Phase-field models for multi-phase fluid flows

by

Hyun Geun Lee

A dissertation submitted in partial satisfaction
of the requirements for the degree of
Doctor of Philosophy
in
Mathematics
in the
GRADUATE DIVISION
of the
KOREA UNIVERSITY

Approved:

Committee Member 1

Committee Member 2

Committee Member 3

Committee Member 4

Committee Member 5

Committee in Charge

2011

Contents

Abstract

Multi-phase, multi-component fluid flows are used in a wide variety of industrial and scientific applications. Problems involving changes in the topology of interfaces between fluids are difficult to study both from the physical and computational points of view since the fluids are complex with the density, viscosity, diffusivity, and surface tension. In order to investigate the complex interactions, we develop numerical algorithms and perform numerical simulations. In particular, the methods allow us to solve more quickly, efficiently, and accurately the important equations in hydrodynamics: the Cahn–Hilliard, Allen–Cahn, and Navier–Stokes equations.

To model multi-component fluid flows, we consider the vector valued phase-field, velocity, and pressure which are governed by the N-component advective Cahn–Hilliard and modified Navier–Stokes equations. For the N-component Cahn–Hilliard equation, we present a practically unconditionally gradient stable conservative nonlinear numerical scheme. The scheme is based on a nonlinear splitting method and is solved by an efficient and accurate nonlinear multigrid method. And the scheme allows us to convert the N-component Cahn–Hilliard equation into a system of $N-1$ binary Cahn–Hilliard equations and significantly reduces the required computer memory and CPU time. The Cahn–Hilliard–Navier–Stokes equations can be applied to any number of fluid components. And, to investigate the buoyancy driven interpenetration of fluids

with different densities, we implement a time-dependent pressure boundary condition through a time-dependent density field at the boundary. Due to the pressure boundary treatment, we can perform long time evolutions resulting in an equilibrium state.

We present an unconditionally stable second-order hybrid numerical method for solving the Allen–Cahn equation. The proposed method is based on operator splitting techniques. The Allen–Cahn equation was divided into a linear and a nonlinear equation. First, the linear equation was discretized using an implicit Euler's scheme and the resulting discrete system of equations was solved by a multigrid method. The nonlinear equation was then solved analytically due to the availability of a closed-form solution. In particular, we apply this technique to dendritic growth simulation. A great challenge in the simulation with various supercoolings is the large difference in time and length scales. This introduces a severe time step restriction for stability. The proposed scheme allows the use of a sufficiently large time step without the technical limitations.

Acknowledgments

Throughout my graduate studies I have been supported by a large number of people.

First I would like to thank my advisor, Junseok Kim, for his guidance and teaching as well as the opportunity to work on fun and exciting projects.

Also, I offer my regards and blessings to all of those who supported me in any respect during the completion of this thesis. And I consider myself very lucky to work in my research group.

Finally, I am indebted to my family for their support and encouragement.

Chapter 1

Introduction

Multi-phase or multi-component fluid flows are used in a wide variety of applications such as extractors [74], polymer-dispersed liquid crystals [75], polymer blends [76], reactors [77], separators [78], sprays [79], and microfluidic technology [80, 81]. Problems involving changes in the topology of interfaces between fluids are difficult to study both from the physical and computational points of view since the fluids are complex with the density, viscosity, diffusivity, and surface tension. For this reason, mathematical modeling and numerical simulation of multi-phase or multi-component flows is a great challenge.

There are two main approaches to simulate multi-phase or multi-component flows: one is an interface tracking and the other is an interface capturing. In interface tracking methods (front-tracking [82, 25], immersed boundary [83, 84], and volume-of-fluid [83, 85, 86]), Lagrangian particles are used to track the interface and are advected by the velocity field. In interface capturing methods (level-set [87, 88, 40, 89] and phase-field [72, 65, 56]), the interface is implicitly captured by a contour of a particular scalar function.

The equations governing the motion of unsteady, viscous, incompressible, immiscible two-fluid system are the Navier-Stokes equations (the subscript i denotes the ith

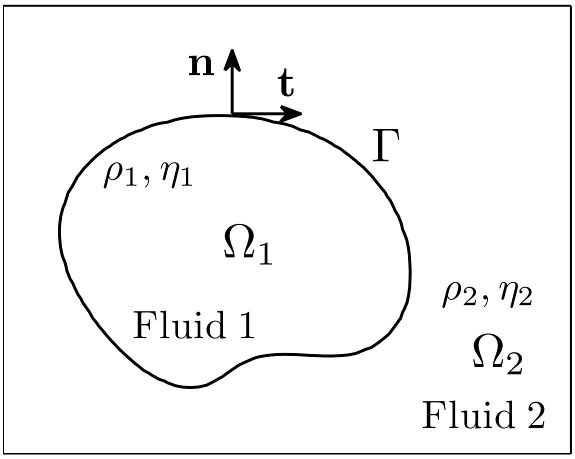


FIGURE 1.1. Schematic of a two-phase domain.

fluid component):

$$\begin{aligned} \rho_i \left(\frac{\partial \mathbf{u}_i}{\partial t} + \mathbf{u}_i \cdot \nabla \mathbf{u}_i \right) &= -\nabla p_i + \nabla \cdot [\eta_i (\nabla \mathbf{u}_i + \nabla \mathbf{u}_i^T)] + \rho_i \mathbf{g} \ \text{ in } \Omega_i, \\ \nabla \cdot \mathbf{u}_i &= 0 \ \text{ in } \Omega_i, \end{aligned}$$

where ρ_i is the density, $\mathbf{u}_i$ is the velocity, p_i is the pressure, and η_i is the viscosity of fluid $i = 1, 2$, the superscript T denotes the transpose, and $\mathbf{g}$ is the gravitational force. A schematic of a two-phase domain is shown in Figure 1.1. Γ is the interface between two fluids, $\mathbf{n} = (n_1, n_2)$ is the outward unit normal vector to the interface, and $\mathbf{t} = (t_1, t_2)$ is the unit tangent vector to the interface.

In fluid flow problems with moving interfaces, part of the interface of the computational domain is unknown and must be determined as part of the solution. On the unknown interface, the boundary conditions should be satisfied. The dynamic condition requires that the forces acting on the fluid at the interface be in equilibrium (momentum conservation at the interface). This means that the normal forces on either side of the interface are of equal magnitude and opposite direction, while the forces in the tangential direction are of equal magnitude and direction (neglecting the surface tension):

$$(\mathbf{n}\cdot\tau)_1\cdot\mathbf{n} = -(\mathbf{n}\cdot\tau)_2\cdot\mathbf{n} \quad \text{and} \quad (\mathbf{n}\cdot\tau)_1\cdot\mathbf{t} = (\mathbf{n}\cdot\tau)_2\cdot\mathbf{t},$$

where τ is the stress tensor (including pressure terms). The kinematic condition requires that the interface be a sharp boundary separating the two fluids that allows no flow through it. This states that the components of the velocity normal (or tangential) to the interface are equal for two fluids:

$$\mathbf{u}_1\cdot\mathbf{n} = \mathbf{u}_2\cdot\mathbf{n} \quad \text{and} \quad \mathbf{u}_1\cdot\mathbf{t} = \mathbf{u}_2\cdot\mathbf{t}.$$

Consider the effect of surface tension on a fluid interface. From the surface stress boundary condition at an interface Γ between two fluids, we have the normal jump condition

$$p_1 - p_2 + \sigma\kappa = \left(2\eta_1 n_k \frac{\partial u_k}{\partial n}\right)_1 - \left(2\eta_2 n_k \frac{\partial u_k}{\partial n}\right)_2,$$

where σ is the fluid surface tension coefficient and κ is the local surface curvature, $R_1^{-1} + R_2^{-1}$, where R_1 and R_2 are the principle radii of curvature of the surface. The above equation implies that both the surface tension and viscous stress at the interface contribute to the pressure jump.

For incompressible and immiscible multi-component fluid flows, the governing equations can be written as

$$\begin{aligned} \rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) &= -\nabla p + \nabla\cdot[\eta(\nabla\mathbf{u} + \nabla\mathbf{u}^T)] + \mathbf{SF}_{sing} + \rho\mathbf{g}, && (1.1) \\ \nabla\cdot\mathbf{u} &= 0 \text{ in } \Omega = \Omega_1\cup\Omega_2, && (1.2) \end{aligned}$$

where $\rho(\mathbf{x},t)$ is the density, $\mathbf{u}(\mathbf{x},t)$ is the velocity, $p(\mathbf{x},t)$ is the pressure, $\eta(\mathbf{x},t)$ is the viscosity, $\mathbf{SF}_{sing} = -\sigma\kappa\delta_\Gamma\mathbf{n}$ is the singular surface tension force (δ_Γ is the surface delta function).

1.1. Phase-field models

Phase-field models are an increasingly popular choice for modeling the motion of multi-phase or multi-component fluid flows [1]. In the phase-field model, sharp fluid interfaces are replaced by thin but nonzero thickness transition regions where the interfacial forces are smoothly distributed [2]. The basic idea is to introduce a conserved order parameter (e.g., mass concentration) that varies continuously over thin interfacial layers and is mostly uniform in the bulk phases (see Fig. 1.2).

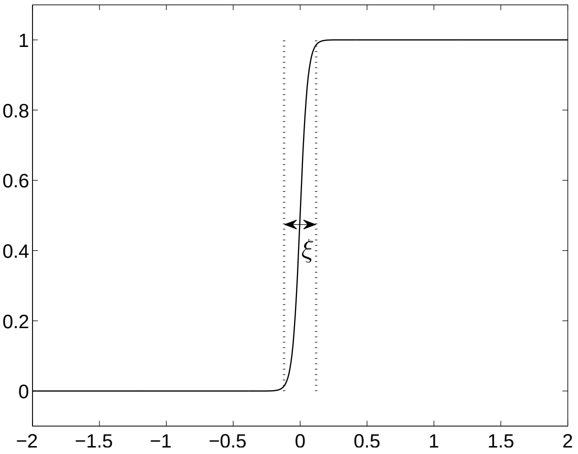


FIGURE 1.2. A concentration profile across an interface with thickness, ξ.

For density-matched binary liquids (let $\rho = 1$ for simplicity), the coupling of the convective Cahn-Hilliard equation for the mass concentration with a modified momentum equation that includes a phase field-dependent surface force is known as Model H [60]. In the case of fluids with different densities a phase-field model has been proposed by Lowengrub and Truskinovsky [61]. Complex flow morphologies and topological transitions such as coalescence and interface break-up can be captured naturally and in a mass-conservative and energy-dissipative fashion since there is an associated free energy functional. The phase field is governed by the following advective Cahn-Hilliard equation:

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = \nabla \cdot [M(c)\nabla(F'(c) - \epsilon^2 \nabla c)],$$

where $M(c) = c(1-c)$ is the mobility, $F(c) = \frac{1}{4}c^2(1-c)^2$ is a Helmholtz free energy that describe the coexistence of immiscible phases, and ϵ is a measure of interface thickness and $\epsilon \sim \xi$ (see Fig. 1.2). It can be shown that in the sharp interface limit $\epsilon \to 0$, the classical Navier-Stokes system equations and jump conditions are recovered [61]. The singular surface tension force is $F_{sing} = -6\sqrt{2}\tau\epsilon\nabla \cdot (\nabla c \otimes \nabla c)$, where τ is the surface tension coefficient [61]. An alternative surface tension force formulation based on the CSF is $F_{sing} = -6\sqrt{2}\tau\epsilon\nabla \cdot \left(\frac{\nabla c}{|\nabla c|}\right)|\nabla c|\nabla c$, with which one can calculate pressure field directly from the momentum equations.

The advantages of the phase-field method are: (1) topology changes are automatically described; (2) the composition field c has a physical meaning not only on near interface but also in the bulk phases; (3) complex physics can easily be incorporated into the framework, the methods can be straightforwardly extended to multicomponent systems, and miscible, immiscible, partially miscible, and lamellar phases can be modeled. For example, the binary model can be straightforwardly extended to describe three component flows as follows. Consider a ternary mixture and denote the composition of components 1, 2, and 3, expressed as mass fractions, by c_1, c_2, and c_3, respectively. Because the concentrations sum to unity, only two of them need to be determined, say c_1, c_2. The evolution of c_1 and c_2 is governed by the following advective ternary Cahn-Hilliard equation:

$$\frac{\partial c_1}{\partial t} + \mathbf{u} \cdot \nabla c_1 = \nabla \cdot \left[M(c_1, c_2)\nabla\left(\frac{F(c_1, c_2)}{\partial c_1} - \epsilon^2 \nabla c_1 - 0.5\epsilon^2 \nabla c_2\right)\right],$$

$$\frac{\partial c_2}{\partial t} + \mathbf{u} \cdot \nabla c_2 = \nabla \cdot \left[M(c_1, c_2) \nabla \left(\frac{F(c_1, c_2)}{\partial c_2} - 0.5\epsilon^2 \nabla c_1 - \epsilon^2 \nabla c_2 \right) \right],$$

where $M(c_1, c_2) = \sum_{i<j}^{3} c_i c_j$ is the mobility, $F(c_1, c_2) = 1/4 \sum_{i<j}^{3} c_i^2 c_j^2$ is the Helmholtz free energy that can be used to model the three immiscible mixtures, and $c_3 = 1 - c_1 - c_2$. In the ternary fluid flows, the singular surface tension force in the momentum should be changed by is $F_{sing} = -6\sqrt{2} \sum_{i=1}^{3} \tau_i \nabla \cdot (\nabla c_i \otimes \nabla c_i)$, where the physical surface tension coefficients τ_{ij} between two fluids i and j are decomposed into the phase specific surface tensions τ_i such that $\tau_{ij} = \tau_i + \tau_j$.

Associated with diffuse-interfaces is a small scale ϵ, proportional to the width of the interface. In real physical systems describing immiscible fluids, ϵ can be vanishingly small. However, for numerical accuracy ϵ must be at least a few grid lengths in size. This can make computations expensive. One way of ameliorating this problem is to adaptively refine the grid only near the transition layer. Such methods are under development by various research groups.

Phase-field methods have been used to model viscoelastic flow, thermocapillary flow, spinodal decomposition, the mixing and interfacial stretching, in a shear flow, droplet breakup process, wave-breaking and sloshing, the fluid motion near a moving contact line, and the nucleation and annihilation of an equilibrium droplet (see the references in the review paper [1]).

Chapter 2

A binary fluid

We consider the incompressible flows of two immiscible fluids (1 and 2) of different density and viscosity. The phase-field is defined to be either the difference between [16] or the fraction of one of [72] the concentrations of the two mixtures.

$$\phi = \frac{m_1 - m_2}{m_1 + m_2} \quad \text{or} \quad c = \frac{m_1}{m_1 + m_2}, \tag{2.1}$$

where m_1 and m_2 are the masses of the fluids 1 and 2. We note that $-1 \leq \phi \leq 1$ and $0 \leq c \leq 1$. Throughout this thesis, we will use ϕ and c as phase-fields defined by Equation (2.1). The phase-field is also called by an order parameter [63]. The governing equation for the phase-field is the advective Cahn–Hilliard (CH) equation:

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = \nabla \cdot (M(c) \nabla \mu), \tag{2.2}$$

$$\mu = F'(c) - \epsilon^2 \Delta c, \tag{2.3}$$

where $\mathbf{u}$ is the bulk velocity and $M(c)$ is the phase-field dependent mobility. $M(c) = c(1-c)$ is used in [72], $M(\phi) = 1-\phi^2$ is used in [55], and $M(c) = 1$ is used in [6]. When we have a constant mobility, the CH dynamics is controlled by bulk diffusion. When we have a variable mobility, the bulk diffusion is severely reduced, which corresponds to the interface-diffusion-controlled dynamics, i.e., the coarsening process is mainly due to the diffusion along the interface between the two phases. $F(c) = 0.25c^2(1-c)^2$ is the Helmholtz free energy (see Figure 2.1). ϵ is a positive constant, which is related to the thickness of the interfacial transition of the phase-field.

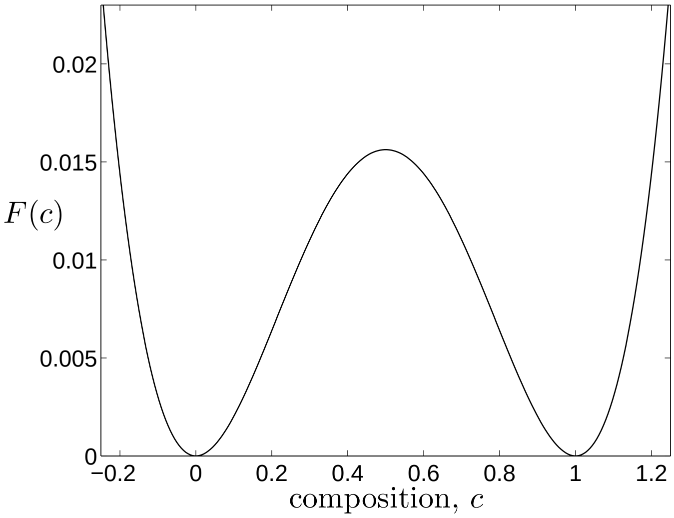


FIGURE 2.1. A double well potential, $F(c) = 0.25c^2(1-c)^2$.

The CH equation without the flow arises from the Ginzburg–Landau free energy

$$\mathcal{E}(c) := \int_{\Omega} \left(F(c) + \frac{\epsilon^2}{2} |\nabla c|^2 \right) d\mathbf{x}, \tag{2.4}$$

where Ω is the region of space occupied by the system. To obtain the CH equation with a variable mobility, one introduces a chemical potential μ as the variational derivative of $\mathcal{E}$,

$$\mu := \frac{\delta \mathcal{E}}{\delta c} = F'(c) - \epsilon^2 \Delta c$$

and defines the flux, $\mathcal{J} := -M(c)\nabla\mu$. As a consequence of mass conservation, we have

$$\frac{\partial c}{\partial t} = -\nabla \cdot \mathcal{J},$$

which is the CH equation with a variable mobility. The natural and no-flux boundary conditions are

$$\frac{\partial c}{\partial \mathbf{n}} = \mathcal{J} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \text{ where } \mathbf{n} \text{ is normal to } \partial\Omega. \tag{2.5}$$

We differentiate the energy $\mathcal{E}$ and the total mass $\int_\Omega c\ d\mathbf{x}$ to get

$$\begin{aligned}\frac{d}{dt}\mathcal{E}(t) &= \int_\Omega (F'(c)c_t + \epsilon^2 \nabla c \cdot \nabla c_t)\ d\mathbf{x} = \int_\Omega \mu c_t\ d\mathbf{x} = \int_\Omega \mu \nabla \cdot (M(c)\nabla\mu)\ d\mathbf{x} \\ &= \int_{\partial\Omega} \mu M(c)\frac{\partial \mu}{\partial \mathbf{n}}\ ds - \int_\Omega \nabla\mu \cdot (M(c)\nabla\mu)\ d\mathbf{x} = -\int_\Omega M(c)|\nabla\mu|^2\ d\mathbf{x} \le 0\end{aligned}$$

and

$$\frac{d}{dt}\int_\Omega c\ d\mathbf{x} = \int_\Omega c_t\ d\mathbf{x} = \int_\Omega \nabla \cdot (M(c)\nabla\mu)\ d\mathbf{x} = \int_{\partial\Omega} M(c)\frac{\partial\mu}{\partial\mathbf{n}}\ ds = 0,$$

where we used the no-flux boundary condition (2.5). Therefore, the total energy is non-increasing in time and the total mass is conserved. If we put all the equations (1.1), (1.2), (2.2), and (2.3) together, then we have

$$\rho(c)(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \nabla \cdot [\eta(c)(\nabla\mathbf{u} + \nabla\mathbf{u}^T)] + \mathbf{SF} + \rho(c)\mathbf{g}, \tag{2.6}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2.7}$$

$$c_t + \mathbf{u}\cdot\nabla c = \nabla \cdot (M(c)\nabla\mu), \tag{2.8}$$

$$\mu = F'(c) - \epsilon^2 \Delta c. \tag{2.9}$$

Singular force $\mathbf{SF}_{sing}$ becomes a regular force $\mathbf{SF}$ in terms of phase-field.

2.1. Variable density and viscosity

The density and viscosity are linear functions of the phase-field [55, 72]. Let ρ_1 and ρ_2 be the densities of each fluid in the mixture, then the variable density of the mixture is defined as

$$\rho(c) = \rho_1 c + \rho_2(1-c) \quad \text{or} \quad \rho(\phi) = \rho_1 \frac{1+\phi}{2} + \rho_2 \frac{1-\phi}{2}.$$

Similarly, the variable viscosity of the mixture is defined as

$$\eta(c) = \eta_1 c + \eta_2(1-c) \quad \text{or} \quad \eta(\phi) = \eta_1 \frac{1+\phi}{2} + \eta_2 \frac{1-\phi}{2}.$$

Then, if we use the linear interpolation, the density and viscosity change across the interface with the same profiles to the scaled phase-field function. In [16], the harmonic interpolation for the variable density and viscosity is used.

$$\frac{1}{\rho(\phi)} = \frac{1+\phi}{2\rho_1} + \frac{1-\phi}{2\rho_2} \quad \text{and} \quad \frac{1}{\eta(\phi)} = \frac{1+\phi}{2\eta_1} + \frac{1-\phi}{2\eta_2}.$$

For the density, the harmonic interpolation is used and the linear interpolation is used for the viscosity in [7].

$$\frac{1}{\rho(\phi)} = \frac{1+\phi}{2\rho_1} + \frac{1-\phi}{2\rho_2} \quad \text{and} \quad \eta(\phi) = \frac{1+\phi}{2}\eta_1 + \frac{1-\phi}{2}\eta_2.$$

One of the reasons to choose the harmonic interpolation is that the solution of the Cahn-Hilliard equation does not satisfy the maximal principle [16]. Hence, the linear interpolation can not be guaranteed to be bounded away from zero. However, due to the L^∞-bound of the solution [8], the harmonic interpolations lead to desired properties. In Figure 2.2, solid and dashed lines represent linear and harmonic interpolations of density, respectively. If the value of c is negative, then the linear interpolated density

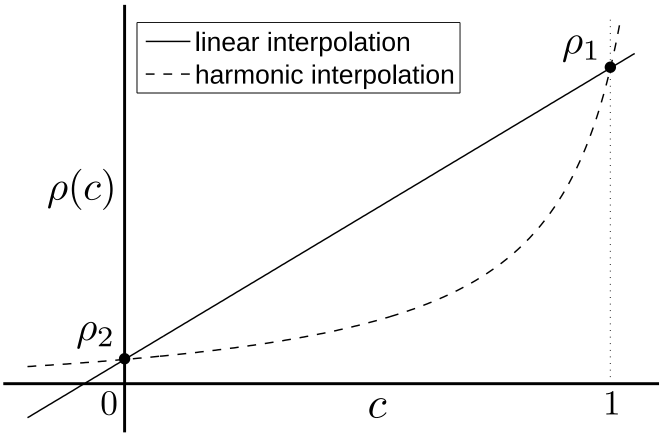


Figure 2.2. Linear (solid line) and harmonic (dotted line) interpolations of density ρ.

can be negative which is not physical value. However, if the harmonic interpolation is used, then we always have positive values of density.

2.2. The Boussinesq approximation

In fluids, the dynamic variables typically required to describe the motion are the velocity, pressure, viscosity, and density. The density is a very important quantity of fluids since it determines fluid properties such as compressibility. Because of the complexity of density variations in fluids, some assumptions have been introduced to reduce the degree of the complexity. One of these assumptions is the Boussinesq approximation [9], which is used in the field of buoyancy-driven flows.

Assume that the densities ρ_1 and ρ_2 (ρ_1 and ρ_2 are the densities of the heavier and lighter fluid, respectively) are uniform on each side of the interface, and that the density difference is small in the sense that the Atwood number $At = (\rho_1 - \rho_2)/(\rho_1 + \rho_2)$ is small. The Boussinesq approximation can be made in this case and is also related to the assumption that all of the accelerations of flow are small compared to the gravitational acceleration. In this approximation, the background density can be treated as a constant ρ_*, i.e., the variation of background density is neglected. And the difference between the actual density and ρ_* will contribute only to the buoyancy force term of the momentum equation [10]. Because of its simplicity in practical implementations (we solve a constant instead of a variable coefficient Poisson equation), the Boussinesq approximation is employed in many previous papers [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24].

Aref and Tryggvason [13] proposed a model for the development of the Rayleigh-Taylor instability in the Boussinesq approximation using concentrations of vorticity along the interface. In their case, the density was not coupled to a scalar field, and the buoyancy term in the Boussinesq Navier-Stokes equations was the only term related to the weak stratification. Han and Tryggvason [14] examined the deformation and breakup of axisymmetric drops, accelerated by a constant body force, for small density differences between the drops and the surrounding fluid. In their work, a density ratio was $\rho_1/\rho_2 = 1.15$ and a front tracking numerical technique was used to solve the unsteady Navier-Stokes equations for both the drops and the surrounding fluid. They showed, in the Boussinesq limit, as the Eötvös number increases the drops break up in a backward facing bag, a transient breakup, and a forward facing bag mode. Young et al. [15] investigated the miscible Rayleigh-Taylor instability in both two and three dimensions. In their work, the fluid was assumed incompressible under the Boussinesq approximation to make the problem more tractable. They found the three-dimensional mixing zone expands two times faster than the two-dimensional mixing zone through the simulation of randomly perturbed interfaces and identified three phases of evolution for the three-dimensional mixing zone: the free-falling phase, the mixing phase, and the another free-falling phase. Liu and Shen [16] modeled the mixture of two incompressible fluids with a phase field model. A semi-discrete Fourier-spectral method for the numerical approximation of a Navier-Stokes system coupled with a Cahn-Hilliard equation was proposed and analyzed. They used the Boussinesq approximation to model the case where the two fluids have different densities.

Vladimirova and Rosner studied the fully nonlinear behavior of premixed flames in a gravitationally stratified medium, subject to the Boussinesq approximation in [18, 19]. Vladimirova [20] simulated a bubble of reaction products rising in the reactant fluid under the influence of gravity using the Boussinesq buoyancy approximation. The Atwood numbers in their experiments were in the range of $0.075 \sim 0.16$. The author showed the evolution of the bubble can be divided into two stages: the bubble grows radially in an essentially motionless fluid in the first stage and the bubble rises and is distorted by the flow during the second stage. Celani et al. [21] studied the Rayleigh-Taylor instability of two immiscible fluids in the limit of small Atwood numbers by means of a phase-field description. They analytically rederived the known gravity-capillary dispersion relation in the limit of vanishing mixing energy density and capillary width. And numerical results were compared with known analytical results, both for the linearly stable and unstable cases, and for the weakly nonlinear stages of the latter. Forbes [22] studied the development of the Rayleigh-Taylor instability for inviscid and viscous fluids. The author used the approximate Boussinesq approach rather than the full Navier-Stokes equations of viscous flow and the density ratio was $\rho_1/\rho_2 = 1.05$. A separate spectral method was presented to study the Rayleigh-Taylor instability in a viscous Boussinesq fluid. The results were shown to agree closely with the inviscid calculations for small to moderate times. In [24], Boffetta et al. investigated the Rayleigh-Taylor turbulence in three dimensions at small Atwood number and at Prandtl number one by means of high resolution direct numerical simulations of the Boussinesq equations. The authors extended the mean-field analysis for velocity and temperature fluctuations and showed that small-scale velocity and temperature fluctuations develop intermittent distributions with structure function scaling exponents consistent with Navier-Stokes turbulence advecting a passive scalar.

In the Boussinesq approximation, the background density can be treated as a constant ρ_*, i.e., the variation of background density is neglected. And the difference between the actual density and ρ_* will contribute only to the buoyancy force term of the momentum equation [10]. Hence, Equation (2.6) becomes

$$\rho_*(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \nabla\cdot[\eta(\phi)(\nabla\mathbf{u} + \nabla\mathbf{u}^T)] + \mathbf{SF} + \rho(\phi)\mathbf{g}, \tag{2.10}$$

where $\rho_* = (\rho_1 + \rho_2)/2$ is the mean density [11, 21].

If we add and subtract the term $\rho_*\mathbf{g}$ to and from Equation (2.10), we get

$$\rho_*(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla p + \rho_*\mathbf{g} + \nabla\cdot[\eta(\phi)(\nabla\mathbf{u} + \nabla\mathbf{u}^T)] + \mathbf{SF} + (\rho(\phi) - \rho_*)\mathbf{g}. \tag{2.11}$$

In the two-dimensional case, we can write Equation (2.11) as follows:

$$\rho_*(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}) = -\nabla(p + \rho_* g y) + \nabla\cdot[\eta(\phi)(\nabla\mathbf{u} + \nabla\mathbf{u}^T)] + \mathbf{SF} + (\rho(\phi) - \rho_*)\mathbf{g}. \tag{2.12}$$

If we reset the pressure field as $p = p + \rho_* g y$ and divide by ρ_*, then Equation (2.12) becomes

$$\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{1}{\rho_*}\nabla p + \frac{1}{\rho_*}\nabla\cdot[\eta(\phi)(\nabla\mathbf{u} + \nabla\mathbf{u}^T)] + \frac{1}{\rho_*}\mathbf{SF} + \frac{\rho(\phi) - \rho_*}{\rho_*}\mathbf{g}.$$

The buoyancy contribution can be rewritten in terms of ρ_1, ρ_2, and ϕ as:

$$\begin{aligned}\frac{\rho(\phi) - \rho_*}{\rho_*}\mathbf{g} &= \frac{\rho_1(\frac{1+\phi}{2}) + \rho_2(\frac{1-\phi}{2}) - \rho_*}{\rho_*}\mathbf{g}\\ &= At\phi\mathbf{g}.\end{aligned}$$

On the other hand, applying the similar procedure to the case of the variable density model, the buoyancy contribution in Equation (2.6) can be rewritten as

$$\frac{\rho(\phi) - \rho_*}{\rho(\phi)}\mathbf{g} = \frac{At\phi}{1 + At\phi}\mathbf{g}.$$

In our phase-field model, ϕ varies from -1 to 1. Note that $At\phi$ (the Boussinesq case) and $At\phi/(1 + At\phi)$ (the variable density case) are linear and nonlinear functions with respect to the phase-field ϕ. Figure 2.3 shows profiles of $At\phi$ and $At\phi/(1+At\phi)$ for two different Atwood numbers $At = 0.01$ and $At = 0.5$. For $At = 0.01$, there is almost no difference between $At\phi$ and $At\phi/(1+At\phi)$. And both $At\phi$ and $At\phi/(1+At\phi)$ are nearly zero. But, for $At = 0.5$, there is a difference between $At\phi$ and $At\phi/(1 + At\phi)$. This difference implies the difference of the buoyancy force. Figures 2.4 (a) and (b) show the buoyancy force for the Boussinesq and the variable density models with $At = 0.5$, respectively. In the case of the Boussinesq model, the buoyancy force is symmetric along the interface. But, in the case of the variable density model, the buoyancy force is not symmetric along the interface although the phase-field is symmetric. This difference of the buoyancy force causes a difference of the interface evolution for the Boussinesq and the variable density models. If the density variation goes to zero, i.e., $At \ll 0$, then the difference between the Boussinesq and the non-Boussinesq fluids disappears. But in general there is a difference between the Boussinesq and the non-Boussinesq fluids.

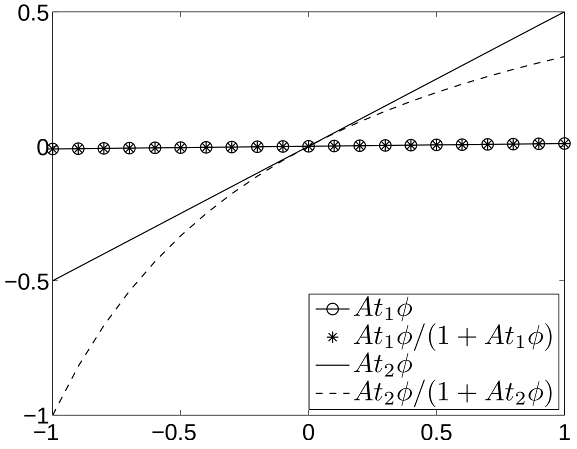


FIGURE 2.3. Profiles of $At\phi$ (the Boussinesq case) and $At\phi/(1 + At\phi)$ (the variable density case) for two different Atwood numbers $At_1 = 0.01$ and $At_2 = 0.5$.

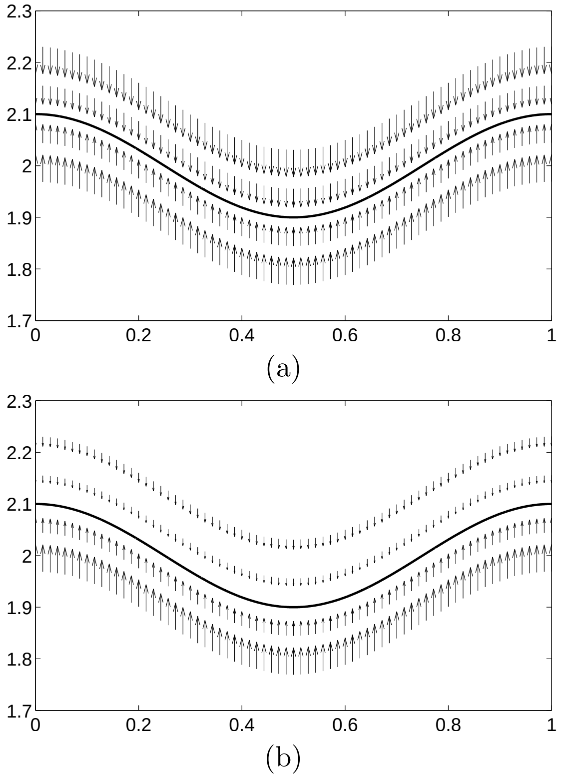


Figure 2.4. The buoyancy force for (a) the Boussinesq approximation and (b) the variable density models with $At = 0.5$. This difference causes a difference of the interface evolution for the two different models.

2.3. Regularized Dirac delta functions

Many numerical techniques, including immersed boundary [90, 91, 92, 93, 94, 95, 96, 97], volume-of-fluid [98, 99, 100, 101, 102, 103, 104, 105, 106, 107], and level-set [108, 109, 110, 111, 112, 113, 114, 115], use the concept of a regularized Dirac delta function to account for interfacial effects and many previous studies show that an appropriate delta function is required to obtain more accurate results.

In the immersed boundary method, the elastic boundary is represented by a set of Lagrangian points and the singular force at the Lagrangian points is determined by the generalized Hooke's law. This force is spread to the surrounding Eulerian points using a delta function. In the immersed boundary method, most commonly used delta functions are 2-point [92, 95, 97], 3-point [94, 95, 97], 6-point [90, 91, 95, 96], 4-point cosine [93, 97], and 4-point [91, 93, 95, 97] functions. For all r, where r is the parameter representing the position of the submerged boundary point and is scaled with respect to the grid size h, the one-dimensional delta functions are listed below.

- 2-point delta function

$$\delta(r) = \begin{cases} 1-|r| & \text{if } |r| \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

- 3-point delta function

$$\delta(r) = \begin{cases} \frac{1}{3}(1+\sqrt{-3r^2+1}) & \text{if } |r| \leq 0.5, \\ \frac{1}{6}(5-3|r|-\sqrt{-3(1-|r|)^2+1}) & \text{if } 0.5 \leq |r| \leq 1.5, \\ 0 & \text{otherwise.} \end{cases}$$

- 6-point delta function

$$\delta(r) = \begin{cases} \frac{61}{112}-\frac{11}{42}|r|-\frac{11}{56}|r|^2+\frac{1}{12}|r|^3+\frac{\sqrt{3}}{336}(243+1584|r| & \\ -748|r|^2-1560|r|^3+500|r|^4+336|r|^5-112|r|^6)^{1/2} & \text{if } 0 \leq |r| \leq 1, \\ \frac{21}{16}+\frac{7}{12}|r|-\frac{7}{8}|r|^2+\frac{1}{6}|r|^3-\frac{3}{2}\delta(|r|-1) & \text{if } 1 \leq |r| \leq 2, \\ \frac{9}{8}-\frac{23}{12}|r|+\frac{3}{4}|r|^2-\frac{1}{12}|r|^3+\frac{1}{2}\delta(|r|-2) & \text{if } 2 \leq |r| \leq 3, \\ 0 & \text{otherwise.} \end{cases}$$

- 4-point cosine delta function

$$\delta(r) = \begin{cases} \frac{1}{4}(1+\cos(\frac{\pi r}{2})) & \text{if } |r| \leq 2, \\ 0 & \text{otherwise.} \end{cases}$$

- 4-point delta function

$$\delta(r) = \begin{cases} \frac{1}{8}(3-2|r|+\sqrt{1+4|r|-4r^2}) & \text{if } 0 \leq |r| \leq 1, \\ \frac{1}{8}(5-2|r|-\sqrt{-7+12|r|-4r^2}) & \text{if } 1 \leq |r| \leq 2, \\ 0 & \text{otherwise.} \end{cases}$$

The above five types of delta functions are shown in Figure 2.5. Shin et al. [95] analyzed the stability regimes of the feedback forcing gains in the feedback forcing method for several types of delta functions and showed that non-growing oscillations became smaller for the delta function supported by more points. Yang et al. [97] found that the non-physical oscillations are mainly due to the fact that the derivatives of the regular discrete delta functions do not satisfy certain moment conditions and demonstrated that the smoothed discrete delta functions can effectively suppress the non-physical oscillations in the volume forces and improve the accuracy of the immersed boundary method with direct forcing in moving boundary simulations.

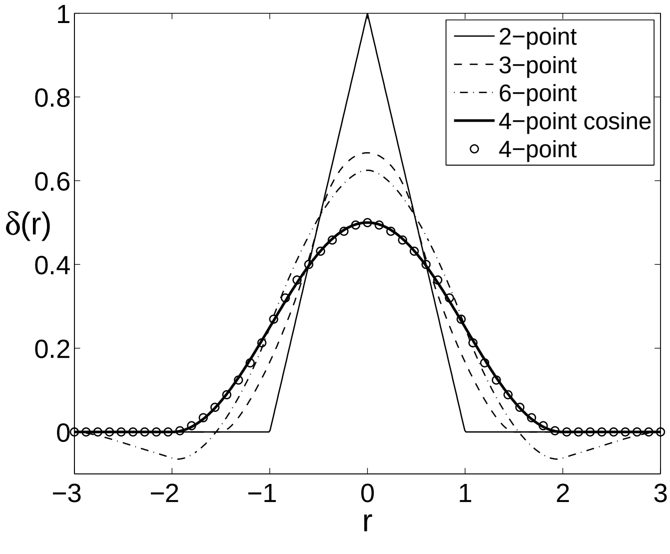


FIGURE 2.5. Five types of delta functions used in the immersed boundary method.

The volume-of-fluid (VOF) method was proposed by Hirt and Nichols [30]. In VOF method, the interface is reconstructed from the values of a color function which represents the volume fraction of one of the fluids in each cell. The continuum surface force (CSF) of Brackbill et al. [120] has been widely used to model surface tension in multi-phase fluid flows in VOF method. In the CSF model [100, 101, 107, 55, 39, 63, 121], the surface tension force is converted into a volume force via a delta function: $\mathbf{f} = \sigma\kappa\mathbf{n}\delta$, where σ is the surface tension coefficient, κ is the curvature, $\mathbf{n}$ is the normal to the surface, and δ is a delta function. In VOF method, the most commonly used delta function is $\delta(\tilde{c}) = |\nabla\tilde{c}|$, where $\tilde{c}$ is a smoothed version of the volume fraction. The CSF model is simple and robust, and involves only the solving of a field equation for a smoothed phase field $\tilde{c}$. However, the method is known to produce strong and spurious currents near the interface. For this reason, many researchers have developed new methods to reduce spurious currents [104, 107, 122, 123, 124]. Meier et al. [104] reduced spurious currents using the piecewise-linear interface construction (PLIC-VOF) method. In [124], a parabolic reconstruction of surface tension (PROST) algorithm is used to gain higher order accuracy for the surface tension force.

In the level-set method, first devised and introduced by Osher and Sethian in [125], delta functions are often used to distribute a singular force or compute a surface area [113, 126, 127, 128]. Most commonly used delta functions are listed below. Here, ϵ is proportional to the grid size, i.e., $\epsilon = mh$ for a positive number m.

- Delta function in [108, 110, 112, 114, 115]

$$\delta_\epsilon(x) \quad = \quad \begin{cases} \frac{1}{\epsilon}\left(1 - \left|\frac{x}{\epsilon}\right|\right) & \text{if } |x| \le \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

- Delta function in [108, 109, 110, 111, 113, 114, 115]

$$\delta_\epsilon(x) \quad = \quad \begin{cases} \frac{1}{2\epsilon}\left(1+\cos(\frac{\pi x}{\epsilon})\right) & \text{if } |x| \leq \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

Tornberg and Engquist [129] pointed out that the most common technique for regularization of delta functions in level-set simulations is not consistent and analyzed the accuracy of regularization of delta functions. Smereka [112] presented methods for constructing consistent approximations to Dirac delta measures concentrated on piecewise smooth curves or surfaces. Towers [130] proposed second-order finite difference methods for approximating Heaviside functions and showed that the methods are more accurate than a commonly used approximate Heaviside function.

We now present eight types of delta functions for phase-field models. From the Ginzburg–Landau free energy (2.4), the phase-field c at the equilibrium state satisfies the following equation:

$$\frac{1}{4}c^2(1-c)^2 = \frac{\epsilon^2}{2}|\nabla c|^2. \tag{2.13}$$

By using Equation (2.13) and delta functions used in phase-field models [65, 131, 132, 133], we get new delta functions for the present model. In this thesis, eight types of delta functions are chosen:

- Delta function in [65, 131] $\delta_1(c) = 6\sqrt{2}\epsilon|\nabla c|^2$.
- New delta function $\delta_2(c) = 3\sqrt{2}c^2(1-c)^2/\epsilon$.
- Delta function in [132, 133] $\delta_3(c) = 0.5(\delta_1(c) + \delta_2(c))$.
- Delta function in [65] $\delta_4(c) = 30\sqrt{2}\epsilon c(1-c)|\nabla c|^2$.
- New delta function $\delta_5(c) = c(1-c)/(\sqrt{2}\epsilon)$.
- New delta function $\delta_6(c) = |\nabla c|$.

- New delta function $\delta_7(c) = 0.5(\delta_5(c) + \delta_6(c))$.
- New delta function $\delta_8(c) = 630\sqrt{2}\epsilon c^3(1-c)^3|\nabla c|^2$.

For a flat interface with an equilibrium profile,

$$\int_{-\infty}^{\infty} \delta_i(c^{\text{eq}})\ dx = 1 \text{ for all } i = 1, 2, \ldots, 8,$$

where $c^{\text{eq}}(x,y) = 0.5(1+\tanh(x/(2\sqrt{2}\epsilon)))$ is an equilibrium profile in the infinite domain when the chemical potential is given by Equation (2.3) [64] and it is a good approximation in the finite domain. Figure 2.6 shows delta functions used in this paper. We note that new delta functions $\delta_5(c)$ and $\delta_8(c)$ have wider and narrower supports than $\delta_1(c)$ and $\delta_4(c)$, respectively.

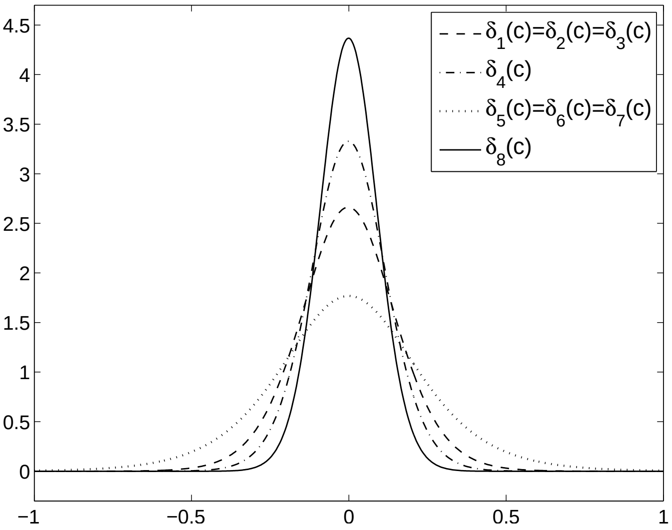


FIGURE 2.6. Delta functions for phase-field models.

We now consider a line of unit length on a unit domain $\Omega = (0,1) \times (0,1)$:

$$c(x,y) = \frac{1}{2}\left(1 + \tanh\left(\frac{0.5 - x}{2\sqrt{2}a}\right)\right)$$

for different values of a. The value of a is related to the interface thickness. For $a = 0.5\epsilon$, ϵ, and 2ϵ, the length of line is given by

$$f_i(a) = \int_\Omega \delta_i(c)\ dx\ dy.$$

This integral is calculated using the Mathematica code given in the Appendix. The results are given in Table 2.1. Next, we consider a circle of radius 0.25 centered at the center of a domain $\Omega = (0, 1) \times (0, 1)$:

$$c(x, y) = \frac{1}{2}\left(1 + \tanh\left(\frac{0.25 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{2\sqrt{2}a}\right)\right).$$

The circumference for different values of a is given in Table 2.1. In both cases, $\delta_6(c)$ is insensitive to the value of a. This means that the length of line or circle can be computed accurately using $\delta_6(c)$ regardless of the interface thickness. We will numerically discuss this in more detail in Section 6.1. Figure 2.7 shows the graphs of $f_i(a)$ for $i = 1, 2, \ldots, 8$.

TABLE 2.1. The length of line and circle with different interface thicknesses for each delta function.

	Case 1: line			Case 2: circle			
a	0.5ϵ	ϵ	2ϵ	0.5ϵ	ϵ	2ϵ	$f_i(a)$
δ_1	2	1	0.5	π	0.5π	0.25π	$f_1(a) = 1/a$
δ_2	0.5	1	2	0.25π	0.5π	π	$f_2(a) = a$
δ_3	1.25	1	1.25	0.625π	0.5π	0.625π	$f_3(a) = (1/a + a)/2$
δ_4	2	1	0.5	π	0.5π	0.25π	$f_4(a) = 1/a$
δ_5	0.5	1	2	0.25π	0.5π	π	$f_5(a) = a$
δ_6	1	1	1	0.5π	0.5π	0.5π	$f_6(a) = 1$
δ_7	0.75	1	1.5	0.375π	0.5π	0.75π	$f_7(a) = (a + 1)/2$
δ_8	2	1	0.5	π	0.5π	0.25π	$f_8(a) = 1/a$

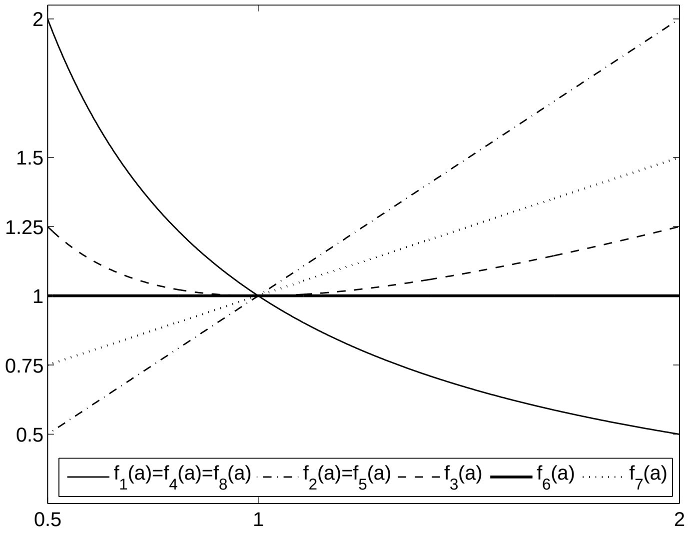


FIGURE 2.7. The graphs of $f_i(a)$ for $i = 1, 2, \ldots, 8$.

2.4. Nondimensionalization

To restate the dimensional NSCH system in dimensionless form, we define the dimensionless variables as

$$x' = \frac{x}{L_c},\ \mathbf{u}' = \frac{\mathbf{u}}{U_c},\ t' = \frac{tU_c}{L_c},\ p' = \frac{p}{\rho_c U_c^2},\ \mathbf{g}' = \frac{\mathbf{g}}{g},\ \rho' = \frac{\rho}{\rho_c},\ \mu' = \frac{\mu}{\mu_c},$$

where L_c, U_c, and ρ_c are the characteristic length, velocity, and density, respectively. g is the gravitational acceleration and μ_c is α, which is $\mu = \alpha(\phi^3 - \frac{\beta}{\alpha}\phi - \frac{k}{\alpha}\Delta\phi) = \mu_c\mu'$. Substituting these variables into the governing equations (2.6) - (2.9) we have

$$\begin{aligned} \rho'(\mathbf{u}'_{t'} + \mathbf{u}' \cdot \nabla'\mathbf{u}') &= -\nabla'p' + \frac{\eta}{\rho_c U_c L_c}\Delta'\mathbf{u}' + \frac{gL_c}{U_c^2}\rho'\mathbf{g}' \\ &\quad -\frac{3\sqrt{2}\sigma\epsilon}{4\rho_c L_c U_c^2}\nabla' \cdot \left(\frac{\nabla'\phi}{|\nabla'\phi|}\right)|\nabla'\phi|\nabla'\phi, && (2.14) \\ \nabla' \cdot \mathbf{u}' &= 0, && (2.15) \\ \phi_{t'} + \nabla' \cdot (\phi\mathbf{u}') &= \frac{M\mu_c}{U_c L_c}\Delta'\mu', && (2.16) \\ \mu' &= \phi^3 - \frac{\beta}{\alpha}\phi - \frac{k}{\alpha L_c^2}\Delta'\phi. && (2.17) \end{aligned}$$

Dropping the primes and considering $\alpha = \beta$, Equations (2.14) - (2.17) become

$$\begin{aligned} \rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla\mathbf{u}) &= -\nabla p + \frac{1}{Re}\Delta\mathbf{u} + \frac{1}{We}\mathbf{SF} + \frac{\rho}{Fr}\mathbf{g}, && (2.18) \\ \nabla \cdot \mathbf{u} &= 0, && (2.19) \\ \phi_t + \nabla \cdot (\phi\mathbf{u}) &= \frac{1}{Pe}\Delta\mu, && (2.20) \\ \mu &= \phi^3 - \phi - \epsilon^2\Delta\phi, && (2.21) \end{aligned}$$

where $\mathbf{SF} = -0.75\sqrt{2}\epsilon\nabla \cdot (\nabla\phi/|\nabla\phi|)\,|\nabla\phi|\nabla\phi$. The dimensionless physical parameters are the Reynolds number, Re, Weber number, We, Froude number, Fr, Peclet number,

Pe, Cahn number, ϵ, given by

$$Re = \frac{\rho_c U_c L_c}{\eta},\ We = \frac{\rho_c L_c U_c^2}{\sigma},\ Fr = \frac{U_c^2}{g L_c},\ Pe = \frac{U_c L_c}{M \mu_c},\ \epsilon = \sqrt{\frac{k}{\alpha L_c^2}}.$$

2.5. Snow crystal growth

In nonlinear dynamical systems, the physics of phase transformations has attracted considerable interest. Crystal growth is an essential part of phase transformations from the liquid phase to the solid phase via heat transfer. To simulate crystal growth, cellular automaton [255, 273, 274, 275, 276], Monte-Carlo [260, 263], boundary integral [256, 257, 264, 266], front-tracking [232, 243, 251], level-set [237, 241, 248, 270], and phase-field [235, 236, 238, 239, 242, 245, 246, 247, 249, 258, 259, 261, 262, 265, 267, 268, 269, 272] methods have been developed. In this thesis, we focus on the phase-field method for crystal growth simulations.

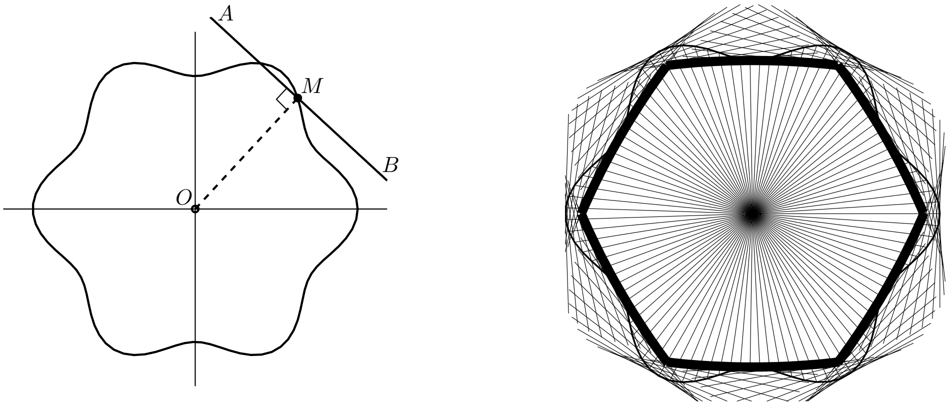
Many numerical methods such as explicit [243, 244, 247, 261, 268], mixed implicit-explicit [259, 269, 272], and adaptive methods [236, 238, 258, 262, 265] have been proposed for crystal growth problems. In particular, a multiple time-step algorithm that uses a larger time step for the flow-field calculations while reserving a finer time step for the phase-field evolution was proposed in [267]. One of the main difficulties in the numerical simulations of crystal growth with various anisotropic interfacial energies is the large difference in time and length scales. The use of space and temporal mesh adaptivity is a natural choice to overcome this problem. Therefore, we need a scheme that allows the use of a sufficiently large time step without the stability limitations. In this paper we present a new accurate and efficient operator splitting algorithm for solving the phase-field model of crystal growth. In particular, we focus on six-fold symmetric crystal growth, which is related to snow crystal. Water has the unique chemical property known as a hydrogen bond. The attractive interaction between the hydrogen and oxygen atoms in different water molecules arranges the solid state water molecules to form a hexagonal shape. For such a reason, in specific temperature, snow crystal grows into six-fold symmetric crystal [254]. The thickness and width of snow crystal are in the ratio of 1 : 50, so snow crystal problems can be simplified into two-dimensional problems. We consider here the solidification of a pure substance from its supercooled melt in two-dimensional space.

2.6. The Wulff construction

The equilibrium crystal shape is determined by minimizing the total interfacial free energy. We use the following k-fold symmetric interfacial energy equation:

$$\epsilon(\theta) = \epsilon_0 \left[1 + \epsilon_k \cos(k\theta)\right],$$

where ϵ_0 is the mean interfacial tension and $0 \leq \epsilon_k < 1$ is the anisotropy parameter.



(a) (b)

Figure 2.8. The Wulff construction. (a) Interfacial free-energy density $\epsilon(\theta)$ in the polar coordinates. (b) Equilibrium crystal shape (bold line) for $k = 6$, $\epsilon_0 = 1$, and $\epsilon_6 = 0.1$.

In this paper, we focus on $k = 6$ case. The equilibrium shape is easily constructed by the Wulff's theorem [271]. We describe the construction of the equilibrium shape geometrically [234]. Let $M = (\epsilon(\theta), \theta)$ be a point on the interfacial energy function in the polar coordinates (see Figure 2.8(a)). The construction starts from the origin O and draw the line segment $\overline{OM}$ to the point M. Draw the perpendicular line $\overleftrightarrow{AB}$ to the line segment $\overline{OM}$. Then the inner convex hull made from all such perpendiculars is an equilibrium crystal shape as shown in Figure 2.8(b).

Conversely, let us assume the equilibrium shape is known and (r, θ) be the polar coordinates of a point T of the crystal boundary S, that is, $T = (r, \theta)$. And let

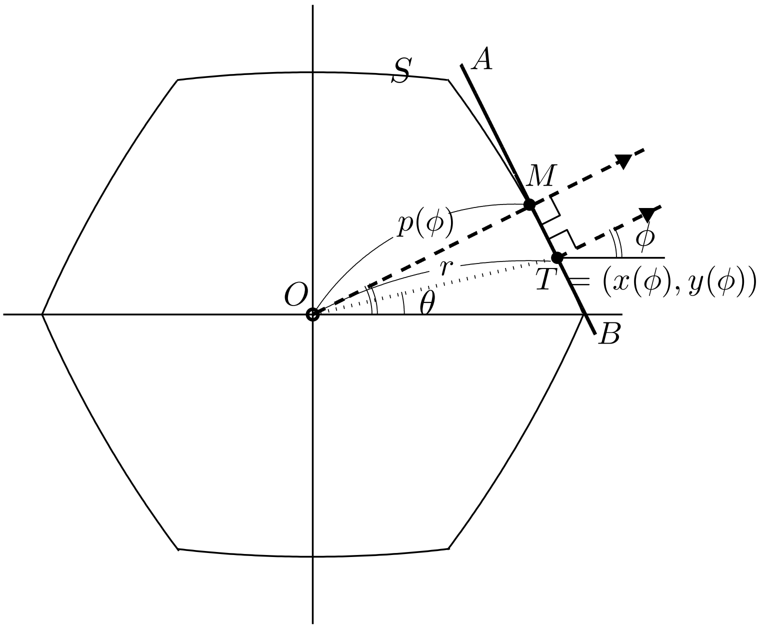


FIGURE 2.9. Parameter definitions.

$T = (x(\phi), y(\phi))$ be the corresponding Cartesian coordinates, where ϕ is a parameter and is the angle between x-axis and the perpendicular line to the tangent line $\overleftrightarrow{AB}$ at the point T. Let M be the intersection point of the line $\overleftrightarrow{AB}$ and the perpendicular line containing the origin to $\overleftrightarrow{AB}$. Let the length of the line segment $\overline{OM}$ be $p(\phi)$. In Figure 2.9, we can see these parameter definitions. Then $p(\phi)$ can be obtained from the right triangle $\triangle OTM$:

$$p(\phi) = r\cos(\phi - \theta) = r\cos\phi\cos\theta + r\sin\phi\sin\theta = x(\phi)\cos\phi + y(\phi)\sin\phi. \tag{2.22}$$

We can express $(x(\phi), y(\phi))$ in terms of $p(\phi)$. First, take a derivative to $p(\phi)$, then we have

$$p_\phi(\phi) = x_\phi(\phi)\cos\phi - x(\phi)\sin\phi + y_\phi(\phi)\sin\phi + y(\phi)\cos\phi. \tag{2.23}$$

Since the normal vector $(\cos\phi, \sin\phi)$ and the tangent vector (x_ϕ, y_ϕ) are orthogonal, that is, $(\cos\phi, \sin\phi)\cdot(x_\phi, y_\phi) = 0$, we can simplify Equation (2.23) as

$$p_\phi = -x\sin\phi + y\cos\phi. \tag{2.24}$$

Now, by solving Equations (2.22) and (2.24) we have

$$x(\phi) = p(\phi)\cos\phi - p_\phi(\phi)\sin\phi,\;\; y(\phi) = p(\phi)\sin\phi + p_\phi(\phi)\cos\phi. \tag{2.25}$$

Let F and A be the total edge free energy and the area of crystal, respectively and be defined as

$$\begin{aligned} F &= \int \epsilon(\phi)\sqrt{(x_\phi(\phi))^2 + (y_\phi(\phi))^2}d\phi, && (2.26)\\ A &= \frac{1}{2}\int (x(\phi)y_\phi(\phi) - y(\phi)x_\phi(\phi))d\phi. && (2.27)\end{aligned}$$

Using Equation (2.25), we can rewrite Equations (2.26) and (2.27) in the form

$$\begin{aligned} F &= \int \epsilon(\phi)(p(\phi) + p_{\phi\phi}(\phi))d\phi,\\ A &= \frac{1}{2}\int p(\phi)(p(\phi) + p_{\phi\phi}(\phi))d\phi.\end{aligned}$$

We want to minimize F with subject to a constant area constraint of A. Using the Lagrange multiplier λ, we seek to minimize

$$F + \lambda A = \int \left(\epsilon(\phi) + \frac{\lambda}{2}p(\phi)\right)(p(\phi) + p_{\phi\phi}(\phi))d\phi.$$

And then, the Euler–Lagrange equation is

$$\frac{\partial Q}{\partial p} - \frac{d}{d\phi}\left(\frac{\partial Q}{\partial p_\phi}\right) + \frac{d^2}{d\phi^2}\left(\frac{\partial Q}{\partial p_{\phi\phi}}\right) = 0, \tag{2.28}$$

where

$$Q = \left(\epsilon + \frac{\lambda}{2}p\right)(p + p_{\phi\phi}). \tag{2.29}$$

From these two Equations (2.28) and (2.29), we get

$$p + p_{\phi\phi} = -\frac{1}{\lambda}(\epsilon + \epsilon_{\phi\phi}). \tag{2.30}$$

A solution of differential equation (2.30) is

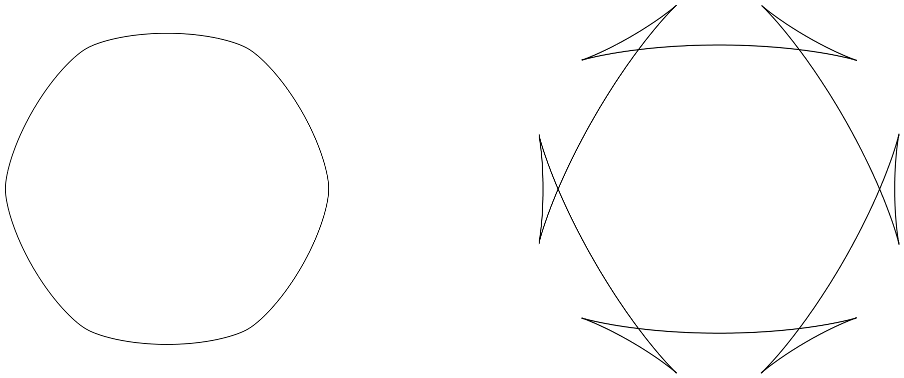
$$p(\phi) = -\frac{1}{\lambda}\epsilon(\phi).$$

This result implies that in a crystal at equilibrium, the distances of the faces from the center of the crystal are proportional to their surface free energies per unit area [234].

For large ϵ_6 values, the crystal shape will be energy minimizing when certain orientations are missing. Missing orientations occur when the polar plot of $r = 1/\epsilon(\theta)$ changes convexity [240]. The curvature of a polar plot $r(\theta)$ is $\kappa = (r^2 + 2r_\theta^2 - rr_{\theta\theta})/(r^2 + r_\theta^2)^{\frac{3}{2}}$. For $r(\theta) = 1/\epsilon(\theta)$, the curvature is $\kappa = (\epsilon + \epsilon_{\theta\theta})/[1 + (\frac{\epsilon_\theta}{\epsilon})^2]^{\frac{3}{2}}$. So convexity changes whenever

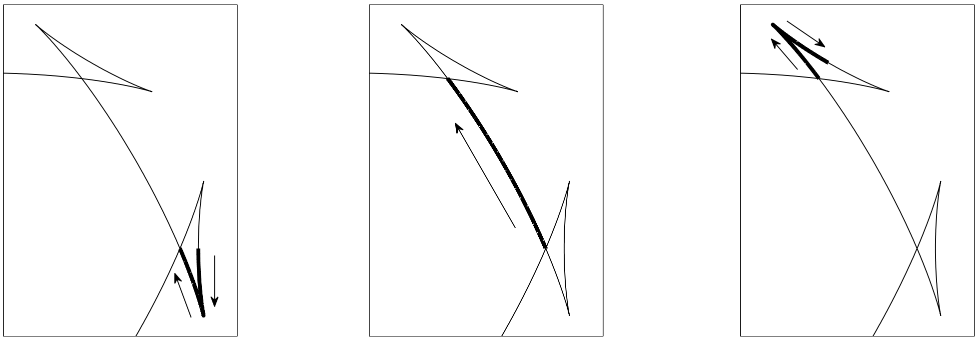
$$\epsilon + \epsilon_{\theta\theta} = \epsilon_0(1 - 35\epsilon_6 \cos 6\theta) < 0.$$

If values of ϵ_6 are larger than 1/35, then missing orientations occur. Figure 2.10 shows the 6-fold Wulff equilibrium shapes $((x(\phi), y(\phi))$ for $0 \leq \phi \leq 2\pi)$ with two different ϵ_6 values: (a) $\epsilon_6 = 1/50$ and (b) $\epsilon_6 = 1/10$ (which shows the missing orientation). Figure 2.11 shows the trace of $(x(\phi), y(\phi))$ with different intervals. ϕ_m is defined as the smallest non-zero value which satisfies $y(\phi_m) = 0$.



(a) $\epsilon_6 = 1/50$ (b) $\epsilon_6 = 1/10$

FIGURE 2.10. The 6-fold Wulff equilibrium shapes with two different ϵ_6 values.



(a) $0 \leq \phi \leq \phi_m$ (b) $\phi_m \leq \phi \leq \pi/3 - \phi_m$ (c) $\pi/3 - \phi_m \leq \phi \leq \pi/3$

FIGURE 2.11. Trace of $(x(\phi), y(\phi))$ with different intervals and $y(\phi_m) = 0$.

2.7. Governing equations of snow crystal growth

The phase-field model for the crystal growth is given by

$$\begin{aligned}
\epsilon^2(c)\frac{\partial c}{\partial t} &= \nabla\cdot(\epsilon^2(c)\nabla c) + [c-\lambda U(1-c^2)](1-c^2) \\
&\quad + \left(|\nabla c|^2\epsilon(c)\frac{\partial\epsilon(c)}{\partial c_x}\right)_x + \left(|\nabla c|^2\epsilon(c)\frac{\partial\epsilon(c)}{\partial c_y}\right)_y \qquad (2.31)\\
\frac{\partial U}{\partial t} &= D\Delta U + \frac{1}{2}\frac{\partial c}{\partial t},
\end{aligned}$$

where c is the order parameter, $\epsilon(c)$ is the anisotropic function, λ is the dimensionless coupling parameter, and $U = c_p(T - T_M)/L$ is the dimensionless temperature field. Here c_p is the specific heat at constant pressure, T_M is the melting temperature, L is the latent heat of fusion, $D = \alpha\tau_0/\epsilon_0^2$, α is the thermal diffusivity, τ_0 is the characteristic time, and ϵ_0 is the characteristic length. The order parameter is defined by $c = 1$ in the solid phase and $c = -1$ in the liquid phase. The interface is defined by $c = 0$ and λ is given as $\lambda = D/a_2$ with $a_2 = 0.6267$ [246, 247]. We define a normal vector of c as (c_x, c_y) and an angle between normal vector and x-axis as ϕ that satisfies $\tan\phi = c_y/c_x$. Then by replacing $\epsilon(c)$ with $\epsilon(\phi) = \epsilon_0(1 + \epsilon_6\cos(6\phi))$, we can simplify the following terms in Equation (2.31):

$$\left(|\nabla c|^2\epsilon(\phi)\frac{\partial\epsilon(\phi)}{\partial c_x}\right)_x = \left((c_x^2+c_y^2)\epsilon(\phi)\epsilon'(\phi)\left(-\frac{c_y}{c_x^2+c_y^2}\right)\right)_x = -\left(\epsilon'(\phi)\epsilon(\phi)c_y\right)_x.$$

In a similar way, we get

$$\left(|\nabla c|^2\epsilon(\phi)\frac{\partial\epsilon(\phi)}{\partial c_y}\right)_y = \left(\epsilon'(\phi)\epsilon(\phi)c_x\right)_y.$$

Hence we can rewrite the governing equations of 6-fold symmetric crystal growth as following:

$$\begin{aligned}\epsilon^2(\phi)\frac{\partial c}{\partial t} &= \nabla \cdot (\epsilon^2(\phi)\nabla c) + [c - \lambda U(1-c^2)](1-c^2) \\ &\quad - \left(\epsilon'(\phi)\epsilon(\phi)c_y\right)_x + \left(\epsilon'(\phi)\epsilon(\phi)c_x\right)_y \qquad (2.32) \\ \frac{\partial U}{\partial t} &= D\Delta U + \frac{1}{2}\frac{\partial c}{\partial t}. \qquad (2.33)\end{aligned}$$

2.8. Derivation of the thin film equation

We consider the dynamics of a thin layer of liquid of thickness $h = h(x, y, t)$ on an inclined surface driven by thermally created surface tension gradients and influenced by gravity. The spatial variables x and y denote the direction of the flow and the direction normal to the flow, respectively. Let α, ρ, g, η, γ, and $\tau = d\gamma/dx$ denote the angle from horizontal of inclination of the plane, the density, the gravitational constant, the dynamic viscosity, the surface tension, and the surface tension gradient of the liquid [278].

We briefly review the derivation of the lubrication model [279]. We start from the three-dimensional Navier-Stokes equations.

$$\begin{aligned}
\rho(u_t + uu_x + vu_y + wu_z) &= -p_x + \eta(u_{xx} + u_{yy} + u_{zz}) - \rho g \sin\alpha, \\
\rho(v_t + uv_x + vv_y + wv_z) &= -p_y + \eta(v_{xx} + v_{yy} + v_{zz}), \\
\rho(w_t + uw_x + vw_y + ww_z) &= -p_z + \eta(w_{xx} + w_{yy} + w_{zz}) - \rho g \cos\alpha, \\
u_x + v_y + w_z &= 0,
\end{aligned}$$

where subscripts denote partial derivatives. In this system the dependent variables are $u(x, y, z, t)$, $v(x, y, z, t)$, $w(x, y, z, t)$, and $p(x, y, z, t)$.

We define characteristic variables: L, H, U, V, W and T are characteristic length (x- and y-direction), height (z-direction), velocity in x-direction, velocity in y-direction, velocity in z-direction, and time, respectively. The lubrication approximation takes advantage of the thickness of the film by employing a small parameter:

$$\frac{H}{L} = \xi.$$

The thin film equation is considered slow flow, so the Navier-Stokes equations reduce to the Stokes equations:

$$\begin{aligned} 0 &= -p_x + \eta(u_{xx} + u_{yy} + u_{zz}) - \rho g \sin\alpha, \\ 0 &= -p_y + \eta(v_{xx} + v_{yy} + v_{zz}), \\ 0 &= -p_z + \eta(w_{xx} + w_{yy} + w_{zz}) - \rho g \cos\alpha, \\ u_x + v_y + w_z &= 0. \end{aligned}$$

Before applying boundary conditions, we drop the terms containing u_{xx}, u_{yy}, v_{xx}, v_{yy}, w_{xx}, w_{yy}, and w_{zz} by comparing their size (in terms of powers of ξ) to other terms in the Stokes equations. In particular, if

$$x = L\hat{x}, \ y = L\hat{y}, \ z = H\hat{z}, \ u = U\hat{u}, \text{ and } t = T\hat{t}, \tag{2.34}$$

the incompressibility condition, $u_x + v_y + w_z = 0$, gives us the following scaling for w:

$$w = \frac{HU}{L}\hat{w} = \xi U \hat{w}$$

and we see that

$$\begin{aligned} u_{xx} &= \frac{U}{L^2}\hat{u}_{\hat{x}\hat{x}} \ll \frac{U}{H^2}\hat{u}_{\hat{z}\hat{z}} = u_{zz}, \ u_{yy} = \frac{U}{L^2}\hat{u}_{\hat{y}\hat{y}} \ll \frac{U}{H^2}\hat{u}_{\hat{z}\hat{z}} = u_{zz}, \\ v_{xx} &= \frac{V}{L^2}\hat{v}_{\hat{x}\hat{x}} \ll \frac{V}{H^2}\hat{v}_{\hat{z}\hat{z}} = v_{zz}, \ v_{yy} = \frac{V}{L^2}\hat{v}_{\hat{y}\hat{y}} \ll \frac{V}{H^2}\hat{v}_{\hat{z}\hat{z}} = v_{zz}, \\ w_{xx} &= \frac{\xi U}{L^2}\hat{w}_{\hat{x}\hat{x}} \ll \frac{\xi U}{H^2}\hat{w}_{\hat{z}\hat{z}} \ll \frac{U}{H^2}\hat{u}_{\hat{z}\hat{z}} = u_{zz}, \\ w_{yy} &= \frac{\xi U}{L^2}\hat{w}_{\hat{y}\hat{y}} \ll \frac{\xi U}{H^2}\hat{w}_{\hat{z}\hat{z}} \ll \frac{U}{H^2}\hat{u}_{\hat{z}\hat{z}} = u_{zz}, \ w_{zz} = \frac{\xi U}{H^2}\hat{w}_{\hat{z}\hat{z}} \ll \frac{U}{H^2}\hat{u}_{\hat{z}\hat{z}} = u_{zz}. \end{aligned}$$

Thus for a thin film we may reduce the Navier-Stokes equations to the following equations:

$$\begin{aligned} p_x &= \eta u_{zz} - \rho g \sin\alpha, && (2.35)\\ p_y &= \eta v_{zz}, && (2.36)\\ p_z &= -\rho g \cos\alpha, && (2.37)\\ u_x + v_y + w_z &= 0. \end{aligned}$$

.

The normal stress boundary condition equates the force exerted by the interface at the liquid-air interface $z = h(x, y, t)$ on the fluid below with the effect of surface tension, through the curvature of the free surface, κ [280]:

$$p - p_{atm} = -\gamma\kappa = -\gamma \frac{(1 + h_y^2)h_{xx} + (1 + h_x^2)h_{yy} - 2h_x h_y h_{xy}}{(1 + h_x^2 + h_y^2)^{3/2}}.$$

Here, surface tension coefficient is denoted by γ. The expression for curvature can be simplified in the lubrication approximation since

$$(h_x)^2 \approx \left(\frac{H}{L}\right)^2 = \xi^2 \ll 1 \quad \text{and} \quad (h_y)^2 \approx \left(\frac{H}{L}\right)^2 = \xi^2 \ll 1.$$

Therefore, the curvature is approximated by Δh and the normal stress boundary condition is

$$p - p_{atm} = -\gamma \Delta h \quad \text{at } z = h(x, y, t). \qquad (2.38)$$

We choose the following linear equation of state to relate the surface tension gradient to the temperature gradient:

$$\Gamma(T_s) = \gamma_0 - \tau(T_s - T_\infty),$$

where T_s is the temperature at the surface, T_∞ is the ambient temperature, τ is the surface tension gradient, and γ_0 is a base value of surface tension. The tangential stress condition at the free surface $z = h(x, y, t)$ is based on the assumption that we drive the film with a constant surface stress, which couples the gradient in surface tension τ to the scalar component of stress T_{zx} in the flow [281]:

$$\tau = T_{zx} = 2\eta D_{zx},$$

where subscript of T and D denotes partial derivative and component, respectively. The strain rate tensor has the form

$$D = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \frac{1}{2}\begin{pmatrix} 2u_x & u_y + v_x & u_z + w_x \\ u_y + v_x & 2v_y & v_z + w_y \\ u_z + w_x & v_z + w_y & 2w_z \end{pmatrix}.$$

Therefore $D_{zx} = \frac{1}{2}u_z + O(\xi)$ and the tangential stress condition takes the form

$$\tau = \eta u_z \quad \text{at } z = h(x, y, t). \tag{2.39}$$

The no slip boundary condition assumes that the liquid at the liquid-solid interface is not moving. This is expressed in the boundary condition:

$$u = v = 0 \quad \text{at } z = 0.$$

We integrate Equation (2.37) with respect to z:

$$p(x, y, z, t) = -\rho g \cos\alpha z + c_1. \tag{2.40}$$

We determine c_1 using the normal stress boundary condition (2.38) which yields

$$c_1 = p_{atm} - \gamma\Delta h + \rho g \cos\alpha h.$$

We differentiate Equation (2.40) with respect to x and y and substitute the resulting expression for p_x and p_y into Equations (2.35) and (2.36), respectively:

$$\begin{aligned}\eta u_{zz} &= \rho g \cos\alpha h_x - \gamma \Delta h_x + \rho g \sin\alpha, \\ \eta v_{zz} &= \rho g \cos\alpha h_y - \gamma \Delta h_y.\end{aligned}$$

We integrate twice with respect to z to derive expressions for the velocity u and v. After the first integration we have the following equation:

$$\begin{aligned}\eta u_z &= (\rho g \cos\alpha h_x - \gamma \Delta h_x + \rho g \sin\alpha) z + c_2, \\ \eta v_z &= (\rho g \cos\alpha h_y - \gamma \Delta h_y) z + c_3.\end{aligned}$$

We determine c_2 and c_3 using the tangential stress boundary condition equation (2.39) which yields

$$\begin{aligned}c_2 &= \tau - (\rho g \cos\alpha h_x - \gamma \Delta h_x + \rho g \sin\alpha) h, \\ c_3 &= -(\rho g \cos\alpha h_y - \gamma \Delta h_y) h.\end{aligned}$$

The second integration with respect to z yields

$$\begin{aligned}\eta u &= (\rho g \cos\alpha h_x - \gamma \Delta h_x + \rho g \sin\alpha) \frac{z^2}{2} + c_2 z + c_4, \\ \eta v &= (\rho g \cos\alpha h_y - \gamma \Delta h_y) \frac{z^2}{2} + c_3 z + c_5,\end{aligned}$$

where $c_4 = c_5 = 0$ due to the no slip boundary condition. This provides the final velocities $u(x, y, z, t)$ and $v(x, y, z, t)$:

$$\begin{aligned}u &= \frac{1}{\eta}\left[(\rho g \cos\alpha h_x - \gamma \Delta h_x + \rho g \sin\alpha)\left(\frac{z^2}{2} - hz\right) + \tau z\right], \\ v &= \frac{1}{\eta}\left[(\rho g \cos\alpha h_y - \gamma \Delta h_y)\left(\frac{z^2}{2} - hz\right)\right].\end{aligned}$$

We model the dynamics of the draining film using the lubrication approximation with a "depth averaged" velocity $\bar{\mathbf{u}} = (\bar{u}, \bar{v})$ [277]:

$$\begin{aligned}
\bar{u} &= \frac{1}{h}\int_0^h u dz = \frac{\tau h}{2\eta} - \frac{\rho g h^2 \sin\alpha}{3\eta} + \frac{\gamma h^2 \Delta h_x}{3\eta} - \frac{\rho g h^2 h_x \cos\alpha}{3\eta}, \\
\bar{v} &= \frac{1}{h}\int_0^h v dz = \frac{\gamma h^2 \Delta h_y}{3\eta} - \frac{\rho g h^2 h_y \cos\alpha}{3\eta}, \\
\bar{\mathbf{u}} &= \left(\frac{\tau h}{2\eta} - \frac{\rho g h^2 \sin\alpha}{3\eta}\right)\vec{e}_x + \frac{\gamma h^2 \nabla\Delta h}{3\eta} - \frac{\rho g h^2 \nabla h \cos\alpha}{3\eta}, \qquad (2.41)
\end{aligned}$$

where $\vec{e}_x = (1, 0)$. In Equation (2.41), the first term is due to surface tension gradient, the second term is due to the tangential component of gravity, the third term is due to curvature, and the fourth term is due to the normal component of gravity. Coupling Equation (2.41) with mass conservation, we obtain

$$h_t + \nabla \cdot (h\bar{\mathbf{u}}) = 0. \qquad (2.42)$$

To non-dimensionalize Equation (2.42), we employ the non-dimensional variables (denoted by hats) defined in Equation (2.34) and $h = H\hat{h}$ to obtain

$$\begin{aligned}
\frac{H}{T}\hat{h}_{\hat{t}} \quad + \quad & \frac{1}{L}\hat{\nabla} \cdot \left[\left(\frac{H^2\tau\hat{h}^2}{2\eta} - \frac{H^3\rho g\hat{h}^3 \sin\alpha}{3\eta}\right)\vec{e}_x\right. \\
& \left. + \frac{H^4\gamma\hat{h}^3\hat{\nabla}\hat{\Delta}\hat{h}}{3L^3\eta} - \frac{H^4\rho g\hat{h}^3\hat{\nabla}\hat{h}\cos\alpha}{3L\eta}\right] = 0. \qquad (2.43)
\end{aligned}$$

Now we define the characteristic variables by balancing terms. First we balance the tangential gravity and Marangoni terms to define H. Note that the independent and dependent variables are all order 1. We obtain the following equation for H

$$\frac{H^2\tau}{2\eta} = \frac{H^3\rho g \sin\alpha}{3\eta}, \text{ i.e., } H = \frac{3\tau}{2\rho g \sin\alpha}.$$

Next we define L such that the Marangoni and surface tension effects balance:

$$\frac{H^2\tau}{2\eta} = \frac{H^4\gamma}{3L^3\eta}, \text{ i.e., } L = \left(\frac{3\gamma\tau}{2\rho^2 g^2 \sin^2\alpha}\right)^{1/3}.$$

Now choose the time scale T so that gravity, Marangoni and surface tension forces balance:

$$\frac{H}{T} = \frac{H^2\tau}{2L\eta}, \text{ i.e., } T = \frac{2\eta}{\tau^2}\left(\frac{4\gamma\tau\rho g\sin\alpha}{9}\right)^{1/3}.$$

We substitute the expressions for H, L, and T into Equation (2.43) and drop the '$\hat{\ }$' to obtain the dimensionless equation thin film equation:

$$h_t + (h^2 - h^3)_x = D\nabla \cdot (h^3\nabla h) - \nabla \cdot (h^3\nabla\Delta h),$$

where $D = \frac{TH^3\rho g\cos\alpha}{3\eta L^2}$. As in [278] we take $D = 0$. Thus we have derived the dimensionless thin film equation

$$h_t + (h^2 - h^3)_x = -\nabla \cdot (h^3\nabla\Delta h). \tag{2.44}$$

We can think of this traveling wave as a viscous regularization of a shock wave, if we rescale the space and the time variables by $(x', y') = \epsilon(x, y)$ and $t' = \epsilon t$, then Equation (2.44) becomes after dropping the $'$ notation

$$h_t + (h^2 - h^3)_x = -\epsilon^3\nabla \cdot (h^3\nabla\Delta h). \tag{2.45}$$

This equation is a fourth order nonlinear singular perturbation of the conservative law $h_t + (h^2 - h^3)_x = 0$ [278]. Here, ϵ parameter is related to the surface tension coefficient γ [282].

Chapter 3

The multi-component Cahn–Hilliard system

The CH equation is the prototypical continuum model of phase separation in a binary alloy. It was originally derived by Cahn and Hilliard [117] to describe spinodal decomposition and has been widely adopted to model many other physical phenomena such as contact angle and wetting problem [148, 64, 149], gravity and capillary waves [150, 151], mixing [63], pinchoff of liquid-liquid jets [152, 153], Rayleigh-Taylor instability [11, 154, 155, 156], solid tumor growth [157, 158, 159], thermocapillary flow [160, 161], and vesicle dynamics [162].

Most of the technologically important alloys are multi-component systems exhibiting multiple phases in their microstructures. Moreover, one or more of these phases are formed as a result of phase transformations induced during processing. Since performance of these multi-component alloys depends crucially on the morphology of the phases, a fundamental understanding of the kinetics of phase transformations is important for controlling the microstructures of these multi-phase alloys [163].

We consider the evolution of the N-component CH system on a domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. Let $\mathbf{c} = (c_1, \ldots, c_N)$ be the phase variables (i.e., the mole fractions of different components). Clearly the total mole fractions must sum to 1, i.e.,

$$c_1 + \cdots + c_N = 1,$$

so that, admissible states will belong to the Gibbs N-simplex

$$G := \left\{ \mathbf{c} \in \mathbb{R}^N \,\middle|\, \sum_{i=1}^{N} c_i = 1,\ 0 \leq c_i \leq 1 \right\}. \tag{3.1}$$

Without loss of generality, we postulate that the free energy can be written as follows:

$$\mathcal{F}(\mathbf{c}) = \int_\Omega \left(F(\mathbf{c}) + \frac{\epsilon^2}{2} \sum_{i=1}^{N} |\nabla c_i|^2 \right) d\mathbf{x}, \tag{3.2}$$

where $F(\mathbf{c}) = 0.25 \sum_{i=1}^{N} c_i^2(1-c_i)^2$ and $\epsilon > 0$ is the gradient energy coefficient. The natural boundary condition for the N-component CH system is the zero Neumann boundary condition:

$$\nabla c_i \cdot \mathbf{n} = 0 \ \text{ on } \partial\Omega, \tag{3.3}$$

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

The time evolution of $\mathbf{c}$ is governed by the gradient of the energy with respect to the $\dot{H}^{-1}$ inner product under the additional constraint (3.1). This constraint has to hold everywhere at any time. In order to ensure this last constraint, we use a variable Lagrangian multiplier $\beta(\mathbf{c})$ [205]. The time dependence of c_i is given by the following CH equation:

$$\begin{aligned} \frac{\partial c_i}{\partial t} &= M \Delta \mu_i, && (3.4) \\ \mu_i &= f(c_i) - \epsilon^2 \Delta c_i + \beta(\mathbf{c}), \quad \text{for } i = 1, \ldots, N, && (3.5) \end{aligned}$$

where M is a mobility, $f(c) = c(c-0.5)(c-1)$, and $\beta(\mathbf{c}) = -\frac{1}{N} \sum_{i=1}^{N} f(c_i)$. We take $M \equiv 1$ for convenience.

The mass conserving boundary condition for the N-component CH system is

$$\nabla \mu_i \cdot \mathbf{n} = 0 \ \text{ on } \partial\Omega. \tag{3.6}$$

We differentiate the energy $\mathcal{F}$ and the total mass of each phase, $\int_\Omega c_i \, d\mathbf{x}$, to get

$$\frac{d}{dt}\mathcal{F}(t) = -M \int_\Omega \sum_{i=1}^{N} |\nabla \mu_i|^2 d\mathbf{x} \leq 0 \tag{3.7}$$

and

$$\frac{d}{dt}\int_\Omega c_i \, d\mathbf{x} = 0, \tag{3.8}$$

where we used the mass conserving boundary condition (3.6). For a more detail description of derivations of (3.7) and (3.8), please refer to Ref. [195]. Therefore, the total energy is non-increasing in time and the total mass of each phase is conserved. That is

$$\mathcal{F}(t) \leq \mathcal{F}(0) \quad \text{and} \quad \int_\Omega c_i(\mathbf{x}, t) \, d\mathbf{x} = \int_\Omega c_i(\mathbf{x}, 0) \, d\mathbf{x} \quad \text{for } i = 1, \ldots, N.$$

The generalization of the CH equation to multi-component systems first appeared in the literature of De Fontaine [164] and Morral and Cahn [165]. Hoyt [166] extended the CH continuum theory of nucleation to multi-component solutions. Elliott and Luckhaus [167] gave a global existence result under constant mobility and specific assumptions. Eyre [168] studied differences between multi-component and binary alloys and discussed the equilibrium and dynamic behavior of multi-component systems. Elliot and Garcke [169] proved a global existence for multi-component systems when the mobility matrix depends on a concentration. Maier-Paape et al. [170] explained the initial-stage phase separation process in multi-component CH systems through spinodal decomposition.

There are many numerical studies with the binary CH equation [149, 153, 156, 171, 172, 65, 173, 174, 175, 72, 176, 177, 178, 179, 180, 181], ternary [163, 171, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193], and quaternary [194, 195, 196, 58, 197] CH systems. One of the main difficulties in solving the CH system numerically is that the system is fourth-order in space which makes difference stencils very large and introduces a severe time step restriction for stability, i.e., $\Delta t \sim (\Delta x)^4$ for explicit methods. And there is a nonlinearity at the lower order spatial derivatives which can also contribute to numerical stability. Copetti [184] considered an explicit finite element approximation of a model for phase separation in a ternary mixture and took $\Delta x = 1.0$ and $\Delta t = 0.01$ in numerical experiments. In [163], Bhattacharyya and Abinandanan used a semi-implicit Fourier spectral method for solving the ternary CH equations. They treated the linear fourth-order terms implicitly and the non-linear terms explicitly and simulations were carried out using $\Delta x = 1.0$ and $\Delta t = 0.05$. Ma [185] performed numerical simulation of the phase separation kinetics in ternary mixtures with different interfacial properties by means of the cell dynamics approach and in the literature Δx and Δt were both set to be unity. In [182], Δx and Δt were also both set to be unity. Kim and Lowengrub [187] developed a stable, conservative, and second-order accurate fully implicit discretization of the ternary CH system. The authors used a nonlinear multigrid method to efficiently solve the discrete ternary CH system at the implicit time-level and a uniform time step $\Delta t \leq 0.25\Delta x$ for a uniform mesh size Δx. Boyer and Lapuerta [189] used the implicit Galerkin finite elements method with $\Delta x = 8.33 \times 10^{-4}$ and $\Delta t = 0.001$ to solve the ternary CH system.

In numerous papers there are stability restrictions on the time step which cause huge computational costs and make the calculation very inefficient. Therefore we need a scheme that allows the use of a sufficiently large time step without the technical limitations. But, though such an algorithm allows the use of a sufficiently large time step, it seems to be less attractive, because we need to invert an $(N-1) \times (N-1)$

matrix to obtain the solution of N-component CH system (except for the explicit method) and the matrix inversion becomes more and more complicated for increasing N. This problem was shown in previous papers. The authors in [195] developed a nonlinear multigrid method to efficiently solve the discrete N-component CH system at the implicit time level, but an iteration step for the nonlinear multigrid method consists of a $2(N-1) \times 2(N-1)$ matrix inversion.

In the literature [198, 199, 200, 201, 202], to overcome this problem, a variety of numerical approaches have been developed for a large number of components. These approaches can significantly reduce the required computer memory and CPU time, but such approaches are limited to the Allen–Cahn system. Adaptive mesh refinement (AMR) [159, 175, 179, 203, 73] has been used in the CH simulations to increase computational efficiency, but implementation of AMR could be difficult and, in general, AMR is employed only a small number of components.

In this thesis, we present a practically unconditionally gradient stable conservative numerical method for solving the CH system representing a model for phase separation in an N-component mixture. This method allows us to solve the N-component CH system in a decoupled way and significantly reduces the CPU time and memory requirements. We emphasize that while the method will allow us to take arbitrarily large time steps, the accuracy of the numerical solution depends on choosing a small enough time step to resolve the dynamics [204].

Chapter 4

The multi-component Allen–Cahn system

The Allen–Cahn (AC) equation [208] was originally introduced as a phenomenological model for antiphase domain coarsening in a binary alloy. The AC equation and its various modified forms have been applied to a wide range of problems, such as phase transitions [208], image analysis [209, 210], motion by mean curvature flows [211, 212, 213, 214, 215, 216], two-phase fluid flows [217], and crystal growth [218, 219, 220]. Therefore, an efficient and accurate numerical solution of this equation is needed to better understand its dynamics.

We consider the evolution of the N-component AC system on a domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. Let $\mathbf{c} = (c_1, c_2, \ldots, c_N)$ be a vector valued phase-field and $c_i = c_i(\mathbf{x}, t)$ for $i = 1, 2, \ldots, N$ is the mole fraction of the ith component in the mixture as a function of space and time. Clearly the total mole fractions must sum to 1, i.e.,

$$c_1 + \cdots + c_N = 1, \tag{4.1}$$

so that, admissible states will belong to the Gibbs N-simplex

$$G := \left\{ \mathbf{c} \in \mathbb{R}^N \;\middle|\; \sum_{i=1}^{N} c_i = 1,\ 0 \le c_i \le 1 \right\}.$$

Without loss of generality, we postulate that the free energy can be written as follows:

$$\mathcal{F}(\mathbf{c}) = \int_\Omega \left(\frac{F(\mathbf{c})}{\epsilon^2} + \frac{1}{2} \sum_{i=1}^{N} |\nabla c_i|^2 \right) d\mathbf{x},$$

where $F(\mathbf{c}) = 0.25\sum_{i=1}^{N} c_i^2(1-c_i)^2$ is the Helmholtz free energy density [221]. The small positive constant ϵ is the gradient energy coefficient related to the interfacial energy. The natural boundary condition for the N-component AC system is the zero Neumann boundary condition:

$$\nabla c_i \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{4.2}$$

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

The time evolution of $\mathbf{c}$ is governed by the L^2-gradient flow of the total free energy under the additional constraint (4.1), which has to hold everywhere at any time. In order to ensure this last constraint, we use a variable Lagrangian multiplier $\beta(\mathbf{c})$ [222] and set $\frac{\partial F}{\partial \mathbf{c}} = \left(\frac{\partial F}{\partial c_1}, \frac{\partial F}{\partial c_2}, \ldots, \frac{\partial F}{\partial c_N}\right) = \mathbf{f}(\mathbf{c}) = (f(c_1), f(c_2), \ldots, f(c_N))$, where $f(c) = c(c-0.5)(c-1)$. Let $\mathbf{1} = (1,1,\ldots,1) \in \mathbb{R}^N$. Using a general smooth vector valued function $\boldsymbol{\xi}$, we set

$$\mathbf{d} = (d_1, d_2, \ldots, d_N) = \boldsymbol{\xi} - \frac{1}{N}\sum_{i=1}^{N} \xi_i \mathbf{1}, \quad \text{then} \ \sum_{i=1}^{N} d_i = 0.$$

Let

$$\beta(\mathbf{c}) = -\frac{1}{N}\sum_{i=1}^{N} f(c_i).$$

Then, the time dependence of $\mathbf{c}$ is given by the following N-component AC system:

$$\frac{\partial \mathbf{c}(\mathbf{x},t)}{\partial t} = -M\left(\frac{F'(\mathbf{c}(\mathbf{x},t)) + \beta(\mathbf{c})\mathbf{1}}{\epsilon^2} - \Delta\mathbf{c}(\mathbf{x},t)\right), \ \mathbf{x} \in \Omega, \ 0 < t \le T,$$

where the coefficient, M, is a constant mobility and $M \equiv 1$ is taken for convenience.

Numerical simulations of the N-component AC system, using explicit methods, impose severe time step restrictions requiring the use of implicit type methods [118, 204, 223]. Restrictions on a time step cause huge computational costs and make the calculation very inefficient. So some authors [224, 225, 226, 227] used adaptive mesh refinement (AMR) methods to increase computational efficiency in the AC simulations, but implementation of AMR could be difficult and, in general, AMR is employed only a small number of components.

The nonlinearly stabilized splitting scheme that involves a semi-implicit time and centered difference space discretizations of the binary AC equation is

$$\frac{c_i^{n+1}-c_i^n}{\Delta t}=\frac{0.25c_i^n-0.75c_i^{n+1}+1.5(c_i^{n+1})^2-(c_i^{n+1})^3}{\epsilon^2}+\Delta_h c_i^{n+1} \tag{4.3}$$

for $i=1,\dots,N$ and $n=0,\dots,N_t-1$. In Ref. [228], the authors reported the energy decreasing property for the corresponding discrete problem by using the eigenvalues of the Hessian matrix of the energy functional. And an implicit Euler's scheme of the one-dimensional AC equation is

$$\frac{c_i^{n+1}-c_i^n}{\Delta t}=\frac{-0.5c_i^{n+1}+1.5(c_i^{n+1})^2-(c_i^{n+1})^3}{\epsilon^2}+\Delta_h c_i^{n+1}.$$

This scheme is much better than the simple explicit Euler's scheme. However, the implicit Euler's scheme suffers from instability if a large time step is used [118]. Equation (4.3) can be reformulated as

$$\frac{c_i^{n+1}-c_i^n}{\frac{\Delta t(2\epsilon)^2}{\Delta t+(2\epsilon)^2}}=\frac{-0.5c_i^{n+1}+1.5(c_i^{n+1})^2-(c_i^{n+1})^3}{\epsilon^2}+\Delta_h c_i^{n+1}.$$

This suggests that the unconditionally gradient stable scheme can be considered a time step rescaling of the implicit Euler's scheme. In addition,

$$\Delta t_{\text{equiv}}=\frac{\Delta t(2\epsilon)^2}{\Delta t+(2\epsilon)^2}\le\min(\Delta t,(2\epsilon)^2).$$

That is, the scaled equivalent time step Δt_{equiv} is bounded by $(2\epsilon)^2$, which is a small value.

In this thesis, we propose an unconditionally stable and accurate numerical method for solving the N-component AC system. The proposed method is based on an operator splitting technique, which is the method for treating the individual terms of a partial differential equation independently. The main benefit of operator splitting methods is that the stability depends on the minimum of each term rather than the stability of all terms combined. The proposed scheme for the the N-component AC system involves two steps. First, $\mathbf{c}_t = \Delta\mathbf{c} - (\beta(\mathbf{c})\mathbf{1})/\epsilon^2$ is solved numerically using an implicit Euler's method. Second, $\mathbf{c}_t = -\mathbf{f}(\mathbf{c})/\epsilon^2$ is solved analytically.

Chapter 5

Numerical solutions

5.1. Cahn–Hilliard solver

In this section, we present fully discrete schemes for the CH equation. We consider an unconditionally gradient stable scheme for time discretization introduced by Eyre [118, 204]. For simplicity of exposition, we shall discretize the CH equation in two-dimensional space, i.e., $\Omega = (a,b) \times (c,d)$. One and three-dimensional discretizations are defined analogously. Let N_x and N_y be positive even integers, $h = (b-a)/N_x$ be the uniform mesh size, and $\Omega_h = \{(x_i, y_j) : x_i = (i-0.5)h,\ y_j = (j-0.5)h,\ 1 \leq i \leq N_x,\ 1 \leq j \leq N_y\}$ be the set of cell-centers. Let c_{ij} and μ_{ij} be approximations of $c(x_i, y_j)$ and $\mu(x_i, y_j)$. We define a discrete energy functional by

$$\mathcal{E}^h(\mathbf{c}^n) = \frac{h^2}{4} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} ((c_{ij}^n)^2 - 1)^2 + \frac{\epsilon^2 h^2}{2} \left(\sum_{i=0}^{N_x} \sum_{j=1}^{N_y} |\nabla_d c_{i+\frac{1}{2},j}^n|^2 + \sum_{i=1}^{N_x} \sum_{j=0}^{N_y} |\nabla_d c_{i,j+\frac{1}{2}}^n|^2 \right).$$

By using the linearly stabilized splitting scheme [204], we present a implicit time and centered difference space discretization of Equations (2.2) and (2.3) without flow:

$$\frac{c_{ij}^{n+1} - c_{ij}^n}{\Delta t} = \Delta_d \mu_{ij}^{n+1}, \tag{5.1}$$

$$\mu_{ij}^{n+1} = 2c_{ij}^{n+1} - \epsilon^2 \Delta_d c_{ij}^{n+1} + f(c_{ij}^n) - 2c_{ij}^n, \tag{5.2}$$

where $f(c) = F'(c)$.

Eyre [118, 204] proved that if $\mathbf{c}^{n+1}$ is the solution of Equations (5.1) and (5.2) with a given $\mathbf{c}^n$, then

$$\mathcal{E}^h(\mathbf{c}^{n+1}) \leq \mathcal{E}^h(\mathbf{c}^n).$$

Furihata et al. [284] have examined the boundedness of the solution of a finite difference scheme [285] using discretized Lyapunov functional. Furihata [283] has shown that the decrease of the total energy implies boundedness of discretized Sobolev norm of the solution, independent Δt and Δx, unconditionally.

We show that boundedness of the solution of finite difference scheme using the decrease of the discrete total energy functional. The decrease of the discrete total energy functional means that the numerical solution for the CH equation is pointwise bounded. If $\mathbf{c}^n$ is a numerical solution for the discrete Equations (5.1) and (5.2), then there exists a constant K, independent of n, such that

$$\|\mathbf{c}^n\|_\infty \leq K. \tag{5.3}$$

We prove Equation (5.3) by a contradiction. Assume on the contrary that there is an integer n_K, dependent on K, such that $\|\mathbf{c}^{n_K}\|_\infty > K$ for all K. Then there is an index i $(1 \leq i \leq N)$ such that $|c_i^{n_K}| > K$. Let K be the largest solution of $hF(K) = \mathcal{E}^h(\mathbf{c}^0)$, i.e., $K = \sqrt{1 + 2\sqrt{\mathcal{E}^h(\mathbf{c}^0)/h}}$. Note that $K \geq 1$. Then, $F(c)$ is a strictly increasing function on (K, ∞) (see Figure 5.1). Since the total energy is non-increasing, we have $\mathcal{E}^h(\mathbf{c}^0) = hF(K) < hF(|c_i^{n_K}|) \leq \mathcal{E}^h(\mathbf{c}^{n_K}) \leq \mathcal{E}^h(\mathbf{c}^0)$ [286]. This contradiction implies that Equation (5.3) should be satisfied.

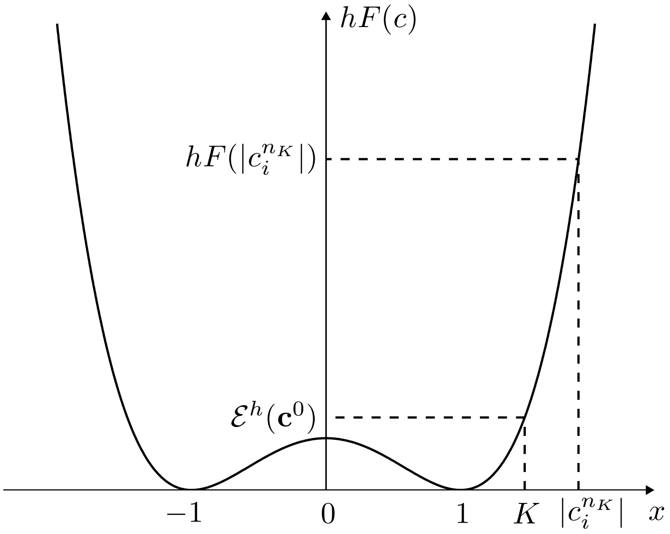


FIGURE 5.1. Graph of $hF(c)$.

5.2. Multigrid methods

In this section, we develop a multigrid method to solve the linear discrete system (5.1) and (5.2) at the implicit time level. A pointwise Gauss-Seidel relaxation scheme is used as the smoother in the multigrid method. See the reference text [71] for additional details. We use the same notations as this reference text. Let us rewrite Equations (5.1) and (5.2) as follows.

$$L(c^{n+1}, \mu^{n+1}) = (\phi^n, \psi^n),$$

where the linear system operator (L) is defined as

$$L(c^{n+1}, \mu^{n+1}) = \left(\frac{c^{n+1}}{\Delta t} - \Delta_d \mu^{n+1},\ \mu^{n+1} - 2c^{n+1} + \epsilon^2 \Delta_d c^{n+1} \right)$$

and the source term is $(\phi^n, \psi^n) = (c^n/\Delta t,\ f(c^n) - 2c^n)$. In the following description of one multigrid cycle, we assume a sequence of grids Ω_k (Ω_{k-1} is coarser than Ω_k by a factor of 2). Given the numbers, ν_1 and ν_2, of pre- and post- smoothing relaxation sweeps, an iteration step for the multigrid method using the V-cycle is formally written as follows:

Multigrid cycle

$\{c_k^{n+1,m+1}, \mu_k^{n+1,m+1}\} = MGcycle(k, c_k^{n+1,m}, \mu_k^{n+1,m}, L_k, \phi_k^n, \psi_k^n, \nu_1, \nu_2)$ on Ω_k grid.

That is, $\{c_k^{n+1,m}, \mu_k^{n+1,m}\}$ and $\{c_k^{n+1,m+1}, \mu_k^{n+1,m+1}\}$ are the approximations of $\{c_k^{n+1}(x_i, y_j), \mu_k^{n+1}(x_i, y_j)\}$ before and after an MGcycle.

Step 1) Presmoothing

$$\{\bar{c}_k^{n+1,m}, \bar{\mu}_k^{n+1,m}\} = SMOOTH^{\nu_1}(c_k^{n+1,m}, \mu_k^{n+1,m}, L_k, \phi_k^n, \psi_k^n) \text{ on } \Omega_k \text{ grid.}$$

This means performing ν_1 smoothing steps. Let us discretize Equations (5.1) and (5.2) as a Gauss-Seidel type.

$$\frac{\bar{c}_{k,ij}^{n+1,m}}{\Delta t} + \frac{4\bar{\mu}_{k,ij}^{n+1,m}}{h^2} = \phi_{k,ij}^n + \frac{\bar{\mu}_{k,i-1,j}^{n+1,m} + \mu_{k,i+1,j}^{n+1,m} + \bar{\mu}_{k,i,j-1}^{n+1,m} + \mu_{k,i,j+1}^{n+1,m}}{h^2}, \tag{5.4}$$

$$-\left(2 + \frac{4\epsilon^2}{h^2}\right)\bar{c}_{k,ij}^{n+1,m} + \bar{\mu}_{k,ij}^{n+1,m} = \psi_{k,ij}^n - \epsilon^2\frac{\bar{c}_{k,i-1,j}^{n+1,m} + c_{k,i+1,j}^{n+1,m} + \bar{c}_{k,i,j-1}^{n+1,m} + c_{k,i,j+1}^{n+1,m}}{h^2} \tag{5.5}$$

One SMOOTH relaxation operator step consists of solving the system (5.4) and (5.5) by a 2×2 matrix inversion for each ij.

Step 2) Coarse grid correction

- Compute the defect: $(\bar{d}_{1k}^m, \bar{d}_{2k}^m) = (\phi_k^n, \psi_k^n) - L_k(\bar{c}_k^{n+1,m}, \bar{\mu}_k^{n+1,m})$.
- Restrict the defect: $(\bar{d}_{1,k-1}^m, \bar{d}_{2,k-1}^m) = I_k^{k-1}(\bar{d}_{1k}^m, \bar{d}_{2k}^m)$.
- Compute an approximate solution $\{\bar{c}_{k-1}^m, \bar{\mu}_{k-1}^m\}$ of the coarse grid equation on Ω_{k-1}:

$$L_{k-1}(c_{k-1}^{n+1,m}, \mu_{k-1}^{n+1,m}) = (\bar{d}_{1,k-1}^m, \bar{d}_{2,k-1}^m). \tag{5.6}$$

If $k = 1$, we apply the smoothing procedure in *Step 1)* to obtain the approximate solution. If $k > 1$, we solve (5.6) by performing a k-grid cycle using the zero grid function as an initial approximation:

$$\{\hat{v}_{1,k-1}^{n+1,m}, \hat{v}_{2,k-1}^{n+1,m}\} = MGcycle(k-1, \mathbf{0}, \mathbf{0}, L_{k-1}, \bar{d}_{1,k-1}^m, \bar{d}_{2,k-1}^m, \nu_1, \nu_2).$$

- Interpolate the correction: $(\hat{v}_{1k}^{n+1,m}, \hat{v}_{2k}^{n+1,m}) = I_{k-1}^k(\hat{v}_{1,k-1}^{n+1,m}, \hat{v}_{2,k-1}^{n+1,m})$.
- Compute the corrected approximation on Ω_k:

$$c_k^{m,\text{ after } CGC} = \bar{c}_k^{n+1,m} + \hat{v}_{1k}^{n+1,m}, \ \mu_k^{m,\text{ after } CGC} = \bar{\mu}_k^{n+1,m} + \hat{v}_{2k}^{n+1,m}.$$

Step 3) Postsmoothing:

$$\{c_k^{n+1,m+1}, \mu_k^{n+1,m+1}\}$$
$$= SMOOTH^{\nu_2}(c_k^{m,\text{ after } CGC}, \mu_k^{m,\text{ after } CGC}, L_k, \phi_k^n, \psi_k^n) \text{ on } \Omega_k \text{ grid.}$$

This completes the description of an MGcycle. Figure 5.2 shows the description of an MGcycle.

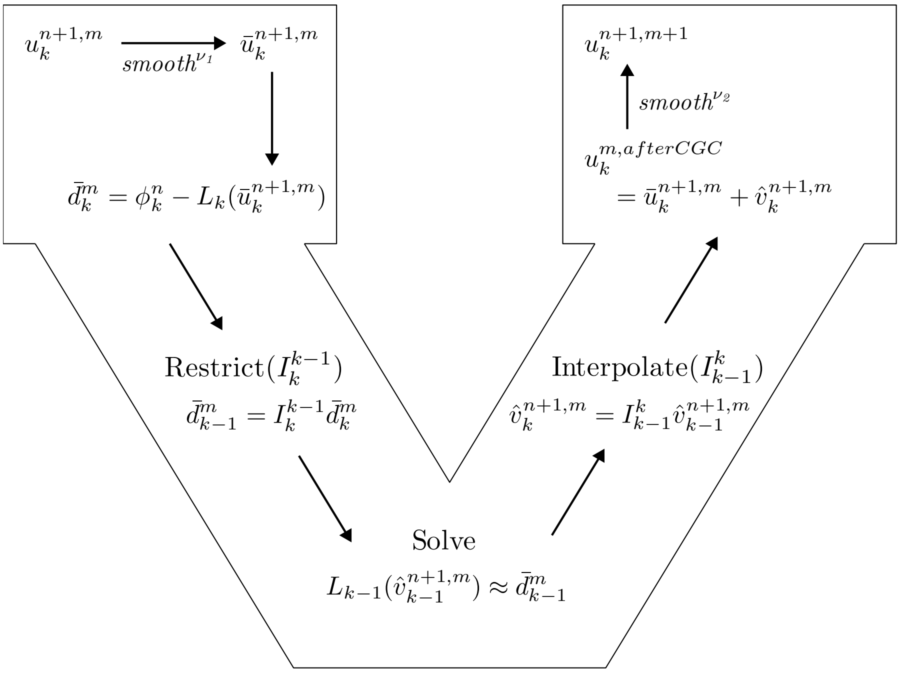


FIGURE 5.2. The description of an MGcycle.

Describing an algorithm for a discrete system is one thing and actual implementation is another. Especially, using recursive function for a multigrid method is non-trivial task for the beginners. We made the code as simple as possible. Therefore, it needs a modification for accounting other features of the equations such as different boundary conditions, variable mobilities, three dimensional extension, and more than two components. These modifications can be done in a straightforward manner.

5.3. Navier–Stokes solver

The simultaneous solution of the large number of discrete equations arising from (2.6) and (2.7) is very costly, especially in three spatial dimensions [69]. An efficient approximation can be obtained by decoupling the solution of the momentum equations from the solution of the continuity equation by a projection method [67, 68]. We will focus on describing the idea in two-dimensions. A staggered marker-and-cell (MAC) mesh of Harlow and Welch [70] is used in which pressure and phase fields are stored at cell centers and velocities at cell interfaces (see Figure 5.3).

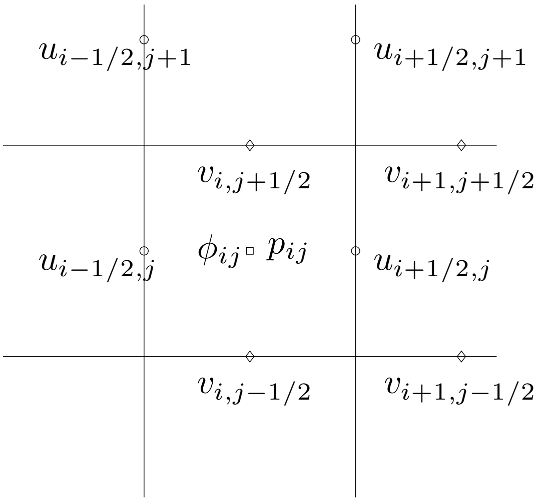


FIGURE 5.3. Velocities are defined at cell boundaries while the pressure and phase field are defined at the cell centers.

Let a computational domain be partitioned in Cartesian geometry into a uniform mesh with mesh spacing h. The center of each cell, Ω_{ij}, is located at $(x_i, y_j) = ((i-0.5)h, (j-0.5)h)$ for $i = 1, \cdots, N_x$ and $j = 1, \cdots, N_y$. N_x and N_y are the numbers of cells in x and y-directions, respectively. The cell vertices are located at $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) = (ih, jh)$.

At the beginning of each time step, given $\mathbf{u}^n$, ϕ^n, and p^n, we want to find $\mathbf{u}^{n+1}$, ϕ^{n+1}, and p^{n+1} which solve the following temporal discretization of dimensionless form of the Equations (2.18) - (2.21) of motion:

$$\begin{aligned}
\rho^n \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= -\rho^n (\mathbf{u} \cdot \nabla_d \mathbf{u})^n - \nabla_d p^{n+1} + \frac{1}{Re} \Delta_d \mathbf{u}^n + \frac{1}{We} \mathbf{SF}^n + \frac{\rho^n}{Fr} \mathbf{g}, \\
\nabla_d \cdot \mathbf{u}^{n+1} &= 0, \\
\frac{\phi^{n+1} - \phi^n}{\Delta t} &= \frac{1}{Pe} \Delta_d \nu^{n+1} - \frac{1}{Pe} \Delta_d \phi^n - \nabla_d \cdot (\phi \mathbf{u})^n, & (5.7) \\
\nu^{n+1} &= (\phi^{n+1})^3 - \epsilon^2 \Delta_d \phi^{n+1}, & (5.8)
\end{aligned}$$

where $\rho^n = \rho(\phi^n)$ and $\mathbf{g} = (0, -1)$.

The outline of the main procedures in one time step is:

Step 1. Initialize $\mathbf{u}^0$ to be the divergence-free velocity field and ϕ^0.

Step 2. Solve an intermediate velocity field, $\tilde{\mathbf{u}}$, which generally does not satisfy the incompressible condition, without the pressure gradient term,

$$\frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\Delta t} = -\mathbf{u}^n \cdot \nabla_d \mathbf{u}^n + \frac{1}{\rho^n Re} \Delta_d \mathbf{u}^n + \frac{1}{\rho^n We} \mathbf{SF}^n + \frac{1}{Fr} \mathbf{g}.$$

The resulting finite difference equations are written out explicitly. They take the form

$$\begin{aligned}
\tilde{u}_{i+\frac{1}{2},j} &= u^n_{i+\frac{1}{2},j} - \Delta t (u u_x + v u_y)^n_{i+\frac{1}{2},j} + \frac{\Delta t}{\rho^n_{i+\frac{1}{2},j} We} SF^{x-edge}_{i+\frac{1}{2},j} \\
&+ \frac{\Delta t}{h^2 \rho^n_{i+\frac{1}{2},j} Re} \left(u^n_{i+\frac{3}{2},j} + u^n_{i-\frac{1}{2},j} - 4u^n_{i+\frac{1}{2},j} + u^n_{i+\frac{3}{2},j+1} + u^n_{i-\frac{1}{2},j-1} \right), \\
\tilde{v}_{i,j+\frac{1}{2}} &= v^n_{i,j+\frac{1}{2}} - \Delta t (u v_x + v v_y)^n_{i,j+\frac{1}{2}} + \frac{\Delta t}{\rho^n_{i,j+\frac{1}{2}} We} SF^{y-edge}_{i,j+\frac{1}{2}} - \frac{\Delta t}{Fr} \\
&+ \frac{\Delta t}{h^2 \rho^n_{i,j+\frac{1}{2}} Re} \left(v^n_{i,j+\frac{3}{2}} + v^n_{i,j-\frac{1}{2},j} - 4v^n_{i,j+\frac{1}{2}} + v^n_{i+1,j+\frac{3}{2}} + v^n_{i-1,j-\frac{1}{2},j} \right),
\end{aligned}$$

where the advection terms, $(uu_x + vu_y)^n_{i+\frac{1}{2},j}$ and $(uv_x + vv_y)^n_{i,j+\frac{1}{2}}$, are defined by

$$\begin{aligned}
(uu_x + vu_y)^n_{i+\frac{1}{2},j} &= u^n_{i+\frac{1}{2},j}\bar{u}^n_{x_{i+\frac{1}{2},j}} \\
&\quad + \frac{v^n_{i,j-\frac{1}{2}} + v^n_{i+1,j-\frac{1}{2}} + v^n_{i,j+\frac{1}{2}} + v^n_{i+1,j+\frac{1}{2}}}{4}\bar{u}^n_{y_{i+\frac{1}{2},j}}, \\
(uv_x + vv_y)^n_{i,j+\frac{1}{2}} &= \frac{u^n_{i-\frac{1}{2},j} + u^n_{i-\frac{1}{2},j+1} + u^n_{i+\frac{1}{2},j} + u^n_{i+\frac{1}{2},j+1}}{4}\bar{v}^n_{x_{i,j+\frac{1}{2}}} \\
&\quad + v^n_{i,j+\frac{1}{2}}\bar{v}^n_{y_{i,j+\frac{1}{2}}}.
\end{aligned}$$

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. The procedure is

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

and

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

The quantities $\bar{v}^n_{x_{i,j+\frac{1}{2}}}$ and $\bar{v}^n_{y_{i,j+\frac{1}{2}}}$ are computed in a similar manner. The surface tension force terms, $SF^{x-edge}_{i+\frac{1}{2},j}$ and $SF^{y-edge}_{i,j+\frac{1}{2}}$, are computed using the procedure derived in Ref. [190].

Then, we solve the following equations for the advanced pressure field at $(n + 1)$ time step.

$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}}{\Delta t} = -\frac{1}{\rho^n}\nabla_d p^{n+1}, \tag{5.9}$$

$$\nabla_d \cdot \mathbf{u}^{n+1} = 0. \tag{5.10}$$

With application of the divergence operator to Equation (5.9), we find that the Poisson equation for the pressure at the advanced time $(n + 1)$.

$$\nabla_d \cdot \left(\frac{1}{\rho^n}\nabla_d p^{n+1}\right) = \frac{1}{\Delta t}\nabla_d \cdot \tilde{\mathbf{u}}, \tag{5.11}$$

where we have made use of the Equation (5.10) and the terms are defined as in the following.

$$\begin{aligned}
\nabla_d \cdot \left(\frac{1}{\rho^n}\nabla_d p_{ij}^{n+1}\right) &= \frac{\frac{1}{\rho^n_{i+\frac{1}{2},j}}p^{n+1}_{i+1,j} + \frac{1}{\rho^n_{i-\frac{1}{2},j}}p^{n+1}_{i-1,j} + \frac{1}{\rho^n_{i,j+\frac{1}{2}}}p^{n+1}_{i,j+1} + \frac{1}{\rho^n_{i,j-\frac{1}{2}}}p^{n+1}_{i,j-1}}{h^2} \\
&\quad - \frac{\frac{1}{\rho^n_{i+\frac{1}{2},j}} + \frac{1}{\rho^n_{i-\frac{1}{2},j}} + \frac{1}{\rho^n_{i,j+\frac{1}{2}}} + \frac{1}{\rho^n_{i,j-\frac{1}{2}}}}{h^2}p^{n+1}_{ij}, \\
\nabla_d \cdot \tilde{\mathbf{u}}_{ij} &= \frac{\tilde{u}_{i+\frac{1}{2},j} - \tilde{u}_{i-\frac{1}{2},j}}{h} + \frac{\tilde{v}_{i,j+\frac{1}{2}} - \tilde{v}_{i,j-\frac{1}{2}}}{h},
\end{aligned}$$

where $\rho^n_{i+\frac{1}{2},j} = (\rho^n_{ij} + \rho^n_{i+1,j})/2$ and the other terms are similarly defined.

The boundary condition for the pressure is

$$\mathbf{n}\cdot\nabla_d p^{n+1} = \mathbf{n}\cdot\left(-\rho^n\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} - \rho^n(\mathbf{u}\cdot\nabla_d\mathbf{u})^n + \frac{1}{Re}\Delta_d\mathbf{u}^n + \frac{1}{We}\mathbf{SF}^n + \frac{\rho^n}{Fr}\mathbf{g}\right),$$

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

In our application of the phase-field to the Rayleigh-Taylor instability, we will use a periodic boundary condition to vertical boundaries and no slip boundary condition to the top and bottom domain. Therefore,

$$\mathbf{n}\cdot\nabla_d p^{n+1} = \mathbf{n}\cdot\left(\frac{1}{We}\mathbf{SF}^n + \frac{\rho^n}{Fr}\mathbf{g}\right).$$

The resulting linear system of Equation (5.11) is solved using a multigrid method [71], specifically, V-cycles with a Gauss-Seidel relaxation. Then the divergence-free normal velocities u^{n+1} and v^{n+1} are defined by

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

We implement the unconditionally gradient stable scheme in Equations (5.7)-(5.8) with a nonlinear multigrid method. For a detailed description of the numerical method used in solving these equations, please refer to Refs. [72, 73].

Since we are interested in long time simulations, mass conservation is an important factor. Therefore, we use a conservative discretization of the convective part of the phase-field equation (5.7).

$$\begin{aligned}((\phi u)_x + (\phi v)_y)_{ij}^n &= \frac{u_{i+\frac{1}{2},j}^n(\phi_{i+1,j}^n + \phi_{ij}^n) - u_{i-\frac{1}{2},j}^n(\phi_{ij}^n + \phi_{i-1,j}^n)}{2h} \\ &+ \frac{v_{i,j+\frac{1}{2}}^n(\phi_{i,j+1}^n + \phi_{ij}^n) - u_{i,j-\frac{1}{2}}^n(\phi_{ij}^n + \phi_{i,j-1}^n)}{2h}.\end{aligned}$$

These complete the one time step. Other projection type methods [67] can be used.

5.4. Modified Allen–Cahn solver

In this section, we propose a robust hybrid numerical method for crystal growth simulation. For simplicity of exposition we shall discretize Equations (2.32) and (2.33) in two-dimensional space, i.e., $\Omega = (-l_1, l_1) \times (-l_2, l_2)$. Let N_x and N_y be positive even integers, $h = 2l_1/N_x$ be the uniform mesh size, and $\Omega_h = \{(x_i, y_j) : x_i = (i - 0.5)h,\ y_j = (j - 0.5)h,\ 1 \le i \le N_x,\ 1 \le j \le N_y\}$ be the set of cell-centers. Let c_{ij}^n be approximations of $c(x_i, y_j, n\Delta t)$, where $\Delta t = T/N_t$ is the time step, T is the final time, and N_t is the total number of time steps. The discrete differentiation operator is $\nabla_d c_{ij} = (c_{i+1,j} - c_{i-1,j}, c_{i,j+1} - c_{i,j-1})/(2h)$. We then define the discrete Laplacian by $\Delta_d c_{ij} = (c_{i+1,j} + c_{i-1,j} - 4c_{ij} + c_{i,j+1} + c_{i,j-1})/h^2$. We discretize Equations (2.32) and (2.33):

$$\begin{aligned}
\epsilon^2(\phi^n)\frac{c^{n+1} - c^n}{\Delta t} &= \epsilon^2(\phi^n)\Delta_d c^{n+1,2} + 2\epsilon(\phi^n)\nabla_d\epsilon(\phi^n) \cdot \nabla_d c^n \\
&\quad - F'(c^{n+1}) - 4\lambda U^n F(c^{n+1,1}) \\
&\quad - \big(\epsilon'(\phi) \cdot \epsilon(\phi) c_y\big)_x^n + \big(\epsilon'(\phi) \cdot \epsilon(\phi) c_x\big)_y^n, \\
\frac{U^{n+1} - U^n}{\Delta t} &= D\Delta_d U^{n+1} + \frac{c^{n+1} - c^n}{2\Delta t},
\end{aligned}$$

where $F(c) = 0.25(c^2 - 1)^2$ and $F'(c) = c(c^2 - 1)$. Here $c^{n+1,k}$ for $k = 1, 2$ are defined in the operator splitting scheme. We propose the following operator splitting scheme:

$$\begin{aligned}
\epsilon^2(\phi^n)\frac{c^{n+1,1} - c^n}{\Delta t} &= 2\epsilon(\phi^n)\nabla_d\epsilon(\phi^n) \cdot \nabla_d c^n \\
&\quad - \big(\epsilon'(\phi) \cdot \epsilon(\phi) c_y\big)_x^n + \big(\epsilon'(\phi) \cdot \epsilon(\phi) c_x\big)_y^n, \\
\epsilon^2(\phi^n)\frac{c^{n+1,2} - c^{n+1,1}}{\Delta t} &= \epsilon^2(\phi^n)\Delta_d c^{n+1,2} - 4\lambda U^n F(c^{n+1,1}), \\
\epsilon^2(\phi^n)\frac{c^{n+1} - c^{n+1,2}}{\Delta t} &= -F'(c^{n+1}). \qquad (5.12)
\end{aligned}$$

We can solve Equation (5.12) analytically by the method of separation of variables [252, 253]. The solution is given as follows:

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

Finally, the proposed scheme can be written as follows:

$$\begin{aligned}
\epsilon(\phi^n)\frac{c^{n+1,1} - c^n}{\Delta t} &= 2\epsilon(\phi^n)_x c_x^n + 2\epsilon(\phi^n)_y c_y^n - \left(\epsilon'(\phi)\cdot c_y\right)_x^n + \left(\epsilon'(\phi)\cdot c_x\right)_y^n, \\
\epsilon^2(\phi^n)\frac{c^{n+1,2} - c^{n+1,1}}{\Delta t} &= \epsilon^2(\phi^n)\Delta_d c^{n+1,2} - 4\lambda U^n F(c^{n+1,1}), \qquad (5.13) \\
c^{n+1} &= \frac{c^{n+1,2}}{\sqrt{e^{-\frac{2\Delta t}{\epsilon^2(\phi^n)}} + (c^{n+1,2})^2\left(1 - e^{-\frac{2\Delta t}{\epsilon^2(\phi^n)}}\right)}}, \\
\frac{U^{n+1} - U^n}{\Delta t} &= D\Delta_d U^{n+1} + \frac{c^{n+1} - c^n}{2\Delta t}. \qquad (5.14)
\end{aligned}$$

Equations (5.13) and (5.14) can be solved by a multigrid method [233, 71].

5.5. The multi-component Cahn–Hilliard solver

The numerical solution of the N-component CH system uses a second-order accurate spatial discretization and a nonlinear splitting time stepping method. For simplicity and clarity of exposition, we will present the numerical method in 2D, but the extension to 3D is straightforward. Note that we only need to solve equations with $c_1, c_2, \ldots, c_{N-1}$ since $c_N = 1 - c_1 - c_2 - \cdots - c_{N-1}$ for the N-component CH system. Let $\mathbf{c} = (c_1, c_2, \ldots, c_{N-1})$ and $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_{N-1})$.

5.5.1. Discretization. Let $\Omega = (a, b) \times (c, d)$ be the computational domain in 2D, N_x and N_y be positive even integers, $h = (b - a)/N_x$ be the uniform mesh size, and $\Omega_h = \{(x_i, y_j) : x_i = (i - 0.5)h,\ y_j = (j - 0.5)h,\ 1 \le i \le N_x,\ 1 \le j \le N_y\}$ be the set of cell-centers.

Let $\mathbf{c}_{ij}$ and $\boldsymbol{\mu}_{ij}$ be approximations of $\mathbf{c}(x_i, y_j)$ and $\boldsymbol{\mu}(x_i, y_j)$. We first implement the zero Neumann boundary condition (3.3) and (3.6) by requiring that

$$\begin{aligned} D_x \mathbf{c}_{\frac{1}{2},j} &= D_x \mathbf{c}_{N_x+\frac{1}{2},j} = D_y \mathbf{c}_{i,\frac{1}{2}} = D_y \mathbf{c}_{i,N_y+\frac{1}{2}} = \mathbf{0}, \\ D_x \boldsymbol{\mu}_{\frac{1}{2},j} &= D_x \boldsymbol{\mu}_{N_x+\frac{1}{2},j} = D_y \boldsymbol{\mu}_{i,\frac{1}{2}} = D_y \boldsymbol{\mu}_{i,N_y+\frac{1}{2}} = \mathbf{0}, \end{aligned}$$

where the discrete differentiation operators are

$$D_x \mathbf{c}_{i+\frac{1}{2},j} = (\mathbf{c}_{i+1,j} - \mathbf{c}_{ij})/h \quad \text{and} \quad D_y \mathbf{c}_{i,j+\frac{1}{2}} = (\mathbf{c}_{i,j+1} - \mathbf{c}_{ij})/h.$$

We then define the discrete Laplacian by

$$\Delta_h \mathbf{c}_{ij} = (D_x \mathbf{c}_{i+\frac{1}{2},j} - D_x \mathbf{c}_{i-\frac{1}{2},j} + D_y \mathbf{c}_{i,j+\frac{1}{2}} - D_y \mathbf{c}_{i,j-\frac{1}{2}})/h$$

and the discrete L^2 inner product by

$$(\mathbf{c}, \mathbf{d})_h = h^2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} (c_{1ij} d_{1ij} + c_{2ij} d_{2ij} + \cdots + c_{N-1_{ij}} d_{N-1_{ij}}). \tag{5.15}$$

We also define a discrete norm associated with (5.15) as

$$\|\mathbf{c}\|^2 = (\mathbf{c}, \mathbf{c})_h.$$

We define $\mathbf{f}(\mathbf{c})$ and $\mathbf{1}$ to $\mathbf{f}(\mathbf{c}) = (f(c_1), f(c_2), \ldots, f(c_{N-1}))$ and $\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{R}^{N-1}$. We discretize Equations (3.4) and (3.5) in time by a nonlinear splitting algorithm:

$$\begin{aligned}
\frac{\mathbf{c}_{ij}^{n+1} - \mathbf{c}_{ij}^{n}}{\Delta t} &= \Delta_h \boldsymbol{\nu}_{ij}^{n+1} + \Delta_h \left(\beta(\mathbf{c}_{ij}^{n})\mathbf{1} - \frac{1}{4}\mathbf{c}_{ij}^{n} \right), \qquad (5.16)\\
\boldsymbol{\nu}_{ij}^{n+1} &= \boldsymbol{\varphi}(\mathbf{c}_{ij}^{n+1}) - \epsilon^2 \Delta_h \mathbf{c}_{ij}^{n+1},
\end{aligned}$$

where the nonlinear function $\boldsymbol{\varphi}(\mathbf{c}) = (\varphi_1(\mathbf{c}), \varphi_2(\mathbf{c}), \ldots, \varphi_{N-1}(\mathbf{c})) = \mathbf{f}(\mathbf{c}) + \mathbf{c}/4$.

In Equation (5.16), the variable Lagrangian multiplier $\beta(\mathbf{c})$ is determined by the solutions at time level n. By treating $\beta(\mathbf{c})$ explicitly, there is no relation between the solutions at time level $n+1$. Thus the N-component CH system can be solved in a decoupled way, i.e.,

$$\begin{aligned}
\frac{c_{k,ij}^{\ n+1} - c_{k,ij}^{\ n}}{\Delta t} &= \Delta_h \nu_{k,ij}^{\ n+1} + \Delta_h \left(\beta(\mathbf{c}_{ij}^{n}) - \frac{1}{4} c_{k,ij}^{\ n} \right),\\
\nu_{k,ij}^{\ n+1} &= \varphi(c_{k,ij}^{\ n+1}) - \epsilon^2 \Delta_h c_{k,ij}^{\ n+1}, \quad \text{for } k = 1, 2, \ldots, N-1.
\end{aligned}$$

This means that we only solve the binary CH equation $N-1$ times to solve the N-component CH system. The above discrete system is solved by a nonlinear multigrid method [71].

5.6. The multi-component Allen–Cahn solver

In this section, an unconditionally stable hybrid method is proposed for the N-component AC system. The unconditional stability means that arbitrarily large time steps can be used in the numerical algorithm. For simplicity and clarity of exposition, we will present the numerical method in 2D, but the extension to 3D is straightforward. Note that we only need to solve equations with $c_1, c_2, \ldots, c_{N-1}$ since $c_N = 1 - c_1 - c_2 - \cdots - c_{N-1}$ for the N-component AC system. Let $\mathbf{c} = (c_1, c_2, \ldots, c_{N-1})$.

Let $\Omega = (a, b) \times (c, d)$ be the computational domain in 2D, N_x and N_y be positive even integers, $h = (b - a)/N_x$ be the uniform mesh size, and $\Omega_h = \{(x_i, y_j) : x_i = (i - 0.5)h,\ y_j = (j - 0.5)h,\ 1 \le i \le N_x,\ 1 \le j \le N_y\}$ be the set of cell-centers. Let $\mathbf{c}_{ij}$ be an approximation of $\mathbf{c}(x_i, y_j)$. We first implement the zero Neumann boundary condition (4.2) by requiring that

$$D_x \mathbf{c}_{\frac{1}{2},j} = D_x \mathbf{c}_{N_x+\frac{1}{2},j} = D_y \mathbf{c}_{i,\frac{1}{2}} = D_y \mathbf{c}_{i,N_y+\frac{1}{2}} = \mathbf{0},$$

where the discrete differentiation operators are

$$D_x \mathbf{c}_{i+\frac{1}{2},j} = (\mathbf{c}_{i+1,j} - \mathbf{c}_{ij})/h \quad \text{and} \quad D_y \mathbf{c}_{i,j+\frac{1}{2}} = (\mathbf{c}_{i,j+1} - \mathbf{c}_{ij})/h.$$

We then define the discrete Laplacian by

$$\Delta_h \mathbf{c}_{ij} = (D_x \mathbf{c}_{i+\frac{1}{2},j} - D_x \mathbf{c}_{i-\frac{1}{2},j} + D_y \mathbf{c}_{i,j+\frac{1}{2}} - D_y \mathbf{c}_{i,j-\frac{1}{2}})/h$$

and the discrete L^2 inner product by

$$(\mathbf{c}, \mathbf{d})_h = h^2 \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} (c_{1ij} d_{1ij} + c_{2ij} d_{2ij} + \cdots + c_{N-1ij} d_{N-1ij}). \tag{5.17}$$

We also define a discrete norm associated with (5.17) as

$$\|\mathbf{c}\|^2 = (\mathbf{c}, \mathbf{c})_h.$$

We define $\mathbf{f}(\mathbf{c})$ and $\mathbf{1}$ to $\mathbf{f}(\mathbf{c}) = (f(c_1), f(c_2), \ldots, f(c_{N-1}))$ and $\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{R}^{N-1}$. We propose the following operator splitting scheme, which is an unconditionally stable hybrid scheme:

$$\begin{aligned} \frac{\mathbf{c}_{ij}^{*} - \mathbf{c}_{ij}^{n}}{\Delta t} &= \Delta_h \mathbf{c}_{ij}^{*} - \frac{\beta(\mathbf{c}_{ij}^{n})\mathbf{1}}{\epsilon^2}, && (5.18) \\ \frac{\mathbf{c}_{ij}^{n+1} - \mathbf{c}_{ij}^{*}}{\Delta t} &= -\frac{\mathbf{f}(\mathbf{c}_{ij}^{n+1})}{\epsilon^2}. && (5.19) \end{aligned}$$

Equation (5.18) is an implicit Euler's scheme for $\mathbf{c}_t = \Delta \mathbf{c} - (\beta(\mathbf{c})\mathbf{1})/\epsilon^2$ with an initial condition $\mathbf{c}^n$. Using a von Neumann stability analysis [229], it can be seen that this scheme is unconditionally stable. The resulting implicit discrete system can be solved by a fast solver such as a multigrid method [230, 71]. Equation (5.19) can be considered as an approximation of the equation

$$\mathbf{c}_t = -\frac{\mathbf{f}(\mathbf{c})}{\epsilon^2} \tag{5.20}$$

by an implicit Euler's method with the initial condition $\mathbf{c}^*$. We can solve Equation (5.20) analytically by the method of separation of variables [231]. The solution is given as follows:

$$\mathbf{c}_{ij}^{n+1} = 0.5 + \frac{\mathbf{c}_{ij}^{*} - 0.5}{\sqrt{e^{\frac{-\Delta t}{2\epsilon^2}} + (2\mathbf{c}_{ij}^{*} - 1)^2 (1 - e^{\frac{-\Delta t}{2\epsilon^2}})}}.$$

The proposed operator splitting algorithm is shown schematically in Fig. 5.4.

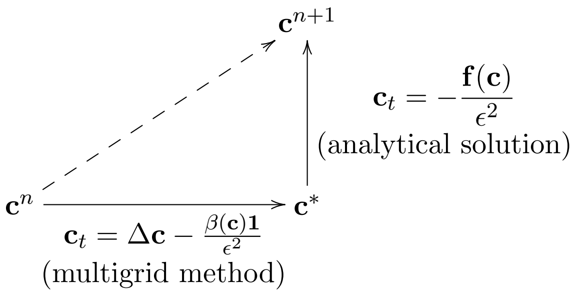


FIGURE 5.4. A hybrid numerical method for the N-component AC system.

In Equation (5.18), the variable Lagrangian multiplier $\beta(\mathbf{c})$ is determined by the solutions at time level n. By treating $\beta(\mathbf{c})$ explicitly, there is no relation between the solutions at time level $*$. Thus the N-component AC system can be solved in a decoupled way, i.e.,

$$\begin{aligned}
\frac{c_{k,ij}^{\;*} - c_{k,ij}^{\;n}}{\Delta t} &= \Delta_h c_{k,ij}^{\;*} - \frac{\beta(\mathbf{c}_{ij}^n)}{\epsilon^2}, \\
c_{k,ij}^{\;n+1} &= 0.5 + \frac{c_{k,ij}^{\;*} - 0.5}{\sqrt{e^{\frac{-\Delta t}{2\epsilon^2}} + (2c_{k,ij}^{\;*} - 1)^2(1 - e^{\frac{-\Delta t}{2\epsilon^2}})}}, \quad \text{for } k = 1, 2, \ldots, N-1.
\end{aligned}$$

This means that we only solve the binary AC equation $N-1$ times to solve the N-component AC system.

Chapter 6

Numerical results

6.1. Numerical experiments for regularized Dirac delta functions

We now present numerical results to show the performance of regularized Dirac delta functions as (1) a postprocessing of the phase-field solution; length of line and circle, Rayleigh-Taylor instability, deformation of a circle by a single vortex, a three-dimensional deformation field, and triply periodic minimal surfaces, and (2) a representation of the surface tension force; reduction of spurious velocities and pressure jump across the drop. And we also demonstrate relation between the interfacial width and grid size. Across the interfacial region, the concentration field varies from 0.1 to 0.9 over a distance of approximately $4\sqrt{2}\epsilon\tanh^{-1}(0.8)$. Therefore, if we want this value to be approximately m (> 0) grid points, the ϵ value needs to be taken as $\epsilon_m = hm/[4\sqrt{2}\tanh^{-1}(0.8)]$. Figure 6.1 shows the concentration $c(x) = 0.5(1 + \tanh(x/(2\sqrt{2}\epsilon_m)))$ with $m = 4$, 8, 12, and 16. We will use various ϵ that is suitable for each problem.

6.1.1. The Dirac delta function as a postprocessing of the phase-field solution. In this section, we investigate the performance of each delta function as an interface length or a surface area calculation tool. The interface length $\mathscr{L}$ and surface

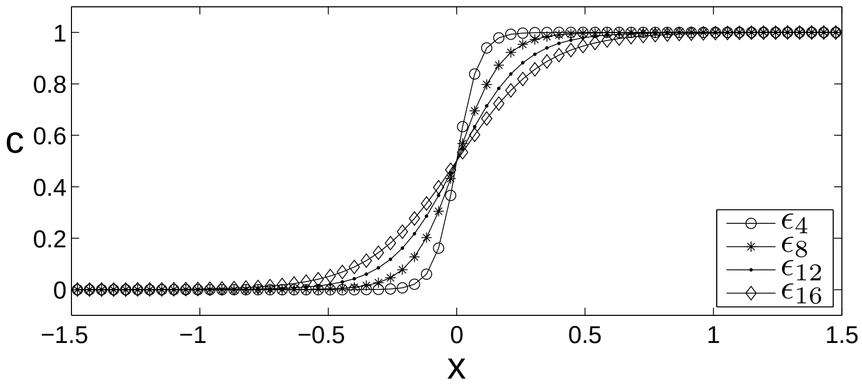


FIGURE 6.1. The concentration $c(x) = 0.5(1 + \tanh(x/(2\sqrt{2}\epsilon_m)))$ with $m = 4$, 8, 12, and 16.

area $\mathscr{A}$ are defined as:

$$\mathscr{L}(c) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \delta(c_{ij})h^2 \quad \text{and} \quad \mathscr{A}(c) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{N_z} \delta(c_{ijk})h^3,$$

where N_x, N_y, and N_z are the number of grid points in the x, y, and z directions, respectively. The discrete composition field c_{ij} (or c_{ijk}) is located at cell-centers and h is the uniform mesh size.

When there is no theoretical value of an interface length or a surface area, we need a reasonable value corresponding to the theoretical value. To obtain a reasonable value, we take the initial condition as

$$c(x, y, 0) = \frac{1}{2}\left(1 + \tanh\left(\frac{0.25 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{2\sqrt{2}\epsilon_4}\right)\right)$$

on the computational domain $\Omega = (0, 1) \times (0, 1)$, which represents a circle with a radius 0.25. We calculate the length of a 0.5-level contour using MATLAB. The results with increasingly finer grids are given in Table 6.1.

TABLE 6.1. Comparison between the theoretical value and the length of contour.

Mesh	16^2	32^2	64^2	128^2	Theoretical value
Interface length	1.565632	1.569770	1.570572	1.570778	$2\pi r \approx 1.570796$

Next, the initial condition is

$$c(x,y,z,0)=\frac{1}{2}\left(1+\tanh\left(\frac{0.25-\sqrt{(x-0.5)^2+(y-0.5)^2+(z-0.5)^2}}{2\sqrt{2}\epsilon_4}\right)\right)$$

on $\Omega=(0,1)\times(0,1)\times(0,1)$, which represents a sphere with a radius 0.25. We compute the area of an isosurface by summation of the areas of all triangle tiles in the isosurface using MATLAB. The results with increasingly finer grids are given in Table 6.2. The results in Tables 6.1 and 6.2 suggest that the length of contour and the area of isosurface agree well with the theoretical value.

Table 6.2. Comparison between the theoretical value and the area of isosurface.

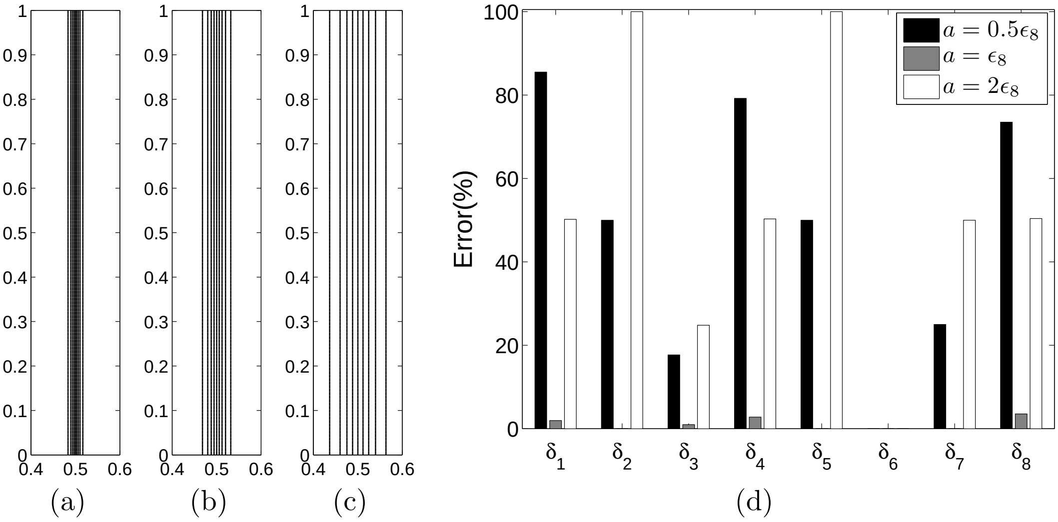
Mesh	32^3	64^3	128^3	256^3	Theoretical value
Surface area	0.781695	0.784493	0.785185	0.785364	$4\pi r^2\approx 0.785398$

6.1.1.1. *Test 1: length of line and circle.* In Section 2.3, we explored the performance of delta functions by the interface profile using Mathematica. To numerically explore the performance, we consider two initial conditions on $\Omega=(0,1)\times(0,1)$;

$$\begin{aligned} c(x,y,0) &= \frac{1}{2}\left(1+\tanh\left(\frac{0.5-x}{2\sqrt{2}a}\right)\right) \quad \text{and} \\ c(x,y,0) &= \frac{1}{2}\left(1+\tanh\left(\frac{0.25-\sqrt{(x-0.5)^2+(y-0.5)^2}}{2\sqrt{2}a}\right)\right) \end{aligned}$$

for $a=0.5\epsilon_8$, ϵ_8, and $2\epsilon_8$. We choose $h=1/128$ and $\Delta t=10h$.

In Figures 6.2 (case of line) and 6.3 (case of circle), (a), (b), and (c) show the contour lines of concentration with $a=0.5\epsilon_8$, ϵ_8, and $2\epsilon_8$, respectively. The length of line and circle for each delta function is given in Table 6.3. From Table 6.3, it is observed that δ_6 accurately calculates the length of line and circle regardless of whether an interface transition is compressed ($a=0.5\epsilon_8$) or stretched ($a=2\epsilon_8$). Figures 6.2 (d) and 6.3 (d) show the percentage error of each delta function of cases of line and circle by representing them as histograms of various sizes, respectively.



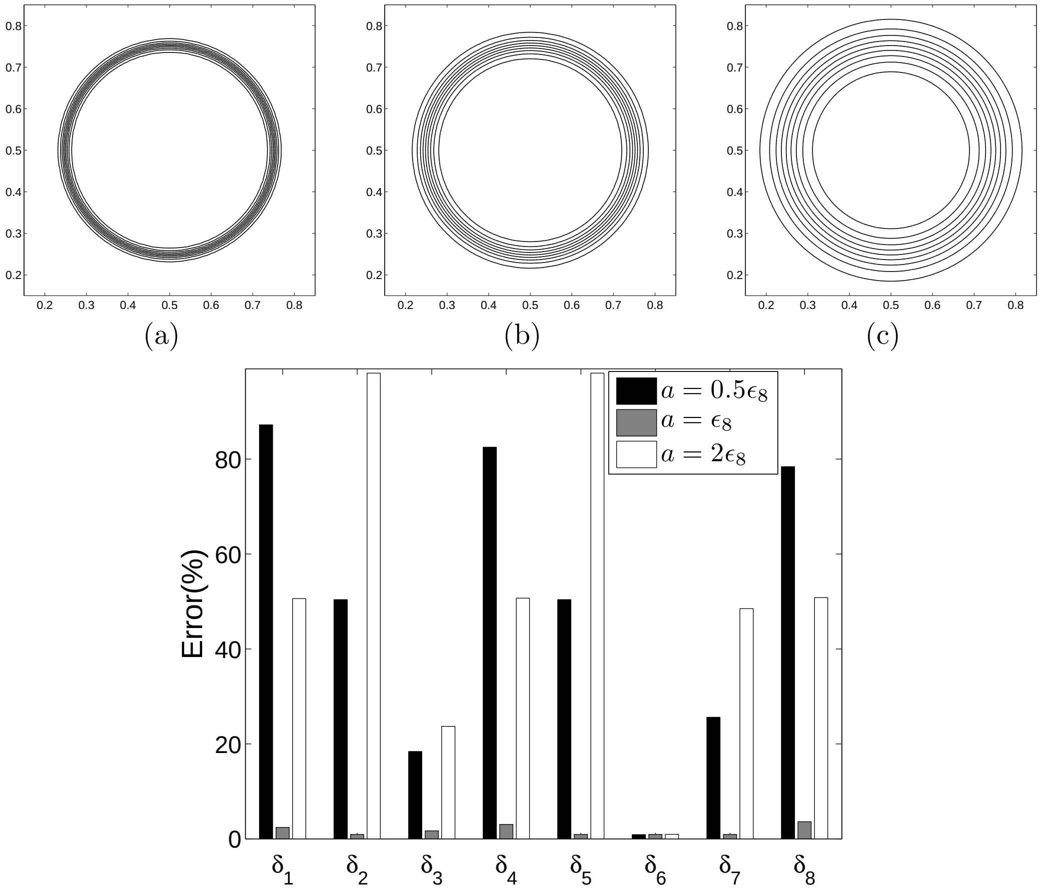
(a) (b) (c) (d)

FIGURE 6.2. The length of line. (a), (b), and (c) show the contour lines of concentration with $a = 0.5\epsilon_8$, ϵ_8, and $2\epsilon_8$, respectively. Contour levels are $0.1, 0.2, \ldots, 0.9$. (d) the percentage error of each delta function.

TABLE 6.3. The length of line and circle for each delta function.

Case	Line			Circle		
a	$0.5\epsilon_8$	ϵ_8	$2\epsilon_8$	$0.5\epsilon_8$	ϵ_8	$2\epsilon_8$
Contour value	1.0000	1.0000	1.0000	1.5849	1.5855	1.5857
δ_1	1.8556	0.9804	0.4975	2.9677	1.5473	0.7823
δ_2	0.5000	1.0000	2.0000	0.7854	1.5708	3.1416
δ_3	1.1778	0.9902	1.2488	1.8766	1.5590	1.9619
δ_4	1.7929	0.9720	0.4964	2.8927	1.5373	0.7810
δ_5	0.5000	1.0000	2.0000	0.7854	1.5708	3.1413
δ_6	1.0000	1.0000	1.0000	1.5708	1.5706	1.5704
δ_7	0.7500	1.0000	1.5000	1.1781	1.5707	2.3558
δ_8	1.7353	0.9645	0.4955	2.8278	1.5284	0.7799

6.1.1.2. *Test 2: Rayleigh–Taylor instability.* When a heavy fluid is superposed over a light fluid in a gravitational field, the fluid interface is unstable. Any perturbation of this interface tends to grow with time, producing the phenomenon known as the Rayleigh–Taylor instability. This phenomenon represents the penetration of both heavy and light fluids into each other. The Rayleigh–Taylor instability for a fluid in a gravitational field was originally introduced by Rayleigh [134] and later applied to



(a) (b) (c)

(d)

FIGURE 6.3. The length of circle. (a), (b), and (c) show the contour lines of concentration with $a = 0.5\epsilon_8$, ϵ_8, and $2\epsilon_8$, respectively. Contour levels are $0.1, 0.2, \ldots, 0.9$. (d) the percentage error of each delta function.

all accelerated fluids by Taylor [135]. In order to simulate the Rayleigh–Taylor instability, the Navier–Stokes–Cahn–Hilliard equations (NSCH) are preferred. The NSCH equations can be written in a dimensionless form

$$\begin{aligned}\rho(c)(\mathbf{u}_t+\mathbf{u}\cdot\nabla\mathbf{u}) &= -\nabla p+\frac{1}{Re}\Delta\mathbf{u}+\frac{\rho(c)}{Fr}\mathbf{g},\\ \nabla\cdot\mathbf{u} &= 0,\\ c_t+\nabla\cdot(c\mathbf{u}) &= \frac{1}{Pe}\Delta\mu,\\ \mu &= f(c)-C\Delta c,\end{aligned}$$

where $\mathbf{u}$ is the velocity, p is the pressure, $\rho(c)=\rho_1 c+\rho_2(1-c)$ is the variable density (ρ_1 and ρ_2 are the densities of the heavier and lighter fluid, respectively), and $\mathbf{g}=(0,-1)$. The dimensionless parameters are the Reynolds number, $Re=\rho_* U_* L_*/\eta$, Froude number, $Fr=U_*^2/(gL_*)$, Peclet number, $Pe=U_*L_*/(M\mu_*)$, and Cahn number, $C=\epsilon^2/(\mu_* L_*^2)$. The values with lower $*$ are characteristic values of corresponding ones, η is the viscosity, and g is the acceleration due to gravity. Here, the effect of the surface tension is negligible. For a detailed description of the numerical method used in solving the NSCH equations, please refer to Ref. [136].

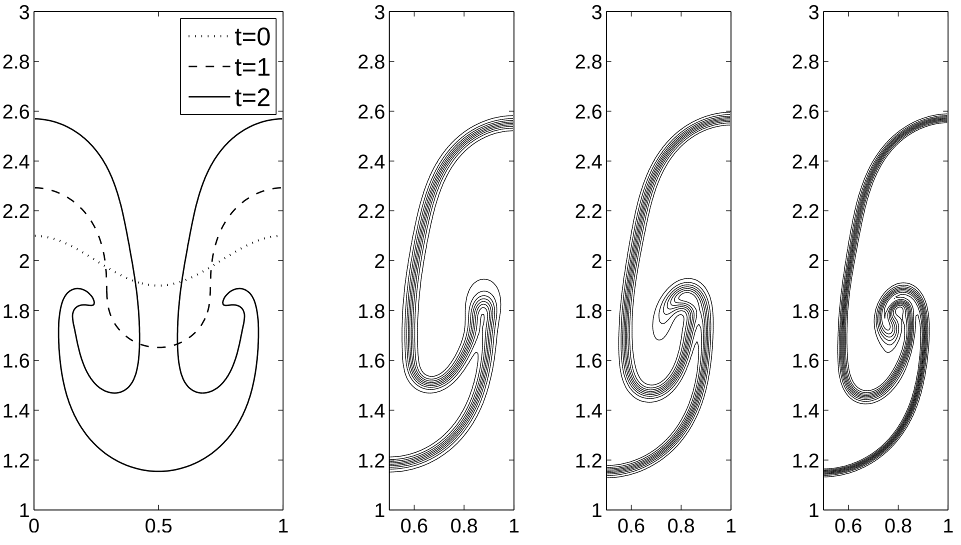
We calculate the length of the interface with three different Peclet numbers. The initial condition is

$$c(x,y,0)=\frac{1}{2}\left(1+\tanh\left(\frac{y-2-0.1\cos(2\pi x)}{2\sqrt{2}\epsilon}\right)\right)$$

on $\Omega=(0,1)\times(0,4)$, which represents a planar interface superimposed by a perturbation of wave number $k=1$ and amplitude 0.1. The density ratio is $\rho_1:\rho_2=3:1$ and we use the simulation parameters such as the uniform grids $h=1/128$, $\Delta t=0.00125$, $\epsilon=0.01$, and $Re=3000$.

Figure 6.4 (a) shows the evolution of the interface with $Pe=1/\epsilon$ at times $t=0$, 1, and 2. The results are given in Table 6.4. In the numerical simulations of the

Rayleigh–Taylor instability, as the Pe number increases, the width of an interface transition becomes non-uniform (see Figures 6.4 (b), (c), and (d)). As a result, the percentage error of almost all delta functions is high (see Figure 6.5). But the result obtained using δ_6 is in better agreement with the contour value, regardless of the Pe number.



(a) (b) (c) (d)

FIGURE 6.4. (a) the evolution of the interface with $Pe = 1/\epsilon$ at times $t = 0$, 1, and 2. The effect of the Peclet number on the temporal evolution of the interface at time $t = 2$: (b) $Pe = 0.1/\epsilon$, (c) $Pe = 1/\epsilon$, and (d) $Pe = 10/\epsilon$. Contour levels are $0.1, 0.2, \ldots, 0.9$.

TABLE 6.4. Rayleigh–Taylor instability: the interface length for each delta function.

Pe	Contour value	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
$0.1/\epsilon$	5.169	4.955	5.478	5.217	4.953	5.798	5.097	5.447	4.950
$1/\epsilon$	5.832	6.385	5.532	5.958	6.222	4.611	5.924	5.267	6.150
$10/\epsilon$	6.455	10.38	4.387	7.386	9.596	2.974	6.901	4.938	9.328

6.1.1.3. *Test 3: deformation of a circle by a single vortex.* In this test, a circle is deformed with a velocity field defined by $u = k\sin^2(\pi x)\sin(2\pi y)$ and $v = -k\sin(2\pi x)\sin^2(\pi y)$. In order to demonstrate the capability of each delta function

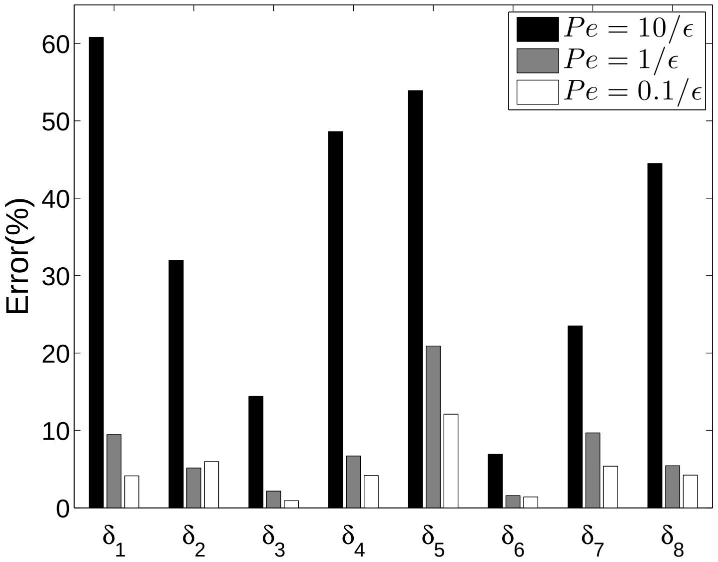
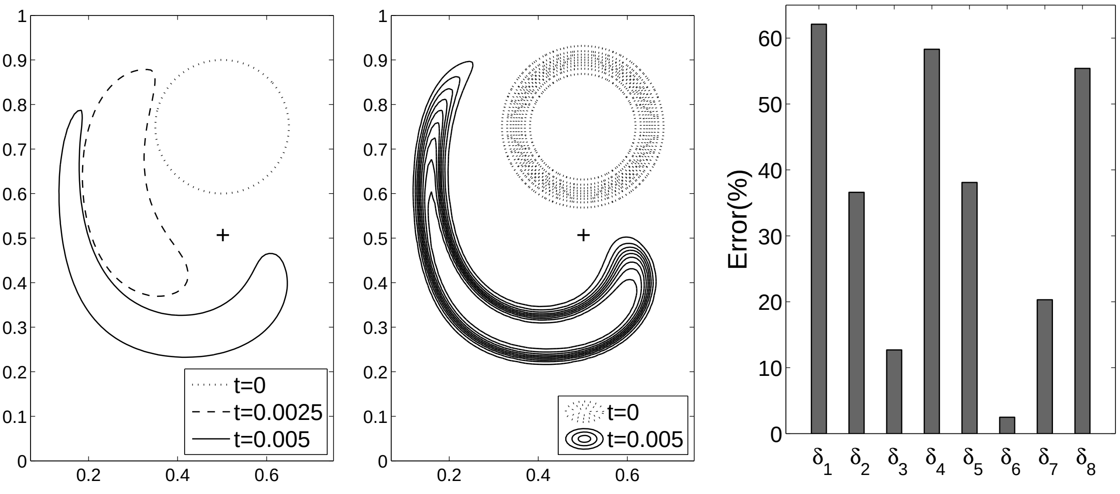


FIGURE 6.5. Rayleigh–Taylor instability: the percentage error of each delta function.

in an extreme velocity field, we choose $k = 100$, which is much larger than values used in the previous experiments. This flow satisfies $u = v = 0$ on the boundaries of the unit square domain. As shown in Figure 6.6 (a) by the dotted line, initially the circle has a radius of 0.15 and is centered at $(0.50, 0.75)$ in the unit square domain. The phase-field is initialized to $c = 1$ and $c = 0$ inside and outside the circle, respectively. The advection by the vorticity field causes the circle to evolve into a filament that spirals toward the vortex center at $(0.5, 0.5)$. The numerical solutions are computed on the uniform grids $h = 1/128$ and the calculation is run with $\Delta t = 0.00125/k$ and $\epsilon = 0.01$. Figure 6.6 (a) shows the evolution of the interface at times $t = 0$, 0.0025, and 0.005. The results are given in Table 6.5. For most of delta functions, except δ_6, the error is large because of the compressed interfacial transition (see Figure 6.6 (b)). But, as seen in Section 2.3, δ_6 yields a good result even though the interfacial transition is compressed (see Figure 6.6 (c)).



(a) (b) (c)

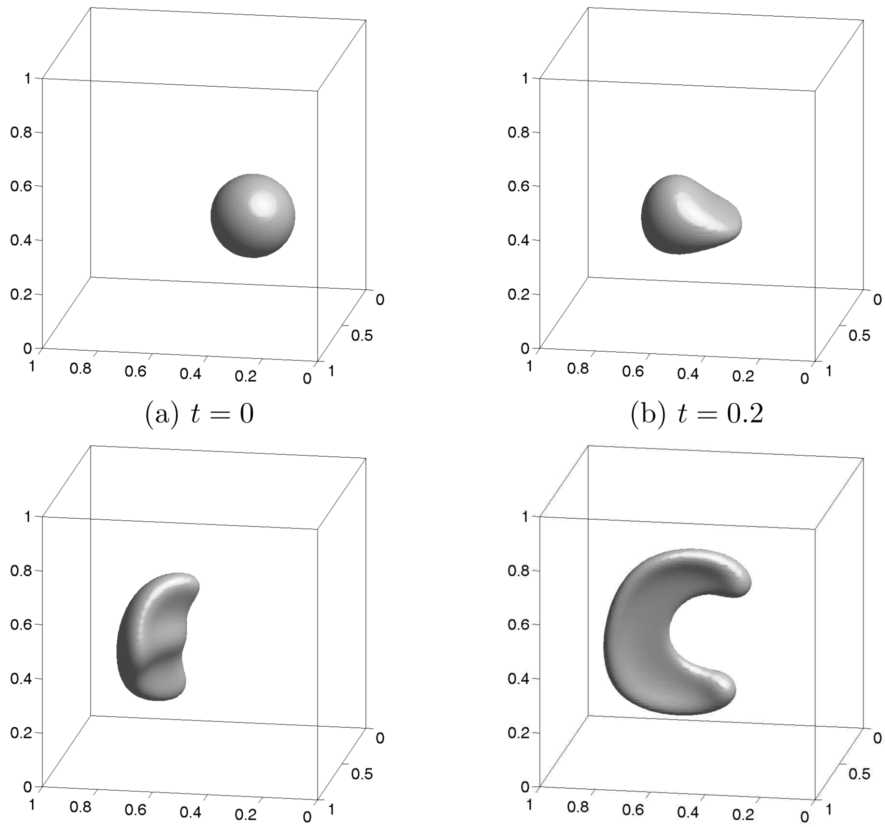
FIGURE 6.6. (a) the evolution of the interface at times $t = 0$, 0.0025, and 0.005. (b) Contour lines of concentration. Contour levels are $0.1, 0.2, \ldots, 0.9$. (c) the percentage error of each delta function.

TABLE 6.5. Deformed circle: the interface length for each delta function.

Contour value	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
1.968	3.192	1.246	2.219	3.117	1.218	1.919	1.568	3.060

6.1.1.4. *Test 4: three-dimensional deformation field.* We consider the problem of a deforming sphere in a velocity field given by $u = 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)$, $v = -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)$, and $w = -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)$ [137]. A simulation of this problem has previously been performed by Enright et al. using the level-set method [138]. This flow satisfies $u = v = w = 0$ on the boundaries of the unit cube domain. A sphere of radius 0.15 is placed within a unit computational domain at $(0.35, 0.35, 0.35)$. We take $h = 1/256$, $\Delta t = 2.5\text{e-}4$, ϵ_3, and $Pe = 0.1/\epsilon_3$.

Figure 6.7 shows the evolution of the interface at times $t = 0$, 0.2, 0.4, and 0.6. Each figure can be compared with the figures from the level-set computation in [138]. The sphere is entrained by two rotating vortices and then is compressed into a pancake-like shape. The surface of the pancake-like shape becomes stretched out. Parts of the interface thin out to about a few grid and almost all delta functions have difficulty to calculate exactly the surface area of this thin interface (see the results in Table 6.6). But δ_6 has very little error even if the surface is deformed. Figure 6.8 shows the percentage error of each delta function by representing them as histograms of various sizes.



(a) $t = 0$ (b) $t = 0.2$

(c) $t = 0.4$ (d) $t = 0.6$

FIGURE 6.7. Deformation test: the evolution of the interface.

TABLE 6.6. Deformed sphere: the surface area for each delta function.

Isosurface value	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
0.586	0.565	0.567	0.566	0.545	0.634	0.596	0.615	0.528

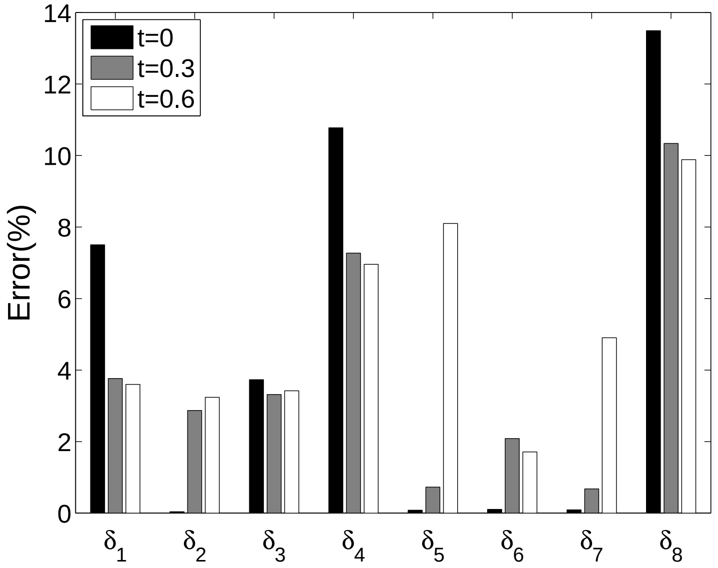


FIGURE 6.8. Deformed sphere: the percentage error of each delta function.

6.1.1.5. *Test 5: triply periodic minimal surfaces.* Triply periodic minimal surfaces (TPMS, see Figure 6.9) are of special interest because they appear in a variety of real structures such as silicates, bicontinuous mixtures, lyotropic colloids, detergent films, lipid bilayers, and biological formations [139]. One important application of TPMS is tissue scaffolds. Tissue scaffolds should have an optimal surface area and pore size to restore function or regenerate tissue more efficiently. The TPMS morphology has been successfully adapted to tissue scaffolds [140]. To calculate surface areas of TPMS, we take the periodic nodal surface approximations of the P, D, and G TPMS [141]:

$$\begin{aligned}
P(x,y,z) &= \cos 2\pi x + \cos 2\pi y + \cos 2\pi z + 0.5,\\
D(x,y,z) &= \cos 2\pi x \cos 2\pi y \cos 2\pi z - \sin 2\pi x \sin 2\pi y \sin 2\pi z + 0.5,\\
G(x,y,z) &= \sin 2\pi x \cos 2\pi y + \sin 2\pi z \cos 2\pi x + \sin 2\pi y \cos 2\pi z + 0.5.
\end{aligned}$$

For these calculations, we employ the computational domain $\Omega = (0,1) \times (0,1) \times (0,1)$ with $h = 1/256$, $\Delta t = 0.5h$, and $\epsilon = 0.01$. We stop the numerical computations when the discrete l_2-norm of the difference between $(n+1)$th and nth time step solutions becomes less than 10^{-6}. That is $||c^{n+1} - c^n|| \leq 10^{-6}$. In Table 6.7, we compare surface areas obtained using each delta function with those obtained by Jung et al. [142]. As we can see from Figure 6.10, the error of δ_6 is small.

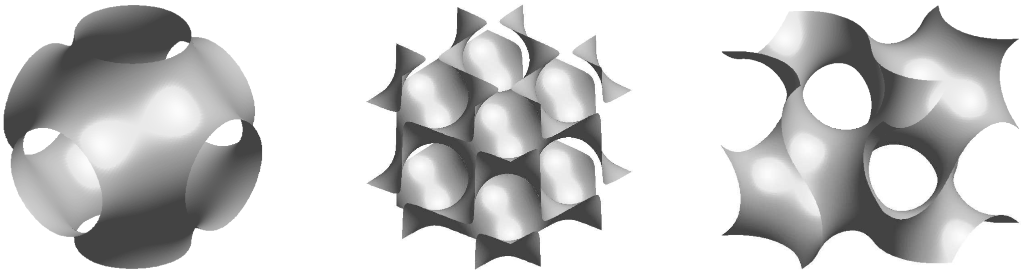

(a) (b) (c)

FIGURE 6.9. Triply periodic minimal surfaces: (a) Schwarz P (primitive), (b) Schwarz D (diamond), and (c) Schoen G (gyroid).

TABLE 6.7. TPMS: the surface area for each delta function.

TPMS (the results in [142])	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
P surface (2.34)	2.21	2.24	2.22	2.11	2.29	2.35	2.32	2.02
D surface (3.84)	3.57	3.63	3.60	3.39	3.80	3.82	3.81	3.22
G surface (3.10)	2.88	2.93	2.91	2.74	3.14	3.08	3.11	2.60

6.1.2. The Dirac delta function in the surface tension force formulation.

An accurate approximation of the surface tension force is essential for solving two-phase incompressible fluid flows. The continuum surface force (CSF) model of Brackbill et al. [120] is employed extensively to model the surface tension force of two-phase incompressible fluid flows in volume-of-fluid [99, 121, 124, 143], level-set [113, 39], and phase-field [72, 65, 55, 63] methods. In the CSF model, surface tension forces acting on the interface are transformed to volume forces in regions near the interface via a delta function, $\mathbf{SF} = \sigma\kappa\delta\mathbf{n}$, where σ is the surface tension coefficient, κ is the curvature, δ

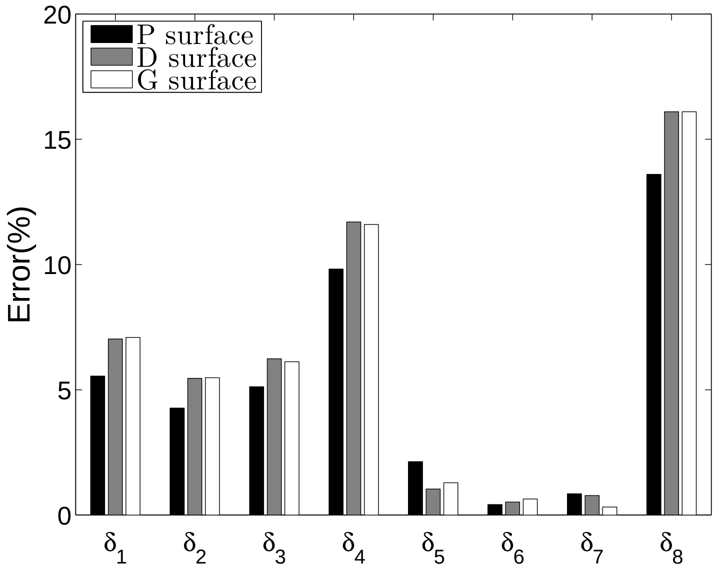


FIGURE 6.10. TPMS: the percentage error of each delta function.

is a delta function concentrated on the interface, and $\mathbf{n}$ is the unit outward normal to the surface and defined by $\mathbf{n} = \nabla c/|\nabla c|$. We note the identity

$$\nabla \cdot \mathbf{n} = -\kappa. \tag{6.1}$$

For a derivation of Equation (6.1), see Ref. [120]. The governing equations for two-phase incompressible fluid flows can be written as [72, 65]

$$\begin{aligned} \rho(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}) &= -\nabla p + \eta\Delta\mathbf{u} - \sigma\nabla\cdot\left(\frac{\nabla c}{|\nabla c|}\right)\delta(c)\frac{\nabla c}{|\nabla c|}, \qquad (6.2) \\ \nabla\cdot\mathbf{u} &= 0, \\ c_t + \nabla\cdot(c\mathbf{u}) &= M\Delta\mu, \\ \mu &= f(c) - \epsilon^2\Delta c. \end{aligned}$$

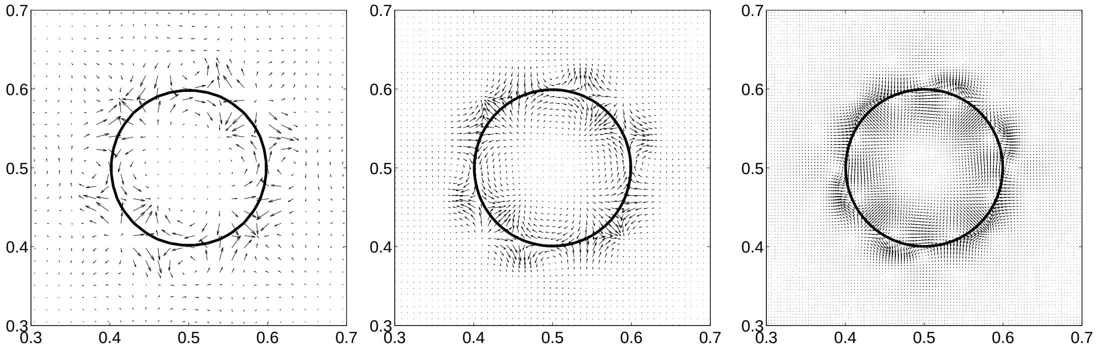
For details of the numerical solution, we refer to Ref. [65]. In this section, we present two tests for the surface tension force.

6.1.2.1. *Reduction of spurious velocities.* Spurious or parasitic velocities are unphysical currents that arise from a slight imbalance between stresses in the interfacial region. There are a number of papers for spurious velocities in incompressible flow problems [102, 144, 145, 146, 147]. Approximating and choosing the surface tension force formulation accurately is important since an improper formulation will lead to spurious velocities. In this section, we compare the performance of each delta function with the similar test problem in [124]. The computational domain is $\Omega = (0,1) \times (0,1)$ and the time step is $\Delta t = 10^{-5}$. The boundary conditions are zero velocity at the top and bottom walls, and periodicity in x-direction. Initially, a circular drop is centered at $(0.5, 0.5)$, with radius $a = 0.1$ and surface tension coefficient $\sigma = 0.357$. Both fluids have equal density, 4, and viscosity, 1. The initial velocity field is zero. The exact solution is zero velocity for all time. In dimensionless terms, the relevant parameter is the Ohnesorge number $Oh = \eta/\sqrt{\sigma\rho a} \sim 2.6463$.

Table 6.8 shows the convergence of spurious velocities as we refine the mesh size for each delta function. Most delta functions show minor decreases as we refine the mesh. But δ_1, δ_6, and δ_7 show a linear convergence of spurious velocities, and δ_6 effectively eliminates spurious velocities than δ_1 and δ_7. In Figure 6.11, scaled velocity vector plots at 200th time step with $\Delta t = 10^{-5}$ are shown. These show the locations of spurious velocities with mesh refinement, (a) $h = 1/64$, (b) $1/128$, and (c) $1/256$ with $\epsilon = 0.64h$. Spurious velocities are mainly concentrated in the neighborhood of the interface. The convergence of spurious velocities is evident as we refine the mesh size h and interface parameter ϵ.

TABLE 6.8. l_2-norm of velocity at 200th time step with $\epsilon = 0.64h$ and $\Delta t = 10^{-5}$.

Case	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
64^2	2.064e-5	4.460e-5	3.219e-5	3.010e-5	3.019e-5	2.232e-5	2.599e-5	4.463e-5
128^2	1.112e-5	4.345e-5	2.663e-5	1.901e-5	1.820e-5	1.018e-5	1.389e-5	2.667e-5
256^2	5.880e-6	4.264e-5	2.350e-5	1.325e-5	1.318e-5	4.653e-6	8.362e-6	2.137e-5



(a) (b) (c)

FIGURE 6.11. Scaled velocity vector plots at 200th time step with $\Delta t = 10^{-5}$. These show the locations of spurious velocities with mesh refinement, (a) $h = 1/64$, (b) 1/128, and (c) 1/256 with $\epsilon = 0.64h$. The solid line in each figure represents the interface. The length of the arrow is proportional to the magnitude of the velocity vector.

6.1.2.2. *Pressure jump across the drop.* Let us consider the equilibrium of a drop placed within another fluid. Let the drop composition be defined as

$$c(x, y) = \frac{1}{2}\left(1 + \tanh\left(\frac{0.1 - \sqrt{(x-0.5)^2 + (y-0.5)^2}}{2\sqrt{2}\epsilon}\right)\right).$$

In the equilibrium state of a droplet, the velocity vanishes ($\mathbf{u} \equiv 0$) and therefore Equation (6.2) reduces to Equation (6.3) and therefore pressure gradient should balance surface tension force.

$$\nabla p = -\sigma \nabla \cdot \left(\frac{\nabla c}{|\nabla c|}\right) \delta(c) \frac{\nabla c}{|\nabla c|}. \tag{6.3}$$

We solve Equation (6.4) numerically by taking the divergence operator to Equation (6.3) with $\sigma = 20$, 128×128 mesh, computational domain $\Omega = (0, 1) \times (0, 1)$, and

$\epsilon = 0.005$.

$$\Delta p = -\sigma \nabla \cdot \left[\nabla \cdot \left(\frac{\nabla c}{|\nabla c|} \right) \delta(c) \frac{\nabla c}{|\nabla c|} \right]. \tag{6.4}$$

From Laplace's formulation, we can obtain the theoretical prediction of the pressure jump inside an infinite cylinder as $\Delta p_{\text{theo}} = \sigma/R$, where R is the drop radius. In this test, the pressure jump Δp_{theo} is 200. This value is compared with the difference between the maximum and minimum computed drop pressures obtained with each delta function, $[p]$, defined as

$$[p] = \max_{i,j} p_{ij} - \min_{i,j} p_{ij}.$$

As shown in Table 6.9, the numerical pressure jump $[p]$ obtained using δ_6 is in excellent agreement with the theoretical prediction.

Table 6.9. The numerical pressure jump $[p]$ across the drop with $\sigma = 20$ and $R = 0.1$.

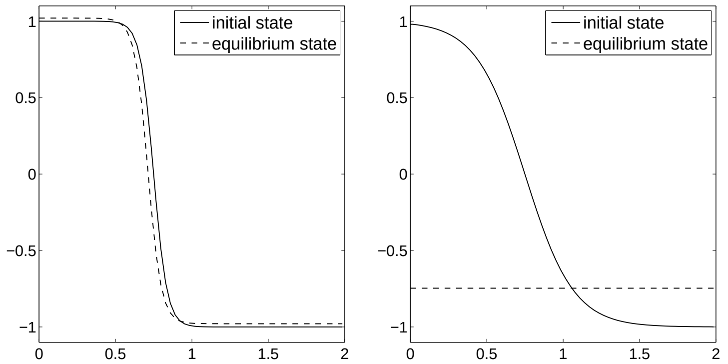
Case	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
$[p]$	190.40	199.08	194.58	182.76	197.31	199.70	198.48	176.90

6.1.3. Relation between the interfacial width and grid size. With the mass conserving boundary condition ($\nabla\mu \cdot \mathbf{n} = 0$ on $\partial\Omega$), the differentiation of the total mass yields

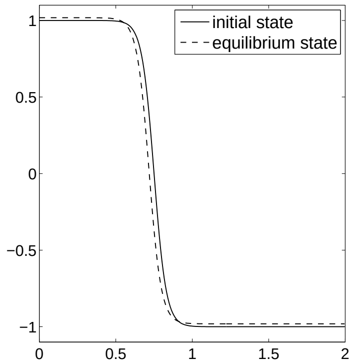
$$\frac{d}{dt}\int_{\Omega} \phi \, d\mathbf{x} = \int_{\Omega} \phi_t \, d\mathbf{x} = \int_{\Omega} \Delta\mu \, d\mathbf{x} = \int_{\partial\Omega} \nabla\mu \cdot \mathbf{n} \, ds = 0, \tag{6.5}$$

where ϕ is the phase-field variable and we redefine ϕ as the difference between the concentration of the two components. Equation (6.5) means that the solution of the Cahn–Hilliard equation conserves mass over the entire domain. Even though the phase-field variable is conserved globally, the mass of a drop is liable to variations as the ϕ evolves. Theoretically, such variations vanish as the interfacial thickness approaches zero. But, in practice, the interfacial thickness is finite. This was pointed out by Yue et al. [133]. To minimize the variations of mass, Yue et al. provided the guidelines on how to pick the interfacial width and grid size relative to the radius of curvature. And the authors calculated the shift of the phase-field variable and found that the shift $\delta\phi$ is proportional to ϵ/r_0, where r_0 is the initial drop radius.

In this section, we perform numerical simulations of the spontaneous shrinking of a drop. Drop with initial radius $r_0 = 0.75$ and interfacial width $\epsilon_{8.3281}$ (this value corresponds to the value in Section 2.2 in Ref. [133]) and ϵ_{30} are simulated. The initial drop is centered at the center of the computational domain $\Omega = (-2, 2) \times (-2, 2)$ and a 128×128 mesh is used. Figures 6.12 (a) and (b) show the phase-field ϕ at $y = 0$ for $\epsilon_{8.3281}$ and ϵ_{30}, respectively. In the case of $\epsilon_{8.3281}$, a grid size h is $\epsilon_{8.3281}/2$ and a 128×128 mesh is sufficient to guarantee numerical accuracy. Thus, we can see that ϕ at the equilibrium state shifts slightly away from the initial state. But, in the case of ϵ_{30}, a 128×128 mesh is too coarse compared with ϵ_{30}. As a result, the phase-field disperses (see Figure 6.12 (b)). In Figure 6.12 (c), we use a 512×512 mesh with ϵ_{30} and obtain more accurate result than result in Figure 6.12 (b). Therefore, to guarantee numerical accuracy, we need to choose properly an interfacial width and grid size.



(a) $\epsilon_{8.3281}$ with a 128^2 mesh. (b) ϵ_{30} with a 128^2 mesh.



(c) ϵ_{30} with a 512^2 mesh.

FIGURE 6.12. Each figure shows the phase-field at $y = 0$. The interfacial width and grid size for each case are given below each figure.

6.2. Numerical experiments for snow crystal growth

In this section we perform numerical experiments for two-dimensional solidification to validate that our proposed scheme is accurate, efficient, and robust. Unless otherwise specified, we take the initial state as

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

The zero level set ($c=0$) represents a circle of radius R_0. From the dimensionless variable definition the value $U=0$ corresponds to the melting temperature of the pure material, while $U=\Delta$ is the initial undercooling. The capillary length, d_0, is defined as $d_0=a_1/\lambda$ [235, 250, 262] with $a_1=0.8839$ [246, 247, 262] and $\lambda=3.1913$ [262].

6.2.1. Convergence test.

To obtain an estimate of the convergence rate, we perform a number of simulations for 6-fold crystal growth problem on a set of increasingly finer grids. The computational domain is $\Omega=(-100,100)^2$ and we take $R_0=15d_0$, $\epsilon_6=0.02$, and $\Delta=-0.55$. The numerical solutions are computed on the uniform grids $h=200/2^n$ and with corresponding time steps $\Delta t=0.6/2^{n-8}$ for $n=8$, 9, 10, and 11. The calculations are run up to time $T=150$. We define the error to be the discrete of l_2-norm of the difference between that grid and the average of the next finer grid cells covering it:

$$e_{h/\frac{h}{2}\,ij}=c_{hij}-(c_{\frac{h}{2}\,2i-1,2j-1}+c_{\frac{h}{2}\,2i-1,2j}+c_{\frac{h}{2}\,2i,2j-1}+c_{\frac{h}{2}\,2i,2j})/4.$$

The rate of convergence is defined as:

$$\log_2(\|\, e_{h/\frac{h}{2}} \,\|_2 \,/\, \|\, e_{\frac{h}{2}/\frac{h}{4}} \,\|_2).$$

The errors and rates of convergence are given in Table 6.10. The results suggest that the scheme is indeed second order accurate in space. Figure 6.13 shows the convergence of numerical results under mesh refinement.

TABLE 6.10. Error and l_2 convergence result.

$256-512$	Rate	$512-1024$	Rate	$1024-2048$
5.477E−4	1.96	1.405E−4	2.01	3.487E−5

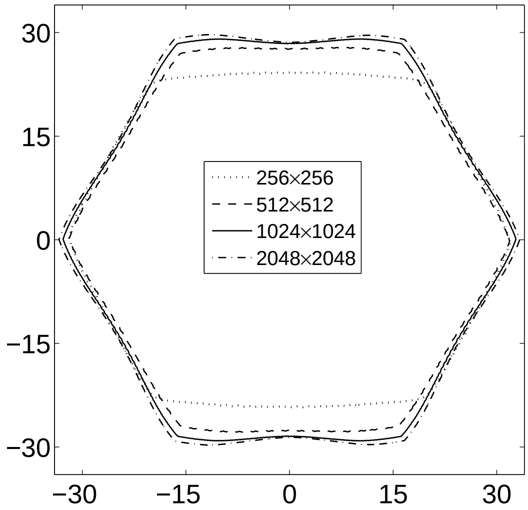
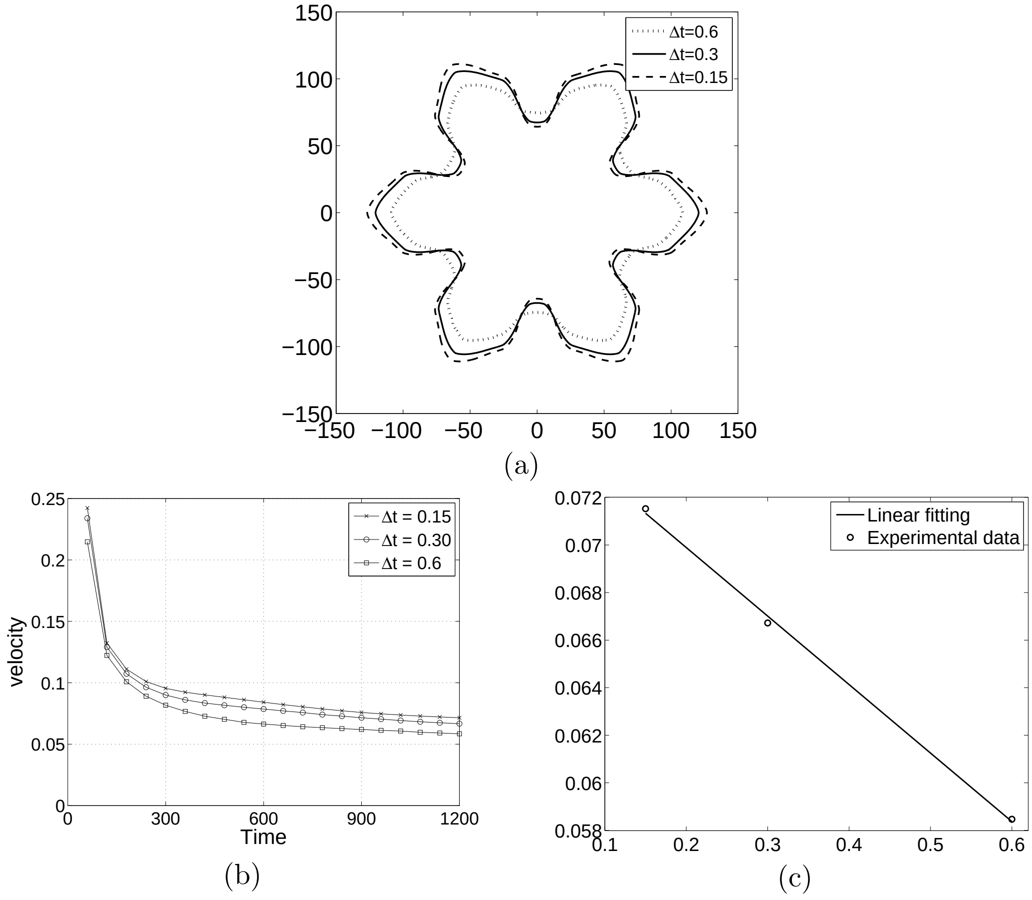


FIGURE 6.13. Convergence of numerical results under mesh refinement.

Next, we consider the evolution of the interface with different time steps in order to investigate the effect of time step. A 1024×1024 mesh is used on the domain $\Omega = (-200, 200)^2$ with $R_0 = 50d_0$, $\epsilon_6 = 0.02$, and $\Delta = -0.55$. Figure 6.14 (a) shows the interfaces at time $T = 1200$ with different time steps $\Delta t = 0.6$, 0.3, and 0.15. Figure 6.14 (b) shows the velocity of the tip versus time. For the calculation of the crystal tip velocity, refer to Ref. [252]. The velocity V of the tip at time $T = 1200$ versus time step is shown in Figure 6.14 (c). Here, we define the error between the fitting velocity $\widetilde{V}$ and V as $E_i = |\widetilde{V}_i - V_i|/V_i$. In Figure 6.14 (c), the linear fit $\widetilde{V}$ is done using the MATLAB function "polyfit" and the errors on the index i are calculated by the MATLAB function "polyval" on the results of the linear fit. In this test, the l_2 error is 0.54%. Therefore the results suggest that the convergence rate of the tip velocity is linear with respect to the time step.



(a)

(b) (c)

FIGURE 6.14. (a) The interfaces at $T = 1200$ for different time steps. (b) shows the velocity of the tip versus time. (c) The numerical experimental and linear fitting velocities versus time step.

6.2.2. Stability test. In this section, we perform a number of simulations on a set of increasingly finer grids to show that our proposed method is more stable than the previous methods which suffer from time restrictions $\Delta t \leq O(h^2)$ for stability. The computational domain is $\Omega = (-200, 200)^2$ and we take $R_0 = 15d_0$, $\epsilon_6 = 0.02$, and $\Delta = -0.55$. The numerical solutions are computed on the uniform grids $h = 400/2^n$ with corresponding time steps $\Delta t = 3h$ for $n = 8$, 9, and 10. Figure 6.15 shows the crystal growth with different time steps at $T = 70.31$. In general, large time steps may cause large truncation errors. However, as can be seen in Figure 6.15, we obtain stable solutions with large time steps.

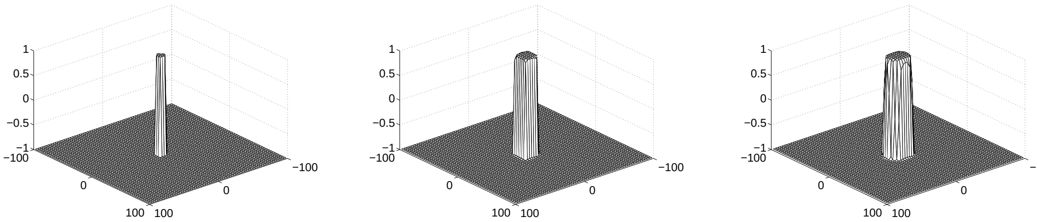
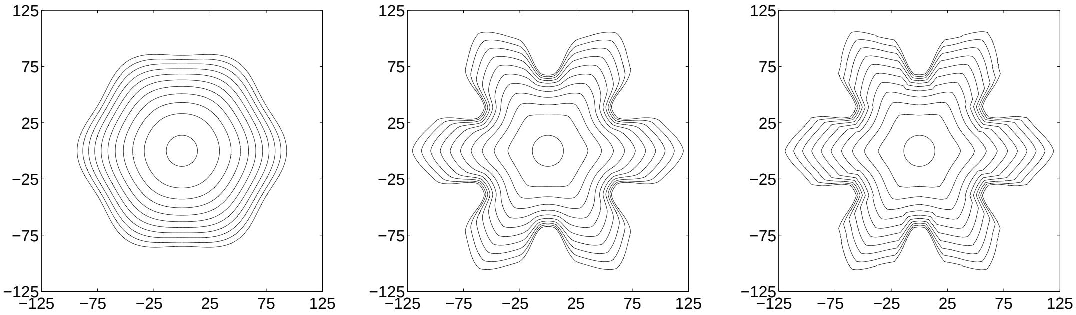


FIGURE 6.15. The stability of crystal growth with different mesh sizes: (a) 256×256 mesh ($\Delta t = 4.68$), (b) 512×512 mesh ($\Delta t = 2.34$), and (c) 1024×1024 mesh ($\Delta t = 1.17$).

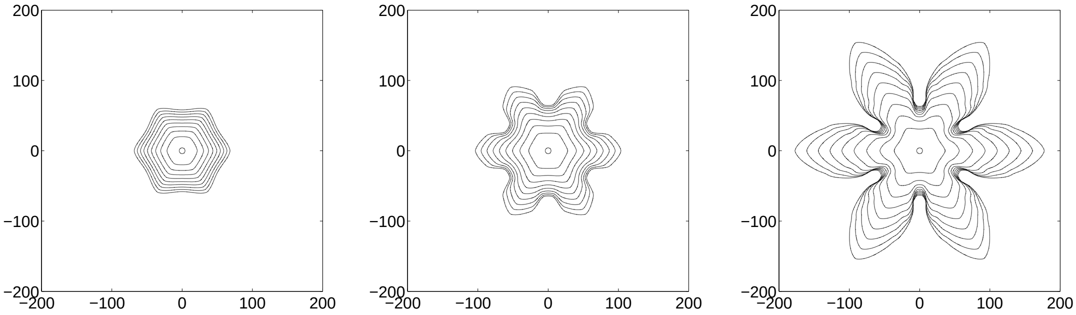
6.2.3. Effect of ϵ_6. To investigate the effect of ϵ_6, we consider the evolution of the interface with different $\epsilon_6 = 0.002$, 0.02, and 0.05. A 1024×1024 mesh is used on the domain $\Omega = (-100, 100)^2$ and we take $R_0 = 50d_0$, $\Delta = -0.55$, $\Delta t = 0.3$, and $T = 1200$. Figures 6.16 (a), (b), and (c) are the evolution of crystal growth with $\epsilon_6 = 0.002$, 0.02, and 0.05, respectively. As advised in the previous paper, If $\epsilon_6 < \frac{1}{35}$, all of tangent planes lie outside and all orientations appear on the equilibrium shape. Detail view is drawn in Figure 6.16(a). Otherwise, there is missing orientations shown in Figure 6.16(c). While if ϵ_6 is not more smaller than $\frac{1}{35}$, the crystal also works well shown in Figure 6.16(b). Thus the Wulff construction is not strictly correlated with ϵ_6 in crystal growth, but provide guidelines for parameter selection.



(a) (b) (c)

FIGURE 6.16. The effect of ϵ_6. (a), (b), and (c) are the evolution of crystal growth with $\epsilon_6 = 0.002$, 0.02, and 0.05, respectively. The times are $t = 0$, 120, 240, 360, 480, 600, 720, 840, 960, 1080, and 1200.

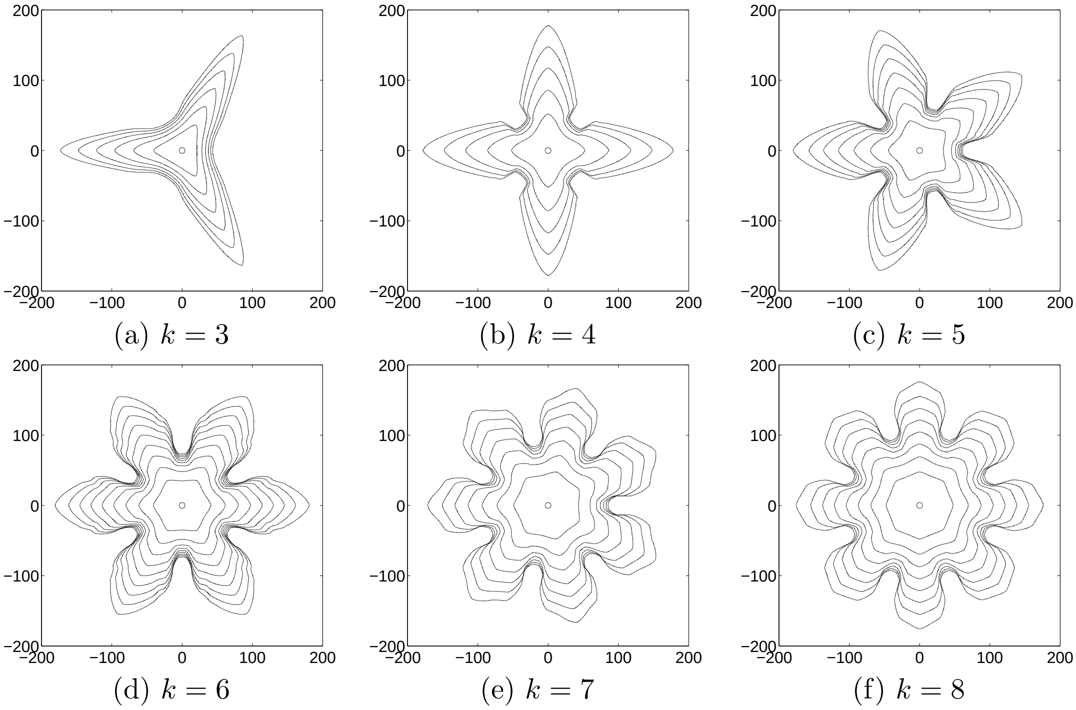
6.2.4. Effect of undercooling. Now we investigate the effects of undercooling of the initial solid seed. For each test, a 1024 × 1024 mesh is used on the domain $\Omega = (-200, 200)^2$ and we choose $R_0 = 15d_0$, $\epsilon_6 = 0.02$, $\Delta t = 0.3$, and $T = 1080$. Figure 6.17 shows sequences of interfaces with different undercooling sizes $\Delta = -0.45$, $\Delta = -0.55$, and $\Delta = -0.65$. We observe that the large initial undercooling causes the dendrite to grow faster.



(a) $\Delta = -0.45$ (b) $\Delta = -0.55$ (c) $\Delta = -0.65$

FIGURE 6.17. Sequences of interfaces with different undercooling sizes $\Delta = -0.45$, $\Delta = -0.55$, and $\Delta = -0.65$.

6.2.5. k-fold symmetric crystal growth. If we set the energy function by $\epsilon(\phi) = \epsilon_0(1 + \epsilon_k \cos(k\phi))$, then our proposed method can simulate the k-fold crystal growth in general. To show this, we simulate sequences of computational experiments of k-fold symmetric crystal growth for $k = 4, \ldots, 9$. A 1024×1024 mesh is used on the domain $\Omega = (-200, -200)^2$ and we take $R_0 = 15d_0$, $\Delta = -0.55$, and $\Delta t = 0.3$. Note that we use $\epsilon_k = 1/(k^2 - 1)$ to respond to the Wulff's algorithm. The evolutions for each k are shown in Figure 6.18.



(a) $k = 3$ (b) $k = 4$ (c) $k = 5$

(d) $k = 6$ (e) $k = 7$ (f) $k = 8$

FIGURE 6.18. The evolutions of k-fold crystal growth after time: (a) $T = 720$, (b) $T = 1200$, (c) $T = 1680$, (d) $T = 2160$, (e) $T = 2520$, and (f) $T = 2880$.

6.3. Numerical experiments for the multi-component Cahn–Hilliard system

6.3.1. The stability of the proposed scheme. We investigate the stability of five different schemes for the quaternary CH system: the explicit Euler's (EE), semi-implicit Euler's (SIE), implicit Euler's (IE), Crank-Nicolson (CN), and nonlinear splitting (NS). For dissipative dynamics such as the CH system, a discrete time stepping algorithm is defined to be gradient stable if the free energy is nonincreasing, $\mathcal{F}(\mathbf{c}^{n+1}) \leq \mathcal{F}(\mathbf{c}^n)$, for each n. Define $\Delta t_{\max}$ as the largest possible time step which allows stable numerical computation. In other words, if the time step is larger than $\Delta t_{\max}$, then the algorithm is not gradient stable. To measure $\Delta t_{\max}$ for each scheme, we perform a number of simulations for a sample initial problem on a set of increasingly finer grids. The initial conditions are

$$\begin{aligned} c_1(x,y,0) &= 0.25 + 0.1\text{rand}(\ ), \\ c_2(x,y,0) &= 0.25 + 0.1\text{rand}(\ ), \\ c_3(x,y,0) &= 0.25 + 0.1\text{rand}(\ ) \end{aligned}$$

on a domain $\Omega = (0,1) \times (0,1)$. Here, rand( ) is a random number between -1 and 1. The numerical solutions are computed on the uniform grids, $h = 1/2^n$ for $n = 5$, 6, 7, and 8. For each case, $\epsilon = 0.64h$ is used. The values of $\Delta t_{\max}$ with different schemes are listed in Table 6.11. From the results, we observe that EE, SIE, IE, and CN schemes are not gradient stable when we use the time step larger than $\Delta t_{\max}$ (see Figure 6.19). However, our proposed scheme (NS) is gradient stable for time steps of any size, i.e., the scheme is practically unconditionally gradient stable.

TABLE 6.11. The values of the maximum time step guaranteeing the stability of each scheme.

Case	32×32	64×64	128×128	256×256
EE	7.6×10^{-5}	1.9×10^{-5}	4.7×10^{-6}	1.1×10^{-6}
SIE	2.9×10^{-2}	7.9×10^{-3}	1.3×10^{-3}	4.3×10^{-4}
IE	3.2×10^{-2}	7.8×10^{-3}	1.7×10^{-3}	4.4×10^{-4}
CN	2.0×10^{-2}	5.0×10^{-3}	1.1×10^{-3}	2.8×10^{-4}
NS	∞	∞	∞	∞

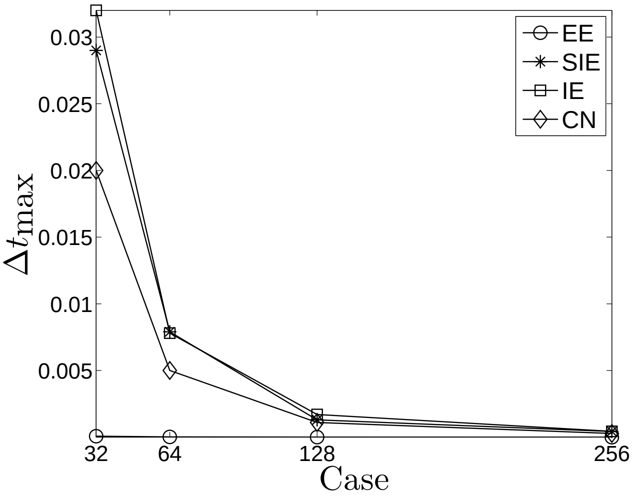


FIGURE 6.19. The maximum time step ($\Delta t_{\max}$) guaranteeing the stability of each scheme.

6.3.2. Linear stability analysis. In this section, we study the short-time behavior of a quaternary mixture. The partial differential Equations (3.4) and (3.5) we wish to solve may be written as

$$\frac{\partial \mathbf{c}(x,t)}{\partial t} = \Delta\left(\psi(\mathbf{c}) - \epsilon^2 \Delta \mathbf{c}\right), \quad \text{for } (x,t) \in \Omega \times (0,T], \tag{6.6}$$

where $\psi(\mathbf{c}) = \mathbf{f}(\mathbf{c}) + \beta(\mathbf{c})\mathbf{1}$. Let the mean concentration take the form $\mathbf{m} = (m_1, m_2, m_3)$. We seek a solution of the form

$$\mathbf{c}(x,t) = \mathbf{m} + \sum_{k=1}^{\infty} \cos(k\pi x)(\alpha_k(t), \beta_k(t), \gamma_k(t)), \tag{6.7}$$

where $|\alpha_k(t)|$, $|\beta_k(t)|$, and $|\gamma_k(t)| \ll 1$. After linearizing $\psi(\mathbf{c})$ about $\mathbf{m}$, we have

$$\psi(\mathbf{c}) \approx \psi(\mathbf{m}) + (\mathbf{c}-\mathbf{m}) \begin{pmatrix} \partial_{c_1}\psi_1(\mathbf{m}) & \partial_{c_1}\psi_2(\mathbf{m}) & \partial_{c_1}\psi_3(\mathbf{m}) \\ \partial_{c_2}\psi_1(\mathbf{m}) & \partial_{c_2}\psi_2(\mathbf{m}) & \partial_{c_2}\psi_3(\mathbf{m}) \\ \partial_{c_3}\psi_1(\mathbf{m}) & \partial_{c_3}\psi_2(\mathbf{m}) & \partial_{c_3}\psi_3(\mathbf{m}) \end{pmatrix}. \tag{6.8}$$

Substituting (6.8) into (6.6) and letting $m_1 = m_2 = m_3 = m$ for simplicity, then, up to first order, we have

$$\frac{\partial \mathbf{c}}{\partial t} = \Delta \mathbf{c} \begin{pmatrix} \frac{18m^2-9m+1}{2} & \frac{3m(4m-1)}{2} & \frac{3m(4m-1)}{2} \\ \frac{3m(4m-1)}{2} & \frac{18m^2-9m+1}{2} & \frac{3m(4m-1)}{2} \\ \frac{3m(4m-1)}{2} & \frac{3m(4m-1)}{2} & \frac{18m^2-9m+1}{2} \end{pmatrix} - \epsilon^2 \Delta^2 \mathbf{c}. \tag{6.9}$$

After substituting $\mathbf{c}(x,t)$ from Equation (6.7) into (6.9), we get

$$\begin{pmatrix} \alpha_k(t) \\ \beta_k(t) \\ \gamma_k(t) \end{pmatrix}' = \mathbf{A} \begin{pmatrix} \alpha_k(t) \\ \beta_k(t) \\ \gamma_k(t) \end{pmatrix}, \ \mathbf{A} = \begin{pmatrix} a & b & b \\ b & a & b \\ b & b & a \end{pmatrix}, \tag{6.10}$$

where $'$ indicates the time derivative and

$$a = \frac{-k^2\pi^2}{2}(18m^2 - 9m + 1) - \epsilon^2 k^4 \pi^4, \ b = \frac{-3k^2\pi^2 m(4m-1)}{2}.$$

The eigenvalues of $\mathbf{A}$ are

$$\begin{aligned} \lambda_1 &= -\frac{k^2\pi^2}{2}(42m^2 - 15m + 1 + 2\epsilon^2 k^2 \pi^2), \\ \lambda_2 &= \lambda_3 = -\frac{k^2\pi^2}{2}(6m^2 - 6m + 1 + 2\epsilon^2 k^2 \pi^2). \end{aligned}$$

The solution to the system of ODEs (6.10) is given by

$$\begin{aligned} \begin{pmatrix} \alpha_k(t) \\ \beta_k(t) \\ \gamma_k(t) \end{pmatrix} &= \frac{\alpha_k(0)+\beta_k(0)+\gamma_k(0)}{3} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} e^{\lambda_1 t} + \frac{-\alpha_k(0)-\beta_k(0)+2\gamma_k(0)}{3} \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} e^{\lambda_2 t} \\ &+ \frac{-\alpha_k(0)+2\beta_k(0)-\gamma_k(0)}{3} \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} e^{\lambda_2 t}. \end{aligned}$$

In Figure 6.20, we plot the evolution of the amplitudes as a function of time. The symbols '-○-', '-◇-', and '-△-' are numerical results that are compared with the theoretical values $\alpha_k(t)$ (point), $\beta_k(t)$ (star), and $\gamma_k(t)$ (plus), respectively, with the

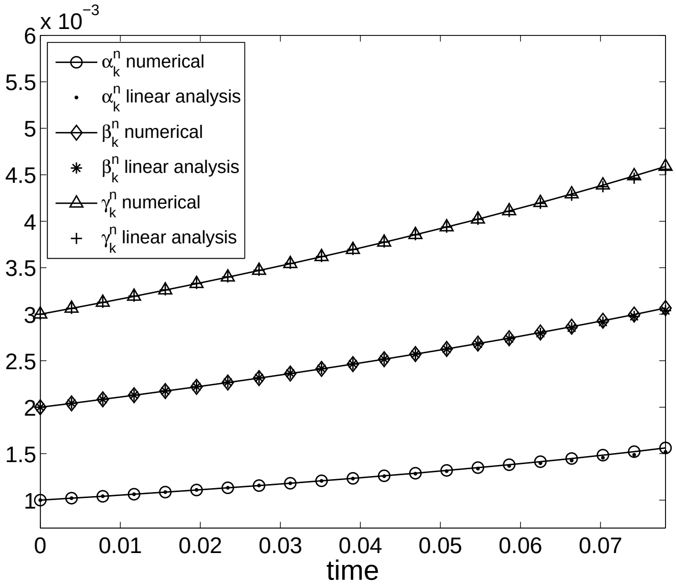


FIGURE 6.20. The symbols '-○-', '-◇-', and '-△-' are numerical results that are compared with the theoretical values $\alpha_k(t)$ (point), $\beta_k(t)$ (star), and $\gamma_k(t)$ (plus), respectively, with the initial conditions of Equations (6.11)-(6.13).

initial conditions:

$$c_1(x,0) = 0.25 + 0.001\cos(3\pi x), \tag{6.11}$$

$$c_2(x,0) = 0.25 + 0.002\cos(3\pi x), \tag{6.12}$$

$$c_3(x,0) = 0.25 + 0.003\cos(3\pi x). \tag{6.13}$$

Here, we used $k = 3$, $m = 0.25$, $\epsilon = 0.005$, $h = 1/256$, $\Delta t = 0.1h$, and $T = 200\Delta t$. The numerical amplitudes are defined by

$$\begin{aligned}\alpha_k^n &= \left(\max_{1\le i\le N_x} c_1^n(x_i) - \min_{1\le i\le N_x} c_1^n(x_i)\right)/2,\\ \beta_k^n &= \left(\max_{1\le i\le N_x} c_2^n(x_i) - \min_{1\le i\le N_x} c_2^n(x_i)\right)/2,\\ \gamma_k^n &= \left(\max_{1\le i\le N_x} c_3^n(x_i) - \min_{1\le i\le N_x} c_3^n(x_i)\right)/2.\end{aligned}$$

The results in Figure 6.20 show that the linear stability analysis and numerical solutions are in good agreement in a linear regime.

6.3.3. The efficiency of the proposed scheme. As mentioned in Section 5.5.1, we can solve the N-component CH system in a decoupled way by using our scheme. In order to show the efficiency of the proposed scheme, we consider phase separation of $N = 3, 4, \ldots, 10$ components in the unit square domain $\Omega = (0, 1) \times (0, 1)$. For each number of components, the initial condition is a randomly chosen superposition of circles. We choose $h = 1/128$, $\Delta t = 10h$, and $\epsilon = 0.0047$ and perform 4000 time steps. The evolution of the interface is shown in Figure 6.27. Rows 1 and 2 correspond to $t = 10\Delta t$ and $4000\Delta t$, respectively. Table 6.13 provides the average CPU time (in seconds) during 4000 time steps for each number of components. The average CPU time versus number of components is shown in Figure 6.28. The results suggest that the convergence rate of average CPU time is linear with respect to number of components.

TABLE 6.12. Average CPU times (*sec.*) for different numbers of components.

N	3	4	5	6	7	8	9	10
Average CPU time	2.060	2.990	4.070	5.028	5.791	6.686	7.492	8.533

6.3.4. Spinodal decomposition - phase separation of a ten-component mixture. We consider phase separation of a ten-component mixture by spinodal decomposition. The initial condition is a randomly chosen superposition of circles. A 128×128 mesh is used on the domain $\Omega = (0, 1) \times (0, 1)$ and we take $\Delta t = 10h$ and $\epsilon = 0.0038$. We compute until the solution becomes numerically stationary. Figure 6.23 shows the evolution of the interface at different times. We observe that three phases meet at one point and the angles between them approach 120° as they approach local

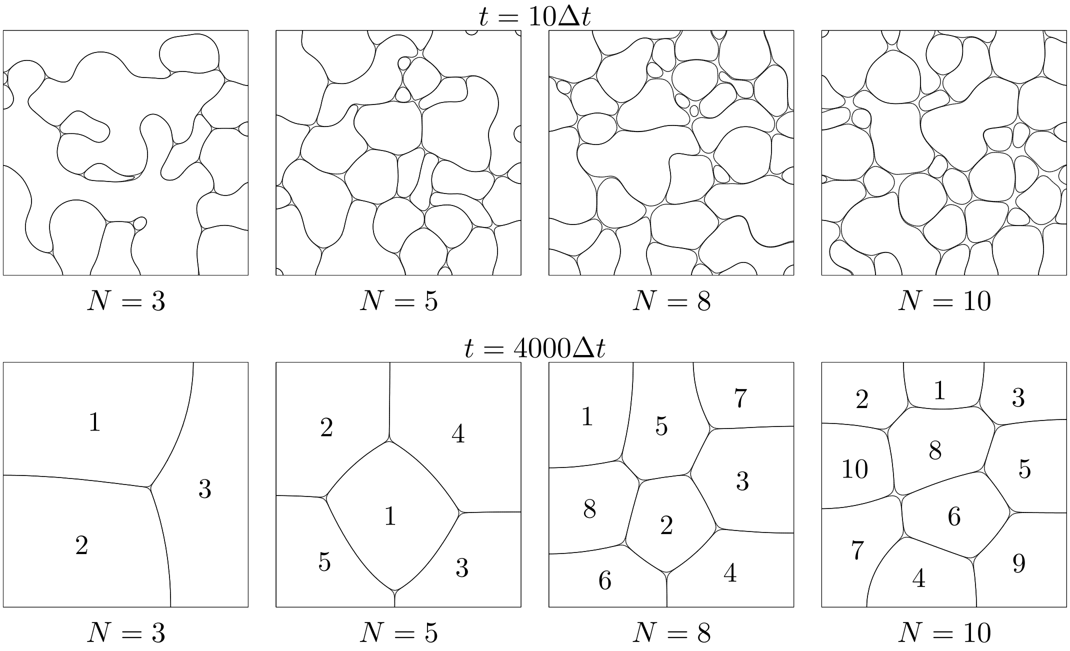


FIGURE 6.21. Phase separation of $N = 3$, 5, 8, and 10 components. Rows 1 and 2 correspond to $t = 10\Delta t$ and $4000\Delta t$, respectively. Numbers in row 2 indicate the number of components.

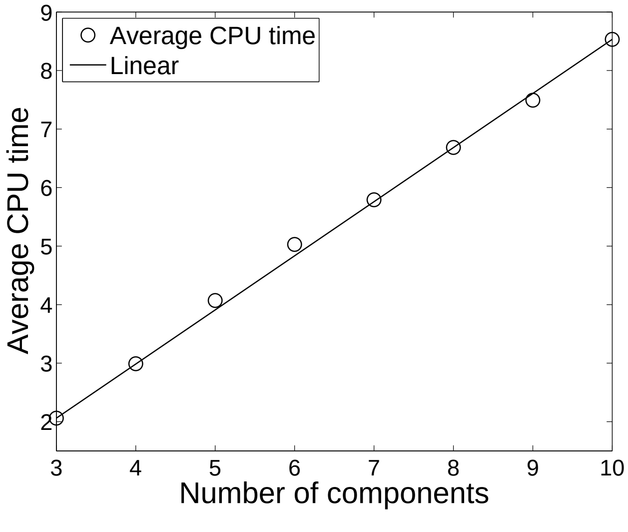
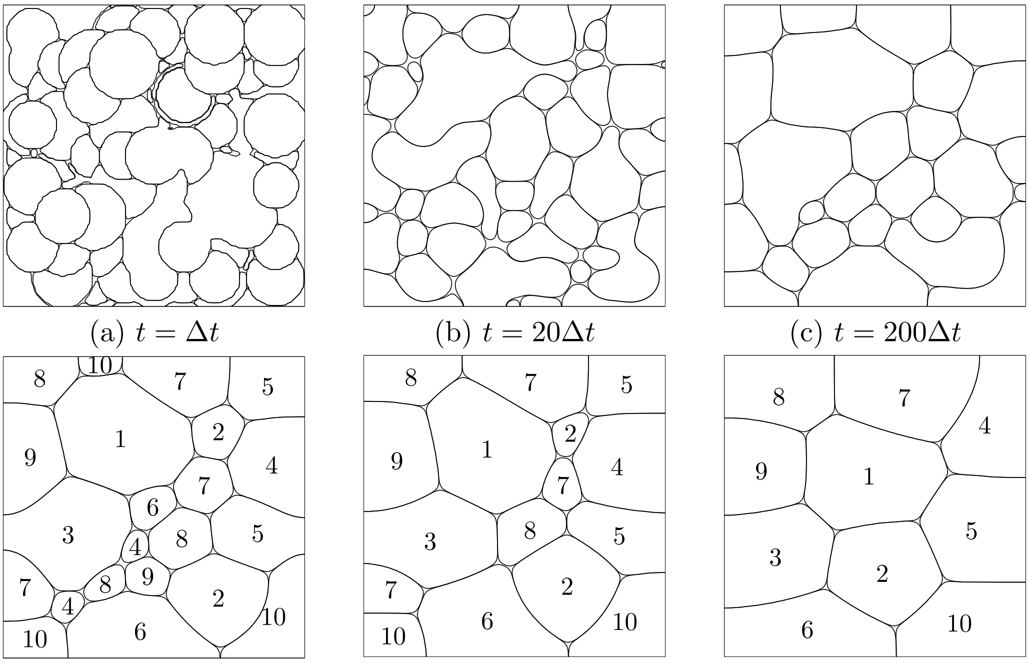


FIGURE 6.22. Average CPU time versus number of components.

equilibrium states. This is due to the fact that in the total energy functional equation (3.2), $\mathcal{F}(\mathbf{c}(\mathbf{x},t))$ is symmetric and the interaction parameter ϵ is constant. This result is in good agreement with the theory in [205].



(a) $t = \Delta t$ (b) $t = 20\Delta t$ (c) $t = 200\Delta t$

(d) $t = 500\Delta t$ (e) $t = 1000\Delta t$ (f) $t = 4000\Delta t$

FIGURE 6.23. Temporal evolution of a ten-component system. Times are shown below each figure. Numbers in (d), (e), and (f) indicate the number of components.

6.4. Numerical experiments for the multi-component Navier–Stokes–Cahn–Hilliard system

6.4.1. Phase separation of a five-component mixture in a gravitational field. We consider multi-component incompressible viscous fluid flow. The fluid dynamics is described by the Navier–Stokes–Cahn–Hilliard (NSCH) equations [64, 63, 153, 11, 154, 155, 156, 65, 187, 189, 58, 55, 16, 56]:

$$\rho(\mathbf{c})\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \eta\Delta\mathbf{u} + \rho(\mathbf{c})\mathbf{g}, \tag{6.14}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{6.15}$$

$$\frac{\partial \mathbf{c}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{c} = M\Delta\boldsymbol{\mu}, \tag{6.16}$$

$$\boldsymbol{\mu} = \mathbf{f}(\mathbf{c}) - \epsilon^2\Delta\mathbf{c} + \beta(\mathbf{c}), \tag{6.17}$$

where $\mathbf{u}$ is the velocity, p is the pressure, $\rho(\mathbf{c})$ is the density, η is the viscosity, $\mathbf{g} = (0, -g)$ is the gravity, M is the mobility, and μ is the generalized chemical potential. In this paper the effect of the surface tension is negligible. We note that even though our phase-field model can deal with a variable viscosity case straightforwardly, we focus on the viscosity matched case.

To make Equations (6.14)-(6.17) dimensionless, we choose the following definitions:

$$\mathbf{x}' = \frac{\mathbf{x}}{L_c},\ \mathbf{u}' = \frac{\mathbf{u}}{U_c},\ t' = \frac{tU_c}{L_c},\ \rho' = \frac{\rho}{\rho_c},\ p' = \frac{p}{\rho_c U_c^2},\ \mathbf{g}' = \frac{\mathbf{g}}{g},\ \boldsymbol{\mu}' = \frac{\boldsymbol{\mu}}{\mu_c},$$

where the primed quantities are dimensionless and L_c is the characteristic length, U_c is the characteristic velocity, ρ_c is the characteristic density and is defined as that of fluid 1, g is the gravitational acceleration, and μ_c is the characteristic chemical potential. Substituting these variables into Equations (6.14)-(6.17) and dropping the primes, we have

$$\begin{aligned}
\rho(\mathbf{c})\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) &= -\nabla p + \frac{1}{Re}\Delta\mathbf{u} + \frac{\rho(\mathbf{c})}{Fr^2}\mathbf{g}, && (6.18)\\
\nabla\cdot\mathbf{u} &= 0, && (6.19)\\
\frac{\partial \mathbf{c}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{c} &= \frac{1}{Pe}\Delta\boldsymbol{\mu}, && (6.20)\\
\boldsymbol{\mu} &= \mathbf{f}(\mathbf{c}) - \epsilon^2\Delta\mathbf{c} + \beta(\mathbf{c}), && (6.21)
\end{aligned}$$

where $\mathbf{g} = (0, -1)$ and ϵ is redefined according to the scaling. The dimensionless parameters are the Reynolds number, $Re = \rho_c U_c L_c/\eta$, Froude number, $Fr = U_c/\sqrt{gL_c}$, and Peclet number, $Pe = U_c L_c/(M\mu_c)$. Using $U_c = \sqrt{gL_c}$, we have $Re = \rho_c U_c L_c/\eta = \rho_c g^{1/2} {L_c}^{3/2}/\eta$ and $Fr = U_c/\sqrt{gL_c} = \sqrt{gL_c}/\sqrt{gL_c} = 1$. By applying our scheme, we can solve the multi-component advective CH system (6.20) and (6.21) in a decoupled way and solving the multi-component NSCH system (6.18)-(6.21) becomes solving the binary NSCH system. For a detailed description of the numerical method used in solving the binary NSCH system, please refer to Ref. [156].

To model phase separation of a five-component mixture in a gravitational field, we take an initial velocity field as zero, $\mathbf{u} = \mathbf{0}$, and the initial conditions for $\mathbf{c}$ are randomly distributed between 0 and 1. $\rho(\mathbf{c}) = \sum_{i=1}^{5} \rho_i c_i$ ($c_5 = 1 - c_1 - c_2 - c_3 - c_4$ and ρ_i is the ith fluid density) and $\rho_i = 6 - i$ for $i = 1, \ldots, 5$. A mesh size 128×128 is used on the unit square domain and we choose $\Delta t = 2.0 \times 10^{-3}$, $\epsilon = 0.0047$, $Re = 3000$, and $Pe = 0.1/\epsilon$. Figure 6.24 shows the time evolution of the five-component mixture system in a gravitational field. Fluid 1 is represented by the black region, fluid 2 by the dark gray region, fluid 3 by the gray region, fluid 4 by the light gray region, and fluid 5 by the white region. From Figure 6.24 we see that the gravity affects a multi-component simulation by pulling the heavy fluid to the bottom of the computational domain.

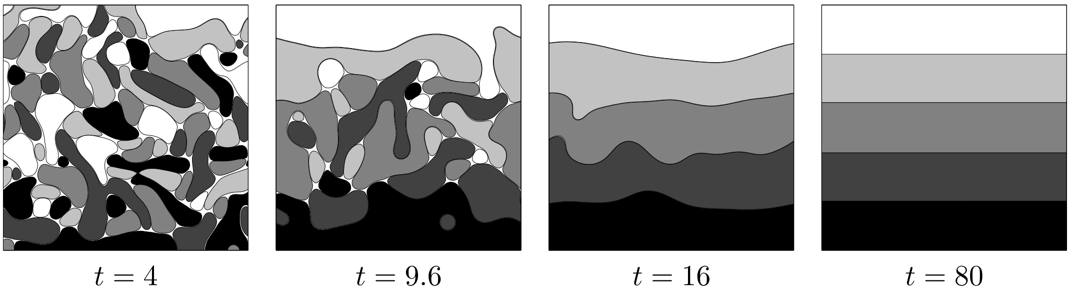

$t = 4$ $t = 9.6$ $t = 16$ $t = 80$

FIGURE 6.24. Phase separation of a five-component mixture in a gravitational field. Fluid 1 is represented by the black region, fluid 2 by the dark gray region, fluid 3 by the gray region, fluid 4 by the light gray region, and fluid 5 by the white region. Times are shown below each figure.

6.4.2. The Rayleigh-Taylor instability of a five-component mixture. When a heavy fluid is superposed over a light fluid in a gravitational field, the fluid interface is unstable. Any perturbation of this interface tends to grow with time, producing the phenomena known as the Rayleigh-Taylor instability. The phenomena are the penetration of both heavy and light fluids into each other. The Rayleigh-Taylor instability for a fluid in a gravitational field was originally introduced by Rayleigh [206] and later applied to all accelerated fluids by Taylor [207].

In this section, we study the Rayleigh-Taylor instability of a five-component mixture. In the simulations, we have two initial states as shown in Figure 6.25 (a) and Figure 6.26 (a). The initial velocity is zero. $\rho(\mathbf{c}) = \sum_{i=1}^{5} \rho_i c_i$ ($c_5 = 1 - c_1 - c_2 - c_3 - c_4$ and ρ_i is the ith fluid density) and $\rho_i = 6 - i$ for $i = 1, \dots, 5$. A mesh size 128×512 is used on a domain $\Omega = (0, 1) \times (0, 4)$ and we choose $\Delta t = 2.0 \times 10^{-3}$, $\epsilon = 0.0047$, $Re = 3000$, and $Pe = 0.1/\epsilon$. The results are presented in Figures 6.25 (b)-(g) and Figures 6.26 (b)-(g). The area shown by black indicates the fluid 1 region, while the dark gray, gray, light gray, and white color regions stand for the fluid 2, 3, 4, and 5 domains, respectively. We observe that our proposed method is a powerful tool to simulate the Rayleigh-Taylor instability between multi-component fluids.

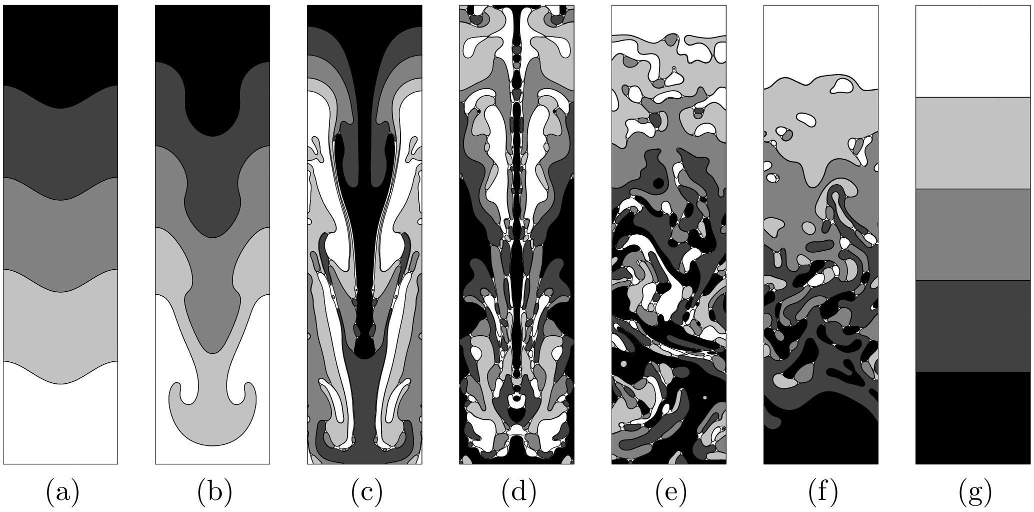

(a) (b) (c) (d) (e) (f) (g)

FIGURE 6.25. The Rayleigh-Taylor instability of a five-component mixture. Times are $t = 0$, 3, 6, 10, 20, 40, and 160 (left to right).

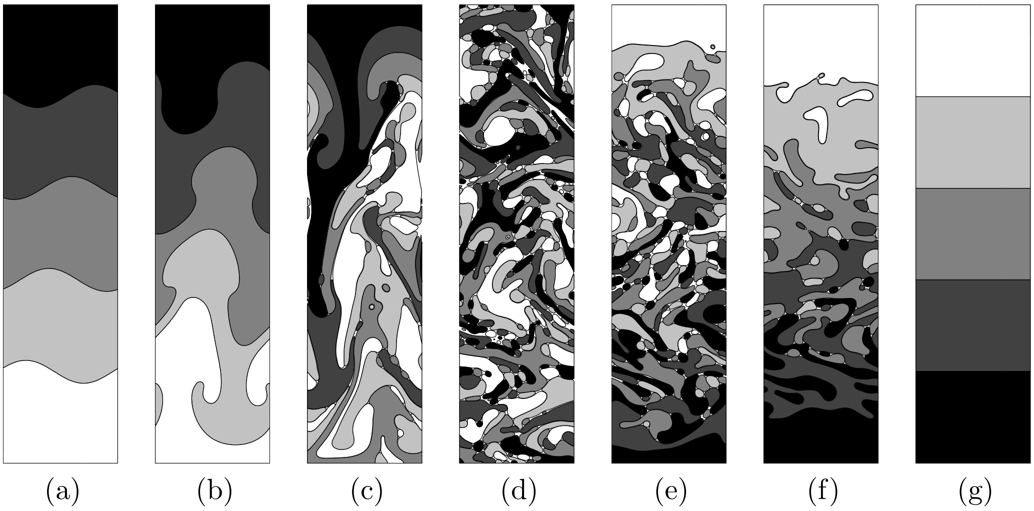


FIGURE 6.26. The Rayleigh-Taylor instability of a five-component mixture. Times are $t = 0$, 3, 6, 10, 30, 50, and 200 (left to right).

6.5. Numerical experiments for the multi-component Allen–Cahn system

6.5.1. The efficiency of the proposed scheme. As mentioned in Section 5.6, we can solve the N-component AC system in a decoupled way by using our scheme. In order to show the efficiency of the proposed scheme, we consider phase separation of $N = 3, 4, \ldots, 10$ components in the unit square domain $\Omega = (0, 1) \times (0, 1)$. For each number of components, the initial condition is a randomly chosen superposition of circles. We choose $h = 1/128$, $\Delta t = 10h$, and $\epsilon = 0.0047$ and perform 30000 time steps. The evolution of the interface is shown in Figure 6.27. Rows 1 and 2 correspond to $t = 3000\Delta t$ and $30000\Delta t$, respectively. Table 6.13 provides the average CPU time (in seconds) during 30000 time steps for each number of components. The average CPU time versus number of components is shown in Figure 6.28. The results suggest that the convergence rate of average CPU time is linear with respect to number of components.

N	3	4	5	6	7	8	9	10
Average CPU time	0.0498	0.0742	0.0998	0.1254	0.1526	0.1779	0.2056	0.2329

TABLE 6.13. Average CPU times (*sec.*) for different numbers of components.

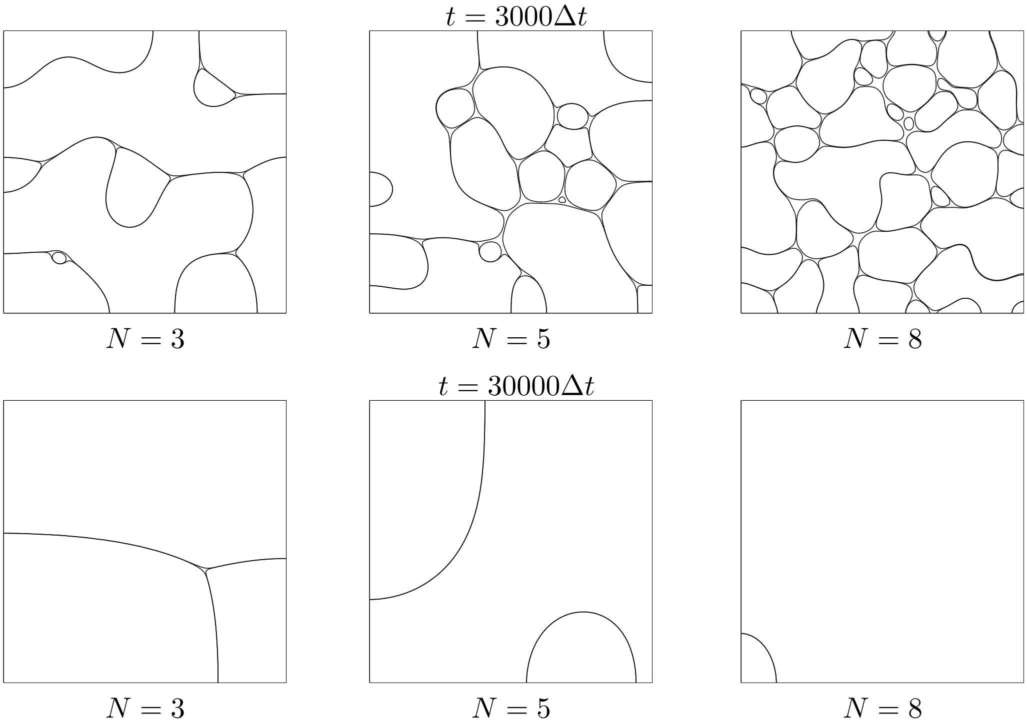


FIGURE 6.27. Phase separation of $N = 3$, 5, and 8 components. Rows 1 and 2 correspond to $t = 3000\Delta t$ and $30000\Delta t$, respectively.

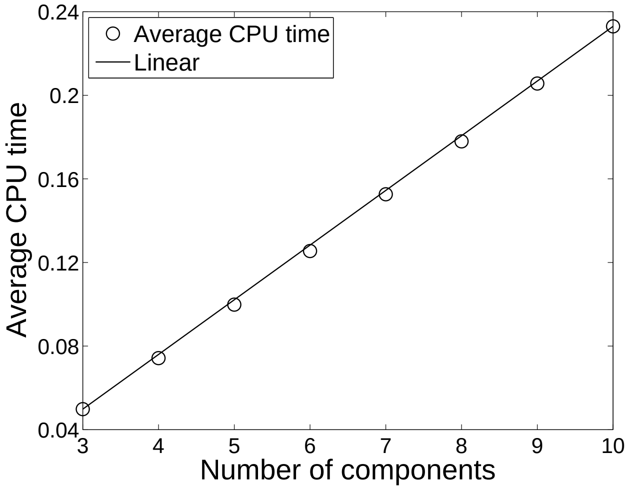


FIGURE 6.28. Average CPU time versus number of components.

Chapter 7

Conclusion

To model multi-component fluid flows, we considered the vector valued phase-field, velocity, and pressure which are governed by the N-component advective Cahn–Hilliard and modified Navier–Stokes equations. For the N-component Cahn–Hilliard equation, we presented a practically unconditionally gradient stable conservative nonlinear numerical scheme. The scheme was based on a nonlinear splitting method and was solved by an efficient and accurate nonlinear multigrid method. And the scheme allows us to convert the N-component Cahn–Hilliard equation into a system of $N-1$ binary Cahn–Hilliard equations and significantly reduces the required computer memory and CPU time. The Cahn–Hilliard–Navier–Stokes equations can be applied to any number of fluid components. And, to investigate the buoyancy driven interpenetration of fluids with different densities, we implemented a time-dependent pressure boundary condition through a time-dependent density field at the boundary. Due to the pressure boundary treatment, we can perform long time evolutions resulting in an equilibrium state.

We presented an unconditionally stable second-order hybrid numerical method for solving the Allen–Cahn equation. The proposed method was based on operator splitting techniques. The Allen–Cahn equation was divided into a linear and a nonlinear equation. First, the linear equation was discretized using an implicit Euler's scheme and the resulting discrete system of equations was solved by a multigrid method. The nonlinear equation was then solved analytically due to the availability of a closed-form solution. In particular, we applied this technique to dendritic growth simulation. A great challenge in the simulation with various supercoolings is the large difference in time and length scales. This introduces a severe time step restriction for stability. The proposed scheme allows the use of a sufficiently large time step without the technical limitations.

Appendix

In this Appendix, we present a source code written in C language for the Cahn-Hilliard equation.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#define nx 128
#define ny 128
#define iloop for(i=1;i<=nx;i++)
#define jloop for(j=1;j<=ny;j++)
#define ijloop int i, j; iloop jloop
#define iloopt for(i=1;i<=nxt;i++)
#define jloopt for(j=1;j<=nyt;j++)
#define ijloopt int i, j; iloopt jloopt
void initialization(double **cn);
void Cahn_Hilliard(double **cn, double **cnp);
void source(double **cn, double **src_c, double **src_mu);
void vcycle(double **cnp, double **mu, double **sor_c, double **sor_mu,
            int nxf, int nyf, int ilevel);
void relax(double **cnp, double **mu, double **sor_c, double **sor_mu,
           int ilevel, int nxt, int nyt);
void defect(double **def_c, double **def_mu, double **cn, double **cnp,
            double **sor_c, double **sor_mu, int nxf, int nyf);
void LS(double **LSc, double **LSmu, double **cnp, double **mu,
        int nxt, int nyt);
void laplace(double **a, double **lap_a, int nxt, int nyt);
void restrict(double **cf, double **cc, double **muf, double **muc,
              int nxt, int nyt);
void prolong(double **cc, double **cf, double **muc, double **muf,
             int nxt, int nyt);
double **dmatrix(long nrl, long nrh, long ncl, long nch);
void zero_matrix(double **a, int nxt, int nyt);
void free_dmatrix(double **m, long nrl, long nrh, long ncl, long nch);
void mat_add(double **a, double **b, double **c, int nxt, int nyt);
void mat_sub(double **a, double **b, double **c, int nxt, int nyt);
void mat_copy(double **a, double **b, int nxt, int nyt);
double error(double **ct, double **cnp);
```

```
void print_data(double **cnp);
int n_level, c_relax;
double **tmp1, **tmp2, **ct, **mu, **sor_c, **sor_mu, xleft, xright,
       yleft, yright, h, h2, dt, epsilon, Cahn;
int main()
{
   extern int n_level, c_relax;
   extern double **tmp1, **tmp2, **ct, **mu, **sor_c, **sor_mu,
                 xleft, xright, yleft, yright, h, h2, dt, epsilon, Cahn;
   int it, max_it, print_interval, count=1;
   double **cn, **cnp;
   FILE *fphi;
   if (nx<ny)  n_level=(int)(log(nx)/log(2)+0.1);
   else        n_level=(int)(log(ny)/log(2)+0.1);
   c_relax=5, max_it=50, print_interval=max_it/2,
   xleft=0.0, xright=1.0, yleft=0.0, yright=1.0;
   h=xright/(double)nx, h2=pow(h,2), dt=h, epsilon=h,
   Cahn=pow(epsilon,2);
   tmp1=dmatrix(1, nx, 1, ny);  tmp2=dmatrix(1, nx, 1, ny);
   ct=dmatrix(1, nx, 1, ny);  mu=dmatrix(1, nx, 1, ny);
   sor_c=dmatrix(1, nx, 1, ny);  sor_mu=dmatrix(1, nx, 1, ny);
   cn=dmatrix(1, nx, 1, ny);  cnp=dmatrix(1, nx, 1, ny);

   initialization(cnp);

   fphi=fopen("phi.m","w");
   fprintf(fphi,"A=[ \n");
   fclose(fphi);

   print_data(cnp);

   for (it=1; it<=max_it; it++) {
       mat_copy(cn, cnp, nx, ny);
       Cahn_Hilliard(cn, cnp);
       if (it % print_interval==0) {
          print_data(cnp);
          printf("print out counts %d \n", count++); }
       printf("Iteration is  %d \n", it); }

   fphi = fopen("phi.m","a");
   fprintf(fphi,"]; surf(A’); shading interp \n");
   fclose(fphi);

   return 0;
}
void initialization(double **cn)
{
   ijloop {
```

```
     cn[i][j]=0.3*(0.5-rand()/(double)RAND_MAX);
     mu[i][j]=0.0; }
}
void Cahn_Hilliard(double **cn, double **cnp)
{
   int max_it=300, iter=1;
   double tol=1.0e-5, resid=1.0;
   source(cn, sor_c, sor_mu);
   mat_copy(ct, cn, nx, ny);
   while (iter <= max_it && resid > tol) {
       vcycle(cnp, mu, sor_c, sor_mu, nx ,ny, 1);
       resid=error(ct, cnp);
       mat_copy(ct, cnp, nx, ny);
       iter++; }
   printf("Error is %12.10f %d",resid,iter-1);
}
void source(double **cn, double **src_c, double **src_mu)
{
   ijloop {
       src_c[i][j]=cn[i][j]/dt;
       src_mu[i][j]=pow(cn[i][j],3)-3.0*cn[i][j];}
}
void vcycle(double **cnp, double **mu, double **sor_c, double **sor_mu,
            int nxf, int nyf, int ilevel)
{
   relax(cnp, mu, sor_c, sor_mu, ilevel, nxf, nyf);

   if (ilevel<n_level) {

   double **def_c, **def_mu, **codef_c, **codef_mu,
          **fidef_c, **fidef_mu;

      def_c=dmatrix(1, nxf/2, 1, nyf/2);
      def_mu=dmatrix(1, nxf/2, 1, nyf/2);
      fidef_c=dmatrix(1, nxf, 1, nyf);
      fidef_mu=dmatrix(1, nxf, 1, nyf);
      codef_c=dmatrix(1, nxf/2, 1, nyf/2);
      codef_mu=dmatrix(1, nxf/2, 1, nyf/2);

      defect(def_c, def_mu, cnp, mu, sor_c, sor_mu, nxf, nyf);
      zero_matrix(codef_c, nxf/2, nyf/2);
      zero_matrix(codef_mu, nxf/2, nyf/2);
      vcycle(codef_c, codef_mu, def_c, def_mu, nxf/2, nyf/2, ilevel+1);
      prolong(codef_c, fidef_c, codef_mu, fidef_mu, nxf/2, nyf/2);
      mat_add(cnp, cnp, fidef_c, nxf, nyf);
      mat_add(mu, mu, fidef_mu, nxf, nyf);
      relax(cnp, mu, sor_c, sor_mu, ilevel, nxf, nyf);
      free_dmatrix(def_c, 1, nxf/2, 1, nyf/2);
```

```
      free_dmatrix(def_mu, 1, nxf/2, 1, nyf/2);
      free_dmatrix(fidef_c, 1, nxf, 1, nyf);
      free_dmatrix(fidef_mu, 1, nxf, 1, nyf);
      free_dmatrix(codef_c, 1, nxf/2, 1, nyf/2);
      free_dmatrix(codef_mu, 1, nxf/2, 1, nyf/2);
   }
}
void restrict(double **cf, double **cc, double **muf, double **muc,
              int nxt, int nyt)
{
   ijloopt {
       cc[i][j]=0.25*(cf[2*i][2*j]+cf[2*i-1][2*j]
                      +cf[2*i][2*j-1]+cf[2*i-1][2*j-1]);
       muc[i][j]=0.25*(muf[2*i][2*j]+muf[2*i-1][2*j]
                       +muf[2*i][2*j-1]+muf[2*i-1][2*j-1]); }
}
void prolong(double **cc, double **cf, double **muc, double **muf,
             int nxt, int nyt)
{
   ijloopt {
       cf[2*i][2*j]=cf[2*i-1][2*j]
       =cf[2*i][2*j-1]=cf[2*i-1][2*j-1]=cc[i][j];
       muf[2*i][2*j]=muf[2*i-1][2*j]
       =muf[2*i][2*j-1]=muf[2*i-1][2*j-1]=muc[i][j]; }
}
void LS(double **LSc, double **LSmu, double **cnp, double **mu,
        int nxt, int nyt)
{
   double **lap_mu, **lap_c;

   lap_mu=dmatrix(1, nxt, 1, nyt);
   lap_c=dmatrix(1, nxt, 1, nyt);
   laplace(cnp, lap_c, nxt, nyt);
   laplace(mu, lap_mu, nxt, nyt);

   ijloopt {
      LSc[i][j]=cnp[i][j]/dt-lap_mu[i][j];
      LSmu[i][j]=- 2.0*cnp[i][j]+Cahn*lap_c[i][j]+mu[i][j]; }

   free_dmatrix(lap_mu, 1, nxt, 1, nyt);
   free_dmatrix(lap_c, 1, nxt, 1, nyt);
}
void defect(double **def_c, double **def_mu, double **cn, double **cnp,
            double **sor_c, double **sor_mu, int nxf, int nyf)
{
   LS(tmp1, tmp2, cn, cnp, nxf, nyf);
   mat_sub(tmp1, sor_c, tmp1, nxf, nyf);
   mat_sub(tmp2, sor_mu, tmp2, nxf, nyf);
```

```
   restrict(tmp1, def_c, tmp2, def_mu, nxf/2, nyf/2);
}
double error(double **ct, double **cnp)
{
   double value=0.0;
   ijloop {
       if (fabs(ct[i][j]-cnp[i][j]) > value)
          value=fabs(ct[i][j]-cnp[i][j]); }

   return value;
}
void relax(double **cnp, double **mu, double **sor_c, double **sor_mu,
           int ilevel, int nxt, int nyt)
{
   int iter;
   double ht2, a[4], f[2], det;
   ht2 = pow(xright/(double) nxt,2);

   for (iter=1; iter<=c_relax; iter++) {
   ijloopt {
    a[0]=1.0/dt, a[1]=0.0, a[2]=-2.0, a[3]=1.0;
    f[0]=sor_c[i][j], f[1]=sor_mu[i][j];

    if (i>1) {a[1]+=1.0/ht2,a[2]-=Cahn/ht2;
              f[0]+=mu[i-1][j]/ht2,f[1]-=Cahn*cnp[i-1][j]/ht2;}
    if (i<nxt) {a[1]+=1.0/ht2,a[2]-=Cahn/ht2;
                f[0]+=mu[i+1][j]/ht2,f[1]-=Cahn*cnp[i+1][j]/ht2;}
    if (j>1) {a[1]+=1.0/ht2,a[2]-=Cahn/ht2;
              f[0]+=mu[i][j-1]/ht2,f[1]-=Cahn*cnp[i][j-1]/ht2;}
    if (j<nyt) {a[1]+=1.0/ht2,a[2]-=Cahn/ht2;
                f[0]+=mu[i][j+1]/ht2,f[1]-=Cahn*cnp[i][j+1]/ht2;}
    det = a[0]*a[3] - a[1]*a[2];
    cnp[i][j] = (a[3]*f[0] - a[1]*f[1])/det;
    mu[i][j] = (-a[2]*f[0] + a[0]*f[1])/det;}
   }
}
void laplace(double **a, double **lap_a, int nxt, int nyt)
{
   double ht2, value;
   ht2 = pow(xright / (double) nxt, 2);

   ijloopt {
       value=0.0;
       if (i<nxt) value += a[i+1][j]-a[i][j];
       if (i>1)   value -= a[i][j]-a[i-1][j];
       if (j<nyt) value += a[i][j+1]-a[i][j];
       if (j>1)   value -= a[i][j]-a[i][j-1];
       lap_a[i][j]=value/ht2;}
```

```
}
void mat_add(double **a, double **b, double **c, int nxt, int nyt)
{
   ijloopt a[i][j] = b[i][j]+c[i][j];
}
void zero_matrix(double **a, int nxt, int nyt)
{
   ijloopt a[i][j]=0.0;
}
void mat_copy(double **a, double **b, int nxt, int nyt)
{
   ijloopt a[i][j]=b[i][j];
}
double **dmatrix(long nrl, long nrh, long ncl, long nch)
{
   double **m;
   long i, nrow=nrh-nrl+1+1, ncol=nch-ncl+1+1;
   m=(double **) malloc((nrow)*sizeof(double*));
   m+=1;
   m-=nrl;
   m[nrl]=(double *) malloc((nrow*ncol)*sizeof(double));
   m[nrl]+=1;
   m[nrl]-=ncl;
   for (i=nrl+1; i<=nrh; i++) m[i]=m[i-1]+ncol;

   return m;
}
void free_dmatrix(double **m, long nrl, long nrh, long ncl, long nch)
{
   free(m[nrl]+ncl-1);
   free(m+nrl-1);
}
void mat_sub(double **a, double **b, double **c, int nxt, int nyt)
{
   ijloopt a[i][j]=b[i][j]-c[i][j];
}
void print_data(double **cnp)
{
   int i, j;
   FILE *fp;
   fp = fopen("phi.m","a");
   iloop {
       jloop { fprintf(fp, "  %f", cnp[i][j]);}
       fprintf(fp, "\n");}
   fclose(fp);
}
```

Bibliography

[1] D.M. Anderson, G.B. McFadden and A.A. Wheeler, Diffuse-interface methods in fluid mechanics, Appl. Math. Lett., 30 (1998), pp. 139-165.

[2] G. Caginalp and X.F. Chen, Phase field equations in the singular limit of sharp interface problems, in On the evolution of phase boundaries (Minneapolis, MN, 1990.91), Springer, New York, 1992, pp. 1-27.

[3] F. Boyer, A theoretical and numerical model for the study of incompressible mixture flows, Computers and Fluids, 31 (2002), pp. 41-68.

[4] M.E. Gurtin, D. Polignone, and J. Viñnals, Two-phase binary fluids and immiscible fluids described by an order parameter,Math. Models Methods Appl. Sci., 6 (1996), pp. 815-831.

[5] J. Lowengrub and L. Truskinovsky, Quasi-incompressible Cahn-Hilliard fluids and topological transitions, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci., 454 (1998), pp. 2617-2654.

[6] R. Borcia and M. Bestehorn, Phase-field for Marangoni convection in liquid-gas systems with a deformable interface, Phys. Rev. E, 67 (2003), 066307.

[7] V.V. Khatavkar, P.D. Anderson, P.C. Duineveld and H.H.E. Meijer, Diffuse-interface modelling of droplet impact, J. Fluid Mech., 581 (2007), 97-127.

[8] L.A. Caffarelli and N.E. Muler, An L^∞ bound for solutions of the Cahn-Hilliard equation, Arch. Rational Mech. Anal., 133 (1995), 129-144.

[9] Boussinesq J (1903) Theorie analytique de la chaleur, Vol. 2. Gauthier-Villars, Paris.

[10] Lighthill J (1978) Waves in fluids. Cambridge University Press, Cambridge.

[11] Jacqmin D (1999) Calculation of two-phase Navier-Stokes flows using phase-field modeling. J Comput Phys 155:96-127.

[12] Tryggvason G (1988) Numerical simulations of the Rayleigh-Taylor instability. J Comput Phys 75:253-282.

[13] Aref H, Tryggvason G (1989) Model of Rayleigh-Taylor instability. Phys Rev Lett 62:749-752.

[14] Han J, Tryggvason G (1999) Secondary breakup of axisymmetric liquid drops. I. Acceleration by a constant body force. Phys Fluids 11:3650-3667.

[15] Young Y-N, Tufo H, Dubey A, Rosner R (2001) On the miscible Rayleigh-Taylor instability: two and three dimensions. J Fluid Mech 447:377-408.

[16] Liu C, Shen J (2003) A phase field model for the mixture of two incompressible fluids and its approximation by a Fourier-spectral method. Physica D 179:211-228.

[17] Kim JS, Lowengrub JS (2005) Phase field modeling and simulation of three-phase flows. Interface Free Bound 7:435-466.

[18] Vladimirova N, Rosner R (2003) Model flames in the Boussinesq limit: The effects of feedback. Phys Rev E 67:066305-1–10.

[19] Vladimirova N, Rosner R (2005) Model flames in the Boussinesq limit: The case of pulsating fronts. Phys Rev E 71:067303-1–4.

[20] Vladimirova N (2007) Model flames in the Boussinesq limit: Rising bubbles. Combust Theor Model 11:377-400.

[21] Celani A, Mazzino A, Muratore-Ginanneschi P, Vozella L (2009) Phase-field model for the Rayleigh-Taylor instability of immiscible fluids. J Fluid Mech 622:115-134.

[22] Forbes LK (2009) The Rayleigh-Taylor instability for inviscid and viscous fluids. J Eng Math 65:273-290.

[23] Forbes LK (2011) A cylindrical Rayleigh-Taylor instability: radial outflow from pipes or stars. J Eng Math 70:205-224.

[24] Boffetta G, Mazzino A, Musacchio S, Vozella L (2010) Statistics of mixing in three-dimensional Rayleigh-Taylor turbulence at low Atwood number and Prandtl number one. Phys Fluids 22:035109-1–8.

[25] Unverdi SO, Tryggvason G (1992) A front-tracking method for viscous, incompressible, multi-fluid flows. J Comput Phys 100:25-37.

[26] Popinet S, Zaleski S (1999) A front-tracking algorithm for accurate representation of surface tension. Int J Numer Meth Fluids 30:775-793.

[27] Glimm J, Grove JW, Li XL, Oh W, Sharp DH (2001) A critical analysis of Rayleigh-Taylor growth rates. J Comput Phys 169:652-677.

[28] Liu X, Li Y, Glimm J, Li XL (2007) A front tracking algorithm for limited mass diffusion. J Comput Phys 222:644-653.

[29] Terashima H, Tryggvason G (2009) A front-tracking/ghost-fluid method for fluid interfaces in compressible flows. J Comput Phys 228:4012-4037.

[30] Hirt CW, Nichols BD (1981) Volume of fluid (VOF) method for the dynamics of free boundaries. J Comput Phys 39:201-225.

[31] Rudman M (1997) Volume-tracking methods for interfacial flow calculations, Int J Numer Meth Fluids 24:671-691.

[32] Gerlach D, Tomar G, Biswas G, Durst F (2006) Comparison of volume-of-fluid methods for surface-tension dominant two-phase flows. Int J Heat and Mass Transfer 49:740-754.

[33] Gopala VR, van Wachem BGM (2008) Volume of fluid methods for immiscible-fluid and free-surface flows. Chem Eng J 141:204-221.

[34] Raessi M, Mostaghimi J, Bussmann M (2010) A volume-of-fluid interfacial flow solver with advected normals. Comput & Fluids 39:1401-1410.

[35] Nie X, Qian Y-H, Doolen GD, Chen S (1998) Lattice Boltzmann simulation of the two-dimensional Rayleigh-Taylor instability. Phys Rev E 58:6861-6864.

[36] He X, Chen S, Zhang R (1999) A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh-Taylor instability. J Comput Phys 152:642-663.

[37] He X, Zhang R, Chen S, Doolen GD (1999) On the three-dimensional Rayleigh-Taylor instability. Phys Fluids 11:1143-1152.

[38] Clark TT (2003) A numerical study of the statistics of a two-dimensional Rayleigh-Taylor mixing layer. Phys Fluids 15:2413-2423.

[39] Chang YC, Hou TY, Merriman B, Osher S (1996) A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. J Comput Phys 124:449-464.

[40] Gomez P, Hernandez J, Lopez J (2005) On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows. Int J Numer Meth Eng 63:1478-1512.

[41] Herrmann M (2008) A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. J Comput Phys 227:2674-2706.

[42] Sheu TWH, Yu CH, Chiu PH (2009) Development of a dispersively accurate conservative level set scheme for capturing interface in two-phase flows. J Comput Phys 228:661-686.

[43] Ding H, Spelt PDM, Shu C (2007) Diffuse interface model for incompressible two-phase flows with large density ratios. J Comput Phys 226:2078-2095.

[44] Sun Y, Beckermann C (2008) A two-phase diffuse-interface model for Hele-Shaw flows with large property contrasts. Physica D 237:3089-3098.

[45] Chiu P-H, Lin Y-T (2011) A conservative phase field method for solving incompressible two-phase flows. J Comput Phys 230:185-204.

[46] Lee HG, Kim K, Kim J (2011) On the long time simulation of the Rayleigh-Taylor instability. Int J Numer Meth Eng 85:1633-1647.

[47] Rayleigh L (1883) Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density. Proc Lond Math Soc 14:170-177.

[48] Taylor G (1950) The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I. Proc Roy Soc Lond A 201:192-196.

[49] Atzeni S, Meyer-Ter-Vehn J (2004) The physics of inertial fusion: beam plasma interaction, hydrodynamics, hot dense matter. Oxford University Press, USA.

[50] Buchler JR, Livio M, Colgate SA (1980) Supernova explosions - The role of a Rayleigh-Taylor instability. Space Science Rev 27:571-577.

[51] Brecht SH, Papadopoulos K (2002) Cross field jetting of energetic ions produced by Rayleigh-Taylor instability. Naval Research Laboratory, Washington.

[52] Debnath L (1994) Nonlinear water waves. Academic Press, Boston.

[53] Keskinen MJ, Ossakow SL, Szuszczewicz EP, Holmes JC (1981) Nonlinear theory and experimental observations of the local collisional Rayleigh-Taylor instability in a descending equatorial spread F ionosphere. J Geophys Res 86:5785-5792.

[54] Ribeyre X, Tikhonchuk VT, Bouquet S (2004) Compressible Rayleigh-Taylor instabilities in supernova remnants. Phys Fluids 16:4661-4670.

[55] Badalassi VE, Ceniceros HD, Banerjee S (2003) Computation of multiphase systems with phase field models. J Comput Phys 190:371-397.

[56] Yue P, Feng JJ, Liu C, Shen J (2004) A diffuse-interface method for simulating two-phase flows of complex fluids. J Fluid Mech 515:293-317.

[57] Kim C-H, Shin S-H, Lee HG, Kim J (2009) Phase-field model for the pinchoff of liquid-liquid jets. J Korean Phys Soc 55:1451-1460.

[58] Kim J (2009) A generalized continuous surface tension force formulation for phase-field models for multi-component immiscible fluid flows. Comput Methods Appl Mech Engrg 198:3105-3112.

[59] Biben T, Kassner K, Misbah C (2005) Phase-field approach to three-dimensional vesicle dynamics. Phys Rev E 72:041921-1–15.

[60] Hohenberg PC, Halperin BI (1977) Theory of dynamic critical phenomena. Rev Mod Phys 49:435-479.

[61] Lowengrub J, Truskinovsky L (1998) Quasi-incompressible Cahn-Hilliard fluids and topological transitions. Proc R Soc Lond A 454:2617-2654.

[62] Ferziger JH, Peric M (2002) Computational methods for fluid dynamics. Springer, Berlin.

[63] Chella R, Viñals J (1996) Mixing of a two-phase fluid by cavity flow. Phys Rev E 53:3832-3840.

[64] Jacqmin D (2000) Contact-line dynamics of a diffuse fluid interface. J Fluid Mech 402:57-88.

[65] Kim J (2005) A continuous surface tension force formulation for diffuse-interface models. J Comput Phys 204:784-804.

[66] Boyer F, Lapuerta C (2006) Study of a three component Cahn-Hilliard flow model. M2AN 40:653-687.

[67] Bell JB, Colella P, Glaz HM (1989) A second-order projection method for the incompressible Navier-Stokes equations. J Comput Phys 85:257-283.

[68] Chorin AJ (1967) A numerical method for solving incompressible viscous flow problems. J Comput Phys 2:12-26.

[69] Li J, Renardy Y (2000) Numerical study of flows of two immiscible liquids at low Reynolds number. SIAM Rev 42:417-439.

[70] Harlow FH, Welch JE (1965) Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Phys Fluids 8:2182-2189.

[71] Trottenberg U, Oosterlee C, Schüller A (2001) Multigrid. Academic Press, USA.

[72] Kim J (2007) A numerical method for the Cahn-Hilliard equation with a variable mobility. Commun Nonlinear Sci Numer Simulat 12:1560-1571.

[73] Kim J, Bae H-O (2008) An unconditionally stable adaptive mesh refinement for Cahn-Hilliard equation. J Korean Phys Soc 53:672-679.

[74] Rieger R, Weiss C, Wigley G, Bart H-J, Marr R. Investigating the process of liquid-liquid extraction by means of computational fluid dynamics. *Computers & Chemical Engineering* 1996; **20**:1467–1475.

[75] West JL. *Polymer-dispersed liquid crystals.* Liquid-Crystalline Polymers Chap. 32, 1990.

[76] Tucker CL, Moldenaers P. Microstructural evolution in polymer blends. *The Annual Review of Fluid Mechanics* 2002; **34**:177–210.

[77] Sundaresan S. Modeling the hydrodynamics of multiphase flow reactors: Current status and challenges. *AIChE Journal* 2000; **46**:1102–1105.

[78] Crowe CT, Sommerfeld M, Tsuji Y. *Multiphase Flows with Droplets and Particles.* CRC Press, 1998.

[79] Dukowicz JK. A particle-fluid numerical model for liquid sprays. *Journal of Computational Physics* 1980; **35**:229–253.

[80] Ménétrier-Deremble L, Tabeling P. Droplet breakup in microfluidic junctions of arbitrary angles. *Physical Review E* 2006; **74**:035303-1–035303-4.

[81] De Menech M. Modeling of droplet breakup in a microfluidic T-shaped junction with a phase-field model. *Physical Review E* 2006; **73**:031505-1–031505-9.

[82] Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, Han J, Nas S, Jan Y-J. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics* 2001; **169**:708–759.

[83] Pederzani J, Haj-Hariri H. A numerical method for the analysis of flexible bodies in unsteady viscous flows. *International Journal for Numerical Methods in Engineering* 2006; **68**:1096–1112.

[84] Udaykumar HS, Kan H-C, Shyy W, Tran-Son-Tay R. Multiphase dynamics in arbitrary geometries on fixed cartesian grids. *Journal of Computational Physics* 1997; **137**:366–405.

[85] Kietzmann CvL, Van Der Walt JP, Morsi YS. A free-front tracking algorithm for a control-volume-based Hele-Shaw method. *International Journal for Numerical Methods in Engineering* 1998; **41**:253–269.

[86] Mehdi-Nejad V, Mostaghimi J, Chandra S. Modelling heat transfer in two-fluid interfacial flows. *International Journal for Numerical Methods in Engineering* 2004; **61**:1028–1048.

[87] Battaglia L, Storti MA, D'Elia J. Bounded renormalization with continuous penalization for level set interface-capturing methods. *International Journal for Numerical Methods in Engineering* 2010; **84**:830–848.

[88] Challis VJ, Guest JK. Level set topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Engineering* 2009; **79**:1284–1308.

[89] Sochnikov V, Efrima S. Level set calculations of the evolution of boundaries on a dynamically adaptive grid. *International Journal for Numerical Methods in Engineering* 2003; **56**:1913–1929.

[90] Griffith BE, Hornung RD, McQueen DM, Peskin CS. An adaptive, formally second order accurate version of the immersed boundary method. *Journal of Computational Physics* 2007; **223**:10–49.

[91] Griffith BE, Peskin CS. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. *Journal of Computational Physics* 2005; **208**:75–105.

[92] LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis* 1994; **31**:1019–1044.

[93] Peskin CS. The immersed boundary method. *Acta Numerica* 2002; **11**:479–517.

[94] Roma AM, Peskin CS, Berger MJ. An adaptive version of the immersed boundary method. *Journal of Computational Physics* 1999; **153**:509–534.

[95] Shin SJ, Huang W-X, Sung HJ. Assessment of regularized delta functions and feedback forcing schemes for an immersed boundary method. *International Journal for Numerical Methods in Fluids* 2008; **58**:263–286.

[96] Stockie JM. *Analysis and computation of immersed boundaries, with application to pulp fibres*. Ph.D. thesis, Institute of Applied Mathematics, University of British Columbia, 1997.

[97] Yang X, Zhang X, Li Z, He G-W. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics* 2009; **228**:7821–7836.

[98] Chen L, Li Y. A numerical method for two-phase flows with an interface. *Environmental Modelling & Software* 1998; **13**:247–255.

[99] Coward AV, Renardy YY, Renardy M, Richards JR. Temporal evolution of periodic disturbances in two-layer Couette flow. *Journal of Computational Physics* 1997; **132**:346–361.

[100] Gao D, Morley NB, Dhir V. Numerical simulation of wavy falling film flow using VOF method. *Journal of Computational Physics* 2003; **192**:624–642.

[101] Gueyffier D, Li J, Nadim A, Scardovelli R, Zaleski S. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics* 1999; **152**:423–456.

[102] Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics* 1994; **113**:134–147.

[103] Lörstad D, Fuchs L. High-order surface tension VOF-model for 3D bubble flows with high density ratio. *Journal of Computational Physics* 2004; **200**:153–176.

[104] Meier M, Yadigaroglu G, Smith BL. A novel technique for including surface tension in PLIC-VOF methods. *European Journal of Mechanics - B/Fluids* 2002; **21**:61–73.

[105] Rudman M. A volume-tracking method for incompressible multifluid flows with large density variations. *International Journal for Numerical Methods in Fluids* 1998; **28**:357–378.

[106] Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *The Annual Review of Fluid Mechanics* 1999; **31**:567–603.

[107] Tang H, Wrobel LC, Fan Z. Tracking of immiscible interfaces in multiple-material mixing processes. *Computational Materials Science* 2004; **29**:103–118.

[108] Engquist B, Tornberg A-K, Tsai R. Discretization of dirac delta functions in level set methods. *Journal of Computational Physics* 2005; **207**:28–51.

[109] Jin S, Wen X. Hamiltonian-preserving schemes for the Liouville equation with discontinuous potentials. *Communications in Mathematical Sciences* 2005; **3**:285–315.

[110] Jin S, Liu H, Osher S, Tsai Y-HR. Computing multivalued physical observables for the semiclassical limit of the Schrödinger equation. *Journal of Computational Physics* 2005; **205**:222–241.

[111] Jin S, Liu H, Osher S, Tsai Y-HR. Computing multi-valued physical observables for high frequency limit of symmetric hyperbolic systems. *Journal of Computational Physics* 2005; **210**:497–518.

[112] Smereka P. The numerical approximation of a delta function with application to level set methods. *Journal of Computational Physics* 2006; **211**:77–90.

[113] Sussman M, Smereka P, Osher S. A level set method for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.

[114] Towers JD. Two methods for discretizing a delta function supported on a level set. *Journal of Computational Physics* 2007; **220**:915–931.

[115] Wen X. High order numerical methods to a type of delta function integrals. *Journal of Computational Physics* 2007; **226**:1952–1967.

[116] Cahn JW. On spinodal decomposition. *Acta Metallurgica* 1961; **9**:795–801.

[117] Cahn JW, Hilliard JE. Free energy of a nonuniform system. I. Interfacial free energy. *Journal of Chemical Physics* 1958; **28**:258–267.

[118] Eyre DJ. An Unconditionally Stable One-Step Scheme for Gradient Systems, http://www.math.utah.edu/~eyre/research/methods/stable.ps.

[119] Vollmayr-Lee BP, Rutenberg AD. Fast and accurate coarsening simulation with an unconditionally stable time step. *Physical Review E* 2003; **68**:066703-1–066703-13.

[120] Brackbill JU, Kothe DB, Zemach C. A continuum method for modelling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.

[121] Renardy YY, Renardy M, Cristini V. A new volume-of-fluid formation for surfactants and simulations of drop deformation under shear at a low viscosity ratio. *European Journal of Mechanics - B/Fluids* 2002; **21**:49–59.

[122] Khismatullin D, Renardy YY, Renardy M. Development and implementation of VOF-PROST for 3D viscoelastic liquid.liquid simulations. *Journal of Non-Newtonian Fluid Mechanics* 2006; **140**:120–131.

[123] Pilliod Jr. JE, Puckett EG. Second-order volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics* 2004; **199**:465–502.

[124] Renardy YY, Renardy M. PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of Computational Physics* 2002; **183**:400–421.

[125] Osher S, Sethian J. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.

[126] Hou TY, Li Z, Osher S, Zhao H. A hybrid method for moving interface problems with application to the Hele-Shaw flow. *Journal of Computational Physics* 1997; **134**:236–252.

[127] Liu X-D, Fedkiw RP, Kang M. A boundary condition capturing method for Poisson's equation on irregular domains. *Journal of Computational Physics* 2000; **160**:151–178.

[128] Peng D, Merriman B, Osher S, Zhao H, Kang M. A PDE-based fast local level set method. *Journal of Computational Physics* 1990; **155**:410–438.

[129] Tornberg A-K, Engquist B. Numerical approximations of singular source terms in differential equations. *Journal of Computational Physics* 2004; **200**:462–488.

[130] Towers JD. Finite difference methods for approximating Heaviside functions. *Journal of Computational Physics* 2009; **228**:3478–3489.

[131] Yang S-D, Lee HG, Kim J. A phase-field approach for minimizing the area of triply periodic surfaces with volume constraint. *Computer Physics Communications* 2010; **181**:1037–1046.

[132] Wang X. *Phase field models and simulations of vesicle bio-membranes*. Ph.D. thesis, Department of Mathematics, Penn State University, 2005.

[133] Yue P, Zhou C, Feng J.J. Spontaneous shrinkage of drops and mass conservation in phase-field simulations. *Journal of Computational Physics* 2007; **223**:1–9.

[134] Rayleigh L. Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density. *Proceedings of the London Mathematical Society* 1883; **14**:170–177.

[135] Taylor G. The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I. *Proceedings of the Royal Society of London A* 1950; **201**:192–196.

[136] Lee HG, Kim K, Kim J. On the long time simulation of the Rayleigh–Taylor instability. *International Journal for Numerical Methods in Engineering* 2011; **85**:1633–1647.

[137] LeVeque R. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis* 1996; **33**:627–665.

[138] Enright D, Fedkiw R, Ferziger J, Mitchell I. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics* 2002; **183**:83–116.

[139] Hyde S, Andersson S, Larsson K, Blum Z, Landh T, Lidin S, Ninham BW. *The Language of Shape: The Role of Curvature in Condensed Matter: Physics, Chemistry and Biology*. Elsevier Science, 1997.

[140] Rajagopalan S, Robb RA. Schwarz meets Schwann: Design and fabrication of biomorphic and durataxic tissue engineering scaffolds. *Medical Image Analysis* 2006; **10**:693–712.

[141] Von Schnering HG, Nesper R. Nodal surfaces of Fourier series: fundamental invariants of structured matter. *Zeitschrift für Physik B Condensed Matter* 1991; **83**:407–412.

[142] Jung Y, Chu KT, Torquato S. A variational level set approach for surface area minimization of triply-periodic surfaces. *Journal of Computational Physics* 2007; **223**:711–730.

[143] Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.

[144] Ganesan S, Matthies G, Tobiska L. On spurious velocities in incompressible flow problems with interfaces. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:1193–1202.

[145] Gerbeau J-F, Le Bris C, Bercovier M. Spurious velocities in the steady flow of an incompressible fluid subjected to external forces. *International Journal for Numerical Methods in Fluids* 1997; **25**:679–695.

[146] Gresho P, Lee R, Chan S, Leone J. A new finite element for incompressible or Boussinesq fluids. *Proceedings of the Third International Conference on Finite Elements in Flow Problems*, Banff, Canada, 1980.

[147] Pelletier D, Fortin A, Camarero R. Are FEM solutions of incompressible flows really incompressible? (Or how simple flows can cause headaches!). *International Journal for Numerical Methods in Fluids* 1989; **9**:99–112.

[148] P. Seppecher, Moving contact lines in the Cahn–Hilliard theory, Int. J. Engng. Sci. 34 (1996) 977-992.

[149] H.G. Lee, J.S. Kim, Accurate contact angle boundary conditions for the Cahn–Hilliard equations, Comput. Fluids 44 (2011) 178-186.

[150] D.M. Anderson, G.B. McFadden, A diffuse-interface description of internal waves in a near-critical fluid, Phys. Fluids 9 (1997) 1870-1879.

[151] L. de Sobrino, J. Peternelj, On capillary waves in the gradient theory of interfaces, Can. J. Phys. 63 (1985) 131-134.

[152] B.T. Nadiga, S. Zaleski, Investigations of a two-phase fluid model, Eur. J. Mech. B/Fluids 15 (1996) 885-896.

[153] C.H. Kim, S.H. Shin, H.G. Lee, J.S. Kim, Phase-field model for the pinchoff of liquid-liquid jets, J. Korean Phys. Soc. 55 (2009) 1451-1460.

[154] H. Ding, P.D.M. Spelt, C. Shu, Diffuse interface model for incompressible two-phase flows with large density ratios, J. Comput. Phys. 226 (2007) 2078-2095.

[155] A. Celani, A. Mazzino, P. Muratore-Ginanneschi, L. Vozella, Phase-field model for the Rayleigh-Taylor instability of immiscible fluids, J. Fluid Mech. 622 (2009) 115-134.

[156] H.G. Lee, K.M. Kim, J.S. Kim, On the long time simulation of the Rayleigh-Taylor instability, Int. J. Numer. Meth. Engng. 85 (2011) 1633-1647.

[157] S.M. Wise, J.S. Lowengrub, H.B. Frieboes, V. Cristini, Three-dimensional multispecies nonlinear tumor growth–I Model and numerical method, J. Theor. Biol. 253 (2008) 524-543.

[158] V. Cristini, X. Li, J.S. Lowengrub, S.M. Wise, Nonlinear simulations of solid tumor growth using a mixture model: invasion and branching, J. Math. Biol. 58 (2009) 723-763.

[159] S.M. Wise, J.S. Lowengrub, V. Cristini, An adaptive multigrid algorithm for simulating solid tumor growth using mixture models, Math. Comput. Model. 53 (2011) 1-20.

[160] D. Jasnow, J. Viñals, Coarse-grained description of thermo-capillary flow, Phys. Fluids 8 (1996) 660-669.

[161] M. Verschueren, F.N. van de Vosse, H.E.H. Meijer, Diffuse-interface modelling of thermocapillary flow instabilities in a Hele-Shaw cell, J. Fluid Mech. 434 (2001) 153-166.

[162] F. Campelo, A. Hernández-Machado, Dynamic model and stationary shapes of fluid vesicles, Eur. Phys. J. E 20 (2006) 37-45.

[163] S. Bhattacharyya, T.A. Abinandanan, A study of phase separation in ternary alloys, Bull. Mater. Sci. 26 (2003) 193-197.

[164] D. de Fontaine, A computer simulation of the evolution of coherent composition variations in solid solutions, Ph.D. thesis, Northwestern University, 1967.

[165] J.E. Morral, J.W. Cahn, Spinodal decomposition in ternary systems, Acta Metall. 19 (1971) 1037-1045.

[166] J.J. Hoyt, The continuum theory of nucleation in multicomponent systems, Acta Metall. 38 (1990) 1405-1412.

[167] C.M. Elliott, S. Luckhaus, A generalised diffusion equation for phase separation of a multi-component mixture with interfacial free energy, IMA Preprint Series, 887 (1991).

[168] D.J. Eyre, Systems of Cahn–Hilliard equations, SIAM J. Appl. Math. 53 (1993) 1686-1712.

[169] C.M. Elliott, H. Garcke, Diffusional phase transitions in multicomponent systems with a concentration dependent mobility matrix, Phys. D 109 (1997) 242-256.

[170] S. Maier-Paape, B. Stoth, T. Wanner, Spinodal decomposition for multicomponent Cahn–Hilliard systems, J. Stat. Phys. 98 (2000) 871-896.

[171] M. Honjo, Y. Saito, Numerical simulation of phase separation in Fe-Cr binary and Fe-Cr-Mo ternary alloys with use of the Cahn–Hilliard equation, ISIJ Int. 40 (2000) 914-919.

[172] J.S. Kim, K.K. Kang, J. Lowengrub, Conservative multigrid methods for Cahn–Hilliard fluids, J. Comput. Phys. 193 (2004) 511-543.

[173] J.S. Kim, A diffuse-interface model for axisymmetric immiscible two-phase flow, Appl. Math. Comput. 160 (2005) 589-606.

[174] E.V.L. de Mello, O.T. Silveira Filho, Numerical study of the Cahn–Hilliard equation in one, two and three dimensions, Phys. A 347 (2005) 429-443.

[175] S. Wise, J.S. Kim, J. Lowengrub, Solving the regularized, strongly anisotropic Cahn–Hilliard equation by an adaptive nonlinear multigrid method, J. Comput. Phys. 226 (2007) 414-446.

[176] Y. He, Y. Liu, Tao Tang, On large time-stepping methods for the Cahn–Hilliard equation, Appl. Numer. Math. 57 (2007) 616-628.

[177] H. Gómez, V.M. Calo, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of the Cahn–Hilliard phase-field model, Comput. Methods Appl. Mech. Engrg. 197 (2008) 4333-4352.

[178] L. Cueto-Felgueroso, J. Peraire, A time-adaptive finite volume method for the Cahn–Hilliard and Kuramoto-Sivashinsky equations, J. Comput. Phys. 227 (2008) 9985-10017.

[179] R.H. Stogner, G.F. Carey, B.T. Murray, Approximation of Cahn–Hilliard diffuse interface models using parallel adaptive mesh refinement and coarsening with C_1 elements, Int. J. Numer. Meth. Engng. 76 (2008) 636-661.

[180] L.-P. He, Y. Liu, A class of stable spectral methods for the Cahn–Hilliard equation, J. Comput. Phys. 228 (2009) 5101-5110.

[181] S.D. Yang, H.G. Lee, J.S. Kim, A phase-field approach for minimizing the area of triply periodic surfaces with volume constraint, Comput. Phys. Commun. 181 (2010) 1037-1046.

[182] T. Ohta, M. Motoyama, A. Ito, The kinetics and morphology of phase-separating copolymer mixtures, J. Phys.: Condens. Matter 8 (1996) A65-A80.

[183] B.F Barton, A.J. McHugh AJ, Kinetics of thermally induced phase separation in ternary polymer solutions. I. Modeling of phase separation dynamics, J. Polym. Sci. Pol. Phys. 37 (1999) 1449-1460.

[184] M.I.M. Copetti, Numerical experiments of phase separation in ternary mixtures, Math. Comput. Simul. 52 (2000) 41-51.

[185] Y.Q. Ma, Domain patterns in ternary mixtures with different interfacial properties, J. Chem. Phys. 114 (2001) 3734-3738.

[186] J.S. Kim, K.K. Kang, J. Lowengrub, Conservative multigrid methods for ternary Cahn–Hilliard systems, Comm. Math. Sci. 2 (2004) 53-77.

[187] J.S. Kim, J. Lowengrub, Phase field modeling and simulation of three-phase flows, Interfaces Free Bound. 7 (2005) 435-466.

[188] P.R. Cha, D.H. Yeon, J.K. Yoon, Phase-field model for multicomponent alloy solidification, J. Cryst. Growth 274 (2005) 281-293.

[189] F. Boyer, C. Lapuerta, Study of a three component Cahn–Hilliard flow model, ESAIM: M2AN 40 (2006) 653-687.

[190] J.S. Kim, Phase field computations for ternary fluid flows, Comput. Methods Appl. Mech. Engrg. 196 (2007) 4779-4788.

[191] Y. Xia, Y. Xu, C.W. Shu, Local discontinuous Galerkin methods for the Cahn–Hilliard type equations, J. Comput. Phys. 227 (2007) 472-491.

[192] J.W. Barrett, H. Garcke, R. Nürnberg, A parametric finite element method for fourth order geometric evolution equations, J. Comput. Phys. 222 (2007) 441-467.

[193] J.S. Kim, K.K. Kang, A numerical method for the ternary Cahn–Hilliard system with a degenerate mobility, Appl. Numer. Math. 59 (2009) 1029-1042.

[194] S. Zhou, M.Y. Wang, Multimaterial structural topology optimization with a generalized Cahn–Hilliard model of multiphase transition, Struct. Multidisc. Optim. 33 (2007) 89-111.

[195] H.G. Lee, J.S. Kim, A second-order accurate non-linear difference scheme for the N-component Cahn–Hilliard system, Phys. A 387 (2008) 4787-4799.

[196] T. Kitashima, J. Wang, H. Harada, Phase-field simulation with the CALPHAD method for the microstructure evolution of multi-component Ni-base superalloys, Intermetallics 16 (2008) 239-245.

[197] R. Nürnberg, Numerical simulations of immiscible fluid clusters, Appl. Numer. Math. 59 (2009) 1612-1628.

[198] R. Kobayashi, J.A. Warren, W.C. Carter, A continuum model of grain boundaries, Phys. D 140 (2000) 141-150.

[199] C.E. Krill III, L.-Q. Chen, Computer simulation of 3-D grain growth using a phase-field model, Acta Mater. 50 (2002) 3057-3073.

[200] J.A. Warren, R. Kobayashi, A.E. Lobkovsky, W.C. Carter, Extending phase field models of solidification to polycrystalline materials, Acta Mater. 51 (2003) 6035-6058.

[201] J. Gruber, N. Ma, Y. Wang, A.D. Rollett, G.S. Rohrer, Sparse data structure and algorithm for the phase field method, Modelling Simul. Mater. Sci. Eng. 14 (2006) 1189-1195.

[202] S. Vedantam, B.S.V. Patnaik, Efficient numerical algorithm for multiphase field simulations, Phys. Rev. E 73 (2006) 016703-1–8.

[203] H.D. Ceniceros, A.M. Roma, A nonstiff, adaptive mesh refinement-based method for the Cahn–Hilliard equation, J. Comput. Phys. 225 (2007) 1849-1862.

[204] D.J. Eyre, Unconditionally gradient stable time marching the Cahn–Hilliard equation, in Computational and Mathematical Models of Microstructural Evolution, The Material Research Society, Warrendale, PA (1998), pp. 39-46.

[205] H. Garcke, B. Nestler, B. Stoth, On anisotropic order parameter models for multi-phase systems and their sharp interface limits, Phys. D 115 (1998) 87-108.

[206] L. Rayleigh, Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density, Proc. Lond. Math. Soc. 14 (1883) 170-177.

[207] G. Taylor, The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I, Proc. Roy. Soc. A 201 (1950) 192-196.

[208] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metall. 27 (1979) 1085-1095.

[209] M. Beneš, V. Chalupecký, K. Mikula, Geometrical image segmentation by the Allen–Cahn equation, Appl. Numer. Math. 51 (2004) 187-205.

[210] J.A. Dobrosotskaya, A.L. Bertozzi, A wavelet-Laplace variational technique for image deconvolution and inpainting, IEEE. Trans. Image Process. 17 (2008) 657-663.

[211] L.C. Evans, H.M. Soner, P.E. Souganidis, Phase transitions and generalized motion by mean curvature, Comm. Pure Appl. Math. 45 (1992) 1097-1123.

[212] T. Ilmanen, Convergence of the Allen–Cahn equation to Brakke's motion by mean curvature, J. Differ. Geom. 38 (1993) 417-461.

[213] M. Katsoulakis, G.T. Kossioris, F. Reitich, Generalized motion by mean curvature with Neumann conditions and the Allen–Cahn model for phase transitions, J. Geom. Anal. 5 (1995) 255-279.

[214] M. Beneš, K. Mikula, Simulation of anisotropic motion by mean curvature-comparison of phase field and sharp interface approaches, Acta Math. Univ. Comenianae 67 (1998) 17-42.

[215] X. Feng, A. Prohl, Numerical analysis of the Allen–Cahn equation and approximation for mean curvature flows, Numer. Math. 94 (2003) 33-65.

[216] T. Ohtsuka, Motion of interfaces by an Allen–Cahn type equation with multiple-well potentials, Asymptotic Anal. 56 (2008) 87-123.

[217] X. Yang, J.J. Feng, C. Liu, J. Shen, Numerical simulations of jet pinching-off and drop formation using an energetic variational phase-field method, J. Comput. Phys. 218 (2006) 417-428.

[218] A.A. Wheeler, W.J. Boettinger, G.B. McFadden, Phase-field model for isothermal phase transitions in binary alloys, Phys. Rev. A 45 (1992) 7424-7439.

[219] M. Cheng, J.A. Warren, An efficient algorithm for solving the phase field crystal model, J. Comput. Phys. 227 (2008) 6241-6248.

[220] Y. Li, H.G. Lee, J.S. Kim, A fast, robust, and accurate operator splitting method for phase-field simulations of crystal growth, J. Cryst. Growth 321 (2011) 176-182.

[221] E.V.L. Melloa, D.F. Msyers, Numerical study of the Cahn–Hilliard equation in one, two and three dimensions, Phys. A. 347 (2005) 429-443.

[222] H. Garcke, B. Nestler, B. Stoth, On anisotropic order parameter models for multi-phase systems and their sharp interface limits, Phys. D 115 (1998) 87-108.

[223] Y. He, Y. Liu, T. Tang, On large time-stepping methods for the Cahn–Hilliard equation, Appl. Numer. Math. 57 (2007) 616-628.

[224] X. Feng, H.-J. Wu, A posteriori error estimates and an adaptive finite element method for the Allen–Cahn equation and the mean curvature flow, J. Sci. Comput. 24 (2005) 121-146.

[225] D.F. Martin, P. Colella, M. Anghel, F.J. Alexander, Adaptive mesh refinement for multiscale nonequilibrium physics, Comput. Sci. Eng. 7 (2005) 24-31.

[226] W.M. Feng, P. Yu, S.Y. Hu, Z.K. Liu, Q. Du, L.Q. Chen, Spectral implementation of an adaptive moving mesh method for phase-field equations, J. Comput. Phys. 220 (2006) 498-510.

[227] D.A. Kay, A. Tomasi, Color image segmentation by the vector-valued Allen–Cahn phase-field model: a multigrid solution, IEEE. Trans. Image Process. 18 (2009) 2330-2339.

[228] J.-W. Choi, H.G. Lee, D. Jeong, J. Kim, An unconditionally gradient stable numerical method for solving the Allen–Cahn equation, Phys. A. 388 (2009) 1791-1803.

[229] D.J. Duffy, Finite difference methods in financial engineering: A partial differential equation approach, Wiley, West Sussex, 2006.

[230] W.L. Briggs, A multigrid tutorial, SIAM, Philadelphia, 1987.

[231] A. Stuart, A.R. Humphries, Dynamical system and numerical analysis, Cambridge University Press, Cambridge, 1998.

[232] N. Al-Rawahi and G. Tryggvason, *Numerical simulation of dendritic solidification with convection: two-dimensional geometry*, Journal of Computational Physics, **180** (2002), 471–496.

[233] W.L. Briggs, *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[234] W.K. Burton, N. Cabrera, and F.C. Frank, *The growth of crystals and the equilibrium structure of their surfaces*, Philosophical Transactions of the Royal Society A, **243** (1951), 299–358.

[235] G. Caginalp, *Stefan and Hele-Shaw type models as asymptotic limits of the phase-field equations*, Physical Review A, **39** (1989), 5887–5896.

[236] C.C. Chen and C.W. Lan, *Efficient adaptive three-dimensional phase-field simulation of dendritic crystal growth from various supercoolings using rescaling*, Journal of Crystal Growth, **311** (2009), 702–706.

[237] S. Chen, B. Merriman, S. Osher, and P. Smereka, *A simple level set method for solving Stefan problem*, Journal of Computational Physics, **135** (1997), 8–29.

[238] 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*, International Journal of Heat and Mass Transfer, **52** (2009), 1158-1166.

[239] J.-M. Debierre, A. Karma, F. Celestini, and R. Guérin, *Phase-field approach for faceted solidification*, Physical Review E, **68** (2003), 041604.

[240] F.C. Frank, *Metal Surfaces*, ASM, Cleveland, OH, 1963.

[241] F. Gibou, R. Fedkiw, R. Caflisch, and S. Osher, *A level set approach for the numerical simulation of dendritic growth*, Journal of Scientific Computing, **19** (2002) 183–199.

[242] J.-H. Jeong, N. Goldenfeld, and J.A. Dantzig, *Phase field model for three-dimensional dendritic growth with fluid flow*, Physical Review E, **64** (2001), 041602.

[243] D. Juric and G. Tryggvason, *A front-tracking method for dendritic solidification*, Journal of Computational Physics, **123** (1996), 127–148.

[244] A. Jacot and M. Rappaz, *A pseudo-front tracking technique for the modelling of solidification microstructures in multi-component alloys*, Acta Materialia, **50** (2002), 1909–1926.

[245] A. Karma, Y.H. Lee, and M. Plapp, *Three-dimensional dendrite-tip morphology at low undercooling*, Physical Review E, **61** (2000) 3996–4006.

[246] A. Karma and W.-J. Rappel, *Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics*, Physical Review E, **53** (1996), 3017–3020.

[247] A. Karma and W.-J. Rappel, *Quantitative phase-field modeling of dendritic growth in two and three dimensions*, Physical Review E, **57** (1998), 4323–4349.

[248] Y.-T. Kim, N. Goldenfeld, and J. Dantzig, *Computation of dendritic microstructures using a level set method*, Physical Review E, **62** (2000) 2471-2474.

[249] R. Kobayashi, *Modeling and numerical simulations of dendritic crystal growth*, Physica D, **63** (1993), 410–423.

[250] J.S. Langer. *Directions in Condensed Matter*, World Scientific, Singapore, 1986, 164–186.

[251] X. Li, J. Glimm, X. Jiao, C. Peyser, and Y. Zhao, *Study of crystal growth and solute precipitation through front tracking method*, Acta Mathematica Scientia, **30** (2010), 377–390.

[252] Y. Li, H.G. Lee, and J.S. Kim, *A fast, robust, and accurate operator splitting method for phase-field simulations of crystal growth*, Journal of Crystal Growth, **321** (2011), 176–182.

[253] Y. Li, H.G. Lee, D. Jeong, and J.S. Kim, *An unconditionally stable hybrid numerical method for solving the Allen-Cahn equation*, Computers and Mathematics with Applications, **60** (2010), 1591–1606.

[254] K.G. Libberecht, *The physics of snow crystal*, Reports on Progress in Physics, **68** (2005), 855–895.

[255] D. Li, R. Li, and P. Zhang, *A cellular automaton technique for modelling of a binary dendritic growth with convection*, Applied Mathematical Modelling, **31** (2007), 971–982.

[256] S. Li, J.S. Lowengrub, P.H. Leo, and V. Cristini, *Nonlinear stability analysis of self-similar crystal growth: control of the Mullins–Sekerka instability*, Journal of Crystal Growth, **277** (2005), 578–592.

[257] D.I. Meiron, *Boundary integral formulation of the two-dimensional symmetric model of dendritic growth*, Physica D, **23** (1986), 329–339.

[258] N. Provatas, N. Goldenfeld, and J. Dantzig, *Efficient computation of dendritic microstructures using adaptive mesh refinement*, Physical Review Letters, **80** (1998), 3308–3311.

[259] N. Provatas, N. Goldenfeld, and J. Dantzig, *Adaptive mesh refinement computation of solidification microstructures using dynamic data structures*, Journal of Computational Physics, **148** (1999), 265–290.

[260] M. Plapp and A. Karma, *Multiscale finite-difference-diffusion-Monte-Carlo method for simulating dendritic solidification*, Journal of Computational Physics, **165** (2000), 592–619.

[261] J.C. Ramirez, C. Beckermann, A. Karma, and H.-J. Diepers, *Phase-field modeling of binary alloy solidification with coupled heat and solute diffusion*, Physical Review E, **69** (2004), 051607.

[262] J. Rosam, P.K. Jimack, A. Mullis, *A fully implicit, fully adaptive time and space discretisation method for phase-field simulation of binary alloy solidification*, Journal of Computational Physics, **225** (2007), 1271–1287.

[263] T.P. Schulze, *Simulation of dendritic growth into an undercooled melt using kinetic Monte Carlo techniques*, Physical Review E, **78** (2008), 020601.

[264] J.A. Sethian and J. Straint, *Crystal growth and dendritic solidification*, Journal of Computational Physics, **98** (1992), 231–253.

[265] C.J. Shih, M.H. Lee, and C.W. Lan, *A simple approach toward quantitative phase field simulation for dilute-alloy solidification*, Journal of Crystal Growth, **282** (2005), 515–524.

[266] J. Strain, *A boundary integral approach to unstable solidification*, Journal of Computational Physics, **85** (1989), 342–389.

[267] X. Tong, C. Beckermann, A. Karma, and Q. Li, *Phase-field simulations of dendritic crystal growth in a forced flow*, Physical Review E, **63** (2001), 061601.

[268] J.A. Warren and W.J. Boettinger, *Prediction of dendritic growth and microsegregation patterns in a binary alloy using the phase-field method*, Acta Metallurgica et Materialia, **43** (1995), 689–703.

[269] S.-L. Wang and R.F. Sekerka, *Algorithms for phase field computation of the dendritic operating state at large supercoolings*, Journal of Computational Physics, **127** (1996), 110–117.

[270] K. Wang, A. Chang, L.V. Kale, and J.A. Dantzig, *Parallelization of a level set method for simulating dendritic growth*, Journal of Parallel and Distributed Computing, **66**, (2006), 1379–1386.

[271] G. Wulff, *Zur frage der geschwindigkeit des wachsturms under auflosung der kristallflachen*, Z Kristallogr, **34** (1901), 449–530.

[272] Y. Xu, J.M. McDonough and K.A. Tagavi, *A numerical procedure for solving 2D phase-field model problems*, Journal of Computational Physics, **218** (2006), 770–793.

[273] H. Yin and S.D. Felicelli, *A cellular automaton model for dendrite growth in magnesium alloy AZ91, Modelling Simul*, Materials Science and Engineering, **17** (2009), 075011.

[274] M.F. Zhu and C.P. Hong, *A modified cellular automaton model for the simulation of dendritic growth in solidification of alloys*, ISIJ International, **41** (2001), 436-445.

[275] M.F. Zhu, S.Y. Lee, and C.P. Hong, *Modified cellular automaton model for the prediction of dendritic growth with melt convection*, Physical Review E, **69** (2004), 061610.

[276] M.F. Zhu, S.Y. Pan, D.K. Sun, and H.L. Zhao, *Numerical simulation of microstructure evolution during alloy solidification by using cellular automaton method*, ISIJ International, **50** (2010), 1851–1858.

[277] Bertozzi AL, Münch A, Fanton X, Cazabat AM. Contact line stability and "Undercompressive shocks" in driven thin film flow. *Physical Review Letters* 1998; **81**:5169–5172.

[278] Bertozzi AL, Münch A, Shearer M. Undercompressive shocks in thin film flows. *Physica D* 1999; **134**:431–464.

[279] Levy R. Partial differential equations of thin liquid films: analysis and numerical simulation. *Ph.D. thesis*, North Carolina State University, Raleigh, NC, 2005.

[280] Lipschutz M. *Differential Geometry.* McGraw-Hill: New York, 1969.

[281] Guyon E, Hulin J-P, Petit L, Mitescu CD. *Physical Hydrodynamics.* Oxford University Press: New York, 2001.

[282] Levy R, Shearer M. Kinetics and nucleation for driven thin film flow. *Physica D* 2005; **209**:145–163.

[283] Furihata D. A stable and conservative finite difference scheme for the Cahn-Hilliard equation. Numer Math 2001;87:675-99.

[284] Furihata D, Onda T, Mori M. A finite difference scheme for the Cahn-Hilliard equation based on a Lyapunov functional. GAKUTO Int Ser Math Sci Appl 1993;2:347-58.

[285] Dehghan M. Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices. Math Comput Simulation 2006;71:16-30.

[286] Choi J-W, Lee HG, Jeong D, Kim J. An unconditionally gradient stable numerical method for solving the Allen-Cahn equation. Phys A 2009;388:1791-803.