2)}$, and $\mathbf{H}^{(3)}$ are

$$\begin{aligned}\lambda_i^{(1)} =& h, \\ \lambda_i^{(2)} =& \frac{4\epsilon^2}{h} \sin^2 \frac{(i-1)\pi}{2N}, \\ \lambda_i^{(3)} =& 3h\phi_i^2,\end{aligned}$$

for $i = 1, \ldots, N$. Note that $\lambda_i^{(1)}$, $\lambda_i^{(2)}$, and $\lambda_i^{(3)}$ are non-negative for all i.

Now, we construct an appropriate functional of our scheme, and then prove the existence and uniqueness of a solution for the minimizer of the functional. Let $\mathcal{F}(\boldsymbol{\phi})$ be a discrete functional such that

$$\mathcal{F}(\boldsymbol{\phi}) = \langle f(\boldsymbol{\phi}), g(\boldsymbol{\phi}) \rangle_h,$$

for functions f and g. Then, the first (discrete) variation $\delta\mathcal{F}(\boldsymbol{\phi}; \boldsymbol{\psi})$ of the $\mathcal{F}$ with respect to $\boldsymbol{\phi}$ is defined as

$$\begin{aligned}\delta\mathcal{F}(\boldsymbol{\phi}; \boldsymbol{\psi}) &= \frac{d}{ds}\mathcal{F}(\boldsymbol{\phi} + s\boldsymbol{\psi})\bigg|_{s=0} \\ &= \frac{d}{ds}\left\langle f(\boldsymbol{\phi} + s\boldsymbol{\psi}), g(\boldsymbol{\phi} + s\boldsymbol{\psi}) \right\rangle_h \bigg|_{s=0}\end{aligned}$$

$$
\begin{aligned}
&=\frac{d}{ds}\left[h\sum_{i=1}^{N} f(\phi_i+s\psi_i)g(\phi_i+s\psi_i)\right]_{s=0}\\
&=h\sum_{i=1}^{N}\left[f_{\phi_i}(\phi_i)g(\phi_i)+f(\phi_i)g_{\phi_i}(\phi_i)\right]\psi_i\\
&=\langle\nabla\mathcal{F}(\boldsymbol{\phi}),\boldsymbol{\psi}\rangle_h,
\end{aligned}
$$

for any vector-valued function $\boldsymbol{\psi}$ and a scalar s where

$$f_{\phi_i}=\frac{\partial f}{\partial\phi_i},$$

and

$$g_{\phi_i}=\frac{\partial g}{\partial\phi_i},$$

or the first variation can be also derived from an increment

$$
\begin{aligned}
\Delta F(\boldsymbol{\phi};\boldsymbol{\psi}) =&F(\boldsymbol{\phi}+\boldsymbol{\psi})-F(\boldsymbol{\phi})\\
=&\delta F(\boldsymbol{\phi};\boldsymbol{\psi})+o(\boldsymbol{\psi}),
\end{aligned}
$$

by using the Taylor's theorem. Note that $o(\boldsymbol{\psi})$ is a small-o notation: if the function $g(\mathbf{x})$ is positive and $f(\mathbf{x})$ is an arbitrary function, $f(\mathbf{x})=o(g(\mathbf{x}))$ implies that $f/g\to 0$ as $\|\mathbf{x}\|_\infty\to\infty$. We define the second variation $\delta^2 F$ as a similar manner:

$$\Delta F(\boldsymbol{\phi};\boldsymbol{\psi})=\delta F(\boldsymbol{\phi};\boldsymbol{\psi})+\delta^2 F(\boldsymbol{\phi};\boldsymbol{\psi})+o(\boldsymbol{\psi}^2),$$

i.e.,

$$\delta^2 F(\boldsymbol{\phi};\boldsymbol{\psi}) = \left.\frac{d^2F(\boldsymbol{\phi}+s\boldsymbol{\psi})}{2ds^2}\right|_{s=0}.$$

Bearing in mind that we want to have Eq. (2.58) as the first variation of a functional, we consider the following functional $G(\boldsymbol{\phi})$ on the Hilbert space

$$H = \{\boldsymbol{\phi} \mid \langle \boldsymbol{\phi}, \mathbf{1}\rangle_h = \langle \boldsymbol{\phi}^n, \mathbf{1}\rangle\}$$

such as

$$G(\boldsymbol{\phi}) = \frac{h}{2}\langle \boldsymbol{\phi},\boldsymbol{\phi}\rangle_{-1,h} - h\langle \boldsymbol{\phi}^n,\boldsymbol{\phi}\rangle_{-1,h} + h\Delta t\mathcal{E}_c^h(\boldsymbol{\phi}) - \Delta t\left\langle \nabla\mathcal{E}_e^h(\boldsymbol{\phi}^n),\boldsymbol{\phi}\right\rangle_h.$$

The first variation of $G(\boldsymbol{\phi})$ is

$$\begin{aligned}\delta G(\boldsymbol{\phi};\boldsymbol{\psi}) =& \left.\frac{dG(\boldsymbol{\phi}+s\boldsymbol{\psi})}{ds}\right|_{s=0} \\ =& h\langle \boldsymbol{\phi},\boldsymbol{\psi}\rangle_{-1,h} - h\langle \boldsymbol{\phi}^n,\boldsymbol{\psi}\rangle_{-1,h} + \Delta t\langle\nabla\mathcal{E}_c^h(\boldsymbol{\phi}),\boldsymbol{\psi}\rangle_h - \Delta t\langle\nabla\mathcal{E}_e^h(\boldsymbol{\phi}^n),\boldsymbol{\psi}\rangle_h \\ =& h\langle \boldsymbol{\phi}-\boldsymbol{\phi}^n,\boldsymbol{\psi}\rangle_{-1,h} + \Delta t\left\langle\nabla\mathcal{E}_c^h(\boldsymbol{\phi}) - \nabla\mathcal{E}_e^h(\boldsymbol{\phi}^n),\boldsymbol{\psi}\right\rangle_h, \qquad (2.60)\end{aligned}$$

where $\boldsymbol{\psi}\in H_0$ is a non-zero vector. In addition, the second variation of $G(\boldsymbol{\phi})$ is

$$\begin{aligned}\delta^2 G(\boldsymbol{\phi};\boldsymbol{\psi}) =& \frac{h}{2}\left\langle\boldsymbol{\psi},\boldsymbol{\psi}\right\rangle_{-1,h} + \frac{\Delta t}{2}\left\langle\nabla^2\mathcal{E}_c^h(\boldsymbol{\phi})\boldsymbol{\psi},\boldsymbol{\psi}\right\rangle_h \\ =& \frac{h}{2}\left\langle\boldsymbol{\psi},\boldsymbol{\psi}\right\rangle_{-1,h} + \frac{\Delta t}{2}\left\langle\left(\mathbf{H}^{(2)}+\mathbf{H}^{(3)}\right)\boldsymbol{\psi},\boldsymbol{\psi}\right\rangle_h > 0,\end{aligned}$$

and it is strictly positive without depending on $\boldsymbol{\phi}$. Now, we split

$$G(\boldsymbol{\phi}) = G_1(\boldsymbol{\phi}) + G_2(\boldsymbol{\phi})$$

to prove that the functional is coercive as follow:

$$\begin{aligned} G_1(\boldsymbol{\phi}) =& \frac{h}{2} \left\langle \boldsymbol{\phi}, \boldsymbol{\phi} \right\rangle_{-1,h} - h \left\langle \boldsymbol{\phi}^n, \boldsymbol{\phi} \right\rangle_{-1,h}, \\ G_2(\boldsymbol{\phi}) =& h\Delta t \mathcal{E}_c^h(\boldsymbol{\phi}) - \Delta t \left\langle \nabla \mathcal{E}_e^h(\boldsymbol{\phi}^n), \boldsymbol{\phi} \right\rangle_h, \end{aligned}$$

and $G_1(\boldsymbol{\phi})$ can be rewritten as

$$G_1(\boldsymbol{\phi}) = \frac{h}{2} \left\langle \boldsymbol{\phi} - \boldsymbol{\phi}^n, \boldsymbol{\phi} - \boldsymbol{\phi}^n \right\rangle_{-1,h} - \frac{h}{2} \left\langle \boldsymbol{\phi}^n, \boldsymbol{\phi}^n \right\rangle_{-1,h}.$$

Since it is clear that

$$h \left\langle \boldsymbol{\phi} - \boldsymbol{\phi}^n, \boldsymbol{\phi} - \boldsymbol{\phi}^n \right\rangle_{-1,h} \to \infty \text{ as } \|\boldsymbol{\phi}\|_2 \to \infty,$$

and

$$G_1(\boldsymbol{\phi}) \to \infty \text{ as } \|\boldsymbol{\phi}\|_2 \to \infty.$$

Moreover, the order of first term in G_2 is higher than the second term's one and it also implies

$$G_2(\boldsymbol{\phi}) \to \infty \text{ as } \|\boldsymbol{\phi}\|_2 \to \infty.$$

In conclusion,

$$G(\boldsymbol{\phi}) \to \infty \text{ as } \|\boldsymbol{\phi}\|_2 \to \infty \text{ for } \boldsymbol{\phi} \in H,$$

which is the definition of coerciveness.

If there is a constant $M > 0$ such that $\|\boldsymbol{\phi}\|_2 < M$, a Fatou's lemma for series yields

$$G(\boldsymbol{\phi}) \leq \liminf_{k\to\infty} G(\boldsymbol{\phi}_k)$$

for any bounded sequence $\{\boldsymbol{\phi}_k\}$ which converges to $\boldsymbol{\phi}$. Thus, the functional G is lower semi-continuous. Next, we prove the existence of a solution. Since G is convex and lower semi-continuous, we can assume that G is bounded below. Then, we define by

$$m = \inf\{G(\boldsymbol{\phi}) \mid \boldsymbol{\phi} \in H\} < +\infty.$$

Now, we consider a sequence $\{\boldsymbol{\phi}_k\}$ in H such that

$$\lim_{k\to\infty} G(\boldsymbol{\phi}_k) = m.$$

Since

$$||\phi_k||_2 < \infty,$$

from coercivity, there exists a subsequence $\{\phi_k^*\}$ whose limit is ϕ^*. From the lower semi-continuity of G, we have

$$G(\boldsymbol{\phi}^*) \leq \lim_{k\to\infty} G(\boldsymbol{\phi}_k^*) = m \leq G(\boldsymbol{\phi}^*),$$

i.e.,

$$G(\boldsymbol{\phi}^*) = m.$$

Next, we assume that $\boldsymbol{\phi}^*$ and $\boldsymbol{\phi}^\circ$ are the distinct vectors with

$$G(\boldsymbol{\phi}^*) = G(\boldsymbol{\phi}^\circ) = m,$$

to show uniqueness. Then,

$$G\left(\frac{1}{2}\boldsymbol{\phi}^* + \frac{1}{2}\boldsymbol{\phi}^\circ\right) < \frac{1}{2}G(\boldsymbol{\phi}^*) + \frac{1}{2}G(\boldsymbol{\phi}^\circ) = m,$$

which is contradiction by strictly convexity. Moreover, the G has the minimum value at its extremum since the functional $G(\boldsymbol{\phi})$ is differentiable. This completes the proof of uniquely existence of a solution of the functional $G(\boldsymbol{\phi})$.

Furthermore, from Eq. (2.60), for any $\boldsymbol{\psi}$, we have

$$\delta G(\boldsymbol{\phi}^*;\boldsymbol{\psi}) = h\left\langle(\boldsymbol{\phi}^* - \boldsymbol{\phi}^n), \boldsymbol{\psi}\right\rangle_{-1,h} - \Delta t\left\langle \Delta_d[\nabla\mathcal{E}_c^h(\boldsymbol{\phi}^*) - \nabla\mathcal{E}_e^h(\boldsymbol{\phi}^n)], \boldsymbol{\psi}\right\rangle_{-1,h}$$

$$
\begin{aligned}
&= \left\langle h(\boldsymbol{\phi}^* - \boldsymbol{\phi}^n) - \Delta t \Delta_d [\nabla \mathcal{E}_c^h(\boldsymbol{\phi}^*) - \nabla \mathcal{E}_e^h(\boldsymbol{\phi}^n)], \boldsymbol{\psi} \right\rangle_{-1,h} \\
&= 0.
\end{aligned}
$$

Therefore, by substituting $\boldsymbol{\phi}^{n+1}$ for $\boldsymbol{\phi}^*$, and it is true if and only if the given equation holds:

$$
\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{\Delta_h}{h} \left(\nabla \mathcal{E}_c^h(\boldsymbol{\phi}^{n+1})_i - \nabla \mathcal{E}_e^h(\boldsymbol{\phi}^n)_i \right) \tag{2.61}
$$

$$
= -\operatorname{grad}_h \mathcal{E}_c^h \left(\boldsymbol{\phi}^{n+1} \right)_i + \operatorname{grad}_h \mathcal{E}_e^h \left(\boldsymbol{\phi}^n \right)_i , \tag{2.62}
$$

for $i = 1, \ldots, N$, which is the same as Eq. (2.58).

In the next step, we consider the stability of the solution for the numerical scheme (2.52). Let $\boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1}$ be expressed in terms of $\mathbf{v}_i$, i.e.,

$$
\boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i .
$$

By the discrete mass conservation and definition of $\mathbf{v}_i$, we have

$$
\begin{aligned}
0 &= \left\langle \boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1}, \mathbf{1} \right\rangle_h \\
&= \sum_{i=1}^{N} \alpha_i \left\langle \mathbf{v}_i, \mathbf{1} \right\rangle_h \\
&= h\sqrt{N} \alpha_1 ,
\end{aligned}
$$

and therefore $\alpha_1 = 0$. If $\boldsymbol{\phi}^{n+1}$ is the solution of Eq. (2.58) with given $\boldsymbol{\phi}^n$, then

$$\mathcal{E}^h(\boldsymbol{\phi}^{n+1}) \le \mathcal{E}^h(\boldsymbol{\phi}^n). \tag{2.63}$$

We now prove the inequality (2.63). With an exact Taylor expansion of $\mathcal{E}^h(\boldsymbol{\phi}^n)$ about $\boldsymbol{\phi}^{n+1}$ up to second order, we have

$$\begin{aligned}\mathcal{E}^h\left(\boldsymbol{\phi}^n\right) =& \mathcal{E}^h\left(\boldsymbol{\phi}^{n+1}\right) + \left\langle \frac{1}{h}\nabla\mathcal{E}^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1} \right\rangle_h \\ &+ \left\langle \frac{1}{2h}\nabla^2\mathcal{E}^h(\boldsymbol{\xi})(\boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1} \right\rangle_h,\end{aligned} \tag{2.64}$$

where

$$\boldsymbol{\xi} = \theta\boldsymbol{\phi}^n + (1-\theta)\boldsymbol{\phi}^{n+1}$$

and

$$0 \le \theta \le 1.$$

Now, by using the mean value theorem and Eqs. (2.58) and (2.59), for the first term of Eq. (2.64), we have

$$\begin{aligned}\left\langle \frac{1}{h}\nabla\mathcal{E}^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1} \right\rangle_h =& \left\langle \mathrm{grad}_h\mathcal{E}^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1} \right\rangle_{-1,h} \cdot \\ =& \left\langle \mathrm{grad}_h\mathcal{E}_c^h(\boldsymbol{\phi}^{n+1}) - \mathrm{grad}_h\mathcal{E}_e^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^n - \boldsymbol{\phi}^{n+1} \right\rangle_{-1,h}\end{aligned}$$

$$
\begin{aligned}
& -\left\langle \frac{\boldsymbol{\phi}^{n+1}-\boldsymbol{\phi}^{n}}{\Delta t} + \mathrm{grad}_h \mathcal{E}_c^h(\boldsymbol{\phi}^{n+1}) - \mathrm{grad}_h \mathcal{E}_e^h(\boldsymbol{\phi}^{n}), \right. \\
& \qquad \left. \boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1} \right\rangle_{-1,h} \\
\geq & \left\langle \mathrm{grad}_h \mathcal{E}_e^h(\boldsymbol{\phi}^{n}) - \mathrm{grad}_h \mathcal{E}_e^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1} \right\rangle_{-1,h} \\
= & \left\langle \frac{1}{h}\nabla \mathcal{E}_e^h(\boldsymbol{\phi}^{n}) - \frac{1}{h}\nabla \mathcal{E}_e^h(\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1} \right\rangle_{h} \\
= & \left\langle \frac{1}{h}\nabla^2 \mathcal{E}_e^h(\boldsymbol{\eta})\left(\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}\right), \boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1} \right\rangle_{h} \\
= & \frac{1}{h}\left\langle \mathbf{H}^{(1)}\left(\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}\right), \boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1} \right\rangle_{h}, \qquad (2.65)
\end{aligned}
$$

where

$$\boldsymbol{\eta} = \theta\boldsymbol{\phi}^{n} + (1-\theta)\boldsymbol{\phi}^{n+1}$$

and

$$0 \leq \theta \leq 1.$$

For the second term of Eq. (2.64), using

$$\mathcal{E}^h = -\mathcal{E}^{(1)} + \mathcal{E}^{(2)} + \mathcal{E}^{(3)},$$

we have

$$\frac{1}{2h}\left\langle \nabla^2\mathcal{E}^h(\boldsymbol{\xi})(\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}), \boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}\right\rangle_h = \frac{1}{2h}\left\langle \left(-\mathbf{H}^{(1)}+\mathbf{H}^{(2)}+\mathbf{H}^{(3)}\right)(\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}),\right.$$

$$
\begin{aligned}
&\left.\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}\right\rangle_{h} \\
&\geq-\frac{1}{2h}\left\langle\mathbf{H}^{(1)}(\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}),\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}\right\rangle_{h}. \qquad (2.66)
\end{aligned}
$$

From inequalities (2.65) and (2.66),

$$
\begin{aligned}
\mathcal{E}^{h}(\boldsymbol{\phi}^{n})-\mathcal{E}^{h}(\boldsymbol{\phi}^{n+1}) &\geq \frac{1}{2h}\left\langle\mathbf{H}^{(1)}(\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}),\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}\right\rangle_{h} \\
&=\frac{1}{2}\left\|\boldsymbol{\phi}^{n}-\boldsymbol{\phi}^{n+1}\right\|_{h}^{2}\geq 0. \qquad (2.67)
\end{aligned}
$$

Therefore, we have proven the decrease of the discrete functional $\mathcal{E}^{h}$ for any time step Δt. Moreover, the decrease of the discrete functional $\mathcal{E}^{h}$ implies the pointwise boundedness of the numerical solution;

$$
\|\boldsymbol{\phi}^{n}\|_{\infty}\leq\sqrt{1+2\sqrt{\frac{\mathcal{E}^{h}(\boldsymbol{\phi}^{0})}{h}}}
$$

for all n [75]. Therefore, we deduce that the proposed numerical scheme is unconditionally stable.

Next, we check the solvability for a two-dimensional case concisely. In [135], the authors prove unique solvability for the unconditional gradient stable scheme. We introduce the proof for NLSS here concisely for a two-dimensional case. For any discrete cell-centered matrix-valued function $\boldsymbol{\phi}$, there exists unique discrete cell-centered matrix-valued solution $\boldsymbol{\psi}$ which has zero Neumann boundary condition mean zero (i.e. $\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\psi_{ij}=0$) for next discrete equation :

$$\mathcal{L}(\boldsymbol{\psi})(:=-\Delta_h\boldsymbol{\psi})=\phi-\frac{h^2}{N_xN_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\phi_{ij}. \tag{2.68}$$

From the zero Neumann boundary condition, we get

$$\begin{aligned}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(-\Delta_h\psi_{ij}) &= -\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{i-1,j}+\psi_{i+1,j}-4\psi_{ij}+\psi_{i,j-1}+\psi_{i,j+1}}{h^2}\\ &= -\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{(\psi_{i-1,j}+\psi_{i+1,j})-4\psi_{ij}+(\psi_{i,j-1}+\psi_{i,j+1})}{h^2}\\ &= -2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{ij}}{h^2}+4\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{ij}}{h^2}-2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{ij}}{h^2}\\ &=0,\end{aligned}$$

It means that both left and right hand sides in (2.68) have zero mean, i.e. the necessity of the unique solution existence for the equation (2.68) is proved. Now, we consider the symmetry and positive definiteness for discrete laplacian operator $\mathcal{L}$ for uniqueness. For distinct zero Neumann vector-valued functions $\boldsymbol{\psi}^1$ and $\boldsymbol{\psi}^2$,

$$\begin{aligned}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\psi_{ij}^1(-\Delta_h\psi_{ij}^2) &= -\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{ij}^1(\psi_{i-1,j}^2+\psi_{i+1,j}^2-4\psi_{ij}^2+\psi_{i,j-1}^2+\psi_{i,j+1}^2)}{h^2}\\ &= -\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{ij}^1(\psi_{i-1,j}^2+\psi_{i+1,j}^2)-4\psi_{ij}^1\psi_{ij}^2+\psi_{ij}^1(\psi_{i,j-1}^2+\psi_{i,j+1}^2)}{h^2}\\ &= -\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{(\psi_{i-1,j}^1+\psi_{i+1,j}^1)\psi_{ij}^2-4\psi_{ij}^1\psi_{ij}^2+(\psi_{i,j-1}^1+\psi_{i,j+1}^1)\psi_{ij}^2}{h^2}\end{aligned}$$

$$
\begin{aligned}
&= -\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \frac{(\psi^1_{i-1,j}+\psi^1_{i+1,j}-4\psi^1_{ij}+\psi^1_{i,j-1}+\psi^1_{i,j+1})\psi^2_{ij}}{h^2} \\
&= \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(-\Delta_h\psi^1_{ij})\psi^2_{ij}
\end{aligned}
$$

which implies the symmetry for the operator and setting $\boldsymbol{\psi}=\boldsymbol{\psi}^1=\boldsymbol{\psi}^2$, then,

$$
\begin{aligned}
\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\psi_{ij}(-\Delta_h\psi_{ij}) &= -\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{ij}(\psi_{i-1,j}+\psi_{i+1,j}-4\psi_{ij}+\psi_{i,j-1}+\psi_{i,j+1})}{h^2} \\
&= -\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{ij}(\psi_{i-1,j}-2\psi_{ij}+\psi_{i+1,j})+(\psi_{i,j-1}-2\psi_{ij}+\psi_{i,j+1})}{h^2} \\
&= \sum_{j=1}^{N_y}\sum_{i=1}^{N_x}\frac{\psi_{ij}(\psi_{ij}-\psi_{i-1,j})-\psi_{ij}(\psi_{i+1,j}-\psi_{ij})}{h^2} \\
&\quad + \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{\psi_{ij}(\psi_{ij}-\psi_{i,j-1})-\psi_{ij}(\psi_{i,j+1}-\psi_{ij})}{h^2}
\end{aligned}
$$

and for the first term of the right hand side (let FRHS),

$$
\begin{aligned}
\text{FRHS} &= \sum_{j=1}^{N_y}\frac{\sum_{i=1}^{N_x}\psi_{ij}(\psi_{ij}-\psi_{i-1,j})-\sum_{i=2}^{N_x+1}\psi_{i-1,j}(\psi_{ij}-\psi_{i-1,j})}{h^2} \\
&\quad + \sum_{i=1}^{N_x}\frac{\sum_{j=1}^{N_y}\psi_{ij}(\psi_{ij}-\psi_{i,j-1})-\sum_{j=2}^{N_u+1}\psi_{i,j-1}(\psi_{ij}-\psi_{i,j-1})}{h^2} \\
&= \sum_{j=1}^{N_y}\frac{\sum_{i=2}^{N_x}(\psi_{ij}-\psi_{i-1,j})^2+\psi_{1,j}(\psi_{1,j}-\psi_{0,j})-\psi_{N_x,j}(\psi_{N_x+1,j}-\psi_{N_x,j}))}{h^2} \\
&\quad + \sum_{i=1}^{N_x}\frac{\sum_{j=2}^{N_y}(\psi_{ij}-\psi_{i,j-1})^2+\psi_{i,1}(\psi_{i,1}-\psi_{i,0})-\psi_{i,N_y}(\psi_{i,N_y+1}-\psi_{i,N_y}))}{h^2} \\
&= \sum_{i=1}^{N_x}\sum_{j=2}^{N_y}\frac{(\psi_{ij}-\psi_{i,j-1})^2}{h^2} \geq 0.
\end{aligned}
$$

By the same process, the second term of the right hand side is also greater than 0, i.e., $\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\psi_{ij}(-\Delta_h\psi_{i}$ 0. Since the equality holds only if both terms are zero which implies ψ is a zero constant function, $\mathcal{L}$ is positive definite, i.e. $\mathcal{L}$ has an inverse operator $\mathcal{L}^{-1}$.

Now, we consider a strictly convex discrete functional $G(\boldsymbol{\phi})$ for $\boldsymbol{\phi}$ which has mean zero such that

$$
\begin{aligned}
G(\boldsymbol{\phi}) =& \frac{h^2}{2}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\psi_{ij}\mathcal{L}(\psi_{ij}) - h^2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\psi_{ij}\mathcal{L}(\psi_{ij}^k) + F_c(\boldsymbol{\phi}) \\
& - h^2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\phi_{ij}(\delta_\phi F_e(\phi_{ij}^k))
\end{aligned}
$$

where $\boldsymbol{\psi}$ is a unique solution of the equation $\mathcal{L}(\boldsymbol{\psi})(:= -\Delta t\Delta_h\boldsymbol{\psi}) = \boldsymbol{\phi}$, for $J(\boldsymbol{\psi}) = h^2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\psi_{ij}f(\psi_{ij})$, $\delta_\phi J(\boldsymbol{\psi}) := f(\boldsymbol{\psi}) + \boldsymbol{\psi}f'(\boldsymbol{\psi})$ is the discrete variational derivative, and $F_c(\boldsymbol{\phi})$ and $F_e(\boldsymbol{\phi})$ are respectively the contractive and the expansive part of the energy where chemical potential $\mu = \delta_\phi F_c(\boldsymbol{\phi}_1) - \delta_\phi F_e(\boldsymbol{\phi}_2) = (f(\boldsymbol{\phi}_1) + \boldsymbol{\phi}_1 - \epsilon^2\Delta_h\boldsymbol{\phi}_1) - \boldsymbol{\phi}_2$ for distinct matrix-valued variables $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$. From the existence of inverse operator such that $\boldsymbol{\psi} = \mathcal{L}^{-1}(\boldsymbol{\phi})$ and symmetry of $\mathcal{L}$ in above, we can rewrite G such that

$$
\begin{aligned}
G(\boldsymbol{\phi}) =& \frac{h^2}{2}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\mathcal{L}^{-1}(\phi_{ij})\phi_{ij} - h^2\sum_{i=1}^{N_x}\sum_{j=1}^{N_x}\phi_{ij}\mathcal{L}^{-1}(\phi_{ij}^k) + F_c(\boldsymbol{\phi}) \\
& - h^2\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\phi_{ij}(\delta_\phi F_e(\phi_{ij}^k))
\end{aligned}
$$

and if we take the variational derivative δ_ϕ to G,

$$\begin{aligned}\delta_\phi G(\phi^{k+1}) =& \frac{1}{2}\left[\mathcal{L}^{-1}(\boldsymbol{\phi}) + \boldsymbol{\phi}\frac{d}{d\boldsymbol{\phi}}\mathcal{L}^{-1}(\boldsymbol{\phi})\right]_{\phi^{k+1}} - \mathcal{L}^{-1}(\boldsymbol{\phi}^k) + \delta_\phi F_c(\boldsymbol{\phi}^{k+1}) - \delta_\phi F_e(\boldsymbol{\phi}^k) - C \\ =& \mathcal{L}^{-1}(\boldsymbol{\phi}^{k+1}) - \mathcal{L}^{-1}(\boldsymbol{\phi}^k) + \delta_\phi F_c(\boldsymbol{\phi}^{k+1}) - \delta_\phi F_e(\boldsymbol{\phi}^k) - C \\ =& \mathcal{L}^{-1}(\boldsymbol{\phi}^{k+1} - \boldsymbol{\phi}^k) + \delta_\phi F_c(\boldsymbol{\phi}^{k+1}) - \delta_\phi F_e(\boldsymbol{\phi}^k) - C\end{aligned}$$

where C is a constant since $\mathcal{L}^{-1}$ is clearly linear operator. From the convexity of G, variational derivative of G is zero at $\boldsymbol{\phi}^{k+1}$ if and only if $\boldsymbol{\phi}^{k+1}$ is the unique minimizer of G. Since $\mathcal{L}^{-1}(\boldsymbol{\phi}^{k+1} - \boldsymbol{\phi}^k)$ has mean zero, it implies that $\delta_\phi F_c(\boldsymbol{\phi}^{k+1}) - \delta_\phi F_e(\boldsymbol{\phi}^k) - C$ has also mean zero, i.e.

$$C = \frac{1}{N_x N_y}\left(\delta_\phi F_c(\boldsymbol{\phi}^{k+1}) - \delta_\phi F_e(\boldsymbol{\phi}^k)\right).$$

Taking the operator $\mathcal{L}$ to above equation taken the variational derivative, we finally get

$$\frac{\boldsymbol{\phi}^{k+1} - \boldsymbol{\phi}^k}{\Delta t} = \Delta_h \delta_\phi F_c(\boldsymbol{\phi}^{k+1}) - \Delta_h \delta_\phi F_e(\boldsymbol{\phi}^k)$$

which is the same form of NLSS since $\mathcal{L}(C) = 0$ and it completes the proof.

Note that the NLSS scheme approximates the following viscous CH equation with an implicit Euler's scheme:

$$\phi_t = \Delta(f(\phi) - \epsilon^2\Delta\phi + \nu\phi_t)$$

where ν is a viscosity. To see this, let us rewrite Eq. (2.51) as

$$\mu_{ij}^{n+1} = f(\phi_{ij}^{n+1}) - \epsilon^2 \Delta_h \phi_{ij}^{n+1} + \Delta t \frac{\phi_{ij}^{n+1} - \phi_{ij}^{n}}{\Delta t}.$$

Therefore, the NLSS scheme approximates the viscous CH equation when ν equals Δt. When using a large time step, we effectively take a large viscous parameter.

In [49], Eyre proved that if $\boldsymbol{\phi}^{n+1}$ is a numerical solution of Eqs. (2.50) and (2.51) with a given $\boldsymbol{\phi}^{n}$, then

$$\mathcal{E}^h(\boldsymbol{\phi}^{n+1}) \leq \mathcal{E}^h(\boldsymbol{\phi}^{n}). \tag{2.69}$$

Furthermore, authors showed the discrete energy decreasing property by using eigenvalues of the Hessian matrix of the energy functional in [76]. Using this decreasing property of the discrete total energy functional, we can show the boundedness of the numerical solution of Eqs. (2.50) and (2.51) [75]. If $\boldsymbol{\phi}^{n}$ is a numerical solution satisfying Eq. (2.69), then there exists a constant K, which is independent on n, such that

$$\|\boldsymbol{\phi}^{n}\|_\infty \leq K. \tag{2.70}$$

Suppose Eq. (2.70) is false then there is an element $\phi_{ij}^{n_K}$ such that $|\phi_{ij}^{n_K}| > K$, where $K = \sqrt{1+2\sqrt{\mathcal{E}^h(\boldsymbol{\phi}^0)/h}}$. Since the total energy is non-increasing, we have

$$\begin{aligned}
\mathcal{E}^h(\boldsymbol{\phi}^0) =& h^2F(K) \\
<& h^2F(|\phi_{ij}^{n_K}|) \\
\le& \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} h^2F(\phi_i^{n_K}) \\
\le& \mathcal{E}^h(\phi^{n_K}) \\
\le& \mathcal{E}^h(\phi^0).
\end{aligned}$$

This contradiction implies that Eq. (2.70) should be satisfied.

2.3.4. Numerical experiments. In this section, we perform the following numerical experiments: finding a relation between the ϵ value and the width of the transition layer, convergence test, linear stability analysis, the non-increase of the total energy, stability tests, and a test for an adaptive time step.

2.3.4.1. *The relation between the ϵ value and the width of the transition layer.* Before presenting the numerical experiment, we first derive an equilibrium profile of the CH equation. When the solution reaches the equilibrium state, the time derivative of the solution is clearly zero. Hence, from Eqs. (2.44) and (2.45), an equilibrium solution ϕ^* satisfies

$$0 = (\phi^*)^3 - \phi^* - \epsilon^2 \Delta \phi^*. \tag{2.71}$$

Suppose that the solution is on the infinite domain, then we can assume that it is an one-dimensional case; therefore, Eq. (2.71) can be rewritten as

$$\epsilon^2 \phi^*_{xx} = (\phi^*)^3 - \phi^*,$$

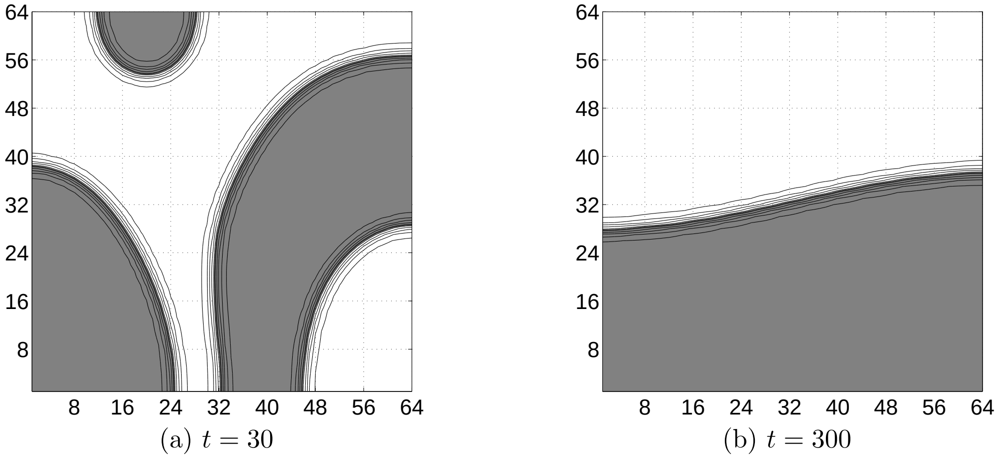
and its solution is

$$\phi^* = \tanh\left(\frac{x}{\sqrt{2}\epsilon}\right).$$

In our first numerical experiment, we consider a relation between the ϵ value and the width of the transition layer for the CH equation. From our choice of the total energy density Eq. (2.47) and an equilibrium profile $\tanh(x/(\sqrt{2}\epsilon))$ on the infinite domain, the concentration field varies from -0.9 to 0.9 over a distance of about $2\sqrt{2}\tanh^{-1}(0.9)$. Therefore, if we want this value to be about m grid points, the ϵ value need to be taken as following [44]:

$$\epsilon_m = \frac{hm}{2\sqrt{2}\tanh^{-1}(0.9)}.$$

To confirm this, we run a simulation with the initial condition $\phi(x, y, 0) = 0.1 \times \text{rand}(x, y)$ on the domain $\Omega = (0, 64) \times (0, 64)$ with $h = 1$, $\Delta t = 0.1$ and ϵ_4. Here, rand$(x, y) \in [-1, 1]$ is a randomly generated number. As can be observed from Fig. 2.13, the transition layer from $\phi = -0.9$ to $\phi = 0.9$ is almost 4 grid points as we expect from our proposed algorithm.



(a) $t = 30$

(b) $t = 300$

FIGURE 2.13. The contours of surface at $\phi = -0.9,\ -0.7,\ \cdots,\ 0.9$ at time (a) $t = 30$ and (b) $t = 300$.

2.3.4.2. *Convergence test.* Tables 2.1 and 2.2 show the discrete l_2 and maximum norms of the errors and convergence rates for space and time, respectively. Here, we refer the results and values in Tables 2.1 and 2.2 from [89]. For the simulation, the initial condition is used as

$$\phi(x, 0) = 0.1\cos(2\pi x) \text{ in } \Omega = (0, 1).$$

$\epsilon = 0.03$ and $T = 0.1$ are used. Since no analytical solutions are available, we use the relative error to calculate the convergence rate.

The spatial convergence rate is computed on the mesh grids with $N = 2^n$ for $n = 6, 7, 8, 9$. We fix the time step size as $\Delta t = 10^{-7}$. We define the error of a grid as the difference between the grid and the average of next finer grid cells as follows:

$$e_i^h := \phi^h(x_i, T) - \frac{\phi^{h/2}(x_{2i-1}, T) + \phi^{h/2}(x_{2i}, T)}{2}.$$

The convergence rate in the discrete l_2 norm sense is defined as the ratio of successive errors

$$\log_2\left(\frac{\|e^h\|_2}{\|e^{h/2}\|_2}\right),$$

and it is similarly defined for the discrete maximum norm.

The second-order accuracy for space is observed.

Table 2.1. Errors and convergence rates for space.

Mesh	64	Rate	128	Rate	256	Rate	512
$\|e^h\|_2$	6.784e-3	2.015	1.678e-3	2.003	4.188e-4	2.001	1.047e-4
$\|e^h\|_\infty$	1.329e-2	1.984	3.359e-3	1.995	8.425e-4	1.999	2.108e-4

To show the convergence of the time integration, we fix the spatial grid as $N = 512$ and choose a set of time steps $\Delta t = 2^n \times 10^{-7}$, for $n = 0$, 1, 2, 3. We define the discrete error as

$$e_i^{\Delta t} := \phi^{\Delta t}(x_i, T) - \phi^{\Delta t/2}(x_i, T)$$

and the convergence rate is defined as the ratio of successive errors,

$$\log_2\left(\frac{\|e^{\Delta t}\|_2}{\|e^{\Delta t/2}\|_2}\right).$$

The first-order accuracy for time is observed.

TABLE 2.2. Errors and convergence rates for time.

Δt	8e-7	Rate	4e-7	Rate	2e-7	Rate	1e-7
$\|e^{\Delta t}\|_2$	4.910e-6	1.000	2.455e-6	1.001	1.226e-6	0.999	6.137e-7
$\|e^{\Delta t}\|_\infty$	2.197e-4	1.000	1.099e-4	1.001	5.490e-5	0.997	2.751e-5

2.3.5. Linear stability analysis. We carry out the linear stability analysis near a solution $\phi = 0$ in one-dimensional space. Let us assume that the solution can be expressed by

$$\phi(x,t) = \sum_{k=1}^{\infty} \beta_k(t)\cos(kx), \tag{2.72}$$

where $\beta_k(t)$ is an amplification factor at the wave number k. After linearizing Eqs. (2.44) and (2.45) about the equilibrium solution, and substituting Eq. (2.72) into the linearized equations, we have

$$\frac{d\beta_k(t)}{dt} = k^2\left(1 - \epsilon^2 k^2\right)\beta_k(t). \tag{2.73}$$

The solution of Eq. (2.73) is $\beta_k(t) = \beta_k(0)\exp(\eta_k t)$, where $\eta_k = k^2(1-\epsilon^2k^2)$ is the growth rate. Note that the growth rate is positive if $\epsilon k < 1$. We denote $k_{\max}$ by the wave number which has the maximal growth rate. In addition, the numerical growth rate is defined by

$$\widetilde{\eta}_k = \frac{1}{T}\log\left(\frac{\|\boldsymbol{\phi}^n\|_\infty}{\|\boldsymbol{\phi}^0\|_\infty}\right).$$

For the numerical test, we take the initial condition

$$\phi(x,0) = 0.01\cos(kx) \text{ in } \Omega = (0,\pi),$$

$\Delta t = 10^{-8}$, $h = 2^{-9}\pi$, $\epsilon = 0.03$, and $T = 10^{-6}$ are used. Figure 2.14 illustrates the growth rate versus the wave number k. Circles and solid line are corresponding numerical results $\widetilde{\eta}_k$ and analytic solutions η_k from the linear stability analysis, respectively. The maximum growth rate is obtained at $k_{max} = 24$.

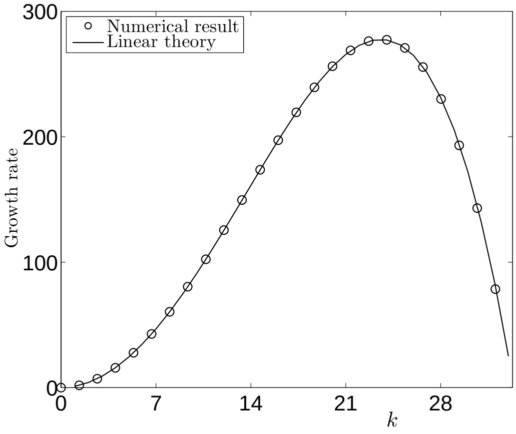


FIGURE 2.14. Growth rate versus the wave number k.

2.3.5.1. *Non-increase of the total energy.* In order to demonstrate that NLSS inherits the energy non-increasing property, we consider the temporal evolution of the discrete total energy. In the simulation, we choose $h = 1/64$, $\Delta t = 0.1$ and ϵ_4. In Fig. 2.15, the temporal evolution of the non-dimensional discrete total energy $\mathcal{E}^h(t)/\mathcal{E}^h(0)$ (solid line) of the numerical solutions with the initial state $\phi(x, y, 0) = 0.1 \times \text{rand}(x, y)$. As shown in Fig. 2.15, the energy is non-increasing during whole time evolution. This numerical result agrees well with the total energy non-increasing property.

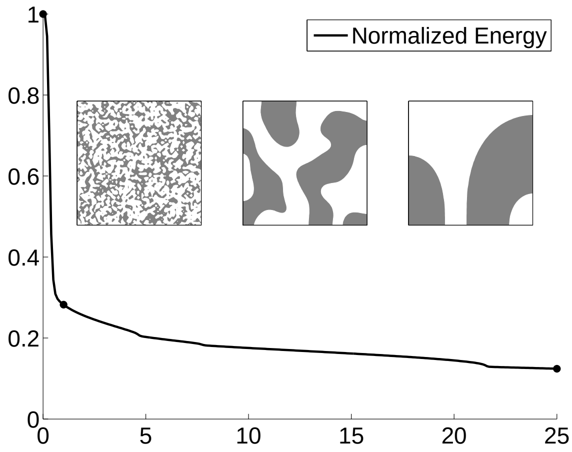


FIGURE 2.15. The time dependent non-dimensional discrete total energy $\mathcal{E}^h(t)/\mathcal{E}^h(0)$ (solid line) of the numerical solutions with the initial state $\phi(x,y,0) = 0.1 \times \text{rand}(x,y)$.

2.3.5.2. *Stability tests.* This section is contained in [89] and we refer the contents and results. We investigate the stability of the different schemes mentioned in section 2.3.2. We consider numerical solutions with random initial condition $\phi(x,0) = \text{rand}(x)$ and $\phi(x,y,0) = \text{rand}\ (x,y)$ on the unit domain for one- and two- dimensional spaces, respectively. Define Δt_c be the largest time step, which satisfies the gradient stable, i.e., $\mathcal{E}^h(\boldsymbol{\phi}^{n+1}) \leq \mathcal{E}^h(\boldsymbol{\phi}^n)$. The numerical simulations are performed on the uniform grids, $h = 1/2^n$ for $n = 5, 6, 7$ and 8. In Table 2.3, we list the values of Δt_c with different schemes for one-dimensional space. From the results, we observe that EE, IE, CN and SIE schemes are not gradient stable when we use the time step larger than Δt_c, whereas both LSS and NLSS are unconditionally gradient stable.

TABLE 2.3. Δt_c with different schemes for the initial condition $\phi(x,0) = \text{rand}(x)$.

Case	32	64	128	256
	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$
EE	3.8×10^{-4}	1.9×10^{-5}	4.7×10^{-6}	1.1×10^{-6}
IE	5.5×10^{-4}	1.1×10^{-3}	2.6×10^{-4}	6.1×10^{-5}
CN	5.6×10^{-4}	1.1×10^{-3}	1.1×10^{-4}	4.8×10^{-4}
SIE	8.2×10^{-4}	8.4×10^{-4}	8.6×10^{-4}	7.3×10^{-4}
LSS	∞	∞	∞	∞
NLSS	∞	∞	∞	∞

TABLE 2.4. Δt_c with different schemes for the initial condition $\phi(x,0) = 10\text{rand}(x)$.

Case	32	64	128	256
	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$
LSS	8.1×10^{-6}	2.0×10^{-6}	3.6×10^{-7}	8.8×10^{-8}
NLSS	∞	∞	∞	∞

Next, we also consider other numerical solutions to investigate gradient stability between LSS and NLSS with the initial data $\phi(x,0) = 10 \times \text{rand}(x,y)$. For simplicity, we perform the comparison of LSS and NLSS in one-dimensional domain instead of two-dimensional one with larger random initial condition than previous simulations. In Table 2.4, we can recognize that there is the different time step constraint in terms of numerical stability for NLSS and LSS; NLSS is still unconditionally gradient stable, but LSS is conditionally stable as shown in the reference [135].

2.3.5.3. *Comparison of the efficience of CN and NLSS schemes.* We compare the efficiency of the CN and the NLSS schemes. In the early stages of spinodal decomposition, a rapid separation of two phases occurs because the free energy $\mathcal{E}$ is high. Thus we take a small time step size in these stages. However, after the free energy has been declined sufficiently, the phases separate leisurely in the late stages. Hence we may take a large time step near the equilibrium state. In this section, we perform two tests. The first case is that the CN is used over whole simulations and the second case is that the CN scheme is used over the early stages and we alter into the NLSS that can adopt a large time step. The reason why we use the CN scheme at the early stages is to accurately evolve the rapid phase separation. We decide to change the scheme when $\|\phi^{n+1} - \phi^n\|_\infty < 2 \times 10^{-4}$. In this test, $\|\phi^{n+1} - \phi^n\|_\infty$ is less than 2×10^{-4} within 10000 iterations. Thus, in second case, we turn on the NLSS schemes after 10000th iteration. We simulate until $t = 0.66$ in the CN scheme case and until $t = 0.9$ in the NLSS scheme case. We continue the computation until $\|\phi^{n+1} - \phi^n\|_\infty$ become less than 10^{-4}. In both cases, a 256×256 mesh is used on the domain $\Omega = (0, 1) \times (0, 1)$, ϵ_4 is taken and a time step for the CN scheme, $\Delta t = 0.00001$ and for the NLSS scheme, $\Delta t = 0.0001$ was employed. Figure 2.16 shows the temporal evolution of each case. In the first case, we reach the $\|\phi^{n+1} - \phi^n\|_\infty < 10^{-4}$ with 66000 iterations. In contrast, only 18000 iterations are needed to reach the same state in the second case. With less iterations, mixed scheme with the CN and the NLSS schemes can obtain the results same as those of the CN scheme.

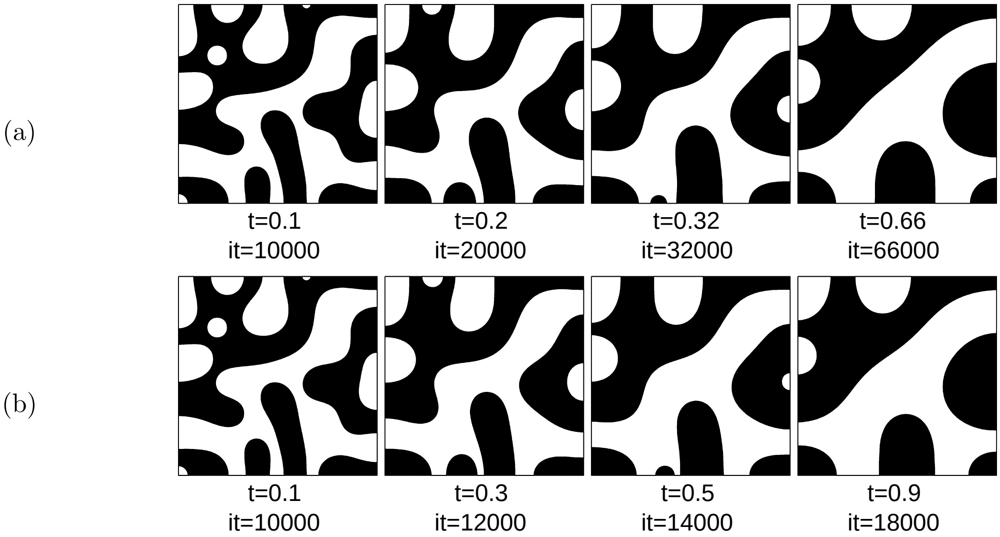


FIGURE 2.16. The temporal evolution of morphologies during a spinodal phase separation of a binary system with different time step size (a) $\Delta t = 0.00001$ using the CN scheme and (b) $\Delta t = 0.00001$ using the CN scheme until 10000th iteration and $\Delta t = 0.0001$ using the Non-linearly stabilized splitting scheme after 10000th iteration, respectively. Both cases run until $\|\phi^{n+1} - \phi^n\|_\infty < 10^{-4}$. With less iterations, mixed scheme with the CN and the NLSS schemes can obtain the results same as those of the CN scheme.

Chapter 3

Dendritic growth

3.1. Introduction

Solidification is one of the most important phenomena in production of alloys. In general, it is called a dendrite when the outcome of solidification has a tree-like structure [85]. Dendritic growth involves complex processes in a micro-scale; the diffusion of the solute, melt convection, motion of the solid phase, and their couplings play important roles in dendritic solidification. Such a complex process determines the mechanical properties; therefore, understanding dendritic growth is of great interest in industrial fields [113].

The phase-field method, described in Chapter 2, is considered as the powerful and accurate numerical tool to simulate microstructural evolution including dendritic growth. The most significant advantages using a phase-field method is the way of representing an interface [29]. As discussed in the above part of this dissertation, the phase-field method tracks an interfaces implicitly. See references [5, 14, 120, 124] for reviewing simulations of dendritic growth using a phase-field method.

In this chapter, we discuss a numerical approach to model a dendrite growth in phase-field simulations applying an operator-splitting method. Here, we consider the numerical scheme introduced in [96], which is known as a fast, robust, and accurate method, and present numerical simulations to show consistency with previous numerical experiments.

3.2. Governing equations

A standard form of the solidification model using phase-field method is given by [96]

$$\begin{aligned} \epsilon^2(\phi)\frac{\partial \phi}{\partial t} =& \nabla \cdot \left(\epsilon^2(\phi)\nabla\phi\right) + \left[\phi - \lambda U\left(1-\phi^2\right)\right]\left(1-\phi^2\right) \\ &+ \sum_{i=1}^{d}\left(|\nabla\phi|^2\,\epsilon^2(\phi)\frac{\partial\epsilon(\phi)}{\partial\phi_{x_i}}\right)_{x_i}, \qquad (3.1) \\ \frac{\partial U}{\partial t} =& D\Delta U + \frac{1}{2}\frac{\partial\phi}{\partial t}, \qquad (3.2) \end{aligned}$$

where $\epsilon(\phi)$ is the anisotropic function, ϕ is the order parameter defined in $[-1, 1]$ with $\phi = 1$ in the solid phase, $\phi = -1$ in the liquid phase, and $\phi = 0$ representing an interface, t is the temporal variable, λ is the dimensionless coupling parameter, $U = c_p(T - T_M)/L$ is the dimensionless temperature field, c_p is the specific heat at a constant pressure, T_M is the melting temperature, L is the Latent heat of fusion, d is a dimension of the domain, $D = \alpha\tau_0/\epsilon_0^2$ is the thermal diffusivity, τ_0 is the characteristic time, ϵ_0 is characteristic length, and x_i is the spatial variable. Note that $(x_1, x_2, x_3) = (x, y, z)$ when we consider the three-dimensional case. For the four-fold symmetry,

$\epsilon(\phi)$ is defined as

$$\epsilon(\phi) = (1 - 3\epsilon_4)\left(1 + \frac{4\epsilon_4}{1 - 3\epsilon_4}\frac{\sum_{i=1}^{d}\phi_{x_i}^4}{|\nabla\phi|^4}\right), \tag{3.3}$$

where ϵ_4 is a parameter for the anisotropy of interfacial energy.

3.3. Discretization

We introduce a robust hybrid numerical scheme for crystal growth proposed in [96]. Here, we focus on the two-dimensional case and present the discretization on the two-dimensional domain. The discrete equation of Eqs. (3.1) and (3.2) is derived as follow:

$$\begin{aligned}\epsilon^2(\phi^n)\frac{\phi^{n+1} - \phi^n}{\Delta t} =& \epsilon^2(\phi^n)\Delta_h\phi^{n+1,2} + 2\epsilon(\phi^n)\nabla_h\epsilon(\phi^n)\cdot\nabla_h\phi^n - F'(\phi^{n+1}) \\ &+ \left(|\nabla_h\phi|^2\epsilon(\phi)\frac{\partial\epsilon(\phi)}{\partial\phi_x}\right)_x^n + \left(|\nabla_h\phi|^2\epsilon(\phi)\frac{\partial\epsilon(\phi)}{\partial\phi_y}\right)_y^n \\ &- 4\lambda U^n F(\phi^{n+1,1}), \end{aligned} \tag{3.4}$$

$$\frac{U^{n+1} - U^n}{\Delta t} = D\Delta_h U^{n+1} + \frac{\phi^{n+1} - \phi^n}{2\Delta t}, \tag{3.5}$$

where $F(\phi) = 0.25(\phi^2 - 1)^2$, $F'(\phi) = \phi(\phi^2 - 1)$, and $\phi^{n+1,k}$ for $k = 1, 2$ are defined in an operator splitting method. The formulation applying the operator splitting method is written as

$$\begin{aligned}\epsilon^2(\phi^n)\frac{\phi^{n+1,1} - \phi^n}{\Delta t} =& 2\epsilon(\phi^n)\nabla_h\epsilon(\phi^n)\cdot\nabla_h\phi^n \\ &+ \left(|\nabla_h\phi|^2\epsilon(\phi)\frac{\partial\epsilon(\phi)}{\partial\phi_x}\right)_x^n + \left(|\nabla_h\phi|^2\epsilon(\phi)\frac{\partial\epsilon(\phi)}{\partial\phi_x}\right)_y^n, \end{aligned} \tag{3.6}$$

$$\epsilon^2(\phi^n)\frac{\phi^{n+1,2}-\phi^{n+1,1}}{\Delta} = \epsilon^2(\phi^n)\Delta_h\phi^{n+1,2} - 4\lambda U^n F(\phi^{n+1,1}), \tag{3.7}$$

$$\epsilon^2(\phi^n)\frac{\phi^{n+1}-\phi^{n+1,2}}{\Delta t} = -F'(\phi^{n+1}). \tag{3.8}$$

Here, we can simplify the last two terms in Eq. (3.6) as follows:

$$\begin{aligned}
|\nabla_h\phi|^2\frac{\partial\epsilon(\phi)}{\partial\phi_x} &= \frac{16\epsilon_4\phi_x(\phi_x^2\phi_y^2-\phi_y^4)}{|\nabla_h\phi|^4},\\
|\nabla_h\phi|^2\frac{\partial\epsilon(\phi)}{\partial\phi_y} &= \frac{16\epsilon_4\phi_y(\phi_x^2\phi_y^2-\phi_y^4)}{|\nabla_h\phi|^4}.
\end{aligned}$$

Next, Eq. (3.8) can be considered as a following continuous equation:

$$\phi_t = \frac{\phi-\phi^3}{\epsilon^2},$$

since it has an approximation form using an implicit Euler's scheme with the initial condition $\phi^{n+1,2}$. Note that we analytically solve the similar equation in previous section 2.2 using the separation of the variable. Applying the same strategy used in above, the analytic solution is derived as

$$\phi^{n+1} = \frac{\phi^{n+1,2}}{\sqrt{e^{-2\Delta t/\epsilon^2(\phi^n)} + (\phi^{n+1,2})^2(1-e^{-2\Delta t/\epsilon^2(\phi^n)})}}.$$

Therefore, the Eqs. (3.4) and (3.5), applying the proposed operator splitting method, can be rewritten as

$$\begin{aligned}\epsilon^2(\phi^n)\frac{\phi^{n+1,1}-\phi^n}{\Delta t} =& \left(\frac{16\epsilon_4\phi_x(\phi_x^2\phi_y^2-\phi_y^4)}{|\nabla_h\phi|^4}\right)_x^n + \left(\frac{16\epsilon_4\phi_y(\phi_x^2\phi_y^2-\phi_y^4)}{|\nabla_h\phi|^4}\right)_y^n \\ &+ 2\epsilon(\phi^n)\nabla_h\epsilon(\phi^n)\cdot\nabla_h\phi^n, \qquad (3.9)\end{aligned}$$

$$\epsilon^2(\phi^n)\frac{\phi^{n+1,2}-\phi^{n+1,1}}{\Delta} =\epsilon^2(\phi^n)\Delta_h\phi^{n+1,2} - 4\lambda U^n F(\phi^{n+1,1}), \qquad (3.10)$$

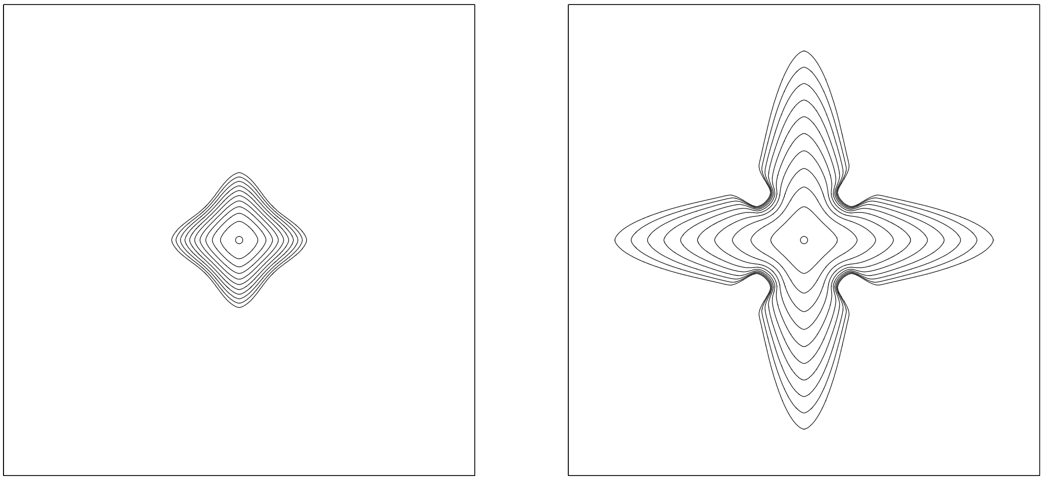
$$\phi^{n+1} =\frac{\phi^{n+1,2}}{\sqrt{e^{-2\Delta t/\epsilon^2(\phi^n)}+(\phi^{n+1,2})^2(1-e^{-2\Delta t/\epsilon^2(\phi^n)})}}, \qquad (3.11)$$

$$\frac{U^{n+1}-U^n}{\Delta t} =D\Delta_h U^{n+1} + \frac{\phi^{n+1}-\phi^n}{2\Delta t}. \qquad (3.12)$$

3.4. Numerical experiments

In the numerical experiments performed in the section, we choose a relatively large time step to show the stability of the used scheme who does not suffer the traditional time step restriction $\Delta t \le O(h^2)$. Figure 3.1 shows the evolution of phase-field contours for a dendrite growing at (a) $\Delta = -0.4$ and (b) $\Delta = -0.6$. Here, a 512×512 meshgrid is used on the computational domain $(0, 500) \times (0, 500)$ with $\Delta t = h$, $\epsilon_4 = 0.05$, $D = 2$, and $\lambda = 1.5957D$. Each contour is drawn per every 800 iterations.

Next, we consider that the underlying temperatures have a gradient; it would give a result of the dendrite growth with a forced flows. Other conditions are set same as previous simulations except the temperature conditions. Let Δ_{top} and Δ_{bottom} be the boundary conditions of temperature

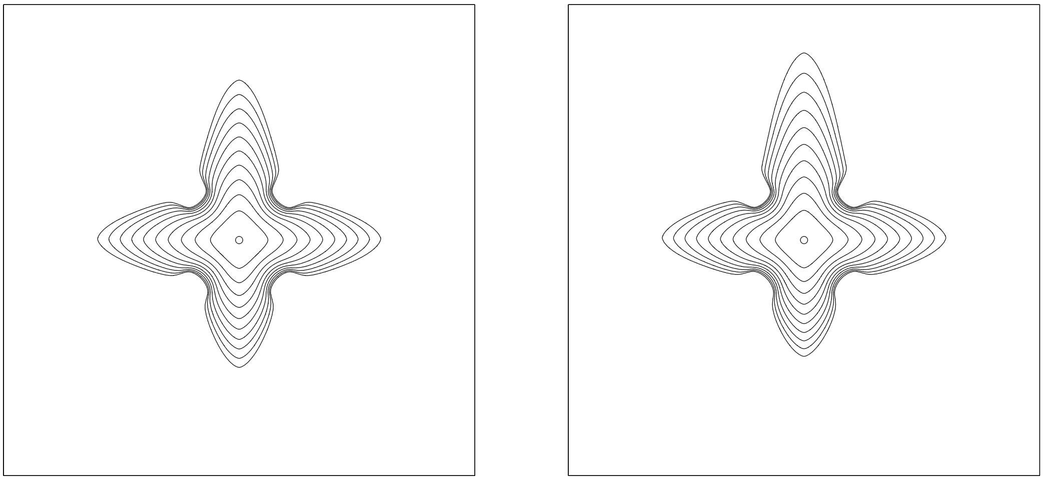


(a) $\Delta = -0.4$

(b) $\Delta = -0.6$

FIGURE 3.1. Evolution of phase-field contours for a dendrite growing at (a) $\Delta = 0.45$ and (b) $\Delta = 0.55$.

at the top and bottom of the domain, respectively. Here, $\Delta_{\text{top}} = 1.1\Delta$, $\Delta_{\text{bottom}} = 0.9\Delta$ and $\Delta_{\text{top}} = 1.2\Delta$, $\Delta_{\text{bottom}} = 0.8\Delta$; i.e., we assume that the approximated forced flow flows from the tom to the bottom. Note that the result in Fig. 3.2 is similar with a dendrite growing presenting in [127], which consider the convective effects for a fixed crystal on a given domain.



(a) (b)

FIGURE 3.2. Evolution of phase-field contours for a dendrite growing with the underlying temperatures who have a gradient (a) $\Delta_{\text{top}} = 1.1\Delta$, $\Delta_{\text{bottom}} = 0.9\Delta$ and (b) $\Delta_{\text{top}} = 1.2\Delta$, $\Delta_{\text{bottom}} = 0.8\Delta$.

Chapter 4

Navier–Stokes equation

4.1. Introduction

The NS equations, named after Claude-Louis Navier and George Gabriel Stokes, describe the motion of viscous fluid substances. Including water and air, the dynamics of almost every viscous liquid and gas are governed by this partial differential equation. The equations arise from the Newton's second law to fluid motion with the assumption: the stress in a fluid is the sum of a diffusing viscous term and a pressure term. Understandably, the solution of the NS equation is a flow velocity field.

Since a fluid does not have a fixed morphology like a solid, but flexible one, the classical dynamics properties cannot be applied directly. Therefore, we need to reorganize the Newtonian mechanics to the suitable form for a fluid; and the result is the NS equation [37]. Nevertheless, the NS equation is one of the most difficult equations to solve analytically among the known partial differential equation. In some specific cases, the exact solutions were derived [15, 38, 131]; however, the general solution is still remained in mysterious one.

Therefore, the numerical solution, which is the approximated solution of the exact NS equation, is the only way to analyze the dynamics of fluid flows and is clearly an important issue in many industrial, physical, and mathematical fields. There have been numerous researches for solving the NS equation numerically [6, 7, 31, 53, 80, 125], and the Chorin's projection method is considered as one of the most useful numerical solver because of its simple implementation [31].

In this dissertation, we only focus on the incompressible Newtonian fluid flow case, which is the appropriate form for the solute of metals and alloys; the dendrite crystal growth.

4.2. Governing equations

There are the basic assumptions of the NS equations: (i) a fluid is a continuum; i.e., a fluid is not made up of discrete particles but rather a continuous substance at the scale of interest and (ii) all fields of interest are (at least weakly) differentiable such as pressure, flow velocity, density, and temperature. The derivation of the equation starts from the conservation of mass, momentum, and energy [126].

The Newton's second law states that the net force on an object is equal to the rate of change of its linear momentum $\mathbf{p}$:

$$\begin{aligned}\mathbf{F} &= \frac{d\mathbf{p}}{dt}\\ &= \frac{d(m\mathbf{v})}{dt},\end{aligned} \tag{4.1}$$

where $\mathbf{F}$ is the net force, m is the mass of a body, and $\mathbf{v}$ is the body's velocity. Moreover, m can be taken outside the differentiation operator in Eq. (4.1) by the constant factor rule in differentiation since the Newton's second law is only valid for a constant mass body:

$$\begin{aligned}\mathbf{F} &= m\frac{d\mathbf{v}}{dt}\\ &= m\mathbf{a},\end{aligned} \tag{4.2}$$

where $\mathbf{a}$ is the body's acceleration. Thus, the net force applied to a body produces a proportional acceleration. Equation (4.2) is rewritten as the following form:

$$(\rho dxdydz)\frac{d\mathbf{u}}{dt} = \mathbf{F}.$$

On the other hand, let $\mathbf{J}_m(\mathbf{x}, t)$ be a mass flux where $\mathbf{x}$ is defined on a domain Ω. and the changes of $\mathbf{J}_m$ be equal to a lost or gained mass through the boundaries $\partial\Omega$ plus a created or dissipated mass by sources or sinks inside Ω. Then, its continuity equation is given as follow:

$$\frac{d}{dt}\int_{\Omega} \mathbf{J}_m dV = -\int_{\partial\Omega} (\mathbf{J}_m\mathbf{u}) \cdot \mathbf{n} dA + \int_{\Omega} \mathbf{s} dV, \tag{4.3}$$

where $\mathbf{u}$ is the flow velocity, $\mathbf{n}$ is the outgoing normal vector proportional to $\partial\Omega$, and $\mathbf{s}$ is the sources and sinks. Note that $\mathbf{J}_m\mathbf{u}$ is a dyad, a special case of tensor product, which results in a second tensor and the divergence of a second rank tensor is again a vector [92]. Using divergence theorem, Eq. (4.3) can be rewritten as follow:

$$
\begin{aligned}
&\frac{d}{dt}\int_{\Omega}\mathbf{J}_m dV = -\int_{\Omega}\nabla\cdot(\mathbf{J}_m\mathbf{u})dV + \int_{\Omega}\mathbf{s}dV,\\
\Leftrightarrow &\int_{\Omega}\frac{\partial\mathbf{J}_m}{\partial t}dV = -\int_{\Omega}\nabla\cdot(\mathbf{J}_m\mathbf{u})dV + \int_{\Omega}\mathbf{s}dV,\\
\Leftrightarrow &\int_{\Omega}\left(\frac{\partial\mathbf{J}_m}{\partial t} + \nabla\cdot(\mathbf{J}_m\mathbf{u}) - \mathbf{s}\right)dV = 0,
\end{aligned}
\tag{4.4}
$$

The integral is zero only if the integrand itself is zero, hence Eq. (4.4) becomes as follow:

$$
\frac{\partial\mathbf{J}_m}{\partial t} + \nabla\cdot(\mathbf{J}_m\mathbf{u}) - \mathbf{s} = 0. \tag{4.5}
$$

Now, we replace the mass flux $\mathbf{J}_m$ with the product of mass density ρ and flow velocity $\mathbf{u}$ (i.e., $\mathbf{J}_m = \rho\mathbf{u}$) in Eq. (4.5):

$$
\begin{aligned}
&\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot(\rho\mathbf{u}\mathbf{u}) - \mathbf{s} = 0,\\
\Leftrightarrow\ &\rho\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\frac{\partial\rho}{\partial t} + \nabla\rho\cdot\mathbf{u}\mathbf{u} + \rho\nabla\mathbf{u}\cdot\mathbf{u} + \rho\mathbf{u}\left(\nabla\cdot\mathbf{u}\right) = \mathbf{s},\\
\Leftrightarrow\ &\mathbf{u}\left(\frac{\partial\rho}{\partial t} + \nabla\rho\cdot\mathbf{u} + \rho\left(\nabla\cdot\mathbf{u}\right)\right) + \rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \mathbf{s},\\
\Leftrightarrow\ &\mathbf{u}\left(\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u})\right) + \rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \mathbf{s}.
\end{aligned}
\tag{4.6}
$$

Here, we consider the mass continuity equation:

$$
\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0. \tag{4.7}
$$

On the other hand, in the case of an incompressible fluid, the density is constant up to temporal evolution, i.e., $d\rho/dt = 0$. It implies that Eq. (4.7) can be rewritten as:

$$\nabla \cdot \mathbf{u} = 0. \tag{4.8}$$

Returning to the derivation from the continuity equation (4.3), we plug Eq. (4.7) into Eq. (4.6) to get

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \mathbf{s}. \tag{4.9}$$

Next, the momentum source $\mathbf{s}$, consisting two parts: surface forces and body forces, can be written as follow:

$$\mathbf{s} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f}, \tag{4.10}$$

where p is the pressure, $\boldsymbol{\tau}$ is the deviatoric stress tensor, $\mathbf{f}$ is the body forces such as gravity using Cauchy momentum equation [83]. For Newtonian fluids, the term of the stress tensor is derived as

$$\nabla \cdot \boldsymbol{\tau} = \nabla \cdot \left(\eta \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right) + \nabla \left(-\frac{2\eta}{3} \nabla \cdot \mathbf{u} \right) + \mathbf{f}, \tag{4.11}$$

where η is the viscosity of fluids with assumptions by Stokes. In conclusion, the NS equations for incompressible Newtonian fluids has the following forms using Eqs. (4.8)–(4.11):

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)=-\nabla p+\nabla\cdot\left(\eta\left(\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right)\right)+\mathbf{f}, \tag{4.12}$$

$$\nabla\cdot\mathbf{u}=0. \tag{4.13}$$

Moreover, if the density and the viscosity is constant and the external force consists of only surface tensional and gravitational forces, we can rewrite Eq. (4.12) as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)=-\nabla p+\eta\Delta\mathbf{u}+\mathbf{f}, \tag{4.14}$$

and we treat this form in this chapter for simplicity.

4.2.1. Non-dimensionalization. The dimensional analysis is an analysis by converting dimensional variables, such as length, mass, time and other quantities, to dimensionless variables. This is a widely used technique for performing comparisons in mathematics, physics, and engineering. Since it is too much complicated if all variables have each physical dimension, the non-dimensionalization is quite important and useful in both theory and experience.

To non-dimensionalize the Eqs. (4.13) and (4.14), we consider the following dimensionless variables:

$$\mathbf{u}^* =\frac{\mathbf{u}}{U^*},$$

$$\mathbf{x}^* = \frac{\mathbf{x}}{L^*},$$

$$t^* = \frac{t}{L^*/U^*},$$

$$p^* = \frac{p}{\rho(U^*)^2},$$

where U^* and L^* are the characteristic velocity and length, respectively. Applying the change of the variables to Eqs. (4.13) and (4.14),

$$\rho\left(\frac{\partial(U^*\mathbf{u}^*)}{\partial\left((L^*/U^*)t^*\right)} + (U^*\mathbf{u}^*)\cdot\nabla(U^*\mathbf{u}^*)\right) = -\nabla\left(\left(\rho(U^*)^2\right)p^*\right) + \eta\Delta(U^*\mathbf{u}^*) + \mathbf{f},$$

$$\nabla\cdot\mathbf{u}^* = 0,$$

or,

$$\frac{\partial\mathbf{u}^*}{\partial t^*} + \mathbf{u}^*\cdot\nabla\mathbf{u}^* = -\nabla p^* + \frac{1}{Re}\Delta\mathbf{u}^* + \frac{1}{We}\mathbf{f}^*,$$

$$\nabla\cdot\mathbf{u}^* = 0,$$

where $Re = \rho U^* L^*/\eta$ is the Reynolds number, $We = \rho(U^*)^2 L^*/\sigma$ is the Weber number, and σ is a coefficient.

For simplicity, we rewrite the non-dimensional equations omitting the * sign as

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \frac{1}{Re}\Delta\mathbf{u} + \frac{1}{We}\mathbf{f}, \tag{4.15}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{4.16}$$

4.3. Discretization

In this section, we describe the discretization of the governing equations (4.15) and (4.16). To simplify the exposition, we consider the equations on the unit square computational domain $\Omega = (0,1) \times (0,1)$. Let h be the uniform mesh grid size, then the center of the cell is

$$\mathbf{x}_{ij} = (x_i, y_j) = ((i - 0.5)h, (j - 0.5)h)$$

for $i, j = 1, \cdots, N$. Here, N denotes the number of grid points in each direction. Let Δt be the time step size. Recall that the discrete gradient and laplacian operators for cell-edged values are defined in Section 2 as follows:

$$\nabla_h^x u_{i+\frac{1}{2},j} = \frac{u_{i+1,j} - u_{ij}}{h},$$

$$\nabla_h^y v_{i,j+\frac{1}{2}} = \frac{v_{i,j+1} - v_{ij}}{h}.$$

Moreover, we define another discrete gradient operators as

$$\nabla_h^{xy} \mathbf{u} = (\nabla_h^x u, \nabla_h^y v),$$

$$\nabla_d p = (\nabla_h^x p, \nabla_h^y p).$$

To discretize the equations, a staggered marker-and-cell (MAC) mesh [55] is used; i.e., the pressure $p(\mathbf{x}, t)$ is stored at the cell-centers and the velocity components $u(\mathbf{x}, t)$ and $v(\mathbf{x}, t)$ are stored at the cell-edges. In a discretized equation, we denote $p(\mathbf{x}_{ij}, n\Delta t)$ as p_{ij}^n. By discretizing the Eqs. (4.15) and (4.16), we can rewrite them as follows:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla_h^{xy} \mathbf{u}^n = -\nabla_d p^{n+1} + \frac{1}{Re}\Delta_h \mathbf{u}^n + \frac{1}{We}\mathbf{f}^n, \tag{4.17}$$

$$\nabla_d \cdot \mathbf{u}^{n+1} = 0. \tag{4.18}$$

Therefore, in two-dimensional Cartesian coordinates, we have

$$\frac{u_{i+\frac{1}{2},j}^{n+1} - u_{i+\frac{1}{2},j}^{n}}{\Delta t} + (\mathbf{u}\cdot\nabla_d u)_{i+\frac{1}{2},j}^n = -\frac{p_{i+1,j}^{n+1} - p_{ij}^{n+1}}{h} + \frac{1}{Re}\Delta_h u_{i+\frac{1}{2},j}^n + \frac{1}{We} f_{1\ i+\frac{1}{2},j}^n, \tag{4.19}$$

$$\frac{v_{i,j+\frac{1}{2}}^{n+1} - v_{i,j+\frac{1}{2}}^{n}}{\Delta t} + (\mathbf{u}\cdot\nabla_d v)_{i,j+\frac{1}{2}}^n = -\frac{p_{i,j+1}^{n+1} - p_{ij}^{n+1}}{h} + \frac{1}{Re}\Delta_h v_{i,j+\frac{1}{2}}^n + \frac{1}{We} f_{2\ i,j+\frac{1}{2}}^n, \tag{4.20}$$

$$(\nabla_d \cdot \mathbf{u})_{ij}^{n+1} = 0. \tag{4.21}$$

Here, we denote

$$f_{m\ i+\frac{1}{2},j}^n = 0.5(f_{m\ i+1,j}^n + f_{m\ ij}^n)$$

for $m = 1, 2, 3$.

We use the projection method to find $\mathbf{u}^{n+1}$ from given $\mathbf{u}^n$ by solving Eqs. (4.17) and (4.18). At first, we compute an intermediate velocity field, $\tilde{\mathbf{u}}$, which generally does not satisfy the incompressible condition. To apply the no-slip boundary condition, we set the values of ghost points as

$$u^n_{-\frac{1}{2},j} = 0 \text{ and } v^n_{0,j} = -v^n_{1,j},$$

in the x-direction. The reason that only u^n has a different type of formula is that only u^n is saved at a cell-edge in the x-direction on the MAC mesh. The values at ghost points in the y-direction are similarly defined. From Eq. (4.19) without the pressure gradient term, we have

$$\frac{\tilde{u}^{n+1}_{i+\frac{1}{2},j} - u^n_{i+\frac{1}{2},j}}{\Delta t} + (\mathbf{u} \cdot \nabla_d u)^n_{i+\frac{1}{2},j} = \frac{1}{Re} \Delta_h u^n_{i+\frac{1}{2},j} + \frac{1}{We} f^n_{1\ i+\frac{1}{2},j}.$$

Then, we define

$$\begin{aligned} \tilde{u}^{n+1}_{i+\frac{1}{2},j} =& u^n_{i+\frac{1}{2},j} - \Delta t (\mathbf{u} \cdot \nabla_d u)^n_{i+\frac{1}{2},j} + \frac{\Delta t}{We} f^n_{1\ i+\frac{1}{2},j} + \frac{\Delta t}{h^2 Re} \Big(u^n_{i+\frac{3}{2},j} + u^n_{i-\frac{1}{2},j} \\ & - 4u^n_{i+\frac{1}{2},j} + u^n_{i+\frac{1}{2},j+1} + u^n_{i+\frac{1}{2},j-1} \Big), \end{aligned}$$

where the advection term is defined as

$$(\mathbf{u} \cdot \nabla_d u)^n_{i+\frac{1}{2},j} = u^n_{i+\frac{1}{2},j} \bar{u}^n_{x_{i+\frac{1}{2},j}} + \frac{v^n_{i,j-\frac{1}{2}} + v^n_{i+1,j-\frac{1}{2}} + v^n_{i,j+\frac{1}{2}} + v^n_{i+1,j+\frac{1}{2}}}{4} \bar{u}^n_{y_{i+\frac{1}{2},j}}.$$

The values $\bar{u}^n_{x_{i+\frac{1}{2},jk}}$ are computed using the following upwind scheme:

$$\bar{u}^n_{x_{i+\frac{1}{2},j}} = \begin{cases} \dfrac{u^n_{i+\frac{1}{2},j} - u^n_{i-\frac{1}{2},j}}{h} & \text{if } u^n_{i+\frac{1}{2},j} > 0, \\ \dfrac{u^n_{i+\frac{3}{2},j} - u^n_{i+\frac{1}{2},j}}{h} & \text{otherwise.} \end{cases}$$

The other values $\tilde{v}_{i,j+\frac{1}{2}}$ are calculated similarly. Then, we solve the following equations for the advanced pressure field at the $(n+1)$-th time step.

$$\frac{\mathbf{u}^{n+1}-\tilde{\mathbf{u}}}{\Delta t} = -\nabla_d p^{n+1} \tag{4.22}$$

$$\nabla_d \cdot \mathbf{u}^{n+1} = 0. \tag{4.23}$$

Applying the divergence operator to Eq. (4.22) and using Eq. (4.23), we have the Poisson equation for the pressure at the advanced time $(n+1)$.

$$\Delta_h p^{n+1} = \frac{1}{\Delta t}\nabla_d \cdot \tilde{\mathbf{u}}, \tag{4.24}$$

where

$$\begin{aligned}
\Delta_h p_{ij}^{n+1} =& \frac{p_{i+1,j}^n + p_{i-1,j}^n - 4p_{ij}^n + p_{i,j+1}^n + p_{i,j-1}^n}{h^2}, \\
(\nabla_d \cdot \tilde{\mathbf{u}})_{ij} =& \frac{\tilde{u}_{i+\frac{1}{2},j} - \tilde{u}_{i-\frac{1}{2},j}}{h} + \frac{\tilde{v}_{i,j+\frac{1}{2}} - \tilde{v}_{i,j-\frac{1}{2}}}{h}.
\end{aligned}$$

Since the Poisson problem is ill-posed under the no-slip boundary condition, the adjustment step is needed for p^{n+1}. Here, we specify that p^{n+1} has zero mean.

The linear system of Eq. (4.24) is solved using a multigrid method [129], specifically, V-cycles using Gauss–Seidel relaxation with a tolerance of 10^{-7}. After solving the pressure field, we update

$u^{n+1}_{i+\frac{1}{2},jk}$ as

$$u^{n+1}_{i+\frac{1}{2},j} = \tilde{u}_{i+\frac{1}{2},j} - \frac{\Delta t}{h}\left(p^{n+1}_{i+1,j} - p^{n+1}_{ij}\right).$$

The variables $v^{n+1}_{i,j+\frac{1}{2}}$ are updated in a similar manner.

This completes one time step update. Refer [30] to see more details of numerical implementation.

4.4. Numerical experiments

4.4.1. Cavity flow. In this section, we consider a lid-driven cavity flow in a two-dimensional domain. Figure 4.1 shows a schematic of a computational domain and the boundary conditions in a driven cavity. The initial condition of the flow inside the domain is given as a zero value and the flow at boundaries at three walls except the top is fixed as a zero velocity. On the other hand, at the top, the lid-driven flow is given as $(u, v) = (1, 0)$. Here, there is no the external force effect, i.e., $\mathbf{f} \equiv \mathbf{0}$.

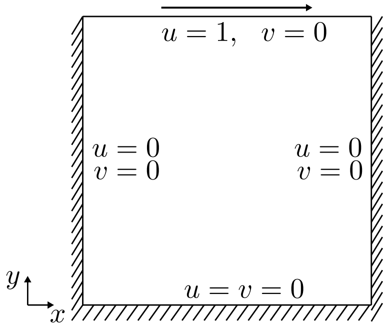
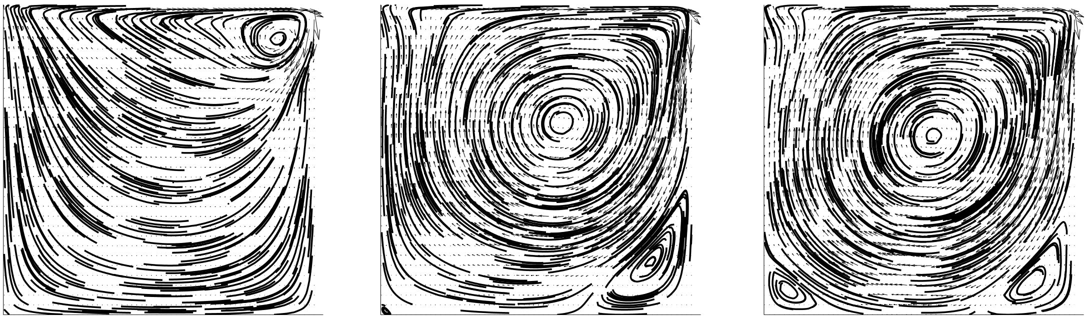


FIGURE 4.1. Schematic illustration of the lid-driven cavity flow.

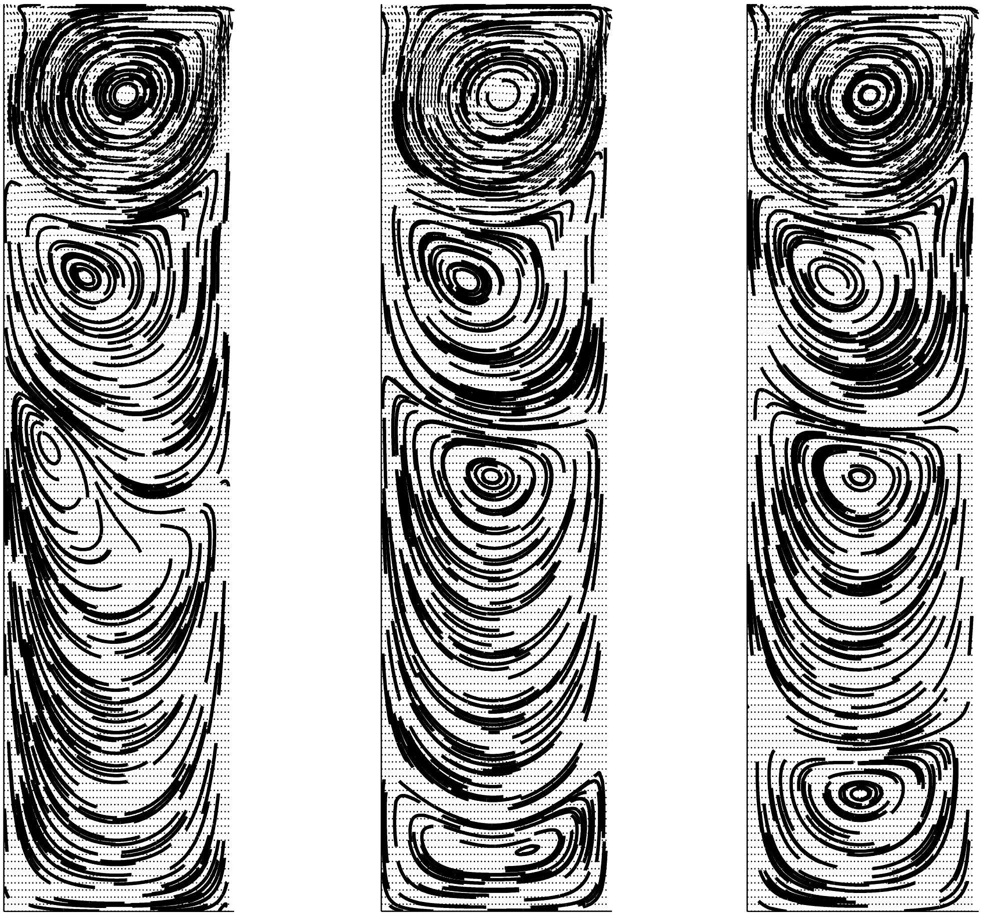
To show the effect of the domain size, we first consider a lid-driven cavity flow in $\Omega = (0,1) \times (0,1)$. The result is shown in Fig. 4.2 with 64×64 mesh grid, i.e., $h = 1/64$, $Re = 5000$, and $\Delta t = 0.01h^2 Re$. We can observe that the eye of principal vortex moves into the core of the cavity and the lower left/right corner-eddies as time evolves.



$t = 200\Delta t$ $t = 2000\Delta t$ $t = 100000\Delta t$

FIGURE 4.2. The evolution of cavity flow on the square domain $\Omega = (0,1) \times (0,1)$. The dimensionless times are shown below each figure.

Next, we perform a numerical simulation on a non-square domain $\Omega = (0,1) \times (0,4)$ with 64×256 mesh grid. We use $h = 1/64$, $Re = 5000$, and $\Delta t = 0.01h^2 Re$. Figure 4.3 shows the numerical results of the lid-driven cavity flow at each time on the rectangle domain. Unlike the results in Fig. 4.2, we observe that more than two vortices with an opposite directional rotation at the lower corner.



$t = 160000\Delta t$ $t = 400000\Delta t$ $t = 800000\Delta t$

FIGURE 4.3. The evolution of cavity flow on the rectangle domain $\Omega = (0,1) \times (0,4)$. The dimensionless times are shown below each figure.

Chapter 5

Convection of crystal growth under a flow

5.1. Introduction

As mentioned in Chapter 1, convection of the crystal in the melt is of great interest for the practical processes to understand the dendritic solidification.

In numerical investigation, the phase-field method, widely applied to model various meso-scale phenomena such as solidification, recrystallization and so on [26], is also a flexible mathematical tool to describe the interfaces in a dendritic crystal growth with convection. Two- and three-dimensional adaptive phase-field simulations of dendritic crystal growth ini a forced flow were presented in [23]. The effect of natural convection in 3D dendritic growth using an efficient adaptive phase-field simulation was investigated by Chen and Lan [24]. A two-dimensional lattice Boltzmann method-cellular automatic model was presented to investigate the dendritic growth of binary alloys in the presence of natural convection [143]. Recently, motion and growth of a dendritic ini the presence of melt convection was modeled using a phase-field-lattice-Boltzmann method [113].

However, the crystal is fixed in the space and cannot be convected, instead the supercooled melt flows around the crystal, which is hard to be realized in the real world experimental setting in the previous studies. Direct application of the advection term to the crystal equation would lead some problems such as the shape deformation and the ambiguity of the crystal orientation for the anisotropy. In recent years, few models have been elaborated for using two computational domains to distinguish between the fluid flow and the phase separation by applying a fictitious domain method [39], a combination of the volume of fluid and the immersed boundary methods [69]. Nevertheless, there are drawbacks in their algorithm; difficulty in matching the grids or computational efficiency.

The main purpose of the present dissertation is to resolve these difficulties by using a moving overset grid. The fluid domain is covered with a fixed Cartesian grid, while a moving overset grid is used to represent the crystal growth and convection. The motion of the crystal is derived by calculating the translational and rotational force of the crystal. Using the fictitious domain method with distributed Lagrange multiplier method, the method in [39] has an advantage that simulations can be performed in a problem involving different scales in time and space. Note that the proposed method would be compared with the method in [39] since our moving overset grid method is similar to their method based on a fictitious domain method. However, our method is simpler to implement and it also has the advantage involving different scales.

5.2. Governing equations

We consider the solidification of a pure substance from its supercooled melt in a two-dimensional flow. This problem has a much more difficult numerical challenge than the previous ones. To model the solidification system, let $\phi(x, y, t)$ be the phase-field function, where $\phi = 1$ and $\phi = -1$ refer to the bulk solid and melt phases, respectively. The phase-field variable ϕ is smoothly changed but has small thickness, and we define the interface by the zero level set of ϕ [127]. The governing equations for crystal growth in the flow are given as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = - \nabla p + \frac{1}{Re} \nabla \cdot [\eta(\phi)(\nabla \mathbf{u} + \nabla \mathbf{u}^T)], \tag{5.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{5.2}$$

$$\begin{aligned} \epsilon^2(\phi) \left(\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi \right) =& \nabla \cdot (\epsilon^2(\phi) \nabla \phi) + [\phi - \lambda U(1 - \phi^2)](1 - \phi^2) \\ &+ \left(|\nabla \phi|^2 \epsilon(\phi) \frac{\partial \epsilon(\phi)}{\partial \phi_x} \right)_x + \left(|\nabla \phi|^2 \epsilon(\phi) \frac{\partial \epsilon(\phi)}{\partial \phi_y} \right)_y, \end{aligned} \tag{5.3}$$

$$\frac{\partial U}{\partial t} + \mathbf{u} \cdot \nabla U = D \Delta U + \frac{1}{2} \frac{\partial \phi}{\partial t}, \tag{5.4}$$

where $\mathbf{u}$ is the velocity, p is the pressure, $\eta(\phi)$ is the variable viscosity, and U is the temperature. Note that Eqs. (5.1) and (5.2) are the Navier–Stokes equations [30] and Eqs. (5.3) and (5.4) are the governing equations for dendrite growth [71]. Here, $\eta(\phi) = 0.5[\eta_s(1 + \phi) + \eta_m(1 - \phi)]$, where η_s and η_m are viscosities of solid and melt, respectively. We take the approach of the increased viscosity [33] which uses a very large viscosity in the solid to describe the resistance to the flow.

The dimensionless parameters are the Reynolds number Re, λ, and D. For the four-fold symmetry, $\epsilon(\phi)$ is defined as:

$$\epsilon(\phi) = (1 - 3\epsilon_4)\left(1 + \frac{4\epsilon_4}{1 - 3\epsilon_4}\frac{\phi_x^4 + \phi_y^4}{|\nabla\phi|^4}\right),$$

where ϵ_4 is a parameter for the anisotropy of interfacial energy.

We will consider the two computational domains to separately represent the crystal growth and fluid flow. The flow with appropriate boundary conditions is defined on the base domain. The phase-field function ϕ for representing the crystal is defined on the relatively small domain, and the crystal growth equation is

$$\begin{aligned}\epsilon^2(\phi)\frac{\partial\phi}{\partial t} =& \nabla\cdot(\epsilon^2(\phi)\nabla\phi) + [\phi - \lambda U(1-\phi^2)](1-\phi^2)\\ &+ \left(|\nabla\phi|^2\epsilon(\phi)\frac{\partial\epsilon(\phi)}{\partial\phi_x}\right)_x + \left(|\nabla\phi|^2\epsilon(\phi)\frac{\partial\epsilon(\phi)}{\partial\phi_y}\right)_y \end{aligned} \tag{5.5}$$

by using Eq. (5.3) without the advection term $\mathbf{u}\cdot\nabla\phi$.

5.2.1. Motion of a rigid body. For translating and rotating of the crystal domain, we use the conservation law of the linear and angular momentums. Since the crystal is governed by a rigid body motion, the conservation of the linear momentum is given as:

$$M_c\mathbf{u}_c = \int_\Omega \mathbf{u}(x,y)\rho(x,y)d\mathbf{x}$$

where M_c is the total mass of a crystal, $M_c = \int_\Omega \rho(x,y)\, d\mathbf{x}$, Ω is a crystal, and $\mathbf{u}_c$ is the velocity of the crystal. With regarding to the constant density field ρ, we can rewrite as

$$\mathbf{u_c} = \frac{\int_\Omega \mathbf{u}(x,y) d\mathbf{x}}{\int_\Omega d\mathbf{x}}. \tag{5.6}$$

Next, the angular momentum $\mathbf{L}_c$ of the crystal is generally defined as a sum of the infinitesimal angular momentum $d\mathbf{L}$ [3]:

$$\begin{aligned} \mathbf{L}_c &= \int_\Omega d\mathbf{L} \\ &= \int_\Omega (\tilde{\mathbf{r}}(x,y) \times \mathbf{u}(x,y))\, \rho(x,y) d\mathbf{x}, \end{aligned}$$

where $\tilde{\mathbf{r}}(x,y)$ is the displacement vector from the center of mass. Moreover, $\mathbf{L}_c$ can be written by a product of the moment of inertia I_c and the angular speed ω_c, i.e., $\mathbf{L}_c = I_c \omega_c$. With $\rho = 1$, ω_c can be written as

$$\omega_c = \frac{\mathbf{L}_c}{I_c} = \frac{\int_\Omega (\tilde{\mathbf{r}} \times \mathbf{u})\, d\mathbf{x}}{\int_\Omega \|\tilde{\mathbf{r}}\|^2 d\mathbf{x}}. \tag{5.7}$$

Therefore, we can translate and rotate a crystal without solving the advection term since the velocity vector and the angular speed of a crystal (or an overset grid) can be derived from Eqs. (5.6) and (5.7).

5.3. Numerical solutions

In this section, we propose a hybrid numerical method using a overset grid for the simulation of the crystal growth in a cavity flow. Let $\Omega_f = (a,b) \times (c,d)$ be a domain for the fluid velocity $\mathbf{u} = (u,v)$, pressure p, and temperature U with proper boundary conditions. Also, let $\Omega_c = (0,\alpha) \times (0,\beta)$ be another domain for the phase-field function ϕ with interpolated boundary conditions. Ω_{moving} is the coordinate transformation of Ω_c, and it represents the location and rotation of Ω_c on Ω_f. Let $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, $\mathbf{X}_4$ be corners of Ω_{moving} on Ω_f corresponding to points $(0,0)$, $(\alpha,0)$, (α,β), $(0,\beta)$ on Ω_c, respectively. We determine the location of Ω_{moving} by setting its center as $\mathbf{m}_c = (\mathbf{X}_1+\mathbf{X}_2+\mathbf{X}_3+\mathbf{X}_4)/4$ and the rotation θ_c as the signed angle measured from the horizontal axis to the vector $\overrightarrow{\mathbf{X}_1\mathbf{X}_2}$ (see Fig. 5.1).

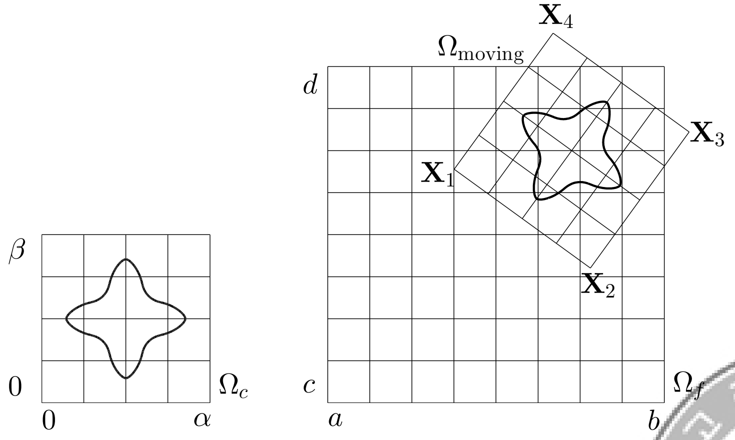


FIGURE 5.1. Schematic illustration of the fluid domain Ω_f, the crystal domain Ω_c, and the moving domain Ω_{moving}.

For the time integration, we consider the uniform time step $\Delta t = T/N_t$, where T is the final time and N_t is the total number of iterations. For Ω_f, we use a uniform mesh with mesh spacing $h = (b-a)/N_x = (d-c)/N_y$, where N_x and N_y are the numbers of cells in the x- and y-directions, respectively. The center of each cell is located at $\mathbf{x}_{ij} = (x_i, y_j) = (a + (i - 0.5)h, c + (j - 0.5)h)$ and we define the computational domain $\Omega_f^h = \{\mathbf{x}_{ij} \mid i = 1, \cdots, N_x, j = 1, \cdots, N_y\}$. Using the marker-and-cell mesh, the pressure p and temperature U are defined at the cell centers and the velocities u and v are defined at the cell edges (see Fig. 5.2). Let p_{ij}^n and U_{ij}^n be approximations of $p(x_i, y_j, n\Delta t)$ and $U(x_i, y_j, n\Delta t)$, respectively. Let $u_{i+1/2,j}^n$ and $v_{i,j+1/2}^n$ be approximations of $u(x_i + h/2, y_j, n\Delta t)$ and $v(x_i, y_j + h/2, n\Delta t)$, respectively.

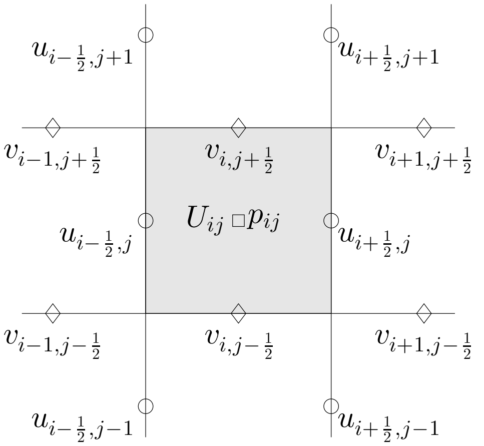


FIGURE 5.2. Schematic of the computational grid for the pressure, velocities, and temperature.

For Ω_c, we also use a uniform mesh with mesh spacing $\bar{h} = \alpha/M_x = \beta/M_y$, where M_x and M_y are the numbers of cells in the x- and y-directions, respectively. The center of each cell is located at $\mathbf{s}_{kl} = (s_k^x, s_l^y) = ((k-0.5)\bar{h}, (l-0.5)\bar{h})$, and we define the computational domain $\Omega_c^h = \{\mathbf{s}_{kl} \mid i = 1, \cdots, N_x, j = 1, \cdots, N_y\}$. And, $\mathbf{m}_c^n$ and θ_c^n represent the center position and rotation of Ω_{moving} at time $n\Delta t$, respectively.

Next, we describe our proposed numerical solution algorithm. At the n-th time step, we have a divergence-free velocity field $\mathbf{u}^n$, the phase-field ϕ^n, and temperature U^n. We seek $\mathbf{u}^{n+1}$, p^{n+1}, ϕ^{n+1}, and U^{n+1}.

5.3.1. Hydrodynamic flow on Ω_f. First, we solve Eqs. (5.1) and (5.2) to update $\mathbf{u}^{n+1}$ and p^{n+1} on the fluid domain Ω_f by using the Chorin's projection method [31, 98]:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla_h^{xy} \mathbf{u}^n = - \nabla_d p^{n+1} + \frac{1}{Re} \nabla_d \cdot \left[\eta^n \left(\nabla_d \mathbf{u}^n + (\nabla_d \mathbf{u}^n)^T \right) \right], \tag{5.8}$$

$$\nabla_d \cdot \mathbf{u}^{n+1} = 0. \tag{5.9}$$

We solve an intermediate velocity field, $\tilde{\mathbf{u}}^n = (\tilde{u}^n, \tilde{v}^n)$:

$$\tilde{u}_{i+\frac{1}{2},j}^n = u_{i+\frac{1}{2},j}^n - \Delta t (u u_x + v u_y)_{i+\frac{1}{2},j}^n + \frac{\Delta t}{Re} \left(2(\eta u_x)_x + (\eta u_y)_y + (\eta v_x)_y \right)_{i+\frac{1}{2},j}^n ,$$

$$\tilde{v}_{i,j+\frac{1}{2}}^n = v_{i,j+\frac{1}{2}}^n - \Delta t (u v_x + v v_y)_{i,j+\frac{1}{2}}^n + \frac{\Delta t}{Re} \left((\eta v_x)_x + (\eta u_y)_x + 2(\eta v_y)_y \right)_{i,j+\frac{1}{2}}^n .$$

The advection terms are defined by

$$(u u_x + v u_y)_{i+\frac{1}{2},j}^n = u_{i+\frac{1}{2},j}^n \bar{u}_{x_{i+\frac{1}{2},j}}^n + v_{i+\frac{1}{2},j}^n \bar{u}_{y_{i+\frac{1}{2},j}}^n ,$$

$$(uv_x + vv_y)^n_{i,j+\frac{1}{2}} = v^n_{i,j+\frac{1}{2}} \bar{v}^n_{y_{i,j+\frac{1}{2}}} + u^n_{i,j+\frac{1}{2}} \bar{v}^n_{x_{i,j+\frac{1}{2}}},$$

where the values $\bar{u}^n_{x_{i+\frac{1}{2},j}}$ and $\bar{u}^n_{y_{i+\frac{1}{2},j}}$ are computed using the upwind procedure

$$\bar{u}^n_{x_{i+\frac{1}{2},j}} = \begin{cases} \frac{u^n_{i+\frac{1}{2},j} - u^n_{i-\frac{1}{2},j}}{h}, & \text{if } u^n_{i+\frac{1}{2},j} > 0, \\ \frac{u^n_{i+\frac{3}{2},j} - u^n_{i+\frac{1}{2},j}}{h}, & \text{otherwise,} \end{cases} \tag{5.10}$$

$$\bar{u}^n_{y_{i+\frac{1}{2},j}} = \begin{cases} \frac{u^n_{i+\frac{1}{2},j} - u^n_{i+\frac{1}{2},j-1}}{h}, & \text{if } v^n_{i+\frac{1}{2},j} > 0, \\ \frac{u^n_{i+\frac{1}{2},j+1} - u^n_{i+\frac{1}{2},j}}{h}, & \text{otherwise.} \end{cases} \tag{5.11}$$

The quantities $\bar{v}^n_{x_{i,j+\frac{1}{2}}}$ and $\bar{v}^n_{y_{i,j+\frac{1}{2}}}$ are similarly computed. The viscosity terms are defined by

$$\begin{aligned}
&(2(\eta u_x)_x + (\eta u_y)_y + (\eta v_x)_y)^n_{i+\frac{1}{2},j} \\
&\quad = 2\left(\eta^n_{i+1,j} \frac{u^n_{i+\frac{3}{2},j} - u^n_{i+\frac{1}{2},j}}{h^2} - \eta^n_{ij} \frac{u^n_{i+\frac{1}{2},j} - u^n_{i-\frac{1}{2},j}}{h^2}\right) \\
&\quad + \eta^n_{i+\frac{1}{2},j+\frac{1}{2}} \frac{u^n_{i+\frac{1}{2},j+1} - u^n_{i+\frac{1}{2},j}}{h^2} - \eta^n_{i+\frac{1}{2},j-\frac{1}{2}} \frac{u^n_{i+\frac{1}{2},j} - u^n_{i+\frac{1}{2},j-1}}{h^2} \\
&\quad + \eta^n_{i+\frac{1}{2},j+\frac{1}{2}} \frac{v^n_{i+1,j+\frac{1}{2}} - v^n_{i,j+\frac{1}{2}}}{h^2} - \eta^n_{i+\frac{1}{2},j-\frac{1}{2}} \frac{v^n_{i,j-\frac{1}{2}} - v^n_{i-1,j-\frac{1}{2}}}{h^2},
\end{aligned}$$

$$\begin{aligned}
&((\eta v_x)_x + (\eta u_y)_x + 2(\eta v_y)_y)^n_{i,j+\frac{1}{2}} \\
&\quad = \eta^n_{i+\frac{1}{2},j+\frac{1}{2}} \frac{v^n_{i+1,j+\frac{1}{2}} - v^n_{i,j+\frac{1}{2}}}{h^2} - \eta^n_{i-\frac{1}{2},j+\frac{1}{2}} \frac{v^n_{i,j+\frac{1}{2}} - v^n_{i-1,j+\frac{1}{2}}}{h^2} \\
&\quad + \eta^n_{i+\frac{1}{2},j+\frac{1}{2}} \frac{u^n_{i+\frac{1}{2},j+1} - u^n_{i+\frac{1}{2},j}}{h^2} - \eta^n_{i-\frac{1}{2},j+\frac{1}{2}} \frac{u^n_{i+\frac{1}{2},j} - u^n_{i+\frac{1}{2},j-1}}{h^2} \\
&\quad + 2\left(\eta^n_{i,j+1} \frac{v^n_{i,j+\frac{3}{2}} - v^n_{i,j+\frac{1}{2}}}{h^2} - \eta^n_{ij} \frac{v^n_{i,j+\frac{1}{2}} - v^n_{i,j-\frac{1}{2}}}{h^2}\right).
\end{aligned}$$

Because the viscosity η^n on Ω_f depends on the phase-field function ϕ^n on Ω_c, we consider the interpolated function $\tilde{\phi}^n$ as follows: First, to check whether $\mathbf{x}_{ij} \in \Omega_f^h$ is inside of Ω_{moving} or not, we compare the sum of areas of four triangles with the area of moving domain Ω_{moving}. That is, if

$$\triangle \mathbf{x}_{ij}\mathbf{X}_1\mathbf{X}_2 + \triangle \mathbf{x}_{ij}\mathbf{X}_2\mathbf{X}_3 + \triangle \mathbf{x}_{ij}\mathbf{X}_3\mathbf{X}_4 + \triangle \mathbf{x}_{ij}\mathbf{X}_4\mathbf{X}_1 > \square \mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4,$$

then it means that $\mathbf{x}_{ij} \notin \Omega_{\text{moving}}$ (Fig. 5.3(a)). If the two areas are same, then $\mathbf{x}_{ij} \in \Omega_{\text{moving}}$ (Fig. 5.3(b)). Next, we estimate the value of $\tilde{\phi}_{ij}^n$ from ϕ^n on Ω_c if $\mathbf{x}_{ij} \in \Omega_{\text{moving}}$ by using the bilinear interpolation and we define $\tilde{\phi}_{ij}^n = 1$ otherwise. For more details, we denote the directional vectors as $\mathbf{a} = \overrightarrow{\mathbf{X}_1\mathbf{X}_2}$, $\mathbf{b} = \overrightarrow{\mathbf{X}_1\mathbf{X}_4}$, and $\mathbf{c} = \overrightarrow{\mathbf{X}_1\mathbf{x}_{ij}}$. If $\mathbf{x}_{ij} \in \Omega_{\text{moving}}$, $\mathbf{x}_{ij}$ is corresponding to the location $((\mathbf{a}\cdot\mathbf{c})/|\mathbf{a}|, (\mathbf{b}\cdot\mathbf{c})/|\mathbf{b}|)$ on Ω_c and then we can calculate the $\tilde{\phi}_{ij}$ by the interpolation. Finally, we define as

$$\eta_{ij}^n = \frac{\eta_s(1+\tilde{\phi}_{ij}^n) + \eta_m(1-\tilde{\phi}_{ij}^n)}{2}.$$

We then solve the pressure field at the $(n+1)$-th time step.

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}^n}{\Delta t} = -\nabla_d p^{n+1}, \tag{5.12}$$

$$\nabla_d \cdot \mathbf{u}^{n+1} = 0. \tag{5.13}$$

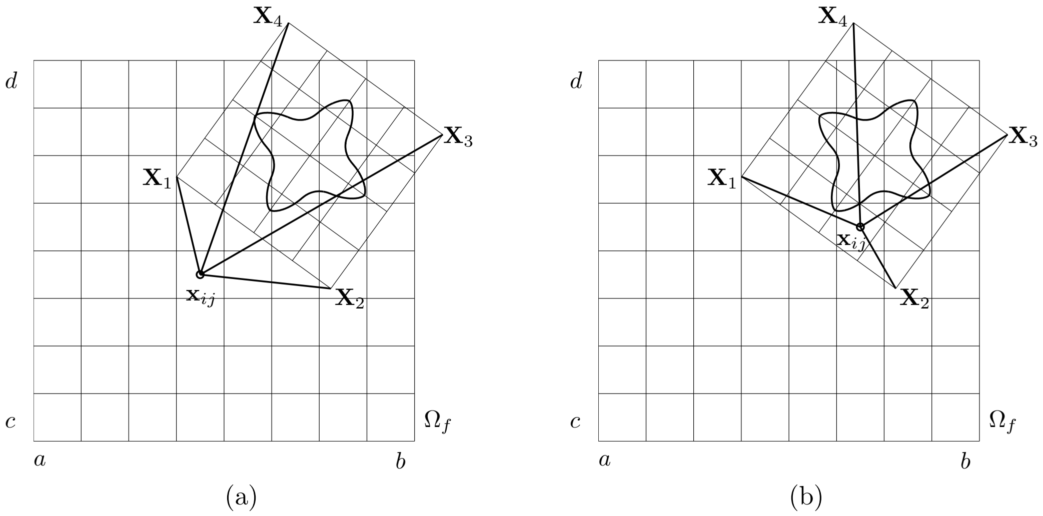


FIGURE 5.3. Schematic illustration of (a) $\mathbf{x}_{ij} \notin \Omega_{\text{moving}}$ and (b) $\mathbf{x}_{ij} \in \Omega_{\text{moving}}$.

Applying the discrete divergence $\nabla_d \cdot$ and divergence-free Eq. (5.13) to Eq. (5.12), we obtain the Poisson's equation with the homogeneous Neumann boundary condition:

$$\Delta_h p^{n+1} = \frac{1}{\Delta t} \nabla_d \cdot \tilde{\mathbf{u}}^n, \tag{5.14}$$

where

$$\Delta_h p_{ij}^{n+1} = \frac{p_{i+1,j}^{n+1} + p_{i-1,j}^{n+1} - 4p_{ij}^{n+1} + p_{i,j+1}^{n+1} + p_{i,j-1}^{n+1}}{h^2},$$

and

$$\nabla_d \cdot \tilde{\mathbf{u}}_{ij}^n = \frac{\tilde{u}_{i+\frac{1}{2},j}^n - \tilde{u}_{i-\frac{1}{2},j}^n + \tilde{v}_{i,j+\frac{1}{2}}^n - \tilde{v}_{i,j-\frac{1}{2}}^n}{h}.$$

We solve the Eq. (5.14) by the multigrid method, and using the updated pressure p^{n+1}, the divergence-free velocities are obtained

$$\begin{aligned} u^{n+1}_{i+\frac{1}{2},j} =& \tilde{u}^{n}_{i+\frac{1}{2},j} - \frac{\Delta t}{h}(p^{n+1}_{i+1,j} - p^{n+1}_{ij}), \\ v^{n+1}_{i,j+\frac{1}{2}} =& \tilde{v}^{n}_{i,j+\frac{1}{2}} - \frac{\Delta t}{h}(p^{n+1}_{i,j+1} - p^{n+1}_{ij}). \end{aligned}$$

5.3.2. Dendritic crystal growth on Ω_c. Next, we solve the crystal equation (5.3) to obtain the updated phase-field function ϕ^n on Ω_c. Note that the convection term $\mathbf{u} \cdot \nabla\phi$ is treated by translating and rotating the moving domain Ω_{moving}. We use the operator splitting scheme [96]:

$$\begin{aligned} \epsilon^2(\phi^n)\frac{\phi^{n+1,1} - \phi^n}{\Delta t} &= 2\epsilon(\phi^n)\epsilon(\phi^n)_x\phi^n_x + 2\epsilon(\phi^n)\epsilon(\phi^n)_y\phi^n_y \qquad (5.15) \\ &+ \left(\frac{16\epsilon_4\epsilon(\phi^n)\phi_x(\phi_x^2\phi_y^2 - \phi_y^4)}{|\nabla_h\phi|^4}\right)^n_x + \left(\frac{16\epsilon_4\epsilon(\phi^n)\phi_y(\phi_x^2\phi_y^2 - \phi_x^4)}{|\nabla_h\phi|^4}\right)^n_y, \\ \epsilon^2(\phi^n)\frac{\phi^{n+1,2} - \phi^{n+1,1}}{\Delta t} &= \epsilon^2(\phi^n)\Delta_h\phi^{n+1,2} - 4\lambda\hat{U}^nF(\phi^{n+1,1}), \qquad (5.16) \end{aligned}$$

where $F(\phi) = 0.25(\phi^2 - 1)^2$. Because the temperature U^n is defined on Ω_f, we consider the interpolated function $\hat{U}^n$ for Ω_c. $\hat{U}^n_{kl}$ at $\mathbf{s}_{kl} \in \Omega^h_c$ is the bilinearly interpolated value from the temperature U^n at the position $\tilde{\mathbf{s}}_{kl} = \mathbf{X_1} + s^x_k\mathbf{a}/|\mathbf{a}| + s^y_l\mathbf{b}/|\mathbf{b}|$ on Ω_f (Fig. 5.4). If $\tilde{\mathbf{s}}_{ij} \notin \Omega_f$, then we define by $\hat{U}^n_{ij} = \Delta$ with a dimensionless undercooling Δ.

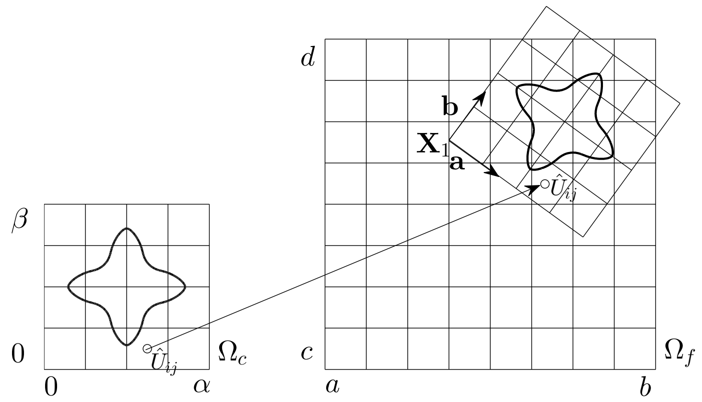


FIGURE 5.4. Schematic of the temperature field interpolation.

And then we update ϕ^{n+1}.

$$\phi^{n+1} = \phi^{n+1,2} \Bigg/ \sqrt{e^{-\frac{2\Delta t^*}{\epsilon^2(\phi^{n+1,2})}} + (\phi^{n+1,2})^2 \left(1 - e^{-\frac{2\Delta t^*}{\epsilon^2(\phi^{n+1,2})}}\right)}\,. \tag{5.17}$$

Equations (5.14) and (5.16) are solved by a multigrid method [129]. For more detail discretizations, please refer to [96].

5.3.3. Translation and rotation of $\Omega_{\mathbf{moving}}$. Also, we update the position of the advected crystal on Ω_f by moving Ω_{moving}. The magnitudes of its rotatory and parallel translations are derived from the conservations of the linear and angular momentums, respectively. Here, we calculate the velocity of the crystal

$$\mathbf{u}_c^n = (u_c^n, v_c^n),$$

and the rotation angle θ_c^n of the crystal to represent the motion of the crystal. Using the interpolated phase-field $\tilde{\phi}_{ij}^n$, fluid velocity $\mathbf{u}^n$, and Eq. (5.6), we can write $\mathbf{u}_c^n$, as

$$\mathbf{u}_c^n = \frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} 0.5\left(1+\tilde{\phi}_{ij}^n\right)\mathbf{u}_{ij}^n h^2}{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} 0.5\left(1+\tilde{\phi}_{ij}^n\right)h^2}. \tag{5.18}$$

Note that the interpolation of $\tilde{\phi}^n$ is described in Section 5.3.1. And, we estimate the fluid velocity at cell centers as

$$\mathbf{u}_{ij}^n = \left(0.5(u_{i+\frac{1}{2},j}^n + u_{i-\frac{1}{2},j}^n), 0.5(v_{i,j+\frac{1}{2}}^n + v_{i,j-\frac{1}{2}}^n)\right). \tag{5.19}$$

Next, we consider the angular momentum $\mathbf{L}_c$ of the crystal to calculate the rotation angle θ_c. Let ω_c^n be an approximation of ω_c at time $n\Delta t$. From Eq. (5.7), ω_c^n is written as

$$\omega_c^n = \frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} 0.5\left(\mathbf{m}_c^n - \mathbf{x}_{ij}\right)\times\mathbf{u}_{ij}^n\left(1+\tilde{\phi}_{ij}^n\right)h^2}{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\|\left(\mathbf{m}_c^n - \mathbf{x}_{ij}\right)\|^2 h^2}. \tag{5.20}$$

Therefore, we can compute the new position and rotation angle of the crystal domain Ω_c on the fluid domain Ω_f from Eqs. (5.18) and (5.20):

$$\mathbf{m}_c^{n+1} = \mathbf{m}_c^n + \mathbf{u}_c^n\Delta t, \tag{5.21}$$

$$\theta_c^{n+1} = \theta_c^n + \omega_c^n\Delta t. \tag{5.22}$$

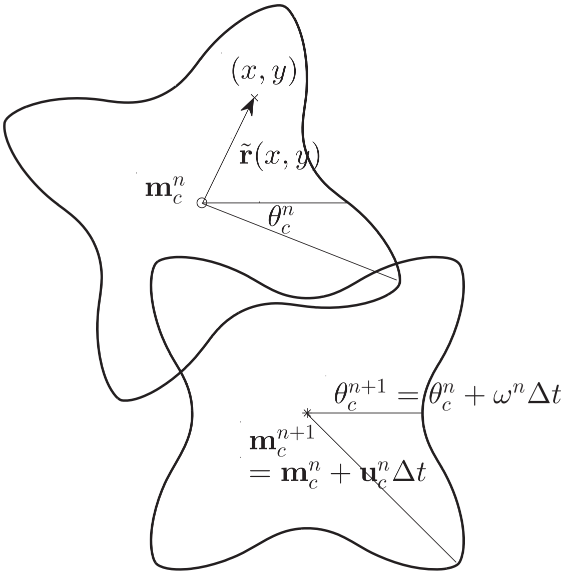

(x, y)
$\tilde{\mathbf{r}}(x, y)$
$\mathbf{m}_c^n$
θ_c^n
$\theta_c^{n+1} = \theta_c^n + \omega^n \Delta t$
$\mathbf{m}_c^{n+1}$
$= \mathbf{m}_c^n + \mathbf{u}_c^n \Delta t$

FIGURE 5.5. New center position and rotation angle of the crystal domain Ω_c.

5.3.4. Temperature field on Ω_f. Finally, we solve the temperature field U on the domain Ω_f with the homogenous Dirichlet boundary condition using the multigrid method:

$$\frac{U_{ij}^{n+1} - U_{ij}^n}{\Delta t} + (\mathbf{u}^n \cdot \nabla_d U^n)_{ij} = D \Delta_h U_{ij}^{n+1} + \frac{\tilde{\phi}_{ij}^{n+1} - \tilde{\phi}_{ij}^n}{2\Delta t}.$$

Here, we apply the upwind scheme for the advection term:

$$U_{x_{ij}}^n = \begin{cases} \frac{U_{ij}^n - U_{i-1,j}^n}{h}, & \text{if } u_{ij}^n > 0, \\ \frac{U_{i+1,j}^n - U_{ij}^n}{h}, & \text{otherwise,} \end{cases}$$

$$U_{y_{ij}}^n = \begin{cases} \frac{U_{ij}^n - U_{ij-1}^n}{h}, & \text{if } v_{ij}^n > 0, \\ \frac{U_{i,j+1}^n - u_{ij}^n}{h}, & \text{otherwise.} \end{cases}$$

Note that the velocity located at a cell center (u^n_{ij}, v^n_{ij}) is defined by using interpolation as

$$\Big(0.5(u^n_{i+\frac{1}{2},j} + u^n_{i-\frac{1}{2},j}), 0.5(v^n_{i,j+\frac{1}{2}} + v^n_{i,j-\frac{1}{2}})\Big)$$

as in Section 5.3.3.

5.3.5. Summary for the implementation. A brief summary of numerical procedures for crystal growth simulation in a cavity flow is as follows: Given the fluid velocity $\mathbf{u}^n$, the phase-field function ϕ^n, the temperature U^n, the crystal location $\mathbf{m}^n_c$, and the crystal rotation θ^n_c, we proceed the following steps:

1. Update $\mathbf{u}^{n+1}$ on Ω_f by Eqs. (5.8) and (5.9).
2. Update ϕ^{n+1} on Ω_c by Eqs. (5.15)–(5.17).
3. Update $\mathbf{m}^{n+1}_c$ and θ^{n+1}_c of Ω_{moving} by Eqs. (5.21) and (5.22).
4. Update U^{n+1} on Ω_f by Eq. (5.23).

This completes the description of the process above by which the quantities $\mathbf{u}^{n+1}$, ϕ^{n+1}, U^{n+1}, $\mathbf{m}^{n+1}_c$, and θ^{n+1}_c.

Moreover, we compare our moving overset grid method for the implementation with the fictitious domain model [39] in Table 5.1. As mentioned above, both methods share the idea using the local and global domains to simulate the growth and convection separately; however, there are differences in describing other physical properties.

Table 5.1. Comparison of the moving overset grid method with the fictitious domain method

	Moving overset grid	Fictitious domain model [39]
Using the local domain for the phase-field	O	O
Solving the NS equations in the local domain	X	O
Advection term in the phase-field equation	X	O
Treating permeability	Using a very large viscosity ratio	Adding a term in the NS equation for the local domain
Enforcing rigid motion	Calculate the translation velocity and the angular velocity separately	Using the correct velocity

5.4. Numerical experiments

In this section, we present an example to numerically demonstrate the efficiency of the proposed methods. The first example compares the result by the proposed translation algorithm with the result by solving the advection equation. And, next examples show temporal evolution of the crystal growth in two-dimensional flows.

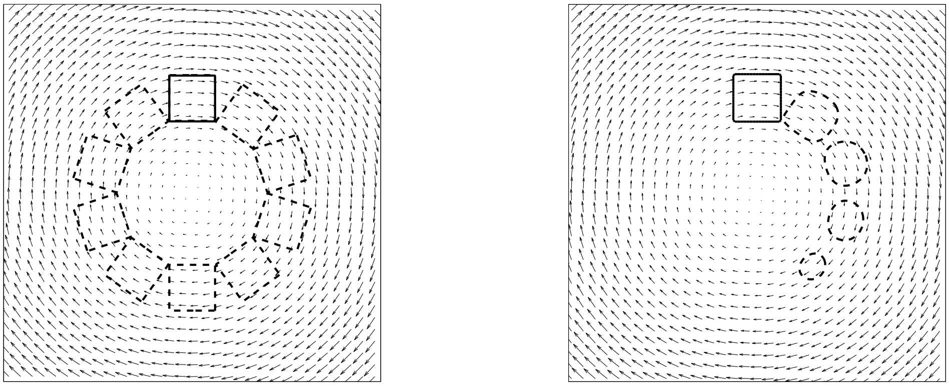
5.4.1. Translation algorithm. We first compare results by our translation algorithm for the moving domain Ω_{moving} with those of the usual advection equation. Here, we just consider an advection equation

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0,$$

i.e., without the crystal growth and the heat diffusion. The underlying velocity field is given by a rotational flow

$$(u, v) = \left(\frac{y}{100}, -\frac{x}{100}\right),$$

which gives a uniform angular velocity in a fluid domain $\Omega_f = (-50, 50) \times (-50, 50)$. The initial translated feature is a square whose one side has a length of 12 and it is defined on a quadruply smaller domain $\Omega_c = (0, 50) \times (0, 50)$. The location of the corresponding moving domain Ω_{moving} is set as defining $\mathbf{m}_c^0 = (50, 75)$ in Ω_f.



(a)

(b)

FIGURE 5.6. Translation of the non-growing crystal under the rotational flow by (a) applying a proposed overset grid and (b) solving the advection equation. The solid contour represents the initial configuration, dotted contours are the configurations at every 400 iterations, and arrows are the underlying velocity field.

Figure 5.6 shows the translation of the non-growing feature under the rotational flow with $\Delta t = 0.05\pi$ up to the final time $T = 200\pi$ by (a) applying a proposed overset grid and (b) solving the advection equation. Here, a upwind scheme is applied to solve the advection equation and it is well known that the smeared phenomena occur by the upwind scheme. We could, of course, obtain the better results by using other time integrating schemes such as Lax–Friedrichs, Lax–Wendroff, Godunov's, Leapfrog, essentially non-oscillatory, weighted essentially non-oscillatory, and so on; however, it is not easy to not only implement but also avoid the drawback perfectly. On the other hand, the overset grid method shows that no unreliable deformation of the structure in contrast with solving the advection equation directly.

Table 5.2. Numerical and theoretical angles at T.

mesh	128^2	256^2	512^2	exact
angle	6.286168	6.283511	6.283191	6.283185

Because of the uniform velocity $\omega = 0.01$ from the underlying velocity, the initial and final configurations should match with each other exactly; i.e., the angle is 2π at the final time. Table 5.2 lists the numerical and theoretical angles at T up to seven significant figures. As the mesh is finer, the angles from the numerical experiments converge to the exact value. In conclusion, the results in this section show that our translation algorithm has a good agreement with the theoretical one.

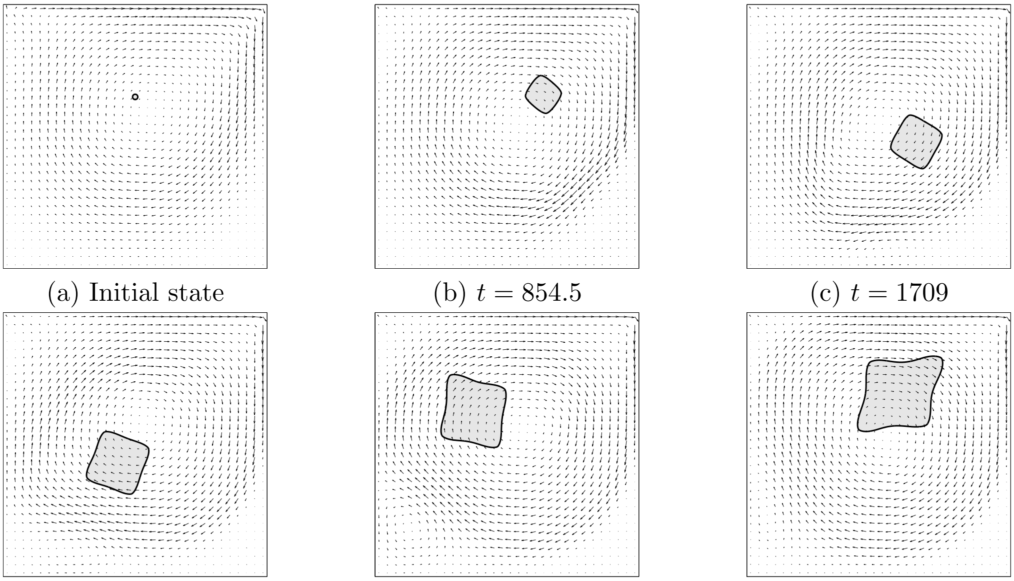
5.4.2. Cavity flow. We perform numerical experiments for the crystal growth in a two-dimensional cavity flow. We consider a sufficiently large domain $\Omega_f = [-300, 300] \times [-300, 300]$ to observe the growth of crystal under the flows for the fluid and temperature and $\Omega_c = [0, 300] \times [0, 300]$ for the crystal. In Ω_f, the time step is restricted to $\Delta t \leq 0.25 Re h^2 \eta_m / \eta_s$ due to the explicit discretization for the diffusion term in Eq. (5.8). In contrast, in Ω_c, the operator splitting method for solving Eqs. (5.15)–(5.16) allows large time step, e.g., $\Delta t \leq 5.5h$ [96]. Thus, we use $\Delta t = \min\big(0.2Reh^2\eta_m/\eta_s, 5.5h\big)$, unless otherwise specified.

For the initial state, we take:

$$\phi(x,y,0) = \tanh\left(\frac{R_0 - \sqrt{x^2+y^2}}{\sqrt{2}}\right) \quad \text{and} \quad U(x,y,0) = \begin{cases} 0 & \text{if } \phi > 0 \\ \Delta & \text{otherwise} \end{cases}$$

The zero level set ($\phi = 0$) of the initial state represents a circle of radius $R_0 = 6$. From the definition of dimensionless variable U, the value of 0 corresponds to the melting temperature of the pure material, while the value of Δ is the initial undercooling. And, we use $\lambda = 3.1913$ like as [96, 97, 114]. The initial center $\mathbf{m}_c^0$ is located at $(0, 120)$ and the initial fluid flow is defined as the steady state solution of the cavity flow $Re = 1$ to reduce the computational time. For other parameters, we set as follows: $\epsilon_4 = 0.05$, $D = 2.0$, $\Delta = -0.3$, $Re = 10$, $\eta_m = 1$, $\eta_s = 25$, and $T = 4272$.

Figure 5.7 shows the temporal evolution of the crystal growth in the cavity flow. The snapshots with the contour of crystal and corresponding fluid vector field are drawn at the specified time. Now, the crystal is not fixed but floats in a liquid, so that the crystal can not grow symmetrically anymore. The flow and crystal affect each other by the viscosity difference, and the crystal follows

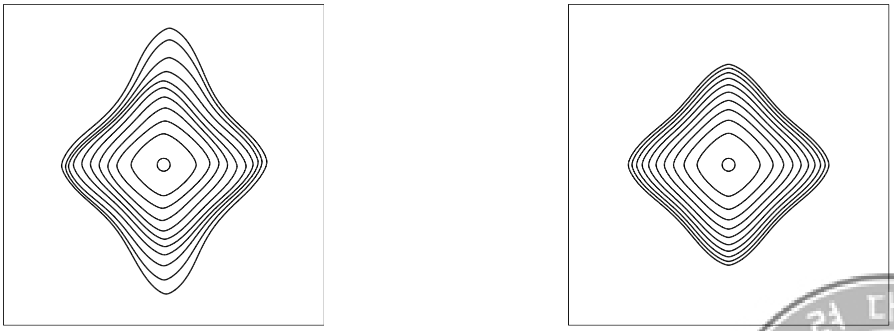


(a) Initial state (b) $t = 854.5$ (c) $t = 1709$

(d) $t = 2563$ (e) $t = 3418$ (f) $t = T$

FIGURE 5.7. Evolutions for the traces of crystal and the flow field.

the fluid flow. The heat distribution is also changed due to the interaction of the flow and crystal, and it makes the non-symmetrical crystal growth.

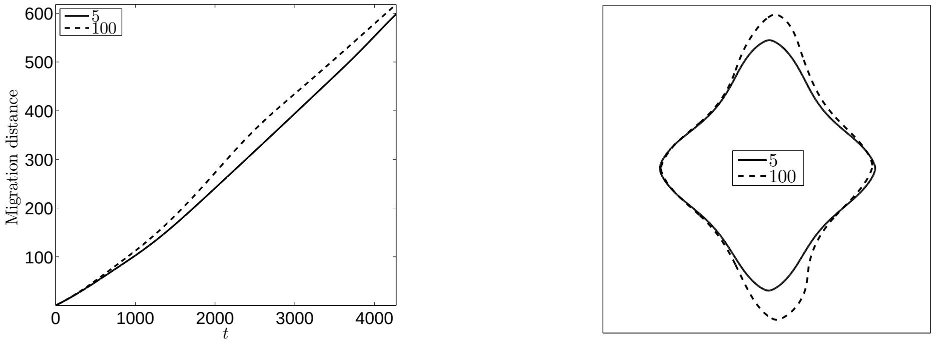


(a) with the flow (b) without the flow

FIGURE 5.8. Contours of crystal growth (a) with and (b) without the cavity flow. The elapsed time for each contour is 427.2.

In addition, Fig. 5.8 displays the temporal evolutions of crystal growth with and without the cavity flow to show the effect of the flow. Same parameters $\epsilon_4 = 0.05$, $D = 2.0$, and $\Delta = -0.3$ are used in a domain $[0, 300] \times [0, 300]$ to simulate the crystal growth without the flow. Small difference of growth is observed in the early stage, however, in the case of with the flow, symmetry in growing branches is getting broken and the crystal is growing faster.

Next, we check the Reynolds number effect for the crystal growth in a cavity flow. The used parameters are same as previous simulation except for Re and Δt. For comparison of the Reynolds number effects, we set $Re = 5$ and $Re = 100$. Since the time step restriction depends on Re, the corresponding Δt is used as 0.02441 and 0.04883 for $Re = 5$ and $Re = 100$, respectively.

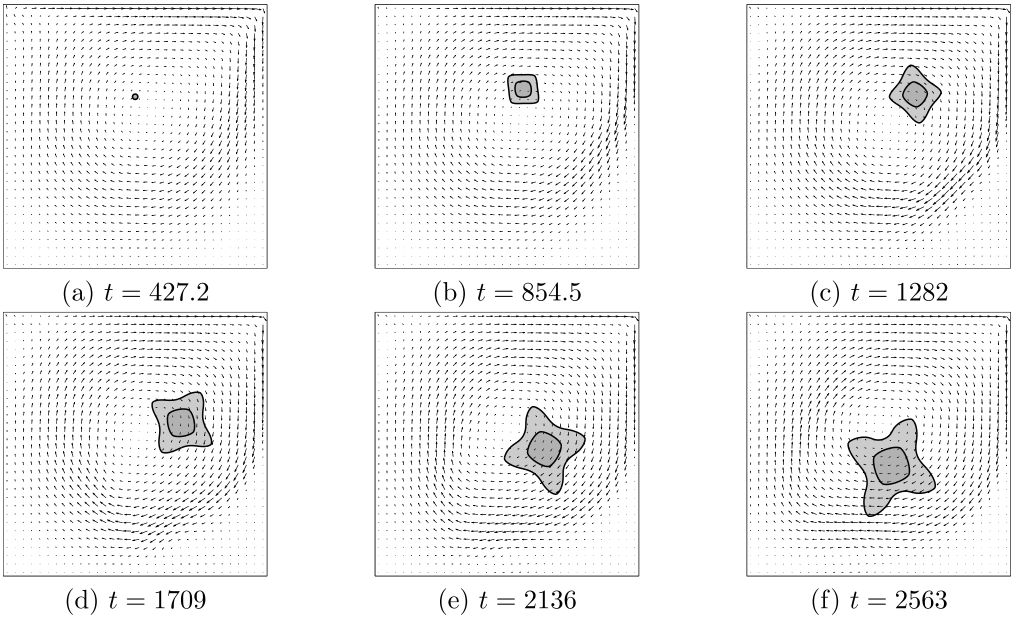


(a) (b)

FIGURE 5.9. (a) Migration distances with respect to time and (b) contours of the crystals at T.

Figures 5.9 (a) and (b) show that migration with respect to time and configurations of the crystals at the final time T. A moving distance is little larger when $Re = 100$. Moreover, there is a difference in the growth rates of the crystals with different Re numbers. Generally, absorbing external heat would give rise to growth of crystal. As shown in the figure, migration distance is changed as Re is also changed and it implies that the crystal absorbs more external heat when $Re = 100$ than $Re = 5$. This agrees well with our result for growth of crystal.

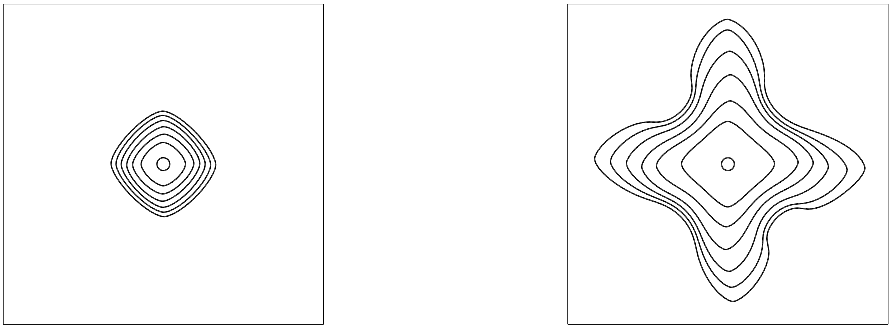
The initial undercooling Δ is also one of the influential parameter to determine the growth of the crystal. We perform simulations to check the effect of Δ in the cavity flow. Figure 5.10 shows that configurations of crystal with $\Delta = -0.2$ and $\Delta = -0.4$ until $t = 2563$. The other parameter settings and initial condition are same as the simulation in the previous result.


(a) $t = 427.2$ (b) $t = 854.5$ (c) $t = 1282$

(d) $t = 1709$ (e) $t = 2136$ (f) $t = 2563$

FIGURE 5.10. Evolution of the crystal growth under the flow fields with $\Delta = -0.2$ (solid line) and $\Delta = -0.4$ (dotted line).

Figure 5.10 shows that the traces of crystal growth under the flow vector fields with $\Delta = -0.2$ (solid line) and $\Delta = -0.4$ (dotted line). Here, the flow vector fields is drawn for better understanding in convection of crystal. As seen in the figures, the crystals with different Δ has apparently different morphology.



(a) $\Delta = -0.2$ (b) $\Delta = -0.4$

FIGURE 5.11. Contours of crystal growth when (a) $\Delta = -0.2$ and (b) $\Delta = -0.4$. The elapsed time for each contour is 427.2.

We also displays the temporal evolutions of crystal growth when $\Delta = -0.2$ and $\Delta = -0.4$ in Fig. 5.11 to compare both cases conveniently.

5.4.3. Gravitational flow. The Boussinesq approximation can be made the gravitational effect, i.e., the accelerations of flow by gravitational force, with a constant density. Hence, the constant density contribute to the buoyancy force in terms of the momentum equation [99]. Because of the simplicity in practical implementations, the approximation has been widely applied in many previous articles [62, 79, 101]. In [88], the variable density and the Boussinesq models are compared on buoyancy-driven flows and we implement the gravitational force as presented in the reference.

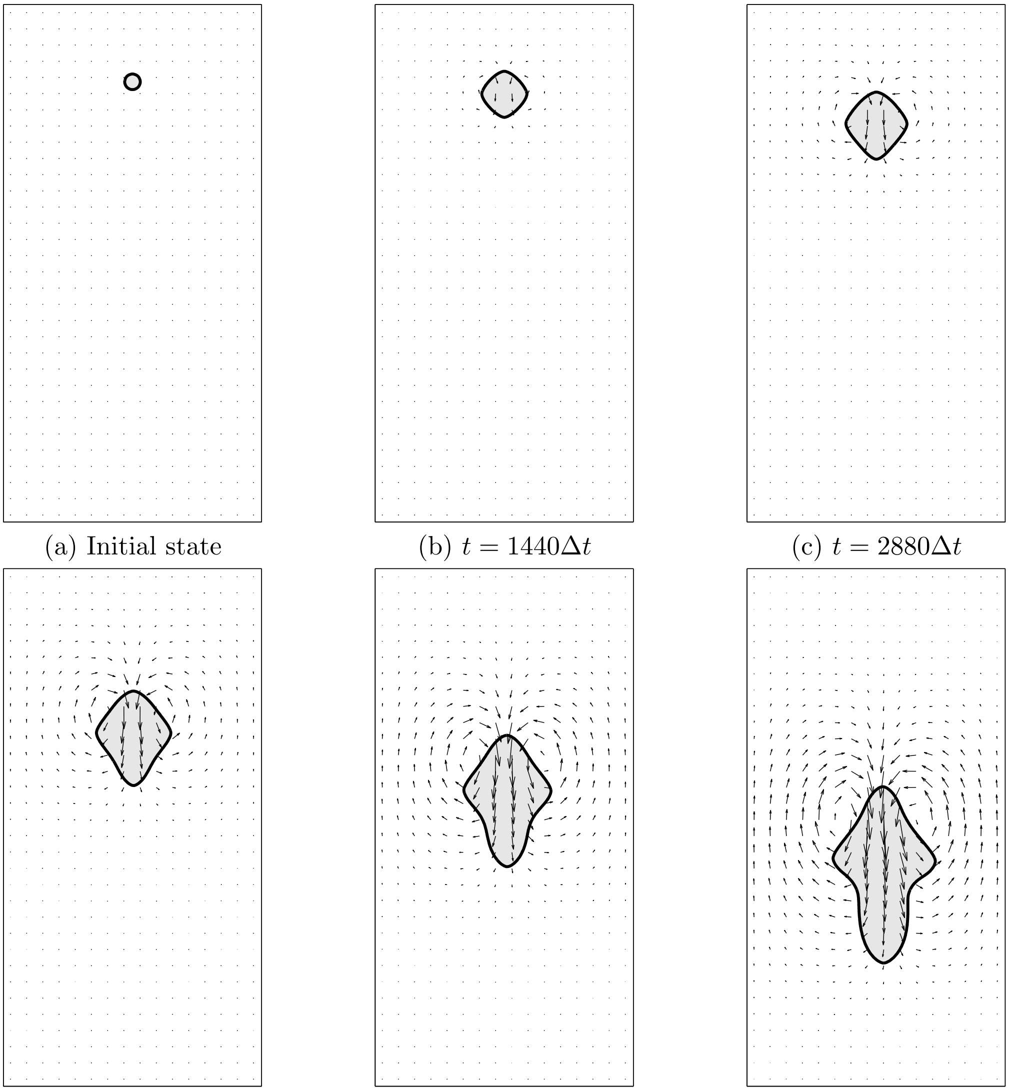
The NS equation with a gravitational force in terms of the Boussinesq approximation can be modified from 5.1 as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla \cdot \left[\eta(\phi)(\nabla \mathbf{u} + \mathbf{u}^T) \right] + At \left(\frac{1+\phi}{2} \right) \mathbf{g}, \tag{5.23}$$

where $\mathbf{g}$ is the gravitational force density. Note that it is written as $(0, -g)$ in a two-dimensional case. To implement this model, we just add the vector term $\mathbf{g}$ in the step for solving an intermediate velocity field.

Figure 5.12 shows the evolutions and the traces of the crystal and the fluid flow fields. Here, a fluid domain $\Omega_f = [0, 200] \times [0, 400]$ and a crystal domain $\Omega_c = [0, 100] \times [0, 100]$ is considers. The mesh grids of Ω_f and Ω_c are 256×512 and 128×128, respectively. Moreover, the parameters $\eta_m = 1$, $\eta_s = 25$, $Re = 10$, $\epsilon_4 = 0.05$, $D = 2$, $\Delta = -0.4$, $\Delta t = 0.2Reh^2\eta_m/\eta_s$, $T = 7200\Delta t$, $At = 0.01$, $\mathbf{g} = (0, -1)$ are used. Because of the gravitational force, generated by the crystal, we can see the force-driven downward flow and the direction of the crystal growth tends to also downward.

Figure 5.13 shows that evolutions and traces of the crystal and the fluid flow fields when the initial seed is not isotropic. Other parameters are same as the previous simulation for Fig. 5.12 except initial conditions. Here, the initial seed is a square whose length is 12 and the initial angle of Ω_c is given by 0.125π. Because of the anisotropic condition of the initial state, the morphology



(a) Initial state

(b) $t = 1440\Delta t$

(c) $t = 2880\Delta t$

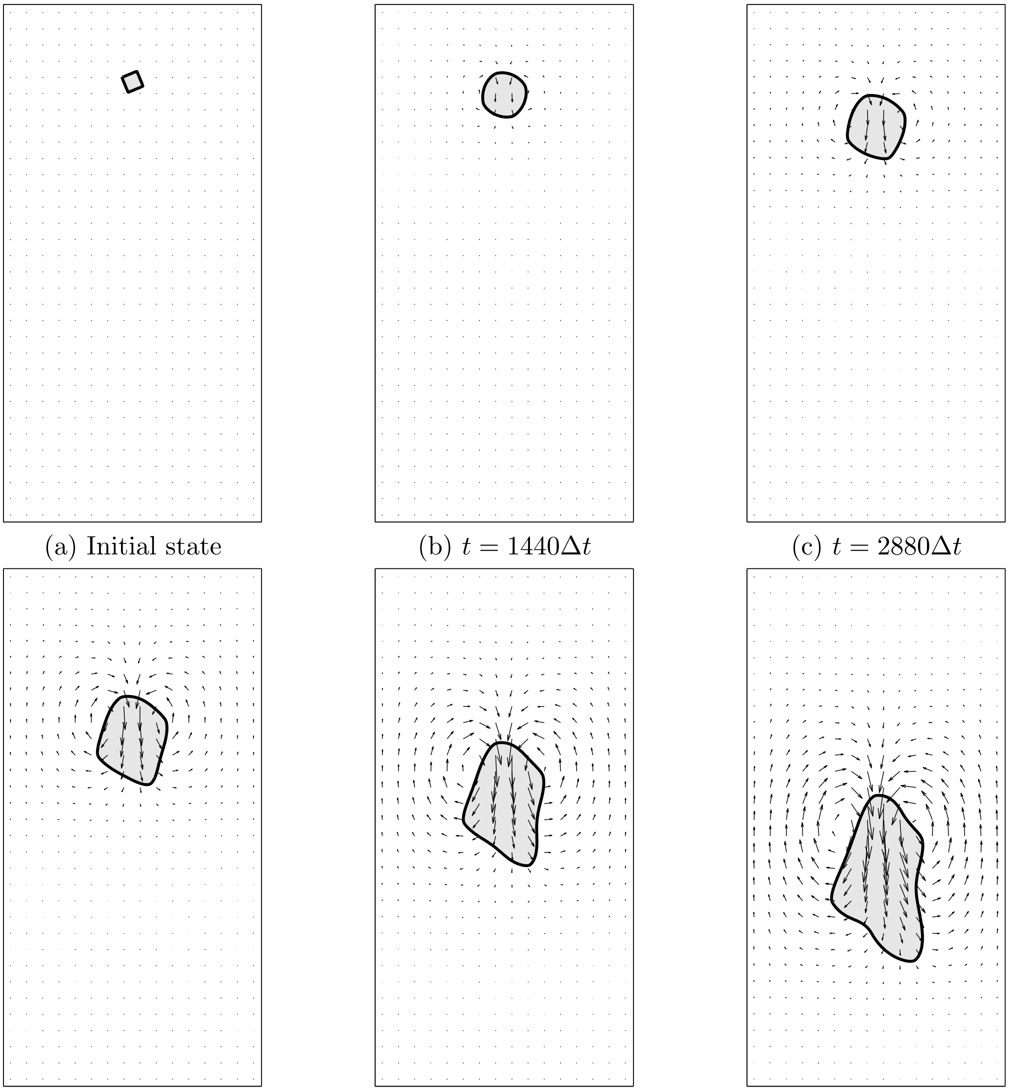
(d) $t = 4320\Delta t$

(e) $t = 5760\Delta t$

(f) $t = 7200\Delta t$

Figure 5.12. Evolutions and traces of the crystal and the fluid flow field.

of the crystal growth is different comparing with the previous one. The direction of growth is not symmetric for left and right tips of the crystal, and the bottom tip of the crystal growths on a skew.



(a) Initial state

(b) $t = 1440\Delta t$

(c) $t = 2880\Delta t$

(d) $t = 4320\Delta t$

(e) $t = 5760\Delta t$

(f) $t = 7200\Delta t$

FIGURE 5.13. Evolutions and traces of the crystal and the fluid flow field with anisotropic initial data.

Chapter 6

Conclusion

We proposed a moving overset grid method to model dendritic growth with convection in phase-field simulations. Unlike most previous studies, the translation and rotation of the crystal were simulated with its deformation. The proposed numerical method was able to be resolve the difficulties in deformation of the crystal shape and ambiguity of the crystal orientation for the anisotropy since the phase-field and fluid flow are solved in distinct domains. Numerical results demonstrated that the proposed method can predict the crystal growth under flows. As future researches, a more accurate algorithm for the motion of fluid and crystal can be applied to the proposed moving overset grid method and the methodology introduced in this thesis can be extended to three-dimensional space.

Appendix A

Multigrid method

A multigrid method is known as one of the fastest method in solving common discretization problems [4, 17, 129]. When applying the Jacobi and Gauss–Seidel iterative methods, high frequencies of the errors are almost removed in a few iteration but the lower frequencies are reduced very slowly. Moreover, the computational cost is $O(N^2)$. In a multigrid method, lower frequencies act like higher frequencies by changing data on a finer grid to a coarser grid. After solving a problem on a coarser grid, we again interpolate the data back to a finer grid. This fine-coarse-fine loop is called a v-cycle and the sub-steps are called restriction and prolongation [122].

In this dissertation, we use the method when solving the AC equation, the CH equation, a Poisson problem, and the heat equation. Here, we present practical examples of the multigrid methods: the heat equation for a linear equation case and the AC equation for a nonlinear equation case.

A.1. Linear multigrid method

The linear multigrid method is one of the fastest iterative solvers for solving linear partial differential equations. Note that the differential equation

$$\mathcal{L}y = f,$$

is called linear when $\mathcal{L}$ is a linear operator, i.e., satisfies

$$\mathcal{L}(y_1 + y_2) = \mathcal{L}(y_1) + \mathcal{L}(y_2).$$

where $\mathcal{L}$ is the differential operator, y is the unknown function, and f is a given function. Therefore, the classical heat equation is a good example of a linear partial differential equation.

The one-dimensional heat equation with homogenous Neumann boundary condition is given as follows:

$$\begin{aligned}
\frac{\partial}{\partial t}u(x,t) =& \Delta u(x,t),\ \ x \in \Omega = (0,1),\ \ t \in (0,T], && \text{(A.1)}\\
\frac{\partial}{\partial x}u(0,t) =& \frac{\partial}{\partial x}u(1,t) = 0, && \text{(A.2)}\\
u(x,0) =& u_0(x), && \text{(A.3)}
\end{aligned}$$

where $u(x,t)$ is the temperature, x is the spatial variable, t is the temporal variable, Ω is the spatial unit domain, T is the final time, and $u_0(x)$ is the given initial condition.

Applying a backward-time central-space scheme (or the implicit Euler's scheme) [106], Eqs. (A.1) can be discretized as follow:

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\frac{u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}}{h^2}, \tag{A.4}$$

where $u_i^n = u\left((i-0.5)h, n\Delta t\right)$ for $i=1,2,\ldots,N_x$ and $n=0,2,\ldots,N_t$. Here $h=1/N_x$ is the spatial step size, N_x is the number of spatial steps, Δt is the temporal step size, and N_t is the number of temporal steps. The homogeneous Neumann boundary conditions are discretized as

$$u_0^n = u_1^n,\ u_{N_x+1}^n = u_{N_x}^n,\ 0 \le n \le N_t.$$

Let us rewrite Eq. (A.4) as

$$\frac{u_i^{n+1}}{\Delta t}-\frac{u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}}{h^2}=\frac{u_i^n}{\Delta t}. \tag{A.5}$$

Using an operator notation, we also let Eq. (A.5) as

$$\mathcal{L}u_i^{n+1}=f_i, \tag{A.6}$$

where

$$\begin{aligned}\mathcal{L}u_i^{n+1} =& \frac{u_i^{n+1}}{\Delta t}-\frac{u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}}{h^2},\\ f_i =& \frac{u_i^n}{\Delta t}.\end{aligned}$$

In order to explain clearly the steps taken during a single V-cycle, we focus on a numerical solution on an eight point grid. We define discrete domains, $\Omega_2,\ \Omega_1$, and Ω_0, where

$$\Omega_k = \{(x_i = (i-0.5)h_k)|1 \le i \le 2^{k+1} \text{ and } h_k = 2^{2-k}h\}.$$

Ω_{k-1} is coarser than Ω_k by a factor of 2. The multigrid solution of the discrete heat equation (A.4) makes use of a hierarchy of meshes ($\Omega_2,\ \Omega_1$, and Ω_0) created by successively coarsening the original mesh, Ω_2 as shown in Fig. A.1. A pointwise Gauss–Seidel relaxation scheme is used as the smoother in the multigrid method. We use notations u_k^n at time $t = n\Delta t$ and $\mathcal{L}_k$ as a numerical solution and the operator defined as (A.6) on the discrete domain Ω_k. We rewrite the above Eq. (A.6) by

$$\mathcal{L}_2(u_2^{n+1}) = f_2 \text{ on } \Omega_2. \tag{A.7}$$

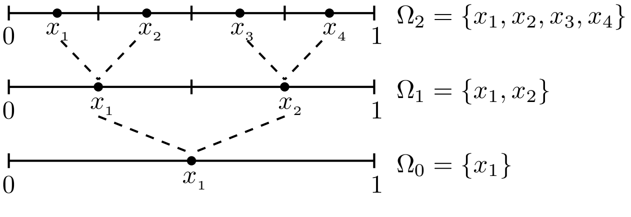


FIGURE A.1. A sequence of coarse grids starting with $h = L/N_x$.

The algorithm of the multigrid method for solving the discrete heat Eq. (A.6) is as follows.

Multigrid cycle

$$u_k^{n+1,m+1} = MGcycle(k, u_k^{n+1,m}, L_k, f_k, \nu_1, \nu_2).$$

That is, $u_k^{n+1,m}$ and $u_k^{n+1,m+1}$ are the approximations of u_k^{n+1} before and after an MGcycle. Given the numbers ν_1 and ν_2 are pre- and post- smoothing relaxation sweeps of an iteration step for the multigrid method using the V-cycle [129]. Starting an initial condition u_2^0, we want to find u_2^n for $n = 1, 2, \cdots$. Given u_2^n, we want to find the u_2^{n+1} solution that satisfies Eq. (A.7). At the very beginning of the multigrid cycle the solution from the previous time step is used to provide an initial guess for the multigrid procedure. First, let $u_2^{n+1,0} = u_2^n$.

Step 1) Presmoothing

$$\bar{u}_k^{n+1,m} = SMOOTH^{\nu_1}(u_k^{n+1,m}, L_k, f_k),$$

means performing ν_1 smoothing steps with the initial approximation $u_k^{n+1,m}$, source terms f_k, and a $SMOOTH$ relaxation operator to get the approximation $\bar{u}_k^{n+1,m}$.

We use the following Gauss–Seidel relaxation scheme.

$$\begin{aligned} u_1^{n+1,m,s+1} &= \left(\frac{u_1^n}{\Delta t} + \frac{u_2^{n+1,m,s}}{h^2} \right) \Bigg/ \left(\frac{1}{\Delta t} + \frac{1}{h^2} \right), \\ u_i^{n+1,m,s+1} &= \left(\frac{u_i^n}{\Delta t} + \frac{u_{i-1}^{n+1,m,s+1} + u_{i+1}^{n+1,m,s}}{h^2} \right) \Bigg/ \left(\frac{1}{\Delta t} + \frac{2}{h^2} \right), \\ & \qquad (2 \le i \le 2^{k-2} N_x - 1), \\ u_{N_x}^{n+1,m,s+1} &= \left(\frac{u_{N_x}^n}{\Delta t} + \frac{u_{N_x-1}^{n+1,m,s+1}}{h^2} \right) \Bigg/ \left(\frac{1}{\Delta t} + \frac{1}{h^2} \right), \end{aligned} \tag{A.8}$$

where s and $s+1$ denote the current and the new approximations, respectively. Therefore, in a multigrid cycle, one smooth relaxation operator step consists of solving Eq. (A.8) given above for $1 \le i \le 2^{k-2} N_x$.

Step 2) Coarse grid correction

- Compute the defect: $\bar{d}_k^m = u_k^n - L_k(\bar{u}_k^{n+1,m})$.
- Restrict the defect and $\bar{u}_k^m$: $\bar{d}_{k-1}^m = I_k^{k-1} \bar{d}_k^m$

The restriction operator I_k^{k-1} maps k-level functions to $(k-1)$-level functions.

$$\begin{aligned} d_{k-1}(x_i, y_j) =& I_k^{k-1} d_k(x_i, y_j) \\ =& \frac{1}{2} [d_k(x_{i-\frac{1}{2}}) + d_k(x_{i+\frac{1}{2}})]. \end{aligned}$$

- Compute an approximate solution $\hat{u}_{k-1}^{n+1,m}$ of the coarse grid equation on Ω_{k-1}, i.e.

$$L_{k-1}(u_{k-1}^{n+1,m}) = \bar{d}_{k-1}^m. \tag{A.9}$$

If $k = 1$, we use a direct or fast iteration solver for (A.9). If $k > 1$, we solve (A.9) approximately by performing k-grid cycles using the zero grid function as an initial approximation:

$$\hat{v}_{k-1}^{n+1,m} = MGcycle(k-1, 0, L_{k-1}, \bar{d}_{k-1}^{m}, \nu_1, \nu_2).$$

- Interpolate the correction: $\hat{v}_k^{n+1,m} = I_{k-1}^{k} \hat{v}_{k-1}^{n+1,m}$.

  Here, the coarse values are simply transferred to the four nearby fine grid points, i.e.,

$$\begin{aligned} v_k(x_i, y_j) =& I_{k-1}^{k} v_{k-1}(x_i, y_j) \\ =& v_{k-1}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}), \end{aligned}$$

  for the i and j odd-numbered integers.

- Compute the corrected approximation on Ω_k

$$u_k^{m,\text{ after } CGC} = \bar{u}_k^{n+1,m} + \hat{v}_k^{n+1,m}.$$

Step 3) Postsmoothing:

$$u_k^{n+1,m+1} = SMOOTH^{\nu_2}(u_k^{m,\text{ after } CGC}, L_k, f_k).$$

This completes the description of a one MGcycle step, which stops if the consequence error $|u^{n+1,m+1} - u^{n+1,m}|$ is smaller than a given tolerance. An illustration of the corresponding two-grid cycle is given in Fig. A.2.

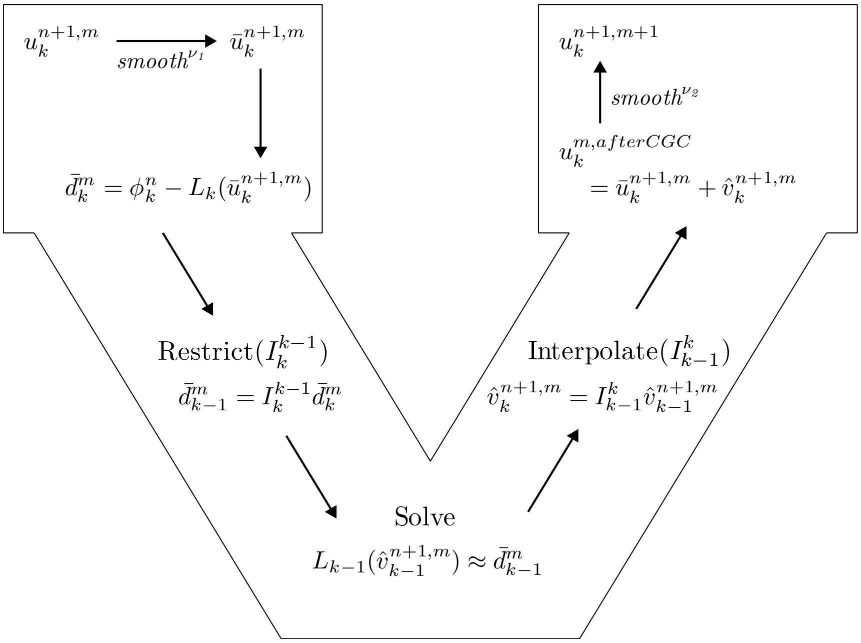

$u_k^{n+1,m}$
$smooth^{\nu_1}$
$\bar{u}_k^{n+1,m}$
$\bar{d}_k^m = \phi_k^n - L_k(\bar{u}_k^{n+1,m})$
Restrict(I_k^{k-1})
$\bar{d}_{k-1}^m = I_k^{k-1}\bar{d}_k^m$
Solve
$L_{k-1}(\hat{v}_{k-1}^{n+1,m}) \approx \bar{d}_{k-1}^m$
Interpolate(I_{k-1}^k)
$\hat{v}_k^{n+1,m} = I_{k-1}^k\hat{v}_{k-1}^{n+1,m}$
$u_k^{m,afterCGC}$
$= \bar{u}_k^{n+1,m} + \hat{v}_k^{n+1,m}$
$smooth^{\nu_2}$
$u_k^{n+1,m+1}$

FIGURE A.2. The *MGcycle* $(k, k-1)$ two-grid method.

To show the numerical result comparing an analytic solution, we set the initial conditions as $u_0(x) = \cos(2\pi x)$ on the space domain $\Omega = (0, 1)$. We take $h = 1/128$ and $\Delta t = 0.5h$. The number of relaxation is 3. Tolerance is 1.0*e*-7. Figure A.3 shows temporal evolution of the numerical solutions at $t = 0, 2\Delta t, 4\Delta t, 6\Delta t, 8\Delta t$, and $10\Delta t$. The exact solution for the heat equation with the given initial conditions can be derived by using the separation of variables. Let $u(x,t) = X(x)T(t)$. Then, Eq. (A.1) is rewritten as

$$X(x)T'(t) = X''(x)T(t),$$

$$\Leftrightarrow \frac{T'}{T} = \frac{X''}{X}.$$

Using the boundary conditions (A.2) and the initial condition (A.3), we can get the solution $\cos(2\pi x)e^{-\pi^2 t}$.

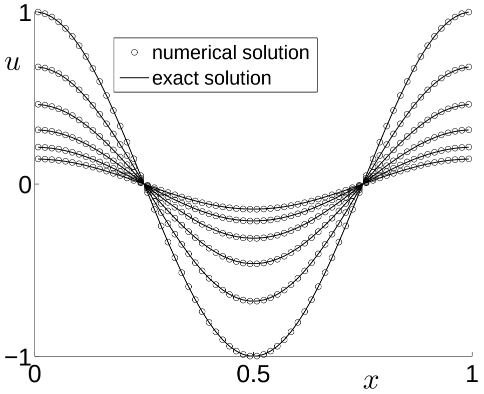


Figure A.3. Temporal evolution of the numerical solutions with the initial condition $u(x,0) = \cos(2\pi x)$. Times are at $t = 0, 2\Delta t, 4\Delta t, 6\Delta t, 8\Delta t$, and $10\Delta t$.

A.2. Nonlinear multigrid method

Next, we study a nonlinear multigrid method with the AC equation as an example, which is the simple nonlinear partial differential equation.

Since the details of AC equation is presented in the section 2.2 of the main text, we just introduce one-dimensional form here:

$$\frac{\partial}{\partial t}\phi(x,t) = -\,\phi^3(x,t) + \phi(x,t) + \epsilon^2 \frac{\partial^2}{\partial x^2}\phi(x,t),\ x \in \Omega = (0,1),\ t \in (0,T] \tag{A.10}$$

$$\frac{\partial}{\partial x}\phi(0,t) = \frac{\partial}{\partial x}\phi(1,t) = 0, \tag{A.11}$$

$$\phi(x,0) = \phi_0(x), \tag{A.12}$$

where $\phi(x,t)$ is the order parameter in $[-1,1]$, x is the spatial variable, t is the temporal variable, ϵ is the coefficient related to interfacial energy, T is the final time. Here, ϕ^3 terms in the right hand side of Eq. (A.10) gives a nonlinearity of the differential equation. To treat a nonlinearity in a multigrid method, we use a linear approximation in a smooth relaxation step. Here, we consider the nonlinearly stabilized splitting scheme [48, 49] for a discretization of the AC equation as follow:

$$\frac{\phi_i^{n+1}-\phi_i^n}{\Delta t} = -\left(\phi_i^{n+1}\right)^3 + \phi_i^n + \epsilon^2 \frac{\phi_{i-1}^{n+1} - 2\phi_i^{n+1} + \phi_{i+1}^{n+1}}{h^2}, \tag{A.13}$$

where $\phi_i^n = \phi(x_i = (i-0.5)h, n\Delta t)$, Δt is the temporal step size, $N_t = T/\Delta t$ is the number of temporal steps, h is the spatial step size, and N_x is the number of spatial steps, for $i = 1, \cdots, N_x$ and $n = 1, \cdots, N_t$.

We describe a nonlinear full approximation storage (FAS) multigrid method to solve Eqs. (A.10), which is the nonlinear discrete system of equations. Refer [18, 129] for the details and the backgrounds of the nonlinear multigrid method.

Let

$$\boldsymbol{\phi}^{n+1} = (\phi_1^{n+1}, \phi_2^{n+1}, \ldots, \phi_{N_x}^{n+1})$$

and

$$\mathbf{f}^n = (f_1^n, f_2^n, \ldots, f_{N_x}^n).$$

Let us rewrite Eq. (A.13) as follow using an operator notation:

$$\mathcal{N}\boldsymbol{\phi}^{n+1} = \mathbf{f}^n, \tag{A.14}$$

where

$$\mathcal{N}\phi_i^{n+1} = \frac{\phi_i^{n+1}}{\Delta t} + \left(\phi_i^{n+1}\right)^3 - \epsilon^2 \frac{\phi_{i-1}^{n+1} - 2\phi_i^{n+1} + \phi_{i+1}^{n+1}}{h^2}$$

and

$$f_i^n = \frac{\phi_i^n}{\Delta t + \phi_i^n}.$$

Given $\boldsymbol{\phi}^n$, we want to calculate $\boldsymbol{\phi}^{n+1}$. We iterate the following FAS multigrid cycle until the discrete l_2-norm of the two consecutive approximations is less than a given tolerance, i.e.,

$$\|\boldsymbol{\phi}^{n+1,m+1} - \boldsymbol{\phi}^{n+1,m}\|_2 < tol.$$

Let

$$\Omega_K = \{x_i | i = 1, \ldots, N_x\},$$

be the original finest grid, where K satisfies $N_x = p \cdot 2^K$ and p is an odd number. Then, for $k = K, \ldots, 1$, we define the successively coarser grids as

$$\Omega_{k-1} = \{y_i | y_i = 0.5(x_{2i-1} + x_{2i}) \text{ and } x_{2i-1}, x_{2i} \in \Omega_k \text{ for } i = 1, \ldots, p \cdot 2^{k-1}\}.$$

Now, we introduce the nonlinear multigrid iteration for solving the discretized problem (A.14) on grid level Ω_k.

$$\boldsymbol{\phi}_k^{n+1,m+1} = FAScycle(\boldsymbol{\phi}_k^{n+1,m}, \mathcal{N}_k, \mathbf{f}_k^n, \nu),$$

which means that $\boldsymbol{\phi}_k^{n+1,m}$ and $\boldsymbol{\phi}_k^{n+1,m+1}$ are the approximations of $\boldsymbol{\phi}^{n+1}$ before and after an FAScycle on grid level Ω_k. By starting from an initial value $\boldsymbol{\phi}^{n+1,0} = \boldsymbol{\phi}^n$, one step of the iteration is given in the following step:

Step 1) Presmoothing

$$\bar{\boldsymbol{\phi}}_k^{n+1,m} = SMOOTH^{\nu}(\boldsymbol{\phi}_k^{n+1,m}, \mathcal{N}_k, \mathbf{f}_k^n) \text{ on } \Omega_k \text{ grid.}$$

This means performing ν smoothing steps with the initial approximations $\boldsymbol{\phi}_k^{n+1,m}$, source terms $\mathbf{f}_k^n$, and $SMOOTH$ relaxation operator to get the approximations $\bar{\boldsymbol{\phi}}_k^{n+1,m}$.

First, let us rearrange discrete Eq. (A.13) as a Gauss–Seidel type.

$$\begin{aligned} &\frac{\phi_i^{n+1,m,s+1}}{\Delta t} + \left(\phi_i^{n+1,m,s+1}\right)^3 + \frac{2\epsilon^2 \phi_i^{n+1,m,s+1}}{h^2} \\ &\qquad\qquad = f_i^n + \epsilon^2 \frac{\phi_{i-1}^{n+1,m,s+1} + \phi_{i+1}^{n+1,m,s}}{h^2}. \end{aligned} \tag{A.15}$$

Here, we denote $\phi_i^{n+1,m,s}$ and $\phi_i^{n+1,m,s+1}$ as current and the new approximations in a Gauss–Seidel iteration.

Since $(\phi_i^{n+1,m,s+1})^3$ is nonlinear, we linearize it at $\phi_i^{n+1,m,s}$, i.e.,

$$(\phi_i^{n+1,m,s+1})^3 \approx (\phi_i^{n+1,m,s})^3 + 3(\phi_i^{n+1,m,s})^2(\phi_i^{n+1,m,s+1} - \phi_i^{n+1,m,s}).$$

Therefore, Eq. (A.15) is rewritten as

$$\begin{aligned}&\left[\frac{1}{\Delta t} + 3\left(\phi_i^{n+1,m,s}\right)^2 + \frac{2\epsilon^2}{h^2}\right]\phi_i^{n+1,m,s+1} \\ &\quad = f_i^n + 2\left(\phi_i^{n+1,m,s}\right)^2 + \epsilon^2\frac{\phi_{i-1}^{n+1,m,s+1} + \phi_{i+1}^{n+1,m,s}}{h^2}.\end{aligned} \tag{A.16}$$

One $SMOOTH$ relaxation operator step consists of solving the system (A.16) for each i on Ω_k grid. After taking ν smoothing steps, we let $\bar{\boldsymbol{\phi}}_k^{n+1,m}$.

Step 2) Compute the defect

$$\boldsymbol{\alpha}_k = \mathbf{f}_k^n - \mathcal{N}_k\bar{\boldsymbol{\phi}}_k^{n+1,m}.$$

Step 3) Restrict the defect and $\bar{\boldsymbol{\phi}}_k^{n+1,m}$

$$\begin{aligned}\boldsymbol{\alpha}_{k-1} =& I_k^{k-1}\boldsymbol{\alpha}_k, \\ \bar{\boldsymbol{\phi}}_{k-1}^{n+1,m} =& I_k^{k-1}\bar{\boldsymbol{\phi}}_k^{n+1,m}.\end{aligned}$$

The restriction operator I_k^{k-1} maps k-level functions to $(k-1)$-level functions.

$$\begin{aligned}\mathbf{d}_{k-1}(i) =& I_k^{k-1}\mathbf{d}_k(i) \\ =& [\mathbf{d}_k(2i) + \mathbf{d}_k(2i-1)]/2,\end{aligned}$$

where $\mathbf{d}_k(i)$ is the i-th component of the vector $\mathbf{d}_k$.

Step 4) Compute the right-hand side

$$\mathbf{f}_{k-1}^{n} = \boldsymbol{\alpha}_{k-1} + \mathcal{N}_{k-1}\bar{\boldsymbol{\phi}}_{k-1}^{n+1,m}.$$

Step 5) Compute an approximate solution $\boldsymbol{\phi}_{k-1}^{n+1,m}$ *of the coarse grid equation on* Ω_{k-1}*, i.e.*

$$\mathcal{N}_{k-1}\boldsymbol{\phi}_{k-1}^{n+1,m} = \mathbf{f}_{k-1}^{n}. \tag{A.17}$$

If $k = 1$, we explicitly invert a 2×2 matrix to obtain the solution. If $k > 1$, we solve (A.17) by performing a FAS k-grid cycle using $\bar{\boldsymbol{\phi}}_{k-1}^{n+1,m}$ as an initial approximation:

$$\hat{\boldsymbol{\phi}}_{k-1}^{n+1,m} = \text{FAScycle}(\bar{\boldsymbol{\phi}}_{k-1}^{n+1,m}, \mathcal{N}_{k-1}, \mathbf{f}_{k-1}^{n}, \nu).$$

Step 6) Compute the coarse grid correction (CGC):

$$\hat{\mathbf{v}}_{k-1}^{n+1,m} = \hat{\boldsymbol{\phi}}_{k-1}^{n+1,m} - \bar{\boldsymbol{\phi}}_{k-1}^{n+1,m}.$$

Step 7) Interpolate the correction:

$$\hat{\mathbf{v}}_{k}^{n+1,m} = I_{k-1}^{k}\hat{\mathbf{v}}_{k-1}^{n+1,m}.$$

Here, the coarse values are simply transferred to the two nearby fine grid points, i.e. $\mathbf{v}_k(2i) = \mathbf{v}_k(2i-1) = I_{k-1}^{k}\mathbf{v}_{k-1}(i) = \mathbf{v}_{k-1}(i)$ for $1 \le i \le p \cdot 2^{k-1}$.

Step 8) Compute the corrected approximation on Ω_k

$$\boldsymbol{\phi}_k^{n+1,m,\text{ after } CGC} = \bar{\boldsymbol{\phi}}_k^{n+1,m} + \hat{\mathbf{v}}_k^{n+1,m}.$$

Step 9) Postsmoothing

$$\boldsymbol{\phi}_k^{n+1,m+1} = SMOOTH^{\nu}(\boldsymbol{\phi}_k^{n+1,m,\text{ after } CGC}, \mathcal{N}_k, \mathbf{f}_k^n).$$

This completes the description of a nonlinear FAScycle.

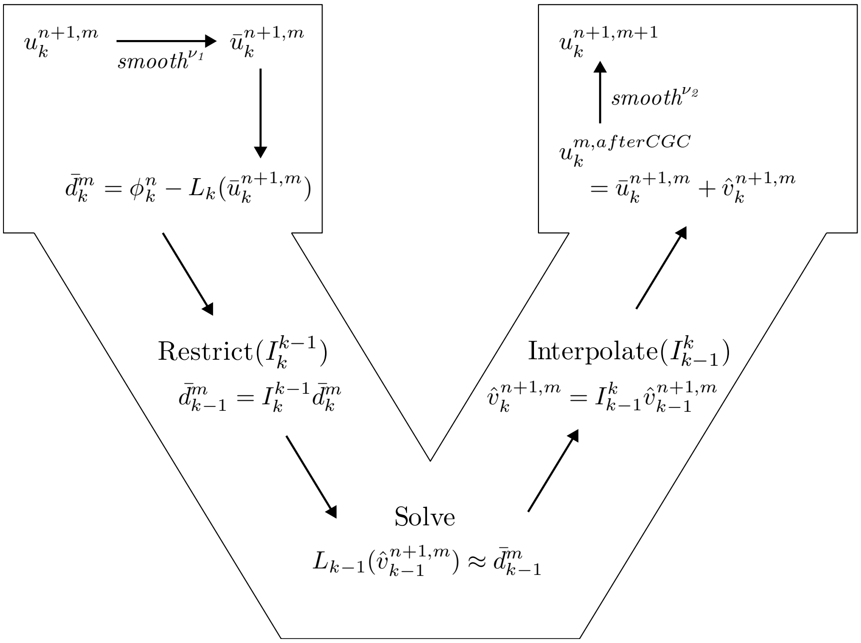

$u_k^{n+1,m}$
$smooth^{\nu_1}$
$\bar{u}_k^{n+1,m}$
$\bar{d}_k^m = \phi_k^n - L_k(\bar{u}_k^{n+1,m})$
Restrict(I_k^{k-1})
$\bar{d}_{k-1}^m = I_k^{k-1}\bar{d}_k^m$
Solve
$L_{k-1}(\hat{v}_{k-1}^{n+1,m}) \approx \bar{d}_{k-1}^m$
Interpolate(I_{k-1}^k)
$\hat{v}_k^{n+1,m} = I_{k-1}^k \hat{v}_{k-1}^{n+1,m}$
$u_k^{m,afterCGC}$
$= \bar{u}_k^{n+1,m} + \hat{v}_k^{n+1,m}$
$smooth^{\nu_2}$
$u_k^{n+1,m+1}$

FIGURE A.4. The MG $(k, k-1)$ two-grid method.

Now, we present the numerical result performed by the algorithm explained above. The initial conditions is $\phi_0(x) = 0.2rand()$ on the space domain $\Omega = (0,1)$, where $rand()$ is the random number uniformly distributed between -1 and 1. We take $h = 1/128$ and $\Delta t = 0.5h$. The number of relaxation is 3. Tolerance is 1.0e-7. Figure A.5 shows temporal evolution of the numerical solutions at $t = 0, 20\Delta t, 40\Delta t, 60\Delta t, 80\Delta t$, and $100\Delta t$.

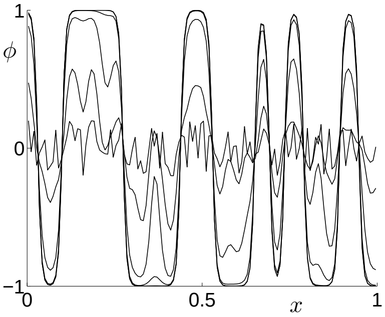


FIGURE A.5. Temporal evolution of the numerical solutions with the initial condition $\phi(x, 0) = 0.2rand()$.

Bibliography

[1] R. Acar, Simulation of interface dynamics: a diffuse-interface model, Vis. Comput. 25 (2009) 101–115.

[2] S.M. Allen and J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta. Metall. 27(6) (1979) 1085–1095.

[3] G.B. Arfken and H.J. Weber, Mathematical Methods for Physicisists, 6th ed., Elsevier, Oxford, 2005.

[4] A.C. Aristotelous, O. Karakashian, and S.M. Wise, A mixed discontinous Galerkin, convex splitting scheme for a modified Cahn–Hilliard equation and an efficient nonlinear multigrid solver, Discrete Continous Dyn. Syst. Ser. B 18 (2013) 2211–2238.

[5] M. Asta, C. Beckermann, A. Karma, W. Kurz, R. Napolitano, M. Plapp, G. Purdy, M. Rappaz, and R. Trivedi, Solidification microstructures and solid-state parallels: Recent developments, future directions, Acta Mater. 57 (2009) 941–971.

[6] O. Axelsson, X. He, and M. Neytcheva, Numerical solution of the time-dependent Navier–Stokes equation for variable density-variable viscousity. Part I, Math. Model. Anal. 20(2) (2015) 232–260.

[7] F. Bassi and S. Rebay, A high-order accuarte discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131(2) (1997) 267–279.

[8] P.W. Bates and J. Jin, Global dynamics of boundary droplets, Discrete Continuous Dyn. Syst. 34 (2013) 1–17.

[9] C. Beckermann, H.-J. Diepers, I. Steinbach, A. Karma, and X. Tong, Modeling melt convection in phase-field simulations of solidification, J. Comput. Phys. 154 (1999) 468–496.

[10] M. Beneš, S. Yazaki, and M. Kimura, Computational studies of non-local anisotropic Allen–Cahn equation, Math. Bohemica 136 (2011) 429–437.

[11] A. Bertozzi, S. Esedoglu, and A. Gillette, Inpainting of binary images using the Cahn–Hilliard equation, IEEE Trans. Image Process 16 (2007) 285–291.

[12] R. Bhargava and H.S. Takhar, Numerical study of heat transfer characteristics of the micropolar boundary layer near a stagnation point on a moving wall, Int. J. Eng. Sci. 38 (2000) 383–394.

[13] W.J. Boettinger, F.S. Biancaniello, and S.R. Coriell, Solutal convection induced marcosegregation and the dendrite to composite transition in off-eutectic alloys, Metall. Trans. A 12 (1981) 321–327.

[14] W.J. Boettinger, J.A. Warren, C. Beckermann, and A. Karma, Phase-field simulation of solidification, Ann. Rev. Mater. Res. 32 (2002) 163–194.

[15] J.F. Brady and A. Acrivos, Steady flow in a channel or tude with an accelerating surface velocity. An exact solution to the Navier–Stokes euqations with reverse flow, J. Fluid Mech. 112 (1981) 127–150.

[16] M. Brassel and B. Bretin, A modified phase field approximation for mean curvature flow with conservation of the volume, Math. Methods Appl. Sci. 34 (2011) 1157–1180.

[17] W.L. Briggs and F. Steve, A multigrid tutorial, SIAM, Philadelphia (1987).

[18] W.L. Briggs and S.F. McCormick, A multigrid tutorial, SIAM, Philadelphia (2000).

[19] L. Bronsard and B. Stoth, Volume-preserving mean curvature flow as a limit of a nonlocal Ginzburg–Landau equation, SIAM J. Math. Anal. 28 (1997) 769–807.

[20] R.L. Burden and J.D. Faires, Numerical Analysis, Cengage Learning, Belmont (2005).

[21] J.W. Cahn and J.E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, J. Chem. Phys. 28(2) (1958) 258–267.

[22] H.D. Ceniceros and A.M. Roma, A nonstiff, adaptive mesh refinement-based method for the Cahn–Hilliard equation, J. Comput. Phys. 225 (2007) 1849–1862.

[23] C.C. Chen, Y.L. Tsai, and C.W. Lan, Adaptive phase field simulation of dendritic crystal growth in a forced flow: 2D vs. 3D morphologies, Int. J. Heat Mass Transfer 52 (2009) 1158–1166.

[24] C.C. Chen and C.W. Lan, Efficient adaptive three-dimensional phase field simulation of free dendritic growth under natural convection, J. Cryst. Growth 312(8) (2010) 1437–1442.

[25] L-Q. Chen, Phase-field models for microstructure evolution, Annu. Rev. Mater. Res. 23 (2002) 113–140.

[26] L. Chen, J. Chen, R.A. Lebensohn, Y.Z. Ji, T.W. Heo, S. Bhattacharyya, and L.Q. Chen, An integrated fast Fourier transform-based phase-field and crystal plasticity approach to model recrystallization of three dimensional poly-crystals, Comput. Methods Appl. Mech. Engrg. 285 (2015) 829–848.

[27] Y. Chen, S.M. Wise, V.B. Shenoy, and J.S. Lowengrub, A stable scheme for a nonlinear, multiphase tumor growth model with an elastic membrane, Int. J. Numer. Meth. Biomed. Engng. 30(7) (2014) 726–754.

[28] J.W. Choi, H.G. Lee, D. Jeong, and J. Kim, An unconditionally gradient stable numerical method for solving Allen–cahn equation, Physica A 388(9) (2009) 1791–1803.

[29] J. Choi, S-K. Park, H-Y. Hwang, and J-Y. Huh, A comparative study of dendritic growth by using the extended Cahn–Hilliard model and the conventional phase-field model, Acta Mater. 84 (2015) 55–64.

[30] Y. Choi, D. Jeong, S. Lee, and J. Kim, Numerical Implementation of the two-dimensional incompressible Navier–Stokes equation, J. KSIAM 9(2) (2015) 103–201.

[31] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[32] M. Copetti and C.M. Elliott, Kinetics of phase decomposition processes: numerical solutions to the Cahn–Hilliard equation, Mater. Sci. Technol. 6 (1990) 273–283.

[33] J.A. Dantzig, Modelling liquid-solid phase changes with melt convection, Int. J. Numer. Methods Engrg. 28(8) (1989) 1769–1785.

[34] J.A. Dantzig and M. Rappaz, Solidification, EPFL press, Lausanne (2009).

[35] J.M. Debierre, A. Karma, F. Celestini, and R. Guérin, Phase-field approach for faceted solidification, Phys. Rev. E 68 (2003) 041604.

[36] M. Dehghan and D. Mizaei, A numerical method based on the boundary integral equation and dual reciprocity methods for one-dimensional Cahn–Hilliard equation, Erg. Anal. Bound. Elem. 33 (2009) 522–528.

[37] C.R. Doering and J.D. Gibbon, Applied analysis of the Navier–Stokes equations, Cambridge University Press, New York, (1995).

[38] J.M. Dorrepaal, An exact solution of the Navier–Stokes equation which describes non-orthogonal stagnation-point flow in two dimensions, J. Fluid Mech. 163 (1986) 141–147.

[39] M. Do-Quang and G. Amberg, Simulation of free dendritic crystal growth in a gravity environment, J. Comput. Phys. 227 (2008) 1772–1789.

[40] B. Drevet, H. Nguyen Thi, D. Camel, B. Billia, and M.D. Dupouy, Solidification of aluminum-lithium alloys near the cell/dendrite transition-influence of solutal convection, J. Cryst. Growth 218 (2000) 419–433.

[41] J. Du, B. Fix, J. Glimm, X. Jia, X. Li, Y. Li, and L. Wu, A simple package for front tracking, J. Comput. Phys. 213 (2006) 613–628.

[42] Q. Du and M. Li, On the stochastic immersed boundary method with an implicit interface formulation, DCDS-B. 15 (2011) 373–389.

[43] Q. Du and R. Nicolaides, Numerical studies of a continuum model of phase transition, SIAM J. Numer. Anal. 28 (1991) 1310–1322.

[44] J.J. Eggleston, G.B. McFadden, and P.W. Voorhees, A phase-field model for highly anisotropic interfacial energy, Phys. D 150 (2001) 91–103.

[45] S. Ei and T. Ishimoto, Dynamics and interations of spikes on smoothly curved boundaries for reaction-diffusion systems in 2D, Jap. J. Ind. Appl. Math. 30(1) (2013) 69–90.

[46] C.M. Elliott and D.A. French, Numerical studies of the Cahn–Hilliard equation for phase separation, IMA J. Appl. Math. 38 (1987) 97–128.

[47] H. Emmerich, The diffuse interface approach in materials science: thermodynamic concepts and applications of phase-field models, Springer-Verlag, Heidelberg (2003).

[48] D.J. Eyre, Systems for Cahn–Hilliard equations, SIAM J. Appl. Math. 53 (1993) 1686–1712.

[49] D.J. Eyre, An unconditionally stable one-step scheme for gradient systems, unpublished paper, http://www.math.utah.edu/∼eyre/research/methods/stable.ps (1998).

[50] D.J. Eyre, Unconditionally gradient stable time marching the Cahn–Hilliard equation, MRS Proceedings 529 Cambridge University Press (1998).

[51] G.J. Fix, Phase field methods for free boundary problems; B. Fasano and M. Primicerio (Eds.) Free boundary problems: theory and applications, Pitman, London (1983).

[52] H. Garcke, T. Preusser, M. Rumpf, A. Telea, U. Weikard, and J. van Wijk, A phase field model for continuous clustering on vector fields, IEEE Trans. Visual. Comput. Graph. 7 (2001) 230–241.

[53] U. Ghia, K.N. Ghia, and C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48(3) (1982) 387–411.

[54] M.E. Glicksman, M.B. Koss, and E.A. Winsa, Dendritic growth velocities in microgravity, Phys. Rev. Lett. 73 (1994) 573–576.

[55] E. Harlow and J. Welch, Numerical calculation of time dependent viscous incompressible flow with free surface, Phys. Fluids 8 (1965) 2182–2189.

[56] C.R. Hird and B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.

[57] D.W. Hoffman and J.W. Cahn, A vector thermodynamics for anisotropic surfaces: I. Fundamentals and application to plane surface junctions, Surf. Sci. 31 (1972) 368–388.

[58] T.Y. Hou, J.S. Lowengrub, and M.J. Shelley, Boundary integral methods for multicomponent fluids and multiphase materials, J. Comput. Phys. 169 (2001) 302–362.

[59] S.C. Huang and M.E. Glicksman, Overview 12: Fundamentals of dendritic growth solidification–I. Steady-state tip growth, Acta Metall. 29(5) (1981) 701–715.

[60] G. Huisken, Flow by mean curvatre of convex surfaces into spheres, J. differential Geom. 20 (1984) 237–266.

[61] J.K. Hunter and B. Nachtergaele, Applied Analysis, World Scientific, Singapore (2001).

[62] D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, J. Comput. Phys. 155 (1999) 96–127.

[63] D. Jeong, J. Shin, Y. Li, Y. Choi, J.-H. Jung, S. Lee, and J. Kim, Numerical analysis of energy-minimizing wavelengths of equilibrium states for diblock copolymers, Curr. Appl. Phys. 14 (2014) 1263–1272.

[64] D. Jeong, S. Lee, Y. Choi, and J. Kim, Energy-minimizing wavelengths of equilibrium states for diblock copolymers in the hex-cylinder phase, Curr. Appl. Phys. 15 (2015) 799–804.

[65] D. Jeong, S. Lee, D. Lee, J. Shin, and J. Kim, Comparison study of numerical methods for solving the Allen–Cahn equation, Comput. Mater. Sci. 111 (2016) 131–136.

[66] J.-H. Jeong, J.A. Dantzig, and N. Goldenfeld, Dendritic growth with fluid flow in pure materials, Metall. Mater. Trans. A 34A (2003) 459–466.

[67] J.-H. Jeong, N. Goldenfeld, and J.A. Dantzig, Phase field model for three dimensional dendritic growth with fluid flow, Phys. Rev. E 64 (2001) 041602.

[68] J.P. Kallungal and A.J. Barduhn, Growth rate of an ice crystal in subcooled pure water, AlChE J. 23(3) (1977) 294–303.

[69] S. Karagadde, A. Bhattacharya, G. Tomar, and P. Dutta, A coupled VOF-IBM-enthalpy approach for modeling motion and growth of equiaxed dendrities in a solidifying melt, J. Comput. Phys. 231 (2012) 3987–4000.

[70] N. Khiari, T. Achouri, M.L. Ben Mohamed, and K. Omrani, Finite difference approximation solutions for the Cahn–Hilliard equation, Numer. Methods Partial Differ. Equ. 23 (2007) 437–455.

[71] J. Kim, An augmented projection method for the incompressible Navier–Stokes equations in arbitrary domains, Int. J. Comput. M. 2(2) (2005) 201–212.

[72] J. Kim, A continuous surface tension force formulation for diffuse-interface models, J. Comput. Phys. 204 (2005) 784–804.

[73] J. Kim, A diffuse-interface model for axisymmetric immiscible two-phase flow, Appl. Math. Comput. 160 (2005) 589–606.

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

[75] J. Kim, Phase-field models for multi-component fluid flows, Comm. Comput. Phys. 12 (2012) 613–661.

[76] J. Kim and H.O. Bae, An unconditionally stable adaptive mesh refinement for Cahn–Hilliard equation, J. Korean Phys. Soc. 53 (2008) 672–679.

[77] J. Kim, K.K. Kang, and J.S. Lowengrub, Conservative multigrid methods for Cahn–Hilliard fluids, J. Comput. Phys. 193 (2004) 511–543.

[78] J. Kim, S. Lee, and Y. Choi, A conservative Allen–Cahn equation with a space-time dependent Lagrange multiplier, Int. J. Eng. Sci 84 (2014) 11–17.

[79] J. Kim and J.S. Lowengrub, Phase field modeling and simulation of three-phase flows, Interfaces Free Bound. 7 (2005) 435–466.

[80] J. Kim and P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equation, J. Comput. Phys. 59(2) (1985) 308–323.

[81] J.W. Kim, D.J. Kim, and H.C. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132–150.

[82] J.Y. Kim, J.K. Yoon, and P.R. Cha, Phase-field model of a morphological instability caused by elastic non-equilibrium, J. Korean Phys. Soc. 49 (2006) 1501–1509.

[83] B.J. Kirby, Micro- and nanoscale fluid mechanics: Trandport in microfluid devices, Cambridge University Press, New York, (2001).

[84] A.J. Kurdila and M. Zabarankin, Convex Functional Analysis, Springer, Heidelberg (2006).

[85] J.S. Langer, Instabilities and pattern formation in crystal growth, Rev. Mod. Phys. 52(1) (1980) 1–28.

[86] J.S. Langer, Models of pattern formation in first-order phase transitions; G. Grinstein and G. Mazenko (Eds.) Directions in condensed matter physics, World Scienfitic, Singapore (1986).

[87] D. Lee, J.Y. Huh, D. Jeong, J. Shin, A. Yun, and J. Kim, Physical, mathematical, and numerical derivations of the Cahn–Hilliard equation, Comput. Mather. Sci. 81 (2014) 216–225.

[88] H.G. Lee and J. Kim, A comparison study of the Boussinesq and the variable density models on buoyancy-driven flows, J. Eng. Math. 75 (2012) 15–27.

[89] S. Lee, C. Lee, H.G. Lee, and J. Kim, Comparison of different numerical schemes for the Cahn–Hilliard equation, J. KSIAM 17(3) (2013) 197–207.

[90] S. Lee, Y. Choi, D. Lee, H.-K. Jo, S. Lee, S. Myung, and J. Kim, A modified Cahn–Hilliard equation for 3D volume reconstruction from two planar cross sections, J. KSIAM 19(1) (2015) 47–56.

[91] S. Lee, D. Jeong, W. Lee, and J. Kim, An immersed boundary method for a contractile elastic ring in a three-dimensional Newtonian fluid, J. Sci. Comput. in press, DOI: 10.1007/s10915-015-0110-8.

[92] L.P. Lebedev and M.J. Cloud, Tensor analysis, World Scientific, Singapore, (2003).

[93] R.J. Leveque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1001–1025.

[94] Y. Li, D. Jeong, J. Choi, S. Lee, and J. Kim, Fast local image inpainting based on the local Allen–Cahn model, Digit. Signal Process. 37 (2015) 65–74.

[95] Y. Li. H.G. Lee, D. Jeong, and J. Kim, An unconditionally stable hybrid numerical method for solving Allen–Cahn equation, Comput. Math. Apple. 60 (2010) 1591–1606.

[96] Y. Li, H.G. Lee, and J. Kim, A fast, robust, and accurate operator splitting method for phase-field simulations of crystal growth, J. Cryst. Growth 321 (2011) 176–182.

[97] Y. Li and J Kim, phase-field simulations of crystal growth with adaptive mesh refinement, Int. J. Heat Mass Trans. 55 (2012) 7929–7932.

[98] Y. Li, A. Yun, D. Lee, J. Shin, D. Jeong, and J. Kim, Three-dimensional volume-conserving immersed boundary model for two-phase fluid flows, Comput. Methods Appl. Mech. Engrg. 257 (2013) 36–46.

[99] J. Lighthill, Waves in fluids, Cambridge Universigy Press, Cambridge, (1978).

[100] F. Liu and H. Metiu, Dynamics of phase-separation of crystal-surfaces, Phys. Rev. B 48 (1993) 5808–5817.

[101] C. Liu and J. Shen, A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method, Physica D 179 (2003) 21–228.

[102] S. L$\ddot{u}$ and Q. Lu, A linear discrete scheme for the Ginzburg–Landau equation, Int. J. Comput. Math. 85(5) (2008) 745–758.

[103] J. Ma, Y. Jiang, and K. Xiang, Numerical simulation of blowup in nonlocal reaction-diffusion equations using a moving mesh method, J. Comput. Appl. Math. 230 (2009) 8–21.

[104] Z. Min, J. Shen, Z. Feng, L. Wang, L. Wang, and H. Fu, Effects of melf flow on the primary dendrite spacing of Pb-Sn binary alloy during directional solidification, J. Cryst. Growth 320 (2011) 41–45.

[105] W.U. Mirihanage, L. Arnberg, and R.H. Mathiesen, In-situ observation of transient columnar dendrite growth in the presence of thermo-solutal convection, IOP C. Ser. Mater. Sci. Engrg. 33(1) (2012) 012033.

[106] K.W. Morton and D. Mayers, Numerical solution of partial differential equations: An introduction (2nd ed.), Cambridge University Press, Cambridge, (2005).

[107] W.W. Mullilns and R.F. Skerka, Morphological stability of a paticle growing by diffusion or heal flow, J. Appl. Phys. 34(2) (1963) 323–329.

[108] K. Murakami, T. Fujiyama, A. Koike, and T. Okamoto, Influence of melt flow on the growth directions of columnar grains and columnar dendrities, Acta Metall. 31 (1983) 1425–1432.

[109] M. Nagayama, L. Ueda, and M. Yadome, Numerical approach to transient dynamics of oscillatory pulsed in a bistable reaction-diffusion system, Jap. J. Ind. Appl. Math. 27(2) (2010) 295–322.

[110] S.J. Osher and R.P. Fedkiw, Level set methods and dynamic implicit surfaces, Springer Verlag, New York, (2003).

[111] M. Rappaz and W. Kurz, Dendrites solidified by computer, Nature 375 (1995) 103.

[112] X. Ren and J. Wei, On a phase field problem driven by interface area and interface curvature, Eur. J. Appl. Math. 20 (2009) 531–556.

[113] R. Rojas, T. Tokaki, and M. Ohno, A phase-field-lattice Boltzmann method for modeling motion and growth of a dendritic for binary alloy solidification in theh presence of melt convection, J. Comput. Phys. 298 (2015) 29–40.

[114] J. Rosam, P.K. Jimack, and A. Mullis, A fully implicit, fully adaptive time and space discretisation method for phase-field simulation of binary alloy solidification, J. Comput. Phys. 225 (2007) 1271–1287.

[115] J. Rubinstein and P. Sternberg, Nonlocal reaction-diffusion equations and nucleation, IMA J. Appl. Math 48 (1992) 249–264.

[116] J.S. Sethian, Level set methods and fast marching methods evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, Cambridgy University Press, Cambridgy, (1999).

[117] J.S. Sethian and Y. Shan, Solving partial differential equations on irregular domains with moving interfaces, with applications to superconformal electrodeposition in semiconductor manufacturing, J. Comput. Phys. 227 (2008) 6411–6447.

[118] Y. Shi and X.P. Wang, Modeling and simulation of dynamics of three component flows on solid surface, Jap. J. Ind. Appl. Math. 31(3) (2014) 611–631.

[119] D.Stanfford, M.J. Ward, and B. Wetton, The dynamics of drops and attached interfaces for the constrained Allen–Cahn equation, Eur. J. Appl. Math. 12 (2001) 1–24.

[120] I. Steinbach, Why solidification? Why Phase-field?, J. Miner. 65 (2013) 1096–1102.

[121] I. Steinbach, F. Pezzolla, B. Nestler, M. Seeßelberg, R. Prieler, G.J. Schmitz, and J.L.L. Rezende, A phase field concept for multiphase system, Phys. D 94 (1996) 135–147.

[122] G. Strang, Introduction to applied mathematics, Wellesley-Cambridgy Press, Wellesley, (1986).

[123] A. Stuart and A.R. Humphries, Dynamical systems and numerical analysis, Cambridge University Press, Cambridge, (1998).

[124] T. Takaki, Phase-field modeling and simulations of dendrite growth, ISIJ Int. 54 (2014) 437–444.

[125] C. Taylor and P. Hood, A numerical solution of the Navier–Stokes equations using the finite element technique, Comput. Fluids 1(1) (1973) 73–100.

[126] T. Temam, Navier–Stokes equations: theory and numerical analaysis, American Mathematical Society, New York, (2001).

[127] X. Tong, C. Beckermann, A. Karma, and Q. Li, Phase-field simulations of dendritic crystal growth in a forced flow, Phys. Rev. E 63 (2001) 061601.

[128] R. Tönhardt and G. Amberg, Dendritic growth of randomly oriented nuclei in a shear flow, J. Cryst, Growth 213 (2000) 161–187.

[129] U. Trottenberg, C. Oosterlee, and A. Schüller, Multigrid, Academic press, London, (2001).

[130] S.O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.

[131] C.Y. Wang, Flow due to a stretching boundary with partial slip–and exact solution of the Navier–Stokes equations, Chem. Eng. Sci. 57(17) (2002) 3745–3747.

[132] M.J. Ward, Metastable bubble solutions for the Allen–Cahn equation with mass conservation, SIAM. J. Appl. Math 56 (1996) 1247–1279.

[133] A.A. Wheeler, W.J. Boettinger, and G.B. McFadden, Phase-field model for isothermal phase transitions in binary alloys, Phys. Rev. A 45(10) (1992) 7424–7439.

[134] S.M. Wise, J.S. Lowengrub, J.S. Kim, K. Thornton. P.W. Voorhees, and W.C. Johnson, Quantum dot formation on a strain-patterned epitaxial thin film, Appl. Phys. Lett. 87 (2005) 133102.

[135] S.W. Wise, C. Wang, and J.S. Lowengrub, An energy-stable and convergent finite-difference scheme for the phase field crystal equation, SIAM J. Numer. Anal. 47(3) (2009) 2269–2288.

[136] Q. Yang, B.Q. Li, J. Shao, and Y. Ding, A phase field numerical study of 3D bubble rising in viscous fluids under an electric field, Int. J. Heat Mass Tran. 78 (2014) 820–829.

[137] X. Yang, J.J. Feng, C. Liu, and J. Shen, Numerical simulations of jet pinching-off and drop formationn using an energetic variational phase-field method, J. Comput. Phys. 218 (2006) 417–428.

[138] P. Yue, Y. Xu, and C.W. Shu, Application of the local discontinous Galerkin method for the Allen–Cahn/Cahn–Hilliard system, Commun. Comput. Phys. 5 (2009) 821–235.

[139] L. Zhang, Long time behavior of differnce approximations for the two-dimensional complex Ginzburg–Landau equation, Numer. Func. Anal. Optim. 31 (2010) 1190–1211.

[140] T. Zhang and Q. Wang, Cahn–Hilliard vs singular Cahn–Hilliard in phase field modeling, Commun. Comput. Phys. 7 (2010) 91–103.

[141] Z. Zhang and H. Tang, An adaptive phase field method for the mixture of two incompressible fluids, Comput. Fluids 36 (2007) 1307–1318.

[142] S. Zhao and G.W. Wei, A unified discontinuous Galerkin framework for time integration, Math. Methods Appl. Sci. 37(7) (2013) 1042–1071.

[143] M. Zhu, D. Sun, S. Pan, Q. Zhang, and D. Raabe, Modelling of dendritic growth during alloy solidification under natural convection, Model. Simul. Sci. Engrg. 22(3) (2014) 034006.