2006

TWO-DIMENSIONAL SIMULATION OF SOLIDIFICATION IN FLOW FIELD USING PHASE-FIELD MODEL|MULTISCALE METHOD IMPLEMENTATION

Ying Xu
University of Kentucky, yxu2@uky.edu

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation
https://uknowledge.uky.edu/gradschool_diss/389

This Dissertation is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Doctoral Dissertations by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.
ABSTRACT OF DISSERTATION

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering in the College of Engineering at the University of Kentucky

By

Ying Xu

Lexington, Kentucky

Co-Directors: Dr. James M. McDonough, Professor of Mechanical Engineering and Mathematics
and Dr. Kaveh A. Tagavi, Professor of Mechanical Engineering

Lexington, Kentucky

Copyright © Ying Xu 2006
ABSTRACT OF DISSERTATION

TWO-DIMENSIONAL SIMULATION OF SOLIDIFICATION IN FLOW FIELD USING PHASE-FIELD MODEL—MULTISCALE METHOD IMPLEMENTATION

Numerous eorts have contributed to the study of phase-change problems for over a century—both analytical and numerical. Among those numerical approximations applied to solve phase-transition problems, phase-field models attract more and more attention because they not only capture two important effects, surface tension and supercooling, but also enable explicitly labeling the solid and liquid phases and the position of the interface. In the research of this dissertation, a phase-field model has been employed to simulate 2-D dendrite growth of pure nickel without a flow, and 2-D ice crystal growth in a high-Reynolds-number lid-driven-cavity flow. In order to obtain the details of ice crystal structures as well as the flow field behavior during freezing for the latter simulation, it is necessary to solve the phase-field model without convection and the equations of motion on two different scales. To accomplish this, a heterogeneous multiscale method is implemented for the phase-field model with convection such that the phase-field model is simulated on a microscopic scale and the equations of motion are solved on a macroscopic scale. Simulations of 2-D dendrite growth of pure nickel provide the validation of the phase-field model and the study of dendrite growth under different conditions, e.g., degree of supercooling, interface thickness, kinetic coefficient, and shape of the initial seed. In addition, simulations of freezing in a lid-driven-cavity flow indicate that the flow field has great effect on the small-scale dendrite structure and the flow field behavior on the large scale is altered by freezing inside it.

KEYWORDS: Solidification, Phase-Field Model, Multiscale Method, Lid-Driven-Cavity Flow, High Reynolds Number

Ying Xu

December 15, 2005
TWO-DIMENSIONAL SIMULATION OF SOLIDIFICATION IN FLOW FIELD USING PHASE-FIELD MODEL—MULTISCALE METHOD IMPLEMENTATION

By

Ying Xu

James M. McDonough, Ph.D.
Co-Director of Dissertation

Kaveh A. Tagavi, Ph.D.
Co-Director of Dissertation

L. S. Stephens, Ph.D.
Director of Graduate Studies

December 15, 2006
RULES FOR THE USE OF DISSERTATION

Unpublished dissertations submitted for the Doctor’s degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the dissertation in whole or in part also requires the consent of the Dean of Graduate School of the University of Kentucky.

A library that borrows this dissertation for use by its patrons is expected to secure the signature of each user.
DISSERTATION

Ying Xu

The Graduate School
University of Kentucky
2006
TWO-DIMENSIONAL SIMULATION OF SOLIDIFICATION IN FLOW FIELD USING PHASE-FIELD MODEL—MULTISCALE METHOD IMPLEMENTATION

DISsertation

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering in the College of Engineering at the University of Kentucky

By

Ying Xu

Lexington, Kentucky

Co-Directors: Dr. James M. McDonough, Professor of Mechanical Engineering and Mathematics
and Dr. Kaveh A. Tagavi, Professor of Mechanical Engineering

Lexington, Kentucky

Copyright © Ying Xu 2006
This dissertation is dedicated to my father and my husband for their love and support, and to the memory of my mother.
Acknowledgments

This dissertation is the results of six years of work and is by far the most significant scientific accomplishment in my life. Fortunately, I am not alone in this long journey to obtaining my degree. There are so many people who gave me support and immense help to complete the dissertation. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

With a deep sense of gratitude, I wish to express my sincere thanks to both of my directors, Dr. James M. McDonough and Dr. Kaveh A. Tagavi, for their stimulating suggestions and encouragement in all the time of research. I would like to thank all committee members for all their help, interest and valuable hints. I also want to thank Dr. Tianliang Yang for his assistance on programming.

This research has been supported and funded by various organizations including NASA/EPSCoR, Center for Computational Science of the University of Kentucky. I also wish to express my gratitude to the University of Kentucky Computing Center for use of their HP N-4000 SMP and HP SuperDome for all computations.
# Table of Contents

Acknowledgments iii  
List of Figures vii  
List of Tables xiii  

**Chapter 1 INTRODUCTION**  
1.1 Importance of Solidification ................................................. 1  
1.2 Models for Solidification and Analytical Solutions .................... 2  
   1.2.1 Classical Stefan Problem ................................................ 4  
   1.2.2 Modified Stefan Problems ............................................... 8  
1.3 Numerical Methods to Solve Solidification Problems .................... 9  
   1.3.1 Front-Tracking Methods .................................................. 10  
   1.3.2 Enthalpy Methods .......................................................... 12  
   1.3.3 Phase-Field Models ....................................................... 13  
1.4 Summary of Current State of the Art ...................................... 15  
   1.4.1 What has been Done ....................................................... 15  
   1.4.2 Important Problems that cannot be Solved ............................ 21  
   1.4.3 How this Research will Help ............................................. 23  
1.5 Brief Outline of Dissertation .............................................. 23  

**Chapter 2 PHYSICS OF SOLIDIFICATION**  
2.1 Conditions for Onset of Nucleation ...................................... 25  
2.2 Dendrite Growth Rates ...................................................... 28  
2.3 Kinetic Coefficient ............................................................ 29  

**Chapter 3 NUMERICAL ANALYSES OF PHASE-FIELD MODEL**  
3.1 Derivation of Phase-Field Model in Flow Field .......................... 33  
3.2 2-D Phase-Field Model in Flow Field ..................................... 41  
3.3 Initial and Boundary Conditions ......................................... 42  
3.4 Numerical Methods and Algorithms ...................................... 43


5.4.4 Effects of Initial Shape ........................................... 110

Chapter 6 SIMULATION OF 2-D DENDRITE GROWTH IN LID-DRIVEN-
CAVITY FLOW .......................................................... 112

6.1 Lid-Driven-Cavity Flow ............................................ 112

6.1.1 Comparisons between Computed Results and Published DNS Results 
at Low Re ............................................................ 113

6.1.2 Implementation of Shuman Filters ................................ 115

6.1.3 Solutions at High Re ............................................. 117

6.1.4 Effects of Filter Parameter $\beta$ ................................ 120

6.2 Dendrite Growth in Lid-Driven-Cavity Flow ...................... 120

6.2.1 Choice of Physical Parameters ................................. 121

6.2.2 Validation of Time-Independent Velocity and Pressure Assumption . 122

6.2.3 Lid-Driven-Cavity Flow with Freezing from Two Discrete Points Near 
Bottom ........................................................................ 124

6.2.4 Lid-Driven-Cavity Flow with Freezing of Three Successive Points Near 
Bottom ........................................................................ 130

6.2.5 Freezing at Different Positions of Lid-Driven-Cavity Flow ...... 136

6.3 Parallel Performance .................................................... 140

Chapter 7 SUMMARY AND CONCLUSIONS ..................... 142

7.1 Summary ................................................................. 142

7.2 Conclusions ............................................................. 146

7.3 Future Work ............................................................. 149

BIBLIOGRAPHY .............................................................. 151

VITA ............................................................................. 158
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Sketch of fixed finite-difference grid.</td>
<td>11</td>
</tr>
<tr>
<td>1.2</td>
<td>Sketch of enthalpy function.</td>
<td>13</td>
</tr>
<tr>
<td>1.3</td>
<td>Sketch of phase field profile.</td>
<td>14</td>
</tr>
<tr>
<td>1.4</td>
<td>Sketch of a line in phase-field profile.</td>
<td>15</td>
</tr>
<tr>
<td>2.1</td>
<td>Free energy of a nucleus as function of radius.</td>
<td>26</td>
</tr>
<tr>
<td>2.2</td>
<td>Step, kink and adsorbed molecule on a crystal surface.</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>Sketch of staggered grid for $u$, $v$ and $p$.</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Sketch of staggered grid for $\phi$ and $T$.</td>
<td>53</td>
</tr>
<tr>
<td>3.3</td>
<td>Fork-join model for OpenMP Fortran API.</td>
<td>55</td>
</tr>
<tr>
<td>3.4</td>
<td>MPI execution model.</td>
<td>56</td>
</tr>
<tr>
<td>3.5</td>
<td>Distribution of processors for two-level Douglas &amp; Gunn time-splitting.</td>
<td>57</td>
</tr>
<tr>
<td>3.6</td>
<td>Example of transferring data block between processors for Douglas &amp; Gunn time-splitting method.</td>
<td>57</td>
</tr>
<tr>
<td>3.7</td>
<td>Example of transferring data between adjacent processors for Douglas &amp; Gunn time-splitting method.</td>
<td>58</td>
</tr>
<tr>
<td>4.1</td>
<td>Sketch of multiscale method.</td>
<td>66</td>
</tr>
<tr>
<td>5.1</td>
<td>Semilog plot of $\epsilon$ versus $\Delta x$ and $\Delta y$.</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>Total energy error as function of spatial step size $\Delta x$ ($\Delta t_y$) at $t = 0.2$.</td>
<td>78</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison of $L^\infty$ residual with number of iterations for sequential and coupled methods with $\Delta x = \Delta y = 0.01$, $\Delta t = 5 \times 10^{-5}$ and $St = 0.5$ at three different times: (a) $t = 0.05$; (b) $t = 0.1$; (c) $t = 0.2$.</td>
<td>79</td>
</tr>
<tr>
<td>5.4</td>
<td>Dendrite surface locations for four different combinations of space and time step sizes with $St = 0.5$ at $t = 0.05$: (a) $\Delta x = \Delta y = 0.01$, $\Delta t = 5 \times 10^{-5}$; (b) $\Delta x = \Delta y = 0.005$, $\Delta t = 2.5 \times 10^{-5}$; (c) $\Delta x = \Delta y = 0.0025$, $\Delta t = 1.25 \times 10^{-5}$; (d) $\Delta x = \Delta y = 0.00125$, $\Delta t = 6.25 \times 10^{-6}$.</td>
<td>81</td>
</tr>
</tbody>
</table>
5.5 Dendrite structures for $St = 0.5$ at time $t = 0.05$, $t = 0.1$ and $t = 0.2$
computed with grid spacing $\Delta x = \Delta y = 0.01$, and $\Delta t = 5 \times 10^{-5}$. 82
5.6 Temperature fields at (a) $t = 0.05$, (b) $t = 0.1$ and (c) $t = 0.2$. 82
5.7 Interface-point computation sketch. 83
5.8 Sketch of interface point on one cell boundary: (a) one interface point; (b) no
interface point; (c) two interface points for finer grid. 87
5.9 Interface shapes at $t = 0.1$ for three different Stefan numbers: (a) $St = 0.5$
dendritic), (b) $St = 1.0$ (scalloped) and (c) $St = 1.4$ (quasi-planar). 88
5.10 Computed interfacial speed as function of time for different Stefan numbers:
(a) $St = 0.1$; (b) $St = 0.5$; (c) $St = 1.0$; (d) $St = 1.4$. 89
5.11 Computed interfacial speed versus Stefan number at $t = 0.1$. 90
5.12 Computed interfacial speed for different interfacial thickness parameter $\epsilon$ at
$St = 0.5$. 91
5.13 Interface structures for different interfacial thickness parameter $\epsilon$ at $St = 0.5$:
(a) $\epsilon = 0.002$; (b) $\epsilon = 0.003$; (c) $\epsilon = 0.004$; (d) $\epsilon = 0.005$. 92
5.14 Computed interfacial speed for different kinetic coefficient at $St = 0.5$: (a)
$m_k = 0.025$; (b) $m_k = 0.05$; (c) $m_k = 0.075$. 93
5.15 Dendrite structures for different kinetic coefficient at $St = 0.5$: (a) $m_k =$
0.025; (b) $m_k = 0.05$; (c) $m_k = 0.075$. 93
5.16 Interfacial propagation speed versus Stefan number: comparison between com-
puted results from the phase-field model, solutions obtained from Brener’s
theory in 2D and 3D, and experimental data. 95
5.17 Approximation to round initial shape. 96
5.18 Dendrite structures for approximately round initial seed at time $t = 0.05$,
t = 0.1 and $t = 0.2$ (outward direction) with grid size $\Delta x = \Delta y = 0.01$ and
$\Delta t = 5 \times 10^{-5}$ for $St = 0.5$. 97
5.19 Solid phase area versus time for nickel, $\Delta T = 3K$: (a) Initial area $A_0 =$
$2.1412 \times 10^{-14} m^2$; (b) Initial area $A_0 = 2.7952 \times 10^{-14} m^2$. 99
5.20 Solid phase area versus time for nickel, $\Delta T = 6K$: (a) Initial area $A_0 =$
$5.2312 \times 10^{-15} m^2$; (b) Initial area $A_0 = 6.8904 \times 10^{-15} m^2$. 100
5.21 Solid phase area versus time for nickel, \(\Delta T = 10K\): (a) Initial area \(A_0 = 1.7489 \times 10^{-15} m^2\); (b) Initial area \(A_0 = 2.7411 \times 10^{-15} m^2\).

5.22 Comparison of analytical and computational critical areas at different \(\Delta T\) for nickel: (a) \(\Delta c_p = 0\), (b) \(\Delta c_p \neq 0\).

5.23 Solid phase area versus time for water, \(\Delta T = 3K\): (a) Initial area \(A_0 = 1.2933 \times 10^{-15} m^2\); (b) Initial area \(A_0 = 1.5465 \times 10^{-15} m^2\).

5.24 Solid phase area versus time for water, \(\Delta T = 6K\): (a) Initial area \(A_0 = 3.1550 \times 10^{-16} m^2\); (b) Initial area \(A_0 = 3.7332 \times 10^{-16} m^2\).

5.25 Solid phase area versus time for water, \(\Delta T = 10K\): (a) Initial area \(A_0 = 1.1150 \times 10^{-16} m^2\); (b) Initial area \(A_0 = 1.4780 \times 10^{-16} m^2\).

5.26 Comparison of analytical and computational critical areas at different \(\Delta T\) for water: (a) \(\Delta c_p = 0\), (b) \(\Delta c_p\) constant, (c) \(\Delta c_p\) a function of temperature.

5.27 Demonstration of diamond initial shape.

5.28 Dendrite structure contours for different degrees of anisotropy with grid sizes \(\Delta x = \Delta y = 3.15 \times 10^{-8} m\) and \(\Delta t = 10^{-12} s\) for Stefan number \(St = 0.5\): (a) \(\epsilon' = 0.01\); (b) \(\epsilon' = 0.02\); (c) \(\epsilon' = 0.03\); (d) \(\epsilon' = 0.04\); (e) \(\epsilon' = 0.05\).

5.29 Interface propagation speed for different degree of anisotropy.

5.30 Dendrite structure contours with grid sizes \(\Delta x = \Delta y = 3.15 \times 10^{-8} m\) and \(\Delta t = 10^{-12} s\) for Stefan number \(St = 0.5\), degree of anisotropy \(\epsilon' = 0.03\) and interfacial thickness \(\delta = 1.26 \times 10^{-8}\).

5.31 Comparison of dendrite structures for different interfacial thickness with grid sizes \(\Delta x = \Delta y = 3.15 \times 10^{-8} m\) and \(\Delta t = 10^{-12} s\) for Stefan number \(St = 0.5\) and \(\epsilon' = 0.03\): (a) \(\delta = 1.05 \times 10^{-8}\); (b) \(\delta = 1.26 \times 10^{-8}\).

5.32 Dendrite structure contours for approximately round initial seed with grid sizes \(\Delta x = \Delta y = 3.15 \times 10^{-8} m\) and \(\Delta t = 10^{-12} s\) for Stefan number \(St = 0.5\), \(\epsilon' = 0.03\) and \(\delta = 1.05 \times 10^{-8}\).

5.33 Interface propagation speed for different initial shape: diamond and round.

6.1 Computed velocity profiles for \(Re = 100\) compared with results of Ghia et al.: (a) \(u\)-velocity component at \(x = 0.05 m\); (b) \(v\)-velocity component at \(y = 0.05 m\).
6.2 Computed velocity profiles for $Re = 400$ compared with results of Ghia et al.:
(a) $u$-velocity component at $x = 0.05m$; (b) $v$-velocity component at $y = 0.05m$.

6.3 Streamlines of lid-driven-cavity flow at steady state with grid sizes $\Delta x = \Delta y = 5 \times 10^{-4}m$ and $\Delta t = 5 \times 10^{-4}s$. Left: $Re = 100$; Right: $Re = 400$.

6.4 Nine-point stencil to create higher-order filter: $\square$: grid points used in the second-order Shuman filter; $\bullet$: grid points used in the fourth-order Shuman filter.

6.5 Computed velocity profiles for $Re = 5000$ compared with results of Ghia et al.:
(a) $u$-velocity component at $x = 0.05m$; (b) $v$-velocity component at $y = 0.05m$.

6.6 Streamlines of lid-driven-cavity flow at steady state with grid sizes $\Delta x = \Delta y = 5 \times 10^{-4}m$ and $\Delta t = 5 \times 10^{-4}s$ at $Re = 5000$.

6.7 Lid-driven-cavity flow at $Re = 5000$ with $\Delta x = \Delta y = 5 \times 10^{-4}m$ and $\Delta t = 5 \times 10^{-3}s$. (a): Phase portrait of $u$ versus $v$; (b): Power spectral density of $u$ velocity.

6.8 Computed velocity profiles at $Re = 5000$ for different filters (fourth-order filter with $\beta = 2000$, fourth-order and second-order filters with $\beta = 200$) compared with data by Ghia et al.: (a) $u$-velocity component at $x = 0.05m$; (b) $v$-velocity component at $y = 0.05m$.

6.9 Velocity vector fields and streamlines for flow across dendrite on small scale:
(a1) velocity vector field using Eq. (4.2); (a2) streamlines using Eq. (4.2); (b1) velocity vector field using Eq. (4.4); (b2) streamlines using Eq. (4.4).

6.10 Position of freezing in large domain from two points near the bottom.

6.11 Velocity vector field and dendrite shape of small-scale computation in: (a) cell $(100,2)$ and (b) cell $(102,2)$ as shown in Figure 6.10 on large-scale domain at $t = 4 \times 10^{-5}s$.

6.12 Velocity vector field and dendrite shape of small-scale computation in: (a) cell $(100,2)$ and (b) cell $(102,2)$ as shown in Figure 6.10 on large-scale domain at $t = 8 \times 10^{-5}s$. 
6.13 Comparison of dendrite shapes for three cases: in grid cell (100, 2), in grid cell (102, 2) and dendrite growth without flow field at (a) $t = 4 \times 10^{-5}$ and (b) $t = 8 \times 10^{-5}$. ................................. 127
6.14 Comparison of solid phase area for three cases: dendrites grow in grid cells (100, 2) and 102, 2) of large scale, and dendrite grows without flow at $t = 8 \times 10^{-5}$. ................................. 128
6.15 Streamlines of lid-driven-cavity flow with freezing from two discrete points on bottom center on small scale at: Left $t = 4 \times 10^{-5}$, Right $t = 8 \times 10^{-5}$. ................................. 128
6.16 Implementation of function $86.5t^{1.34} + 9.0 \times 10^{-4}$ to fit the computed solid area. 129
6.17 Streamlines of lid-driven-cavity flow at $Re = 5000$: Left, with freezing from two discrete points at bottom center on small scale; Right, without freezing. 131
6.18 Position of freezing in large domain from three successive points near the bottom. ................................................................. 131
6.19 Velocity vector field and dendrite shape of small-scale computation in: (a) cell (100, 2), (b) cell (101, 2) and (c) cell (102, 2) as shown in Figure 6.18 on large-scale domain at $t = 4 \times 10^{-5}$. ................................. 132
6.20 Velocity vector field and dendrite shape of small-scale computation in: (a) cell (100, 2), (b) cell (101, 2) and (c) cell (102, 2) as shown in Figure 6.18 on large-scale domain at $t = 8 \times 10^{-5}$. ................................. 133
6.21 Streamlines of lid-driven-cavity flow with freezing from three successive points at bottom center on small scale at: Left $t = 4 \times 10^{-5}$, Right $t = 8 \times 10^{-5}$. 134
6.22 Streamlines of lid-driven-cavity flow at $Re = 5000$: Left, with freezing from three successive points at bottom center on small scale; Right, without freezing. 135
6.23 Comparison of velocity vector fields at $Re = 5000$ for: Left: freezing from two discrete points (blue vector), three successive points (red vector) around $t = 0.05$; Right: without freezing. ................................. 135
6.24 Locations of grid cells (191, 161), (101, 191), (31, 31) and (161, 31) on large-scale domain. ................................. 137
6.25 Velocity vector field and dendrite shape of small-scale computation in grid cells: (a) (191, 161), (b) (101, 191), (c) (31, 31) and (d) (161, 31) on large-scale domain at $t = 4 \times 10^{-5}$ s. ................................................................. 138

6.26 Comparison of solid phase area for dendrite growth in grid cells (191, 161), (101, 191), (31, 31) and (161, 31) on large scale, and dendrite growth without flow at $t = 4 \times 10^{-5}$ s. ................................................................. 139

6.27 Solid phase area of dendrite variation with velocity magnitude in large-scale lid-driven-cavity flow. ................................................................. 139

6.28 Speedup of parallelized phase-field model with convection. .............. 141
List of Tables

1.1 Summary of physical factors in phase changes; assumptions for modified Stefan problems and phase-field model ........................................... 3

5.1 Thermo-physical parameters of Nickel and Water ................................. 75

5.2 Grid function convergence tests ....................................................... 80

5.3 Analytical critical area of nickel ..................................................... 98

5.4 Analytical critical area of water ..................................................... 102
Chapter 1
INTRODUCTION

In this chapter, we first state the importance of solidification phenomena in several different areas. We then introduce mathematical models—Classical and Modified Stefan Problems—to formulate phase-change processes, followed by analytical solutions of some prototype problems. Since analytical solutions are obtainable only for simple problems in 1D, however, most realistic solidification problems do not fall into this category. Therefore, numerical methods are required to model the mathematical problems. We will introduce three numerical methods: front-tracking methods, enthalpy methods, and phase-field models, which have played important roles during the history of simulating phase-transformation problems. Finally, we will summarize the current state of the art, i.e., what has been done, what needs to be done, and what cannot be done. A brief outline of the dissertation will be provided at the end of this chapter.

1.1 Importance of Solidification

Solidification phenomena play important roles in many processes ranging from production engineering to solid-state physics, and including hydrodynamics. One of their major practical applications, casting, is a very economical method of forming a component if the melting point of the metal is not too high. Some important processes which involve solidification are casting, welding, soldering/brazing, rapid solidification processing, and directional solidification. Good control of solidification processes at the outset is of utmost importance. If the properties of casting were easier to control, then solidification would be an even more important process. In this respect, solidification theory plays a vital role since it forms the basis for influencing the microstructure, and hence, improving the quality of cast products.

In addition, crystallization of certain pure substances is of great importance. For instance, preparation of semiconductor-grade silicon crystals is an essential step in modern solid-state physics and technology. Indeed, requirements of semiconductor physics have enormously influenced solidification theory and practices. Therefore, during the past forty years, solidification has evolved from a purely technological, empirical field, to a science.

Another important issue of concern is ice formation in rivers and oceans in winter. One
important ice form in rivers and oceans is called frazil ice as described by Martin [1], which, by definition, consists of small discs of ice measuring $1 - 4 mm$ in diameter and $1 - 100 \mu m$ in thickness that form in turbulent, slightly supercooled water. Once they form, the discs rapidly aggregate and adhere to foreign material in water, and their production rate can be as large as $10^6 m^3$ per day. Therefore, these crystals cause serious problems for hydroelectric facilities, including the reduction of available head by 25\%, the blocking of turbine intakes, the blockage of hydroelectric reservoirs, and the freezing opens of gates. Ocean frazil ice has become a source of economic concern only in the last decade with the onset of oil and gas development in the Arctic.

1.2 Models for Solidification and Analytical Solutions

In this section, we introduce two important models simulating solidification problems: the classical Stefan problem and a modified Stefan problem. We will also display some analytical solutions obtained from the models for simple solidification problems.

Solidification phenomena can be incorporated into the phase-change group in which a substance has a transformation point at which it changes from one phase to another with emission or absorption of heat. The mathematical model of a simple melting or freezing process incorporating only the most basic phenomena was formulated by Stefan [2] to determine the temperature distribution of solid and liquid phases and the position of the boundary between phases when solidifying a material. Later this model became known as the classical Stefan problem and constituted the foundation of more complex models that incorporate some of the effects initially left out. Overview of the origins of such problems, referred to as “moving boundary problems” or “free boundary problems,” can be found in works by Rubinstein [3] and Carslaw and Jaeger [4].

Before introduction of the classical and modified Stefan problems, we list the physical factors involved in a phase-change process and the simplifying assumptions made in modified Stefan problems and the phase-field model of the current work in Table 1.1. It is crucial to make reasonable assumptions such that the model can at best be as good as its underlying physical assumptions.

We note that the first assumption for modified Stefan problems works well for most com-
Table 1.1  Summary of physical factors in phase changes; assumptions for modified Stefan problems and phase-field model

<table>
<thead>
<tr>
<th>Physical Factors Involved in Phase Change</th>
<th>Assumptions for Modified Stefan Problems</th>
<th>Assumption for Phase-Field Model we used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Heat and mass transfer by conduction, convection, radiation with possible gravitational, elastic, chemical, and electromagnetic effects.</td>
<td>Heat transfer isotropically by conduction only, all other effects assumed negligible.</td>
<td>Heat transfer mainly by conduction, and convection introduced in buoyancy force term.</td>
</tr>
<tr>
<td>2. Release or absorption of latent heat</td>
<td>Latent heat is constant; It is released or absorbed at the phase-change temperature.</td>
<td>Latent heat is constant</td>
</tr>
<tr>
<td>3. Variation of phase-change temperature</td>
<td>Phase-change temperature ( T_m ) is a fixed known temperature, a property of the material.</td>
<td>Phase-change temperature is fixed</td>
</tr>
<tr>
<td>4. Nucleation difficulties, supercooling effects</td>
<td>Supercooling effect considered</td>
<td>Supercooling effect considered</td>
</tr>
<tr>
<td>5. Interface thickness and structure</td>
<td>Assume a sharp interface at the freezing temperature</td>
<td>Assume finite thickness but very thin</td>
</tr>
<tr>
<td>6. Surface tension and curvature effects at the interface</td>
<td>Surface tension and curvature effects considered</td>
<td>Surface tension and curvature effects considered</td>
</tr>
<tr>
<td>7. Variation of thermodynamical properties</td>
<td>Assume constant in each phase (( c_L \neq c_S, k_L \neq k_S )).</td>
<td>Assume constant in each phase (( c_L \neq c_S, k_L = k_S )).</td>
</tr>
<tr>
<td>8. Density changes</td>
<td>Assume constant (( \rho_L = \rho_S ))</td>
<td>Assume constant in each phase (( \rho_L \neq \rho_S ))</td>
</tr>
</tbody>
</table>
mon cases, and it is especially reasonable for pure substances and moderate temperature gradients. But in our phase-field model, since we will consider unequal densities, convective effects will be introduced in a buoyancy force term using Boussinesq approximation. The assumption for the latent heat is also reasonable, since although the latent heat varies with temperature, the amount of variation is usually negligible. Furthermore, since the phase-change temperature is assumed to be fixed, the latent heat is obtained at the phase-change temperature. It is assumed in modified Stefan problems that a sharp interface exists, separating the solid and liquid phases, which is reasonable for many pure materials without internal heating. However, we assume that the interface is thin with finite thickness in the phase-field model. This assumption removes the “stiffness” of modified Stefan problems. To make this assumption, we introduce a high-degree polynomial to represent the interface (see details in Chapter 3). Thermophysical properties such as heat conductivity $k$ and heat capacity $c$ are assumed to be constants in each phase for simplicity in both modified Stefan problems and the phase-field model. This is an assumption of convenience only, but it is reasonable for most materials under moderate ranges of temperature variation. This assumption can be relaxed by expressing $k$ and $c$ as functions of temperature, location or time. In modified Stefan problems, density is a constant, while in the phase-field model density is constant in each phase; but $\rho_L \neq \rho_S$. The purpose of introducing unequal densities is to make this assumption more reasonable without adding complexity to the computation. Movement of solid material caused by density differences will not be considered here; therefore, drag and lift forces will not be applied to the solid body. Nucleation difficulties (supercooling effects), surface tension, and curvature effects at the interface are incorporated in both modified Stefan problems and the phase-field model. It should be mentioned that these assumptions comprise the only difference between the classical Stefan problem and modified Stefan problems. As a result of relaxing these assumptions, we obtain $T = T_m$ at the interface in the classical Stefan problem.

1.2.1 Classical Stefan Problem

In this subsection, we state a mathematical problem modeling a phase-change process that satisfies the classical Stefan problem assumptions listed before. The problem is considered
in a region $\Omega \subset \mathbb{R}^d$, $d = 1, 2$ and $t \in \mathbb{R}_+$ with an interface $\Gamma(t) \subset \Omega$, and it is presented in dimensional form:

\begin{align}
T_i &= \alpha \Delta T \quad \text{in} \quad \Omega \setminus \Gamma(t), \quad (1.1a) \\
\rho L_0 v_n &= (k_S \nabla T_S - k_L \nabla T_L) \cdot \mathbf{n} \quad \text{on} \quad \Gamma(t), \quad (1.1b) \\
T &= T_m \quad \text{on} \quad \Gamma(t). \quad (1.1c)
\end{align}

Here $T$, $\alpha$, $\rho$ and $L_0$ are temperature, thermal diffusivity, density, and latent heat at the equilibrium melting temperature $T_m$, respectively; $k_S$ and $k_L$ are thermal conductivities in solid and liquid, respectively; $\mathbf{n}$ is the unit normal to $\Gamma$ (in the direction from solid to liquid for freezing and from liquid to solid for melting), and $v_n$ is the normal velocity of a point on the interface. Subscripts $S$ and $L$ represent solid and liquid, respectively; $\Delta$ is the Laplacian, and the subscript $t$ denotes partial differentiation with respect to time. Stefan condition (1.1b) reflects the energy balance across the interface. It states that the latent heat released due to the interface displacement equals the net amount of heat flux delivered to (or from) the interface.

Analytical solutions can be obtained for some simple problems in 1D with prescribed boundary and initial conditions as in Alexiades and Solomon [5] and Crank [6]. Here, we will present analytical solutions for two cases: one is a 1-D melting problem of a semi-infinite slab; the other is a 1-D supercooled solidification of a semi-infinite slab.

First we consider the melting of a semi-infinite slab, $0 \leq x < \infty$, which is initially solid at a uniform temperature $T_S < T_m$. At time $t = 0$, a temperature $T_L > T_m$ is imposed on the face $x = 0$. Thus, the boundary and initial conditions for this problem are as follows:

\begin{align}
\text{Initial condition} \\
T(x, 0) &= T_S < T_m, \quad \Gamma(0) = 0, \quad x > 0, \quad (1.2a) \\
\text{Boundary conditions} \\
T(0, t) &= T_L > T_m, \quad \lim_{x \to \infty} T(x, t) = T_S, \quad t > 0. \quad (1.2b)
\end{align}

As a consequence, we can find a solution in terms of the similarity variable $\xi = x/\sqrt{t}$.  


Therefore, the solution of the classical Stefan problem is

\[ \Gamma(t) = 2\lambda \sqrt{\frac{\alpha_L}{t}}, \quad t > 0, \quad (1.3a) \]

**Interface location**

\[ T(x, t) = T_L - (T_L - T_m) \frac{\text{erf} \left( \frac{\xi}{2\sqrt{\alpha_L}} \right)}{\text{erf}(\lambda)}, \quad (1.3b) \]

**Temperature in liquid**

\[ 0 \leq x \leq \Gamma(t), \quad t > 0: \]

\[ T(x, t) = T_S + (T_m - T_S) \frac{\text{erfc} \left( \frac{\xi}{2\sqrt{\alpha_S}} \right)}{\text{erfc}(\nu \lambda)}. \quad (1.3c) \]

**Temperature in solid**

\[ x > \Gamma(t), \quad t > 0: \]

Here \( \lambda \) is the solution to the transcendental equation

\[ \frac{St_L}{\exp(\lambda^2) \text{erf}(\lambda)} - \frac{St_S}{\nu \exp(\nu^2\lambda^2) \text{erfc}(\nu \lambda)} = \lambda \sqrt{\pi} , \]

with \[ St_L = \frac{c_L (T_L - T_m)}{L_0}, \quad St_S = \frac{c_S (T_m - T_S)}{L_0}, \quad \nu = \sqrt{\frac{\alpha_L}{\alpha_S}}. \]

Now we consider the solidification of a semi-infinite slab, \( 0 \leq x < \infty \). Initially supercooled liquid occupies the domain at a uniform temperature \( T_{\text{init}} < T_m \). At time \( t = 0 \), a temperature \( T_S \leq T_m \) is imposed at \( x = 0 \). Thus, the boundary and initial conditions for this problem are prescribed below:

**Initial condition**

\[ T(x, 0) = T_{\text{init}} < T_m, \quad \Gamma(0) = 0, \quad x > 0, \quad (1.4a) \]

**Boundary conditions**

\[ T(0, t) = T_S \leq T_m, \quad \lim_{x \to \infty} T(x, t) = T_{\text{init}}, \quad t > 0. \quad (1.4b) \]
Its similarity solution is

**Interface location**

\[ \Gamma(t) = 2\lambda \sqrt{\alpha_S t}, \quad t > 0, \]  

(1.5a)

**Temperature in solid** \( 0 \leq x \leq \Gamma(t), \quad t > 0 : \)

\[ T(x, t) = T_S + (T_m - T_S) \frac{\text{erf}(\xi/2\sqrt{\alpha_S})}{\text{erf}(\lambda)}, \]  

(1.5b)

**Temperature in liquid** \( x > \Gamma(t), \quad t > 0 : \)

\[ T(x, t) = T_{\text{init}} + (T_m - T_{\text{init}}) \frac{\text{erfc}(\xi/2\sqrt{\alpha_L})}{\text{erfc}(\nu \lambda)}, \]  

(1.5c)

where \( \lambda \) is a root of the transcendental equation

\[ \frac{St_S}{\exp(\lambda^2 \text{erf}(\lambda))} + \frac{St_{\text{init}}}{\nu \exp(\nu^2 \lambda^2 \text{erfc}(\nu \lambda))} = \lambda \sqrt{\pi}, \]

with

\[ St_S = \frac{c_S (T_m - T_S)}{L_0}, \quad St_{\text{init}} = \frac{c_L (T_m - T_{\text{init}})}{L_0}, \quad \nu = \sqrt{\frac{\alpha_S}{\alpha_L}}. \]

We note that the transcendental equation has a unique positive solution if and only if \( 0 \leq St_{\text{init}} < 1. \) Therefore, the supercooled solidification problem has a unique similarity solution if and only if the liquid is not hypercooled initially, i.e., if

\[ T_m - \frac{L_0}{c_L} < T_{\text{init}} \leq T_m. \]

The physical significance for this condition is very clear. The sensible heat in the amount of \( c_L(T_m - T_{\text{init}}) \) is required to raise the temperature to \( T_m \) for freezing to occur. Therefore, it is necessary (and sufficient) to have \( c_L(T_m - T_{\text{init}}) < L_0, \) since the only source of heat is the latent heat \( L_0. \)

This condition adds an additional constraint to the classical Stefan problem and makes it impossible to solve some realistic problems with large supercooling. Fortunately, this limitation can be removed in modified Stefan problems by incorporating two important effects left out in the classical Stefan problem: nucleation difficulties and curvature effects at the interface. The contribution of these effects is to make the freezing temperature \( T_f \) lower than \( T_m. \) Therefore, although it is necessary (and sufficient) to have \( c_L(T_f - T_{\text{init}}) < L_0, \) there is no limitation on \( T_m - T_{\text{init}}, \) which implies that the Stefan number \( St_{\text{init}} \) can be chosen such that \( 0 \leq St_{\text{init}} < 1, St_{\text{init}} = 1 \) or \( St_{\text{init}} > 1. \)
1.2.2 Modified Stefan Problems

Similar to the formulation of the classical Stefan problem, in the modified Stefan problems, a mathematical formulation for phase-transition problems considered in a region \( \Omega \subset \mathbb{R}^d, d = 1, 2 \) and \( t \in \mathbb{R}^+ \) with an interface \( \Gamma(t) \subset \Omega \) is presented in dimensional form:

\[
T_t = \alpha \Delta T \quad \text{in} \quad \Omega \setminus \Gamma(t), \quad (1.6a)
\]
\[
\rho L_0 v_n = (k_s \nabla T_S - k_L \nabla T_L) \cdot \mathbf{n} \quad \text{on} \quad \Gamma(t), \quad (1.6b)
\]
\[
T = T_m - \gamma \kappa - \frac{v_n}{\mu_k} \quad \text{on} \quad \Gamma(t). \quad (1.6c)
\]

Here \( \gamma = \sigma T_m / \rho L_0 \) is the interfacial energy with \( \sigma \) denoting the surface tension; \( \kappa \) is the sum of principal curvatures at a point on the interface; \( \mu_k \) is the interfacial kinetic coefficient. The above equations are based on the mathematical model described by Davis [7]. Other similar models can be found in Caginalp [8] and Caginalp and Socolovsky [9] in dimensionless form.

In modified Stefan problems, Eq. (1.6c) is called the Gibbs–Thomson condition; it reflects the effects of kinetic supercooling and curved interface on the interface temperature. These effects cause the freezing temperature on the interface to be lower than the melting temperature \( T_m \), which differs by the amount of \(-\gamma \kappa \) (curvature effect) and \(-v_n/\mu_k \) (kinetic effect). Therefore, it is difficult to obtain analytical solutions because of the complexity of the interface.

Here, we will consider a simple problem of 1-D freezing into a supercooled melt. We assume that temperature in the solid is a constant—the interface temperature. Thus, Eqs. (1.6) are reduced to

\[
T_t = \alpha_L T_{xx} \quad \text{in} \quad x > \Gamma(t), \quad (1.7a)
\]
\[
\rho L_0 v_n = -k_L T_x \quad \text{on} \quad \Gamma(t), \quad (1.7b)
\]
\[
T = T_m - \frac{v_n}{\mu_k} \quad \text{on} \quad \Gamma(t). \quad (1.7c)
\]
with boundary and initial conditions:

**Initial condition**

\[ T(x, 0) = T_{init} < T_m, \quad \Gamma(0) = 0 \quad x > 0, \quad (1.8a) \]

**Boundary conditions**

\[ T(x, t) = T(\Gamma(t), t) \quad \text{in} \quad 0 \leq x \leq \Gamma(t), \quad (1.8b) \]

\[ \lim_{x \to \infty} T(x, t) = T_{init} < T_m. \quad (1.8c) \]

The solution of Eqs. (1.7), subject to conditions (1.8) is [7]

\[ T = T_{init} + (T_m - T_{init}) St \exp[-(1 - 1/St)(x - v_n t)], \quad (1.9) \]

with

\[ St = \frac{c_L (T_m - T_{init})}{L_0}, \quad v_n = 1 - \frac{1}{St}. \]

The above equation requires \( v_n = 1 - 1/St > 0 \), i.e., \( St > 1 \). Therefore, the solutions for \( St > 1 \) can be obtained, and the interface propagates at a constant speed.

The modified Stefan problems contain all important effects that should be involved in solidification. Their assumptions listed in Table 1.1 are reasonable, implying a good approximation to real physical problems. Therefore, they constitute the foundation for most current mathematical models of phase-change problems.

### 1.3 Numerical Methods to Solve Solidification Problems

As we have mentioned, explicit solutions for phase-change problems exist only in one-dimensional cases of an infinite or semi-infinite region with simple initial and boundary conditions. They usually take the form of functions of a similarity variable \( \xi = x/\sqrt{t} \) as shown in Eqs. (1.3) and (1.5). Even problems with constant imposed flux boundary conditions do not have exact solutions. Very few explicit solutions are known in cylindrical and spherical geometries, and none for finite domains and higher dimensions. Thus, approximate solutions are necessary to solve more realistic problems. There are two distinct categories of approximations: analytical and numerical.

Analytical approximations include steady-state and quasi-stationary approximations, the Megerlin method, perturbation methods, etc. [5] [6]. The applicability of analytical approximation depends on being able to simplify the problem to the form that fits the method.
Such simplification is achieved by making physical and/or mathematical assumptions on the underlying processes and/or mathematical modeling. As a consequence, the solution may on the one hand be quite accurate, especially for standard, simple processes; on the other hand, we have no way to know, a priori, how accurate the solution will be, since there is no way to check the validity of physical simplifications and estimate the error for the mathematical model. Therefore, the main usefulness of analytical approximation is to reveal qualitative behaviors.

Numerical approximations provide general, versatile and relatively easy ways to obtain quantitative information. Their application is limited by the methods employed and capabilities of the computer. Digital computers can deal only with discrete approximation of continuum concepts such as time and length. Therefore, the mathematical concepts, such as derivatives, integrals, and limits, must be reapprroximated by finite differences, quadratures, and approximate values, respectively. We will introduce three major numerical approximations modeling phase-change problems: front-tracking methods and fixed-domain methods, including enthalpy methods and phase-field methods. Their advantages and restrictions will be demonstrated here.

1.3.1 Front-Tracking Methods

Front-tracking methods are numerical schemes to compute the position of the moving boundary at each time step, and they attempt to explicitly track the interface using the Stefan condition. Such methods require the phase front to evolve smoothly in time and space, and one will usually have to draw on some a priori knowledge of the solution based on the physical model in order to judge whether the front is trackable, as in Meyer [10]. Numerical techniques of front-tracking methods include fixed finite-difference grids and modified grids [6].

Suppose the heat-conduction equation is to be solved by using finite-difference methods for the derivatives in order to compute temperature, $T_i^n$, at discrete points $(i\Delta x, n\Delta t)$ on a fixed grid in the $(x, t)$ plane. In fixed finite-difference grid techniques, at any time $n\Delta t$, the phase-change boundary will usually be located between two adjacent grid points, say $i\Delta x$ and $(i + 1)\Delta x$ as shown in Figure 1.1. This can be allowed by using a modified finite-
difference formula which incorporates unequal space intervals near the moving boundary via, e.g., Lagrange interpolation. For points other than \(i\Delta x, s(t)\) and \((i + 1)\Delta x\), the usual finite-difference formula for equally-spaced intervals is used. However, the disadvantage of the interpolation scheme is that it generally causes loss of accuracy. Therefore, various finite-difference schemes have been applied to approximate both the Stefan condition on the moving boundary and the partial differential equation at the neighboring grid point in order to remedy this. For example, Furzeland [11] suggested an approximation for \(\partial T/\partial x\) centered on the moving boundary and containing fictitious values of \(T^n_i\) and \(T^n_{i+1}\) to be eliminated later.

Moreover, modified grids have been proposed to avoid the increased complexity and loss of accuracy associated with unequal space intervals near the moving boundary. One approach is to fix the spatial step, \(\Delta x\), but allow the time step, \(\Delta t_n\), to “float” in such a way that the front always passes through a node \((i\Delta x, n\Delta t_n)\). Another approach is to fix the time step and allow the spatial step to “float”—in fact, use two distinct and time-varying space steps for the two phases. In addition, adaptive meshes and/or finite element methods can be applied in space and time for non-rectangular \((x, t)\) grids.

All front-tracking approaches work well, more or less, for simple Stefan problems in which we know what to expect. It may be difficult or even impossible to track the moving boundary directly if it does not move smoothly or monotonically with time. Such cases include sharp peaks, multiple fronts, disappearing phases, non-predictable behaviors, and internal heating. All these reasons make front-tracking schemes unviable as general simulation tools for modeling realistic phase-change processes. The possibility, therefore, exists in reformulating the problem in such a way that the Stefan condition is implicitly bound up in a new form of
the equation, which applies over the whole fixed domain. Thus, the position of the moving boundary appears as one feature of the solution.

1.3.2 Enthalpy Methods

Enthalpy methods are one of the fixed-domain methods, in which an enthalpy or total heat function is introduced. In these approaches the jump condition (Stefan condition) is not forced on the solution, but obeyed automatically by it as a “natural boundary condition;” thus such methods bypass the explicit tracking of the interface. Their theoretical basis consists of a formulation of the Stefan problem different from the classical one, the so-called weak or enthalpy formulation, which is similar to the weak formulations commonly used in gas dynamics for shocks.

The enthalpy approach is based on the fact that the energy conservation law, expressed in terms of energy (enthalpy) and temperature, together with the equation of state, contain all the physical information needed to determine the evolution of the phases. The localized differential form of heat balance can be expressed as

\[ E_t + \nabla \cdot \overline{q} = 0 \quad (1.10) \]

The procedure of enthalpy formulation is to introduce an enthalpy function, \( E(x,t) \), which is the total heat content, i.e., the sum of the sensible heat and the latent heat required for a phase change. The heat jump at the moving boundary is incorporated in the definition of \( E(x,t) \) as follows [5] [6]:

\[ E(x,t) = \int_{T_{m}}^{T} \left[ \rho c_p(\theta) + \rho L_0 \delta (\theta - T_m) \right] d\theta, \quad (1.11) \]

where \( \delta \) is the Dirac function. Here we assume that density \( \rho \) is a constant. The form of \( E(x,t) \) is shown graphically in Figure 1.2.

Thus, the phases are described by

\[ E(x,t) \leq 0 \quad \Rightarrow \quad \text{solid} \quad (1.12a) \]
\[ 0 < E(x,t) < \rho L_0 \quad \Rightarrow \quad \text{interface} \quad (1.12b) \]
\[ E(x,t) \geq \rho L_0 \quad \Rightarrow \quad \text{liquid} \quad (1.12c) \]
Figure 1.2 Sketch of enthalpy function.

Therefore, the front location may be recovered a posteriori from the values of the enthalpy. In addition, a convenient phase-indicator, viz., the liquid fraction of a control volume, can be introduced and defined as

\[
\lambda^n_i = \begin{cases} 
0, & \text{if } E^n_i \leq 0 \quad \text{(solid)} \\
\frac{E^n_i}{\rho L_0}, & \text{if } 0 < E^n_i < \rho L_0 \quad \text{(mushy)} \\
1, & \text{if } E^n_i \geq \rho L_0 \quad \text{(liquid)},
\end{cases}
\]

with \(0 < \lambda^n_i < 1\) corresponding to the mushy zone.

1.3.3 Phase-Field Models

Enthalpy methods are unable to solve all phase-change problems. Excluded are problems which we do not know how to formulate weakly due to their special interface conditions. These are problems with supercooling, where the instability of the interface introduced by supercooling must be studied. The so-called phase-field approach is another fixed-domain type formulation to numerically simulate such problems successfully.

In phase-field models, the front is allowed to be diffuse rather than of zero thickness as in front-tracking methods, and the fields of interest, such as temperature \(T\) and concentration \(C\), are supposed to have well-defined bulk behaviors away from the interfacial region, and rapid, though continuous, variations within it. In addition, we intend to require the model to satisfy the laws of thermodynamics, appropriately extend into the nonequilibrium regions,
and regain the interfacial properties and jump conditions when the interfacial thickness approaches zero. This diffuse-interface approach makes it possible to simulate complex microstructures numerically.

In general, phase-field models provide a method to solve time-dependent free-boundary problems without explicit front-tracking of the interface position. In phase-field models, an additional variable, phase field $\phi$, is introduced to explicitly label the solid-liquid interface as follows:

$$
\phi = \begin{cases} 
0, & \text{(solid)} \\
1, & \text{(liquid)} 
\end{cases}
$$

where $0 < \phi < 1$ corresponds to the interface region, and $\phi$ varies smoothly between 0 and 1. Therefore, a system of continuum equations that governs concentration, temperature, and an extra equation, the phase equation, must be derived and subsequently solved. It is conventional to assign $\phi = 0.5$ at the interface. A simulation of the interface then involves the locations of $\phi = 0.5$ and not the solution of a free-boundary problem, per se.

Figure 1.3 (taken from Chen [12]) provides a sketch of how a phase-field profile appears, where the blue region ($\phi = 0$) denotes solid phase, and the red one ($\phi = 1$) denotes liquid phase. If we draw a line at $y = b$ and parallel to $x$-axis as shown in Figure 1.4 (also from [12]), we observe that the phase field $\phi$ varies from 0 (solid) to 1 (liquid) smoothly, but sharply, as indicated by $\phi(x, b)$. The width of interface is determined by the capillary length $d_0 = c_p \sigma T_m / \rho L_0^2$.

![Figure 1.3](Sketch of phase field profile.)
1.4 Summary of Current State of the Art

We will report a thorough literature survey on important works of phase-field models, including mathematical modeling, asymptotic analysis of phase-field models for pure substances and binary alloys, with and without convection in liquid, and numerical simulations. We then list some problems that can be done using phase-field models but have not yet been done. In addition, we will present what we have done to improve the situation.

1.4.1 What has been Done

The basic idea of “phase field” was initiated in the late 1970s. Numerous studies introducing phase-field models were reported in the 1980s leading to increasing interest in solving phase-transition problems. Fix [13] proposed a method in 1983 including two important effects, surface tension and supercooling, not captured in classical Stefan problems. Surface tension is a stabilizing factor, while supercooling tends to amplify any perturbation introduced in the system. Presence of surface tension and supercooling imposes certain difficulties for numerical solutions of Stefan problems, and early attempts to remove the difficulties were not very successful. In [13], Fix introduced a phase variable, a so-called order parameter, $\phi$, in his model that is to be determined by an appropriate field equation and varies sharply but smoothly from one value to another over the solid-liquid interface. Langer [14] presented a simple, intuitive description of the phase-field model without proposing any numerical solution approach; Caginalp [15] presented a mathematical analysis of the phase-field approach.
to solidification problems, proving the existence and regularity of solutions. He also provided a rigorous asymptotic analysis leading to the Gibbs–Thomson condition, which relates the temperature at the interface to the surface tension and curvature. Caginalp and Fife [16] introduced a phase-field model that incorporated surface tension, anisotropy, curvature and dynamics of the interface, and supercooling, by employing the Landau–Ginzburg [17] approach. Later, Caginalp [8] showed that as a limiting case of the continuous representation of phase transitions based on microscopic considerations, the phase-field equations can lead to the sharp-interface models, e.g., classical Stefan models, modified Stefan models, Hele–Shaw models, and the Cahn–Allen model. He and Socolovsky [9] also showed in detail that the sharp-interface problem arising from any of the major phase transition problems (classical or modified Stefan problems, etc.) can be recovered mathematically by using the phase-field approach as a numerical method to smooth solutions over a thin interface with finite thickness, in which effects of surface tension and supercooling are included. Therefore, the phase-field method converts the sharp interface problem to a stiff system of partial differential equations for the evolution of phase and temperature fields.

In 1990, Penrose and Fife [18] derived a “thermodynamically-consistent” phase-field model using an analogous entropy functional instead of the Landau–Ginzburg free energy functional. A “thermodynamically-consistent” phase-field model, based on the first and second laws of thermodynamics, was also presented by Wang et al. [19]. McFadden et al. [20] provided an asymptotic analysis in the sharp-interface limit of the phase-field model studied by Kobayashi [21] and Wheeler et al. [22], including an anisotropic mobility. They concluded that the surface free energy and the thickness of the diffuse interface have the same anisotropy, whereas the kinetic coefficient has an anisotropy characterized by the product of the interface thickness with the intrinsic mobility of the phase field. Bates et al. [23] conducted a formal asymptotic analysis on phase-field models for hypercooled (\(St > 1\)) solidification. They considered anisotropy and nonlocal variations on the phase-field model and examined their effects on the free-boundary problem. They also showed that in this context, the analysis predicts without extra assumptions certain properties of the interface.

It is worth mentioning that Karma and Rappel [24] [25] developed a “thin-interface” limit of the phase-field model. This is intended to circumvent two limitations of the phase-field
approach, namely, necessity of nonzero interface kinetics and computational complexity, to a large degree by modifying the expression for the inverse of the kinetic coefficient. The modified result in dimensional form is

$$\beta = \frac{c_p}{\mu_k L_0} - a_2 \frac{\delta}{\alpha},$$

(1.15)

where $\delta$ is an interface thickness parameter, and $a_2$ is a positive constant of order unity that depends on the choice of free-energy functional forms, $F$, in the phase-field model. By way of contrast, in Caginalp’s asymptotic analysis [8], the following simpler expression was obtained:

$$\beta = \frac{c_p}{\mu_k L_0}.$$

(1.16)

Here we have used the notations of the current work, but Eqs. (1.15) and (1.16) are equivalent (modulo notation) to the corresponding ones in [25] and [8], respectively.

This thin-interface limit has two computational advantages compared with currently used phase-field models. First, it is able to perform simulations with a larger $\delta/d_0$ ratio (where $d_0 = c_p \sigma T_m / \rho L_0^2$ is the capillary length), which reduces the computation time, especially when the degree of supercooling is small, since it allows larger spatial steps. Second, the inverse of the kinetic coefficient, $\beta$, can be made zero or even negative, permitting simulation of negligible interface kinetics cases, especially important for metallic systems with fast kinetics.

Later, McFadden et al. [26] applied the asymptotic methodology in [25] to a phase-field model with unequal conductivities, derived using a thermodynamically consistent approach. They also compared scalings of surface tension and kinetics between the classical asymptotic analysis [8] and the thin-interface analysis [25]. They found that as interfacial thickness tends to zero, surface tension and kinetics effects remain fixed for the classical asymptotic analysis, while they vanish for the thin interface analysis. Vetsigian and Goldenfeld [27] presented a modification of the phase-field model based on [25], which allows the interface thickness to be many times larger than the capillary length.

Elder et al. [28] recently provided a detailed derivation of the sharp interface limits of phase-field models. They pointed out that the main difficulty of connecting phase-field models and the sharp-interface description is how to account for finite thickness $\delta$ of the diffuse
interface of the continuum phase-field models. They also stated that the approach, which
takes the limit of the interface width of the phase-field model to zero [8], is not very useful
since the interface width is always finite, while those introducing some artificial form of the
free energy functional $F$ to include an interface of nonzero width [24] [25] is counterintu-
itive. They provided one set of equations that related the parameters of the phase-field
equations to those of the sharp interface equations for phenomena including: order-disorder
transitions, dendritic growth, phase separation in binary alloys, eutectic growth, and sur-
face roughening. In this approach, they avoided a zero interface width and eliminated the
counterintuitive necessity of the thin-interface analysis [25]. In particular, thermodynamic
consistency is automatically satisfied in this approach.

Initially, most researchers focused their work on pure substances. Kobayashi [21] in-
troduced “a kind of phase field model” to simulate directional solidification and dendritic
crystal growth, with and without anisotropy. He also investigated the stability of the shape
of interfaces against noise by introducing low-amplitude random fluctuations. Wheeler et
al. [22] applied the phase-field model to a one-dimensional spherically-symmetric geometry
and a planar two-dimensional rectangular region to numerically simulate a solid front grow-
ing into a supercooled melt. Murray et al. [29] presented their computation of dendritic
solidification based on the model provided in [19]. The numerical study of Kupferman et
al. [30] focused on the asymptotic late-stage growth in the large supercooling limit. Cagi-
nalp and Socolovsky [31] applied a computational method for smoothing a sharp interface
problem within the scaling of distinguished limits of the phase-field equations that preserves
the physically important parameters. By varying model parameters, they demonstrated that
the single system of equations can be used to compute a broad spectrum of phenomena that
are generally associated with sharp interface problems, from motion by mean curvature, to
stable anisotropic crystal growth, to single-needle dendrites. González-Cinca et al. [32] pre-
sented numerical simulations of a phase-field model based on [22] and [20] to study strong
anisotropy and faceted interfaces. They compared the dendrites of smectic-B germs growing
into the supercooled nematic phase with experiments, and they showed that the simulations
qualitatively reproduced a rich variety of morphologies observed in the experiments for dif-
ferent values of supercooling from the faceted equilibrium shape to fully-developed dendrites.
Provatas et al. [33] simulated dendrite growth using the phase-field model employed in [24] at low undercooling. They compared their simulations with experimental data on pivalic acid (PVA) obtained from NASA’s USMP-4 isothermal dendritic growth experiment (IDGE), described in detail by Koss [34], finding that the time-dependent evolution of a 2-D dendrite is self-affine in time, and that the dendrite tip position and maximum dendrite width exhibit a power law dependence on time. Recently, Nestler et al. [35] performed 2-D and 3-D simulations of dendritic structures using a phase-field model for non-isothermal solidification in multicomponent systems introduced by Garcke et al. [36]. They compared their results with the analytical predictions of the Brener theory [37] and recent experimental measurements of solidification in pure nickel by Herlach et al. [38]. They also compared the interface velocities for different undercooling between thin and sharp interface asymptotics.

Phase transitions in binary alloys have recently attracted increasing interest. In 1992, Wheeler et al. [39] presented a phase-field model of solute trapping during solidification, by introducing a species equation into the phase-field model. As the gradient energy coefficient of the phase field becomes small, it showed that their asymptotic model recovered classical sharp interface model results for alloy solidification when interface layers are thin. Moreover, they related parameters in the phase-field model to material and growth parameters in real systems. Later, Wheeler et al. [40] developed a phase-field model for isothermal solidification of a binary alloy containing a gradient energy term for both the phase field and the solute field, instead of for the phase field only as in [22]. They considered the gradient-energy effect of the concentration since the length scale of the solute field approaches atomic dimensions for velocities on the order of 100\(cm\cdot s^{-1}\), which is common in rapid solidification experiments. Caginalp and Xie [41] also presented a phase-field model of a binary mixture or alloy that has a phase boundary. The phase-field model is the simplest system of equations that has the characteristic behavior of alloys and also has the appropriate scaling and coefficients. They obtained a new interface relation for concentration that is discontinuous across the interface in the simplest phase-field system.

Recently, more and more works have been focused on phase-field models of solidification with convection in the liquid phase. Anderson et al. [42] derived a phase-field model for the solidification of a pure material with convection and, based on the model they obtained,
investigated two important simple cases: density-change flow and shear flow. They modeled the solid phase as an extremely viscous liquid and employed the formalism of irreversible thermodynamics to derive the governing equations. Later, Anderson et al. [43] examined, in detail, a simplified version of the phase-field model in the sharp-interface limit to derive the interfacial conditions of the associated free-boundary problem. Their analysis revealed the underlying physical mechanisms built into the phase-field model in the context of a free-boundary problem and provided a further validation of the model. Beckermann et al. [44] provided a phase-field model with convection using the volume or ensemble averaging methods. They exhibited two examples, coarsening in an isothermal mush of a binary alloy, where both the interface curvature and the flow permeability evolve with time, and dendritic growth in the presence of melt convection, with particular emphasis on the operating point of the tip. Jeong et al. [45] studied the effect of a fluid flow on 3-D dendrite growth using a phase-field model on an adaptive finite element grid. They used an averaging method for the flow problem coupled to the phase-field model introduced in [44]. They also validated 2-D dendrite-growth results with solvability theory and previous results, finding them in good agreement. Furthermore, they found that the physics of dendrite growth within a flow field in 3D is very different from that in 2D. Nestler et al. [46] discussed two phase-field models for solidification of monotectic alloys, in which a liquid phase simultaneously transformed into a new liquid phase and a solid phase. The first one used three different phase fields to characterize the three phases in the system and, in addition, a concentration field. The second one distinguished the two liquid phases by their concentrations, using a Cahn–Hilliard type model, and employed only one phase field to characterize the system as solid or liquid. The difference between the two models was that the first one restricted the validity of the model to the vicinity of the monotectic temperature, while the second one enabled the model to represent a wider temperature range.

In contrast, Juric and Tryggvason [47] simulated dendritic solidification with a somewhat different approach, namely, a front-tracking method related to level sets, thus providing an independent check of numerical predictions. The front-tracking method explicitly provides the location of the interface at all times, and the Gibbs–Thomson condition on the interface temperature is also explicitly satisfied. The front-tracking method they presented can eas-
ily handle discontinuous material properties between the liquid and solid phases, topology changes, and anisotropy of interfacial energy and kinetics. It allows a detailed description of the interface microstructure. Al-Rawahi and Tryggvason [48] simulated the dendritic growth of pure substances in the presence of flow using a front-tracking method similar to [47]: the liquid-solid interface was explicitly tracked, and the latent heat released during solidification was calculated using the normal temperature gradient near the interface. They validated their method through a comparison with an exact solution for a Stefan problem, a grid refinement test, and a comparison with a solution obtained by a boundary integral method.

Finally, we will emphasize some works on adaptive griding. We note that the small physical length and time scales of problems to which phase-field models are applied require the spatial and time step sizes to be very small, which increases the computational complexity for uniform grid discretizations. However, since most variations occur over the interface, there is no necessity to use such small scales in the bulk phases as those needed on the interface. Therefore, adaptive griding is a good method to consider. McCarthy [49] investigated the advantages of using adaptive grid techniques to solve the phase-field model for pure substances in 1D. Provatas et al. [50] [51] studied dendritic microstructure evolution using an adaptive finite element method applied to a phase-field model. They discussed the details of their algorithm in [51] and showed that it greatly reduced the computational cost of solving the phase-field model at low supercooling. Simulations using adaptive finite element grids can also be found in [45].

1.4.2 Important Problems that cannot be Solved

Although numerous works have been done to study phase-change processes numerically and experimentally, there are many problems that currently cannot be solved. For example, because of the limitation of experimental tools and unpredictable behaviors of dendrite structures, experimental data do not guarantee the correctness of measured results. They will be affected by such unavoidable factors as gravity, impurity of the substances, unexpected perturbations introduced into the system, etc. In addition, although phase-field models can be employed to solve many realistic problems in two or three dimension, there are still some problems left out.
Direct numerical simulations of dendrite growth have been studied in 2D, and even in 3D, but most of the results are exhibited without perturbations. However, different kinds of perturbations exist in reality, and such perturbations are not appropriately introduced by random number generators. Therefore, one task to simulate more realistic dendrite structure and study how the dendrite behaves under perturbations is to introduce small-scale fluctuations that contain physics to describe the small-scale behavior of solidification processes.

Numerical simulations on solidification with convection in the liquid phase by using phase-field models have been performed by numerous researchers as discussed in Section 1.4.1. All of the previous works were focused on the influence of convection on the dendrite structures. Moreover, we note that in those studies, the Reynolds number has to be very small in order to avoid unreasonably large velocity since the length scale is extremely small (on the order of a micro-, sometimes even a nano-meter). Therefore, solidification problems with convection are restricted to those with small Reynolds numbers and simple flow cases, such as Stokes’ flow or stationary flow, which are not realistic problems. The shortcoming of these works arises in inducing the flow field based on the length and time scales of dendrite growth. One remedy to this is to implement multiscale (two-level or three-level scales) methods.

Simulations of phase-change problems are restricted because of their computational intensity. To remedy this, one possibility is to use adaptive gridding since most of the computations occur over the interface, and the other is to implement parallelization techniques. Some works on adaptive gridding have been described in Section 1.4.1. Parallel procedures will be implemented in the current work and reported in Section 3.5.

Recall that in the assumptions of phase-field models in Table. 1.1, the thermodynamical properties (heat capacity $c$ and heat conductivity $k$) are constant in each phase to reduce computational complexity. This assumption, together with the assumption that the latent heat is constant, can be relaxed, as will be done later. Other heat transfer effects, such as radiation and other physics, viz., gravity, elastic, electromagnetic, etc., can be included if good approximations are to be made. Drag and lift forces should be imposed on dendrites in a flow field.
1.4.3 How this Research will Help

Although numerous studies have been conducted using phase-field methods to solve phase-change problems, specific and detailed attention has seldom been given to the numerical analysis, per se. In particular, it has been typical to present results without citing details of the methods and parameters used to obtain them. Moreover, most of the works have shown the convergence of asymptotic analysis, but they do not show the convergence of numerical solutions to the differential equations. This makes the grid function convergence test important because of its ability to show the convergence of numerical methods and to compute the order of accuracy. Therefore, one purpose of this research is to provide a complete numerical simulation of dendrite growth into a supercooled melt and, in doing so, to carefully present details of the numerical analysis.

Another task of this research is to simulate dendrite growth in a high Reynolds-number flow (in this case lid-driven-cavity flow) and later more complex flows, hopefully in the future, to simulate ice rivers. We have mentioned before that previous solidification with convection problems have been solved with small Reynolds numbers. We will implement a two-level multiscale method, i.e., a macroscale for flow field and a microscale for dendrite growth, which enables us to solve phase-change problems in a high-$Re$ flow.

1.5 Brief Outline of Dissertation

The structure of this dissertation will be as follows. In Chapter 2, we will present some important physics during solidification processes, such as conditions for nucleation, dendrite growth rate, etc. Importance of the kinetic coefficient will be emphasized here, and the methods to estimate it will be briefly provided. The coupled Navier–Stokes, phase-field, and energy equations will be presented in Chapter 3, together with the prescribed initial and boundary conditions. Derivations leading to the governing equations will be provided, and numerical methods to solve the discrete form of the governing equations will be given. Some major algorithms to implement the numerical methods will also be provided. A two-level heterogeneous multiscale method for the phase-field model in a flow field will be discussed in detail and implemented in Chapter 4. Results and analyses are presented in Chapters 5 and 6. In these chapters, we first conduct some computational tests to verify that the
numerical schemes, i.e., the discrete equations, are convergent to the governing equations (i.e., consistency). Validation of the phase-field model will be provided by the demonstration of how these results capture the physics in solidification and how they compare with the experimental data. In other words, questions such as “Are the results from the phase-field model describing the correct solidification phenomena?” will be answered in these chapters. Dendritic structures for both isotropic and anisotropic surface energy will be shown as part of the solutions. Finally, we will demonstrate the solidification processes in a lid-driven-cavity flow. Summary, conclusions, and proposed future work will be given at the end of the dissertation.
Chapter 2  
PHYSICS OF SOLIDIFICATION

In this chapter we provide a description of the physics that the phase-field model must capture. Kurz and Fisher [52] note that solidification generally occurs in two stages: the first stage is nucleation controlled, and the second is growth controlled. Once nucleation has occurred, atom transfer to the crystals must continue in order to ensure their growth, which, if successful, begins the second stage. In Section 2.1 we consider nucleation, and in Section 2.2 we discuss the dendrite growth rate of the second stage.

2.1 Conditions for Onset of Nucleation

A phase transition (e.g., transition from liquid to solid) is usually initiated by a nucleation process, which proceeds via rearrangement of atoms to form a new crystal structure. Nucleation of a crystal from its melt depends mainly on two processes: thermal fluctuations that lead to the creation of variously-sized crystal embryos and creation of an interface between the liquid and solid. Hence, the critical condition for nucleation of a spherical nucleus is derived by summing the volume and interface terms for the Gibbs free energy $\Delta G_V$ and $\Delta G_I$, respectively, given by Kurz [52] and Papon et al. [53]:

$$
\Delta G = \Delta G_I + \Delta G_V
= 4\pi r^2 \sigma + \frac{4\pi r^3}{3} \Delta g_V,
$$

(2.1)

where $r$ is the radius of a 3-D spherical nucleus, and $\sigma$ is the surface tension; $\Delta g_V$ denotes the Gibbs free energy of formation per unit volume of the new phase. Here $\Delta G$ behaves as a potential barrier and must be lowered for a nucleus to be formed more easily.

$\Delta g_V$ is positive if $T > T_m$, where $T_m$ is the melting temperature, which implies the new phase is unstable since $\Delta G$ is positive; then nuclei of the new phase do not tend to grow. On the contrary, $\Delta g_V$ is negative if $T < T_m$. There exist values of $r$ for which $\partial \Delta G / \partial r < 0$, and the corresponding nuclei will tend to stabilize the new phase since their formation reduces the free energy of the material. Therefore, a critical radius $r^*$ exists as shown in Figure 2.1, and the nuclei is stable as long as $r \geq r^*$. For a supercooled melt, i.e., $T < T_m$ ($\Delta T = T_m - T > 0$), $\Delta G$ achieves a maximum as $r \rightarrow r^*$. Formation of nuclei of
Figure 2.1  Free energy of a nucleus as function of radius.

increasing size results in stabilization of the solid phase once \( r > r^* \); conversely, a nucleus with an initial radius less than \( r^* \) tends to shrink and then disappear. The critical radius can be obtained by computing \( \partial \Delta G / \partial r = 0 \) in Eq. (2.1) as follows:

\[
    r^* = -\frac{2\sigma}{\Delta g_V}. \tag{2.2}
\]

The analogous derivation applied to a 2-D circular nucleus leads to the free energy

\[
    \Delta G = 2\pi r \sigma + \pi r^2 \Delta g_V. \tag{2.3}
\]

Hence, the corresponding critical radius \( r^* \) and area \( A^* \) in 2D are

\[
    r^* = -\frac{\sigma}{\Delta g_V}, \quad A^* = \pi \left( \frac{\sigma}{\Delta g_V} \right)^2. \tag{2.4}
\]

If \( \Delta h_V \) and \( \Delta s_V \) represent changes in enthalpy and entropy, respectively, associated with the formation of a unit volume of a new phase, we have

\[
    \Delta g_V = \Delta h_V - T \Delta s_V.
\]
In general, we can write

$$\Delta h_V(T) = \Delta h_V(T_m) - \int_T^{T_m} \Delta \rho c_p \, d\tau$$

$$= -\rho L_0 - \int_T^{T_m} \rho \Delta c_p \, d\tau,$$  \hspace{1cm} (2.5a)

$$\Delta s_V(T) = \Delta s_V(T_m) - \int_T^{T_m} \frac{\rho \Delta c_p}{\tau} \, d\tau$$

$$= -\frac{\rho L_0}{T_m} - \int_T^{T_m} \frac{\rho \Delta c_p}{\tau} \, d\tau,$$  \hspace{1cm} (2.5b)

with the postulation that $\Delta h_V = -\rho L_0$ and $\Delta s_V = -\rho L_0/T_m$ at the melting temperature $T_m$, where $\rho$ is the density, and $L_0$ is the latent heat of fusion per unit mass at the melting temperature; $\Delta c_p$ is the difference in specific heat between the liquid and solid phases. Hence,

$$\Delta g_V = -\frac{\rho L_0 \Delta T}{T_m} - \int_T^{T_m} \rho \Delta c_p \, d\tau + T \int_T^{T_m} \frac{\rho \Delta c_p}{\tau} \, d\tau.$$  \hspace{1cm} (2.6)

Here, $\Delta T = T_m - T$ is the degree of supercooling of the liquid.

In some cases, e.g., metal, we can assume that $\Delta c_p = 0$, then Eq. (2.6) reduces to

$$\Delta g_V = -\frac{\rho L_0 \Delta T}{T_m}.\hspace{1cm} (2.7)$$

Hence, the critical radius $r^*$ is

$$r^* = \frac{2\sigma T_m}{\rho L_0 \Delta T},\hspace{1cm} (2.8)$$

and in 2D we obtain the critical radius and critical area as follows:

$$r^* = \frac{\sigma T_m}{\rho L_0 \Delta T};\hspace{1cm} A^* = \pi \left( \frac{\sigma T_m}{\rho L_0 \Delta T} \right)^2.\hspace{1cm} (2.9)$$

In other cases $\Delta c_p$ is assumed to be a constant; we then obtain

$$\Delta g_V = -\frac{\rho L_0 \Delta T}{T_m} - \rho \Delta c_p \Delta T + T \rho \Delta c_p (\ln T_m - \ln T)$$

$$\approx -\frac{\rho L_0 \Delta T}{T_m} - \rho \Delta c_p \frac{\Delta T^2 (T_m + \Delta T)}{2T_m^2}.\hspace{1cm} (2.10)$$

Here we take the difference in the specific heats at the melting point $\Delta c_p^f$ as the value of $\Delta c_p$. In the above equation, we can apply Taylor expansion about $T_m$ to $\ln T$ if the degree of supercooling $\Delta T$ is small and obtain an approximation to second-order accuracy in $\Delta T$.  

27
It is often more realistic to assume $c_p$ is related to temperature as

$$\Delta c_p(T) = A + BT + CT^2 + DT^3 + E/T^2,$$  \hspace{1cm} (2.11)

where $A, B, C, D$ and $E$ are constants. In this case Eq. (2.6) becomes

$$\Delta g_v = \rho \left[ -\frac{L_0 \Delta T}{T_m} + AT (\ln T_m - \ln T) - A\Delta T - \frac{B\Delta T^2}{2} - \frac{C\Delta T^2(2T_m + T)}{6} ight. \
- \left. \frac{D\Delta T^2(T^2 + 2TT_m + 3T_m^2)}{12} - \frac{E\Delta T^2}{2T_m^2} \right].$$ \hspace{1cm} (2.12)

### 2.2 Dendrite Growth Rates

Once a nucleus is formed, it will continue to grow. The growth rate is limited by the kinetics of attachment of atoms to the interface, capillarity, and diffusion of heat and mass [52]. In other words, the crystal growth rate depends on the net difference between the rates of attachment and detachment of atoms at the interface. Moreover, it depends on the diffusion rate in the liquid and the interface roughness on microscopic scales. Cahn [54] has shown theoretically that there are two major mechanisms related to crystal growth: stepwise and continuous. The mechanism that actually occurs for a specific interface depends on the (thermodynamic) driving force and diffuseness of the interface. Cahn et al. [55] have identified three different regimes according to the magnitude of the driving force: classical regime, transitional regime, and continuous regime, which correspond to low driving force (lateral mechanism), intermediate driving force (lateral spreading mechanism), and high driving force (continuous mechanism), respectively. They also presented expressions for the growth rate in the three regimes. Here, we introduce the expressions only for classical and continuous regimes for simplicity; the transitional regime is more complicated because of the necessity to determine the transition undercooling.

In the classical regime the Gibbs free energy is given by

$$G = \frac{\beta(1 + 2\sqrt{g})}{g} \frac{DL^2\Delta T^2}{4\pi RT^3\sigma V_m},$$

or

$$G = \frac{\beta DL^2\Delta T^2}{4\pi g\sigma RT^3V_m} \quad \text{for} \quad g \ll 1,$$ \hspace{1cm} (2.13)
where $g$ is the diffuseness parameter ($g \sim 1$ implies that the interface is sharp); $L$ is the latent heat of fusion per mole, with $V_m$ the molar volume of solid, $a$ the step height and $R$ the gas constant (see Hillig [56]). In the continuous regime

$$G = \frac{\beta DL \Delta T}{aRT^2},$$  \hspace{1cm} (2.14)

the assumption that the transport process, by which molecules add to the solid at the step, is related to liquid diffusion introduces the parameter $\beta$ that expresses the degree to which the assumption is not strictly satisfied. An empirical relationship for this parameter is [55]

$$\beta = 6 \left( \frac{a}{\lambda} \right)^2 \left( \frac{\nu_i}{\nu_l} \right),$$

where $\lambda$ is the mean jump distance that a molecule moves to the next diffusion step in the liquid, and $\nu_i, \nu_l$ are the jump frequencies at the interface and in the liquid, respectively. The growth rate equations (2.13) and (2.14) imply a quadratic relation between the growth rate and the degree of supercooling for a lateral mechanism and a linear relation for a continuous mechanism.

### 2.3 Kinetic Coefficient

The theory of perfect crystal growth has been under development for almost a century. The essential development of the theory can be found in the works of Volmer [57], Stranski and Kaischew [58], Becker and Döring [59], and Burton et al. [60]. It was stated in the theory that when all stepped surfaces have disappeared, crystal will continue to grow by a 2-D nucleation of new molecular layers on saturated surfaces. In addition, the probability of the formation of these 2-D nuclei is very sensitive to supersaturation. Burton et al. [60] studied the theory of crystal growth, including the movement of steps on a crystal surface and the growth rate. They also analyzed the equilibrium structure of steps and its dependence on temperature, binding energy parameters, and crystallographic orientation.

Fundamental processes of crystal growth can be summarized in three steps: migration of adsorbed molecules, advance of steps, and nucleation on a growing surface as described by Ookawa [61]. The sketch of adsorbed molecules, kinks, and steps is shown in Figure 2.2. Surface of crystal can be automatically flat or rough. An automatically flat surface grows step by step, and its normal growth rate is determined by the step source energy and the
step velocity; whereas the addition of new particles to the lattices on a rough interface takes place practically at each surface position, and the growth rate depends on the step roughness and the kinetics of atomic crystallization processes such as desolvation, chemical reactions, surface diffusion, impurity adsorption and desorption, liquid boundary layer structure, etc. The step roughness, i.e., the kink density, strongly depends on crystallization conditions. For example, the kink density is low at low temperatures, and it decreases also with the increase of the step height. Increasing kink density leads to an increasing kinetic coefficient as supersaturation increases and to a superlinear increase of the step velocity and crystal growth rate at near-equilibrium conditions.

The kinetic coefficient, $\mu_k$, reflects the kink density at the step and the atom exchange rate at each kink according to Chernov [62]. To understand the physics of the kinetic coefficient, we first denote frequencies of atom (molecule) addition and subtraction at the kink by $\nu_+$ and $\nu_-$, respectively. The former exponentially depends on the activation energy, and the latter contains the dissolution heat. Probability of the kink density at an arbitrary atomic position at the step is $a/\lambda_0$, where $a$ is the interatomic distance and $\lambda_0$ is the interkink distance. If we assume that kinks are smoothly spread along the step, and the solution concentration $c$ is a constant, the probability of crystalline aggregation in the kink vicinity is $\sim ca^3(a/\lambda_0)$ [62]; hence, the aggregation flux into the crystalline phase is $W_+ = \nu_+ca^3a/\lambda_0$. In addition, the dissolution flux becomes $W_- = \nu_-(1-ca^3)a/\lambda_0$. As a consequence, the diffusion flux per
atomic site is
\[ a^2 D \frac{\partial c}{\partial r} = W_+ - W_- = \nu_+ c a^3 a/\lambda_0 - \nu_- (1 - c a^3) a/\lambda_0, \tag{2.15} \]
where \( r \) is the radius vector perpendicular to the step line, and \( D \) is the diffusivity in solution. We remark here that all parameters introduced are dimensionless. At equilibrium we have \( \partial c/\partial r = 0 \); then the equilibrium concentration of the solution \( c_e \) can be obtained from Eq. (2.15) as follows:
\[ c_e = \frac{\nu_-}{a^3(\nu_+ + \nu_-)}. \tag{2.16} \]
Therefore, the diffusion flux equation (2.15) becomes
\[ D \frac{\partial c}{\partial r} = \frac{a^2}{\lambda_0(\nu_+ + \nu_-)} (c - c_e). \tag{2.17} \]
Moreover, since we have
\[ D \frac{\partial c}{\partial n} = \mu_k(c - c_e) \tag{2.18} \]
from the physical interpretation of the kinetic coefficient, we obtain the kinetic coefficient
\[ \mu_k = \frac{a^2}{\lambda_0(\nu_+ + \nu_-)} = \frac{\nu_-}{c_e a \lambda_0}. \tag{2.19} \]
We emphasize here that the kinetic coefficient in Eq. (2.19) is suitable for crystallization primarily by the exchange between kinks and bulk solutions.

The difficulty in determining the value of the kinetic coefficient is that the parameters \( \nu_+ \), \( \nu_- \), \( a \) and \( \lambda_0 \) in Eq. (2.19) are on the molecular level; hence, it is almost impossible to obtain the exact values of these parameters. As a consequence, the value of the kinetic coefficient computed from Eq. (2.19) cannot be exact; therefore, it is necessary to estimate this value. Linear variation of the kinetic coefficient with temperature might be a reasonable assumption for a molecularly rough interface. However, the kinetic coefficient strongly depends on the orientation of the surface for faceted interfaces [14]. For simplicity, the kinetic coefficient is usually assumed to be a constant because it is difficult to determine its dependence on other parameters either theoretically or experimentally.

We suggest that molecular dynamics simulations might be appropriate for estimating the value of the kinetic coefficient. Hoyt et al. [63] reviewed several different molecular dynamics simulation methods for computing the kinetic coefficient in solid-liquid systems.
They divided those techniques into three different groups: forced velocity simulations (by Broughton et al. [64], and Celestini and Debierre [65]), free solidification simulations (by Hoyt et al. [66]), and fluctuation analyses (by Briels and Tepper [67]). We found that it is actually the front propagation speed (i.e., dendrite growth rate) and not the kinetic coefficient that was being computed in the above papers. Therefore, we must estimate the kinetic coefficient by the propagation speed obtained from these molecular dynamics simulations. However, the resultant kinetic coefficient might be under- or overestimated if an improper relation between the kinetic coefficient and the growth rate is used. For example, Hoyt et al. [66] overestimate the kinetic coefficient by using the relation $\mu_k = V_n/\Delta T$. We will estimate the kinetic coefficients of pure nickel and water in Chapters 5 and 6, respectively, based on previous works by peers and the numerical tests conducted by using the phase-field model in this work.
Chapter 3  
NUMERICAL ANALYSES OF PHASE-FIELD MODEL

In this chapter, we first provide a detailed derivation of phase-field model with convection in the melt, then present a mathematically well-posed problem consisting of the governing equations of the phase-field model with boundary and initial conditions, followed by details of the numerical methods applied to solve the problem and the algorithms to implement the numerical methods.

3.1 Derivation of Phase-Field Model in Flow Field

In this section, we provide the detailed derivation of a thermodynamically consistent phase-field model in a flow field.

Cahn and Hilliard stated that the local free energy in a region of nonuniform composition depends both on the local composition and on the composition of the immediate environment. Therefore, they expressed free energy as the sum of two contributions that are functions of the local composition and its derivatives, respectively [68] [69]. Hence, we can express the Gibbs free energy in the form of

\[ G = \int_V \left[ \rho g(\rho, \phi, T) + \frac{1}{2} \epsilon_G^2 \Phi(\nabla \phi)^2 \right] dV . \]  

(3.1)

Similarly the bulk internal energy and entropy can be expressed as

\[ E = \int_V \left[ \rho e(\rho, \phi, T) + \frac{1}{2} \epsilon_E^2 \Phi(\nabla \phi)^2 \right] dV , \]  

(3.2)

\[ S = \int_V \left[ \rho s(\rho, \phi, T) + \frac{1}{2} \epsilon_S^2 \Phi(\nabla \phi)^2 \right] dV . \]  

(3.3)

Here \( g, e \) and \( s \) are the Gibbs free energy, internal energy, and entropy density per unit mass respectively; \( \rho \) is the density. \( \epsilon_G^2, \epsilon_E^2 \) and \( \epsilon_S^2 \) are the gradient energy coefficients corresponding to the Gibbs free energy, internal energy, and entropy, respectively, and we use a power of two to demonstrate that they are positive. \( \Phi(\nabla \phi) = \xi \cdot \nabla \phi \) is a homogeneous degree one function of its argument. A \( \xi \)-vector is introduced as a homogeneous function of \( \nabla \phi \) of...
degree zero given by $\xi(\nabla \phi) = \partial \Phi(\nabla \phi)/\partial \nabla \phi$. For an isotropic surface energy, $\xi$ becomes a unit vector.

Mass, momentum, energy conservation laws and entropy balance take the form

$$\frac{d}{dt} \int_V \rho \, dV = 0,$$  \hspace{1cm} (3.4a)

$$\frac{d}{dt} \int_V \rho \mathbf{u} \, dV = \int_A F_s \cdot \mathbf{n} \, dA + \int_V b \, dV,$$  \hspace{1cm} (3.4b)

$$\frac{dE_{tot}}{dt} = - \int_A q_E \cdot \mathbf{n} \, dA + \int_A F_s \cdot \mathbf{u} \, dA + \int_V b \cdot \mathbf{u} \, dV + \int_V \dot{r} \, dV,$$  \hspace{1cm} (3.4c)

$$\frac{dS}{dt} = - \int_A q_S \cdot \mathbf{n} \, dA + \int_V \dot{s}_{\text{prod}} \, dV.$$  \hspace{1cm} (3.4d)

where $E_{tot} = E + \int_V \frac{1}{2} \rho \mathbf{u}^2 \, dV$. Substitution of the internal energy and entropy by Eqs. (3.2) and (3.3), and application of the constitutive theory to Eqs. (3.4) lead to the equations in the form of control volume as

$$\frac{D \rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0,$$  \hspace{1cm} (3.5a)

$$\rho \frac{Du}{Dt} = \nabla \cdot F_s + b,$$  \hspace{1cm} (3.5b)

$$\rho \frac{De}{Dt} + \epsilon^2_E \Phi \frac{D \Phi}{Dt} + \frac{1}{2} \epsilon^2_E \Phi^2 \nabla \mathbf{u} = - \nabla \cdot q_E + F_s \cdot \nabla \mathbf{u} + \dot{r},$$  \hspace{1cm} (3.5c)

$$\rho \frac{Ds}{Dt} = \dot{s}_{\text{prod}} + \epsilon^2_S \Phi \frac{D \Phi}{Dt} + \frac{1}{2} \epsilon^2_S \Phi^2 \nabla \mathbf{u} - \nabla \cdot q_S + \frac{\dot{r}}{T}.$$  \hspace{1cm} (3.5d)

Here, $F_s$ and $b$ are surface and body forces, respectively. $\mathbf{n}$ is the unit normal to $A$, and $\mathbf{u}$ is the velocity vector. $T$ is temperature, and $\dot{r}$ is the radiation rate in the control volume. Fluxes $q_E$ and $q_S$ are heat flux and entropy flux across the boundary of the control volume, respectively. $\dot{s}_{\text{prod}}$ is the entropy production rate in the control volume, and it should always obey the second law of thermodynamics, i.e., $\dot{s}_{\text{prod}} \geq 0$.

Next, we will use the relations between the Gibbs free energy, Helmholtz free energy, and internal energy per unit mass to obtain $\dot{s}_{\text{prod}}$. We then apply the second law of thermodynamics to obtain the fundamental form of the governing equations. The relations between $g$, $f$ and $e$ are given by

$$e = Ts - \frac{p}{\rho} + g,$$  \hspace{1cm} (3.6a)

$$f = e - Ts,$$  \hspace{1cm} (3.6b)
where $f$ is the Helmholtz free energy per unit mass. Differentiations of $e$ with respect to $s$, $\rho$ and $\phi$ and $g$ with respect to $\phi$, $T$ and $p$, using Eq. (3.6a), provide

$$de = \frac{\partial e}{\partial s} ds + \frac{\partial e}{\partial \rho} d\rho + \frac{\partial e}{\partial \phi} d\phi = Tds + \frac{p}{\rho^2} d\rho + \frac{\partial g}{\partial \phi} d\phi,$$

$$dg = \frac{\partial g}{\partial \phi} d\phi + \frac{\partial g}{\partial T}dT + \frac{\partial g}{\partial p}dp = \frac{\partial e}{\partial \phi} d\phi - sdT + \frac{1}{\rho} dp. \tag{3.7a}$$

In addition, if we apply the chain rule to the substantial derivative $De/Dt$ and make use of the relation in Eq. (3.7a), we obtain

$$De = \frac{\partial e}{\partial s} Ds + \frac{\partial e}{\partial \rho} D\rho + \frac{\partial e}{\partial \phi} D\phi = TDs + \frac{p}{\rho^2} D\rho + \frac{\partial g}{\partial \phi} D\phi. \tag{3.8}$$

Hence, we obtain an expression for $Ds/Dt$ in terms of the relations for $De/Dt$, $D\rho/Dt$ and $D\phi/Dt$. We then substitute the expression for $Ds/Dt$ back to Eq. (3.5d) and rearrange Eqs. (3.5a) and (3.5c) to obtain expressions for $De/Dt$ and $D\rho/Dt$, respectively. Finally, we have the rate of entropy production

$$\dot{s}_{prod} = \frac{1}{T} \left[ F_s + \left( p - \frac{\epsilon^2_G}{2} \Phi^2 \right) I + \epsilon^2_G \Phi \xi \otimes \nabla \phi \right] : \nabla \mathbf{u} + \nabla \cdot \left( q_s - \frac{q_E}{T} - \frac{\epsilon^2_G}{T} \Phi \xi \cdot \frac{D\phi}{Dt} \right)$$

$$+ \left( q_E + \epsilon^2_G \Phi \xi \cdot \frac{D\phi}{Dt} \right) \cdot \nabla \left( \frac{1}{T} \right) + \frac{1}{T} \left( \epsilon^2_G \nabla \cdot (\Phi \xi) - \rho \frac{\partial g}{\partial \phi} \right) \frac{D\phi}{Dt} \geq 0, \tag{3.9}$$

if the relation $\epsilon^2_G = \epsilon^2_E + T\epsilon^2_S$ is implemented. We note that the entropy flux, i.e., the second term in Eq. (3.9), across the boundary being zero and other terms being positive guarantee that the entropy density production rate is nonnegative. To accomplish this, we assume the entropy flux to be zero and introduce two positive variables $\hat{k}$ and $M$ such that

$$qs - \frac{q_E}{T} - \frac{\epsilon^2_G}{T} \Phi \xi \cdot \frac{D\phi}{Dt} = 0, \tag{3.10a}$$

$$q_E + \epsilon^2_G \Phi \xi \cdot \frac{D\phi}{Dt} = \hat{k} \nabla \left( \frac{1}{T} \right), \tag{3.10b}$$

$$\epsilon^2_G \nabla \cdot (\Phi \xi) - \rho \frac{\partial g}{\partial \phi} = M \frac{D\phi}{Dt}, \tag{3.10c}$$

$$F_s + \left( p - \frac{\epsilon^2_G}{2} \Phi^2 \right) I + \epsilon^2_G \Phi \xi \otimes \nabla \phi = \tau. \tag{3.10d}$$

In the above equations, we choose $F_s$ such that Eq. (3.10d) can be satisfied, where $\tau$ is the shear stress. Therefore, the property $\tau : \nabla \mathbf{u} \geq 0$ for shear stress guarantees the first term in
Eq. (3.9) to be positive. In incompressible flow, \( \rho \) can be considered as a constant in each bulk phase. Eq. (3.5a) is then reduced to the divergence-free condition. We rearrange Eqs. (3.5) and make use of Eqs. (3.10) to cancel out \( \nabla \cdot q_E \) and \( \nabla \cdot q_S \), then express the governing equations can be expressed as follows:

\[
\begin{align*}
\nabla \cdot u &= 0, \\
\rho \frac{Du}{Dt} &= \nabla \cdot \left[ \tau + \left(-p + \frac{1}{2} \epsilon_G^2 \Phi^2 \right) I - \epsilon_G^2 \Phi \xi \otimes \nabla \phi \right] + b, \\
\rho \frac{De}{Dt} &= \nabla \cdot (k\nabla T) + \left[ \tau + \left(-p + \frac{1}{2} T \epsilon_S^2 \Phi^2 \right) I - T \epsilon_S^2 \Phi \xi \otimes \nabla \phi \right] : \nabla u \\
&\quad + \epsilon_E^2 \nabla \cdot (\Phi \xi) \frac{D\phi}{Dt} + \dot{\epsilon} , \\
M \frac{D\phi}{Dt} &= \epsilon_G^2 \nabla \cdot (\Phi \xi) - \rho \frac{\partial g}{\partial \phi} .
\end{align*}
\]  

Here \( I \) is the unit tensor; \( k \) is the thermal conductivity, and \( M \) is the mobility, i.e., inverse of the kinetic coefficient, where \( k \) and \( M \) are positive constants. If we assume that no heat source exists in the control volume, i.e., \( \dot{\epsilon} = 0 \), the unknowns in Eqs. (3.11) are \( \epsilon_E^2, \epsilon_S^2, \epsilon_G^2, M, \rho(\phi, T), \mu(\phi) \) and \( \partial g/\partial \phi \).

In order to determine the functions to represent the density and dynamic viscosity, we first introduce a function \( P(\phi) \) to denote volume fraction of liquid such that \( P(0) = 0 \) and \( P(1) = 1 \). Moreover, \( P(\phi) \) has the property to smooth the phase-field solutions on the interface. Since we are considering a density-change problem, density is no longer a constant. It is a function of phase field and temperature and takes on a constant value in each bulk phase. We apply the Boussinesq approximation to determine density as a function of temperature. Dependence of density on phase field has the property that \( \rho = \rho_S \) in solid when \( \phi = 0 \) and \( \rho = \rho_L \) in liquid when \( \phi = 1 \). In all terms of the governing equations except the buoyancy force term, density is considered to be a constant and denoted by a referenced density \( \rho_0 \). We can then express \( \rho(\phi, T) \) as

\[
\rho(\phi, T) = \left[ 1 - P(\phi) \right] \rho_S + P(\phi) \rho_L \left[ 1 + \beta(T - T_m) \right] ,
\]  

where \( \beta \) is the coefficient of thermal volumetric expansion, and \( T_m \) is the melting temperature. We note that in Eq. (3.12), \( P(\phi) \) is indeed volume fraction of liquid. In addition, we express the specific heat \( c_p \) as a function of \( \phi \) such that it remains constants in bulk phases and
varies over the interface. Thus a mass fraction of liquid is introduced as follows:

\[ w(\phi) = [1 - P(\phi)] P(\phi) \frac{\rho_s}{\rho_L} + P(\phi)^2. \]  

(3.13)

Therefore,

\[ c_p(\phi) = [1 - w(\phi)] c_{pS} + w(\phi) c_{pL}, \]  

(3.14)

where \( c_{pS} \) and \( c_{pL} \) are the specific heat in solid and liquid, respectively. Moreover, the dynamic viscosity can be expressed as

\[ \mu(\phi) = [1 - w(\phi)] \mu_S + w(\phi) \mu_L, \]  

(3.15)

where \( \mu_S \) and \( \mu_L \) are the dynamic viscosity in solid and liquid, respectively.

Furthermore, it is reasonable to make the assumption that the internal energy density does not depend on density; then we have

\[ e = e_S(T) + P(\phi)L(T), \]  

(3.16)

where \( L(T) = e_L(T) - e_S(T) \). We can also assume that the internal energy density in solid phase \( e_S(T) \) is a linear function of temperature. Hence,

\[ e = e_S(T_m) + c_{pS}(T - T_m) + P(\phi)L(T). \]  

(3.17)

Differentiation of Eq. (3.6b) with respect to \( T \) at a constant \( \phi \) leads to

\[ \frac{\partial [f/T]}{\partial T} = \frac{1}{T} \left( \frac{\partial e}{\partial T} \right)_{\phi} - \frac{e}{T^2} - \left( \frac{\partial s}{\partial T} \right)_{\phi}. \]  

Since \( e \) does not depend on \( \rho \) as indicated by Eq. (3.17), we obtain \( (\partial e/\partial s)_{\phi} = T \) from Eq. (3.7a). Moreover, \( (\partial e/\partial T)_{\phi} = (\partial e/\partial s)_{\phi} (\partial s/\partial T)_{\phi} = T (\partial s/\partial T)_{\phi} \) leads to cancellation of the first and third terms in the above equations. Therefore, the above formula can be rewritten as

\[ \frac{\partial [f/T]}{\partial T} = -\frac{e}{T^2}. \]  

(3.18)

Integration of Eq. (3.18) leads to

\[ f(T, \phi) = -T \int_{T_m}^{T} \frac{e(\xi, \phi)}{\xi^2} d\xi + T \Psi(\phi) \]

\[ = -T \int_{T_m}^{T} \frac{e_S(\xi)}{\xi^2} d\xi - TP(\phi)Q(T) + T \Psi(\phi), \]  

(3.19)
with the substitution of Eq. (3.16). Here we introduce $\Psi(\phi)$ as an undetermined function of $\phi$, and $Q(T) = \int_{T_m}^{T} [L(\xi)/\xi^2] d\xi$. Moreover, the assumption that $L(T) = L(T_m) = L_0$ is reasonable for $T$ in a neighborhood of the melting temperature and provides an explicit form for $Q(T)$ such that

$$Q(T) = -\frac{L(T_m) T_m - T}{T_m}.$$  \hfill (3.20)

The second procedure to determine the unknowns is to expand the Gibbs free energy with respect to a reference pressure and make use of properties of the “double-well” potential $f(T, \phi)$ to obtain the expressions for $P(\phi)$, $\Psi(\phi)$ and $\partial g/\partial \phi$. The Gibbs free energy $g(p, T, \phi)$ can be expanded with respect to a reference pressure as

$$g(p, T, \phi) = g_0(T, \phi) + \frac{\partial g}{\partial p} (p - p_0),$$  \hfill (3.21)

by substituting the relation in Eq. (3.6a). Here $p_0$ is the reference pressure, and $g_0(T, \phi) = f(T, \phi)$ at $p_0$. Hence, at equilibrium, with $T = T_m$ and $p = p_0$, the properties of $f$ give

$$f(T_m, 0) = f(T_m, 1),$$

$$\frac{\partial g}{\partial \phi} = \frac{\partial f}{\partial \phi} \bigg|_{\phi=0,1} = 0,$$

$$\frac{\partial^2 g}{\partial \phi^2} = \frac{\partial^2 f}{\partial \phi^2} \bigg|_{\phi=0,1} > 0.$$  

If we choose $\Psi(\phi) = \psi(\phi)/4a = \phi^2(1-\phi)^2/4a$ and $P(\phi) = \phi^2(6\phi^3 - 15\phi^2 + 10\phi)$, the above conditions can be satisfied, with $a$ a positive constant. In addition, differentiation of Eq. (3.21) with respect to $\phi$ leads to

$$\frac{\partial g}{\partial \phi} = \frac{\partial g_0}{\partial \phi} - \frac{p - p_0}{\rho^2} \frac{\partial \rho}{\partial \phi} = \frac{\partial f}{\partial \phi} - \frac{p - p_0}{\rho^2} \frac{\partial \rho}{\partial \phi}. \hfill (3.22)$$

We substitute the differential forms of density $\partial \rho/\partial \phi$ computed from Eq. (3.12) and $\partial f/\partial \phi$ from Eq. (3.19) into above equation to obtain $\partial g/\partial \phi$ such that

$$\frac{\partial g}{\partial \phi} = \frac{L_0(T_m - T)}{T_m} P'(\phi) + T \Psi'(\phi) - \frac{p - p_0}{\rho^2} \left[ (\rho_L - \rho_S) + \beta \rho_L(T - T_m) \right] P'(\phi).$$

Finally, the steady-state solution of $\phi$ in one dimension at equilibrium ($T = T_m$ and $p = p_0$) without a flow field is computed from Eq. (3.11d) as

$$\phi(x) = \frac{1}{2} \tanh \left( \sqrt{\frac{T_m p_0}{8}} \cdot \frac{x}{\sqrt{\sigma \epsilon G}} \right) + \frac{1}{2} = \frac{1}{2} \tanh \left( \frac{x}{\delta} \right) + \frac{1}{2}. \hfill (3.23)$$
Here we introduce $\delta = \sqrt{a\epsilon_G^2 / \rho_0 T_m}$ to denote the interfacial thickness. Furthermore, according to Cahn and Hilliard [68], surface tension for 1D can be written as

$$\sigma = \int_{-\infty}^{\infty} \left[ \Delta f + K \left( \frac{d\phi}{dx} \right)^2 \right] dx,$$

where $\Delta f$ is the free energy crossing the interface, and $K$ is a positive gradient coefficient. In our case, $K = \epsilon_G^2 / 2$. Therefore, $a$ is computed by differentiating Eq. (3.23) and substituting it into Eq. (3.24); then we have

$$a = \frac{\sqrt{2}\rho_0 T_m \delta}{12\sigma}.$$

Since $\epsilon_G$, $\epsilon_E$ and $\epsilon_S$ are related to each other by $\epsilon_G^2 = \epsilon_E^2 + T \epsilon_S^2$, it is reasonable to assume that $\epsilon_E^2 = \epsilon_G^2 / 2$ and $\epsilon_S^2 = \epsilon_G^2 / 2T$. Therefore, the gradient coefficient $\epsilon^2$ can be expressed as

$$\epsilon^2 = \epsilon_G^2 = 6\sqrt{2}\sigma\delta.$$

Mobility $M$ is determined by approximating the phase-field equation (3.11d) by the limiting modified Stefan model with $M \to 0$, $\delta \to 0$, and $M/\sigma\delta$ being fixed as discussed in [8]. We then compare it with dimensional form of interface temperature

$$T = T_m - \gamma \kappa - \frac{v_n}{\mu_k},$$

where $\gamma = \sigma T_m / \rho L_0$ is the interfacial energy; $\kappa$ is the curvature; $v_n$ is the interfacial speed in the normal direction, and $\mu_k$ is the kinetic coefficient. Hence, we have

$$M = \frac{6\sqrt{2}\rho_0 L_0 \delta}{T_m \mu_k}.$$

Therefore, the governing equations can be written as

$$\rho \frac{Du}{Dt} = -\nabla p + \nabla \cdot \tau + b + X(\phi),$$

$$M \frac{D\phi}{Dt} = \epsilon^2 \nabla \cdot (\Phi \xi) + \frac{\rho_0 L_0 P'(\phi)}{T_m} (T - T_m) - \frac{\rho_0 \psi'(\phi)}{4a} T + Y(\phi, T),$$

$$\rho_0 c_p \frac{DT}{Dt} = \nabla \cdot (k \nabla T) - \left[ \rho_0 L_0 P'(\phi) - \frac{1}{2} \epsilon^2 \Delta \phi \right] \frac{D\phi}{Dt} + W(\mathbf{u}, \phi).$$

Here, source terms $X(\phi)$, $Y(\phi, T)$ and $W(\mathbf{u}, \phi)$ are defined as follows:

$$X(\phi) = \nabla \cdot \left( \frac{\epsilon^2}{2} \Phi^2 I - \epsilon^2 \Phi \xi \otimes \nabla \phi \right),$$

$$Y(\phi, T) = \frac{30(p - p_0)}{\rho_0} \psi(\phi) [\rho_L - \rho_S + \beta \rho_L (T - T_m)],$$

$$W(\mathbf{u}, \phi) = \left[ \tau + \left( -p + \frac{\epsilon^2}{4} \Phi^2 \right) I - \frac{\epsilon^2}{2} \Phi \xi \otimes \nabla \phi \right] : \nabla \mathbf{u},$$
with

\[ P(\phi) = \phi^2(6\phi^3 - 15\phi^2 + 10\phi), \quad (3.28a) \]
\[ \psi(\phi) = \phi^2(1 - \phi)^2. \quad (3.28b) \]

The forcing term \( X(\phi) \) in Eq. (3.26a) can be interpreted in a similar way as the forcing term in the momentum equation introduced in the immersed boundary (IB) method. When fluid flows over a body, it exerts a normal force (pressure) on the surface and; if the surface is no-slip, it also exerts a shear force. Conversely, the surface exerts a (localized) force of opposite sign on the fluid. In the moving-boundary-problem case, an additional normal force exerted on the fluid comes from the propagation of the moving boundary, i.e., the solid-liquid interface in current research. Therefore, the effect of certain boundary conditions can be modeled with an external force field as in the immersed boundary method, introduced by Peskin and McQueen [70] [71] [72] and later applied to different problems by Goldstein et al [73], Calhoun and Leveque [74], Balaras [75], etc, rather than with a specification of boundary parameter values.

In the immersed boundary method, the forcing term in the momentum equations is obtained by the force exerted by the body on the fluid multiplying a Dirac \( \delta \)-function. We note that the force on solid-liquid interface in the case of solidification is actually the surface tension \( \sigma \). In addition, the expression of \( X(\phi) \) in Eq. (3.27a) can be rewritten as

\[ X(\phi) = \epsilon^2 \nabla \cdot \left( \frac{\Phi^2 I}{2} - \Phi \xi \otimes \nabla \phi \right), \quad (3.29) \]

where the surface tension is contained in \( \epsilon^2 \) as shown in Eq. (3.25), and the remaining part of \( X(\phi) \) is a \( \delta \)-function. Therefore, the source term \( X(\phi) \) in the momentum equation has the same role as the forcing term in the immersed boundary method. Moreover, modeling the governing equations (3.26) containing the forcing term \( X(\phi) \) is equivalent to solve the flow over a solid body with moving boundary.
3.2 2-D Phase-Field Model in Flow Field

Since we are currently solving 2-D solidification problems in a flow field, we rewrite the phase-field model (3.26) as follows:

\[
\begin{align*}
\frac{\partial u}{\partial t} + (uu)_x + (uv)_y &= -\frac{p_x}{\rho_0} + \frac{1}{\rho_0} \left[ 2(\mu u_x)_x + (\mu u_y)_y + (\mu v)_x \right] + X_1(\phi), \\
\frac{\partial v}{\partial t} + (uv)_x + (vv)_y &= -\frac{p_y}{\rho_0} + \frac{1}{\rho_0} \left[ (\mu v_x)_x + 2(\mu v_y)_y + (\mu u)_x \right] + X_2(\phi) + B(\phi, T), \\
\frac{\partial \phi}{\partial t} + (u\phi)_x + (v\phi)_y &= \frac{\varepsilon^2}{M} N(\phi) + \frac{\rho_0 L_0 p(\phi)'}{M T_m} (T - T_m) - \frac{\rho_0 \psi'(\phi) T}{4aM} + \frac{Y(\phi, T)}{M}, \\
\frac{\partial T}{\partial t} + (uT)_x + (vT)_y &= \alpha \Delta T + \left[ \frac{\varepsilon^2 N(\phi)}{2\rho_0 c_p} - \frac{30L_0 \psi(\phi)}{c_p} \right] \frac{\partial \phi}{\partial t} + \frac{W(u, v, \phi)}{\rho_0 c_p}.
\end{align*}
\]

Here, \( u \) and \( v \) are velocity components in \( x \) and \( y \) directions, respectively; \( \phi \) and \( T \) are phase field and temperature. Subscripts \( t \), \( x \) and \( y \) denote partial differentiation with respect to time, \( x \) direction and \( y \) direction, respectively, and \( \alpha = k/\rho_0 c_p \) is the thermal diffusivity with the thermal conductivity \( k \) a constant. In the above equations, we denote

\[
N(\phi) = \xi^2(\phi_{xx} + \xi^2(\phi_{yy} + (2\xi_1 \xi_{1x} + \xi_2 \xi_{1y} + \xi_1 \xi_{2y}) \phi_x \\
+ (\xi_1 \xi_{2x} + \xi_2 \xi_{2x} + 2\xi_2 \xi_{2y}) \phi_y + 2\xi_1 \xi_{2y} \phi_{xy}).
\]

\[
X_1(\phi) = -\frac{\varepsilon^2}{\rho_0} \left( \xi^2(\phi_{xx} + 2\xi_1 \xi_{2y} \phi_x + \xi^2(\phi_{yy}) \right),
\]

\[
X_2(\phi) = -\frac{\varepsilon^2}{\rho_0} \left( \xi^2(\phi_{xx} + 2\xi_1 \xi_{2y} \phi_x + \xi^2(\phi_{yy}) \right),
\]

\[
B(\phi, T) = -\frac{\rho(\phi, T) - \rho_L}{\rho_0} g,
\]

\[
Y(\phi, T) = \frac{30(p - p_0)}{\rho_0} \psi(\phi) \left[ \rho_L - \rho_S + \beta \rho_L (T - T_m) \right],
\]

\[
W(u, v, \phi) = \mu(\phi) \left[ 2u_x^2 + (u_y + v_x)^2 + 2v_y^2 \right] + \frac{\varepsilon^2}{4} \left( \xi^2(\phi_x^2 - \xi^2(\phi_y)^2) \right) (v_y - u_x) \\
- \frac{\varepsilon^2}{2} \left[ v_x \left( \xi_1 \xi_{2y} \phi_x + \xi^2(\phi_x \phi_y) \right) + u_y \left( \xi^2(\phi_x \phi_y + \xi_1 \xi_{2y} \phi_y) \right) \right].
\]

As discussed in Section 3.1, \( \mu(\phi), \rho(\phi, T) \) and \( c_p(\phi) \) are no longer constants and are expressed by Eqs. (3.15), (3.12) and (3.14), respectively. The \( \xi \)-vector introduced in Section 3.1 can be written in component form \( \xi = (\xi_1, \xi_2) \) in 2D, where \( \xi_1 \) and \( \xi_2 \) are components in \( x \) and \( y \) directions, respectively. Notations \( \xi_{1x} \) and \( \xi_{1y} \) in Eq. (3.31a) denote derivatives of \( \xi_1 \) with
respect to x and y, so are \( \zeta_{2x} \) and \( \zeta_{2y} \). Vector \( \xi \) is chosen to be of the form

\[
\xi = 1 + \epsilon' \cos m(\theta - \alpha_0),
\]

where \( \epsilon' \) and \( m \) are the degree of anisotropy and number of folds, respectively. Notation \( \theta \) is the angle between the normal to the front and the x axis, and it is determined by the gradient of phase field as follows:

\[
\cos \theta = \frac{\phi_x}{|\nabla \phi|} = \frac{\phi_x}{\sqrt{\phi_x^2 + \phi_y^2}}, \quad \sin \theta = \frac{\phi_y}{|\nabla \phi|} = \frac{\phi_y}{\sqrt{\phi_x^2 + \phi_y^2}}.
\]

Here, phase angle \( \alpha_0 \) represents the anti-clockwise rotating angle of a main branch from the positive x direction, and \( \alpha_0 = 0 \) indicates that one main branch lies in positive x axis. Moreover, components of \( \xi \) are given by \( \xi_1 = \xi \cdot \cos \theta \) and \( \xi_2 = \xi \cdot \sin \theta \). Eq. (3.32) indicates that the anisotropy parameter in the direction such that \( \cos m(\theta - \alpha_0) = 1 \) is \( \epsilon' \) times larger than in other directions. For example, if \( m = 4 \) and \( \alpha_0 = 0 \), the main branches grow in the directions of \( \theta = 0, \pi/2, \pi \) and \( 3\pi/2 \), and their anisotropy parameters \( \xi \) in the four directions are \( \epsilon' \) times larger than in other directions.

In addition, three parameters of the phase-field model are defined by

\[
\epsilon^2 = 6\sqrt{2}\sigma\delta, \quad M = \frac{6\sqrt{2}\rho_0 L_0 \delta}{T_m \mu_k}, \quad a = \frac{\rho_0 T_m \delta}{6\sqrt{2}\sigma}.
\]

Here, the physical meanings of \( \epsilon^2 \), \( M \) and \( a \) were provided in Section 3.1.

### 3.3 Initial and Boundary Conditions

Freezing in a lid-driven-cavity flow will be simulated in the current work. Therefore, for a square domain \( \Omega \equiv [0, L] \times [0, L] \) with length \( L \) on each side, the prescribed boundary and
Initial conditions for the lid-driven-cavity flow are

\textit{Initial conditions}

\begin{equation}
\begin{align*}
u = v = p & = 0 \quad \text{in} \quad \Omega, \\
\end{align*}
\end{equation}

\textit{Boundary conditions}

\begin{equation}
\begin{align*}
u = 0 & \quad \text{on} \quad \partial\Omega \setminus \{ y = L \}, \\
u = U & \quad \text{on} \quad \{ y = L \}, \\
v = 0 & \quad \text{on} \quad \partial\Omega, \\
\frac{\partial p}{\partial n} & = 0 \quad \text{on} \quad \partial\Omega. \\
\end{align*}
\end{equation}

Here, \( U \) provides the Reynolds number as \( Re = \frac{UL}{\nu_L} \), and \( \nu_L \) is the kinematic viscosity of liquid. In addition, the prescribed boundary and initial conditions for freezing are

\textit{Initial conditions}

\begin{equation}
\begin{align*}
\phi & = 0 \quad \text{in} \quad \Omega_0, \\
\phi & = 1 \quad \text{in} \quad \Omega \setminus \Omega_0, \\
T & = T_m \quad \text{in} \quad \Omega_0, \\
T & = T_{init} < T_m \quad \text{in} \quad \Omega \setminus \Omega_0. \\
\end{align*}
\end{equation}

\textit{Boundary conditions}

\begin{equation}
\begin{align*}
\frac{\partial \phi}{\partial n} = \frac{\partial T}{\partial n} & = 0 \quad \text{on} \quad \partial\Omega. \\
\end{align*}
\end{equation}

In the above conditions, \( \Omega_0 \) is the domain of a small seed which indicates the onset of freezing.

### 3.4 Numerical Methods and Algorithms

In this section, we first describe in detail the implicit numerical procedures employed in this study. We then provide an outline of the derivations leading to the discrete form of the 2-D coupled nonlinear reaction-diffusion equations (3.30), followed by the pseudolanguage algorithm to perform one time step. But we first briefly note the types of algorithms used by investigators in previous studies. Wheeler et al. [22] employed an alternating-direction implicit method (ADI) for the energy equation and an explicit time-differencing
scheme for the phase-field equation. Both equations are discretized spatially using second-order finite differences on a uniform grid. In Karma and Rappel’s work [25], governing equations are discretized using second-order finite differences, except that the Laplacian of $\phi$ is approximated using a nine-point formula with nearest and next nearest neighbors to reduce effects of grid anisotropy. The time differencing is first-order Euler and second-order implicit Crank–Nicolson scheme for phase-field and energy equations, respectively. Beckermann et al. [44] solved the phase-field and energy equations using the explicit method of Karma and Rappel in [24].

### 3.4.1 Equations of Motion

We discretize the governing equations using generalized trapezoidal integration in time and centered differencing in space, leading to second-order accuracy in space and first-order accuracy in time for backward Euler method or second-order accuracy in time for Crank–Nicolson approach. In addition, we implement a $\delta$-form Douglas & Gunn time-splitting method [76] in the current work. This method is guaranteed to maintain second-order accuracy and stability of the unsplit scheme, and it is very efficient, especially in $\delta$-form. Furthermore, it can be extended to 3-D problems in a straightforward way.

We implement a projection method first suggested by Chorin [77] (but in a form analyzed by Gresho [78]) to solve the equations of motion (3.30a–3.30c). In this method, we assume to start from time level $n$ and construct fractional steps such that in the first step we ignore the pressure gradient terms and solve a system of Burgers’ equations:

\[
\begin{align*}
\hat{u}_t + (\hat{u}^2)_x + (\hat{v}\hat{u})_y &= \frac{1}{\rho_0} \left[ 2(\mu\hat{u}_x)_x + (\mu\hat{u}_y)_y + (\mu\hat{v}_x)_y \right] + X_1(\phi), \\
\hat{v}_t + (\hat{u}\hat{v})_x + (\hat{v}^2)_y &= \frac{1}{\rho_0} \left[ (\mu\hat{v}_x)_x + 2(\mu\hat{v}_y)_y + (\mu\hat{v}_y)_x \right] + X_2(\phi) + \hat{B}(\phi,T),
\end{align*}
\]

(3.35a, 3.35b)

and in the second step we solve

\[
\frac{\partial \mathbf{u}}{\partial t} = -\frac{\nabla p}{\rho_0},
\]

(3.36)

The discretization of the above equation leads to

\[
\Delta p = \frac{\rho_0 \nabla \cdot \mathbf{u}}{\Delta t},
\]

(3.37)
if we take the divergence of Eq. (3.36) and apply the divergence-free condition (3.30a) at the end of each time step, i.e., \( \nabla \cdot \mathbf{u}^{n+1} = 0 \). Here, \( \hat{u} \) and \( \hat{v} \) denote the velocity components obtained from the Burgers’ equation in \( x \) and \( y \) direction, respectively, and \( \Delta t \) is the time step size. Finally, we apply Leray projection to obtain

\[
\mathbf{u}^{n+1} = \hat{\mathbf{u}} - \frac{\Delta t \nabla p}{\rho_0},
\]

where \( \mathbf{u}^{n+1} \) is the divergence-free velocity at time level \( n + 1 \). This completes a time step.

We note that \( p \) here is not true pressure; hence, we use the notation \( \varphi \) instead and solve the pseudo pressure Poisson equation (PPE) given as

\[
\Delta \varphi = \frac{\rho_0 \nabla \cdot \hat{\mathbf{u}}}{\Delta t},
\]

with boundary condition \( \partial \varphi / \partial n = 0 \) on \( \partial \Omega \). Therefore, Eq. (3.38) becomes

\[
\mathbf{u}^{n+1} = \hat{\mathbf{u}} - \frac{\Delta t \nabla \varphi}{\rho_0}.
\]

In some cases, if more accurate pressure is needed, e.g., in tracking free surfaces, it is probably necessary to solve the true PPE at the end of each time step after getting a mass conserved velocity field as follows:

\[
\Delta p = -\rho_0 \nabla \cdot (\nabla \cdot (\mathbf{u}^2)) ,
\]

with boundary condition \( \partial p / \partial n = \mu \Delta u - \rho_0 \nabla \cdot (\nabla u^2) - \rho_0 \partial u / \partial t \). Implementation of the projection method can be summarized as a pseudo-language algorithm presented below.

**Algorithm 1** Suppose \( n \) time steps have been computed, and \( u^n, v^n, \varphi^n \) have been obtained using Eqs. (3.35), (3.39) and (3.40). To advance the numerical solution to time level \( n + 1 \), carry out the following steps.

1. Solve the “Burgers’ equation” form of momentum equations (3.35) using \( \delta \)-form Douglas–Gunn time splitting to obtain \( \hat{u}^{n+1} \) and \( \hat{v}^{n+1} \);

2. Filter Burgers’ equation solution \( \hat{u}^{n+1} \) and \( \hat{v}^{n+1} \) to obtain \( \hat{u}^{n+1} \) and \( \hat{v}^{n+1} \);

3. Solve PPE (3.39) for pseudo pressure \( \varphi^{n+1} \);
4. Project $\tilde{u}^{n+1}$, $\tilde{v}^{n+1}$ to obtained mass-conserved $u^{n+1}$ and $v^{n+1}$ using Eq. (3.40);

5. Solve true PPE (3.41) if and when needed.

In step 2 of Algorithm 1, we employ a Shuman filter [79] to mollify the solutions since it works very well to the Burgers’ equation [80]. For simplification, we extend the 1-D Shuman filter to 2D under the assumption of uniform grids, i.e., $h_x = h_y$. Hence, solutions obtained from the filter are

\[
\tilde{u}_{i,j} = \frac{\hat{u}_{i+1,j} + \hat{u}_{i-1,j} + \beta_u \hat{u}_{i,j} + \hat{u}_{i,j+1} + \hat{u}_{i,j-1}}{4 + \beta_u}, \tag{3.42a}
\]
\[
\tilde{v}_{i,j} = \frac{\hat{v}_{i+1,j} + \hat{v}_{i-1,j} + \beta_v \hat{v}_{i,j} + \hat{v}_{i,j+1} + \hat{v}_{i,j-1}}{4 + \beta_v}, \tag{3.42b}
\]

where $\beta_u$ and $\beta_v$ are the filter parameters for $u$ and $v$.

We now describe implementation of Douglas–Gunn time splitting applied to Eqs. (3.35). Here we suppress notation “^” for convenience. The first step of Douglas–Gunn procedure is discretization. We use centered differencing for all spatial derivatives and backward Euler or trapezoidal integration in time. Therefore, a variable $\theta$ is introduced such that $\theta = 0$, $0.5$ and $1$ represents the explicit (forward Euler) scheme, trapezoidal integration, and backward Euler in time, respectively. We denote the nonlinear terms in Eqs. (3.35) by $N_1$ and $N_2$ such that

\[
N_1(u, u_x, u_y) = \left(u^2\right)_x + (uv)_y, \tag{3.43a}
\]
\[
N_2(v, v_x, v_y) = (uv)_x + \left(v^2\right)_y. \tag{3.43b}
\]

We then apply \(\delta\)-form quasilinearization to $N_1$ and $N_2$ by expanding nonlinear terms in Fréchet–Taylor series and neglecting higher-order terms as follows:

\[
N_1(u, u_x, u_y) = N_1^{(m)} + \left(\frac{\partial N_1}{\partial u}\right)^{(m)} (\delta u) + \left(\frac{\partial N_1}{\partial u_x}\right)^{(m)} (\delta u)_x + \left(\frac{\partial N_1}{\partial u_y}\right)^{(m)} (\delta u)_y = \left(u^{(m)}\right)_x + \left(u^{(m)}v^{(m)}\right)_y + 2 \left(u^{(m)}\delta u\right)_x + \left(v^{(m)}\delta u\right)_y, \tag{3.44a}
\]
\[
N_2(v, v_x, v_y) = N_2^{(m)} + \left(\frac{\partial N_2}{\partial v}\right)^{(m)} (\delta v) + \left(\frac{\partial N_2}{\partial v_x}\right)^{(m)} (\delta v)_x + \left(\frac{\partial N_2}{\partial v_y}\right)^{(m)} (\delta v)_y = \left(u^{(m)}v^{(m)}\right)_x + \left(v^{(m)}\right)_y + 2 \left(u^{(m)}\delta v\right)_x + \left(v^{(m)}\delta v\right)_y, \tag{3.44b}
\]

46
where superscripts \((m)\) denote the value at the \(m\)th nonlinear iteration. Applying generalized trapezoidal integration to Eqs. (3.35), and replacing the advanced time level values \(u^{n+1}, \ v^{n+1}\) with \(u^{(m)} + \delta u\) and \(v^{(m)} + \delta v\), respectively, yield

\[
\begin{align*}
\{ I - \frac{\theta \Delta t}{\rho_0} \left\{ 2 \left[ D_{0,x} \left( \mu^{(m)} D_{0,x} \right) \right] + \left[ D_{0,y} \left( \mu^{(m)} D_{0,y} \right) \right] \right\} + \theta \Delta t \left[ 2D_{\mp,x} \left( u^{(m)} \right) + \\
D_{\pm,y} \left( v^{(m)} \right) \right] \} \delta u &= u^n - u^{(m)} + \frac{\theta \Delta t}{\rho_0} \left[ 2 \left( \mu^{(m)} u_x^{(m)} \right)_x + \left( \mu^{(m)} u_y^{(m)} \right)_y \right] \\
&+ \frac{(1 - \theta) \Delta t}{\rho_0} \left[ 2 \left( \mu^n u_x^n \right)_x + \left( \mu^n u_y^n \right)_y \right] - \theta \Delta t \left[ (u^{(m)})^2 \right]_x + (u^{(m)})^2 \right]_y \\
&- (1 - \theta) \Delta t \left[ (u^{n+1})_x + (u^n v^n)_y \right] + \theta \Delta t X_1 \left( \phi^{(m)} \right) + \left( (1 - \theta) \Delta t X_1 \left( \phi^n \right) \right),
\end{align*}
\tag{3.45a}
\]

\[
\begin{align*}
\{ I - \frac{\theta \Delta t}{\rho_0} \left\{ \left[ D_{0,x} \left( \mu^{(m)} D_{0,x} \right) \right] + 2 \left[ D_{0,y} \left( \mu^{(m)} D_{0,y} \right) \right] \right\} + \theta \Delta t \left[ D_{\mp,x} \left( u^{(m)} \right) + \\
2D_{\mp,y} \left( v^{(m)} \right) \right] \} \delta v &= v^n - v^{(m)} + \frac{\theta \Delta t}{\rho_0} \left[ \left( \mu^{(m)} v_x^{(m)} \right)_x + 2 \left( \mu^{(m)} v_y^{(m)} \right)_y \right] \\
&+ \frac{(1 - \theta) \Delta t}{\rho_0} \left[ (\mu^n v_x^n)_x + 2 (\mu^n v_y^n)_y \right] - \theta \Delta t \left[ (u^{(m)})^2 \right]_x + (v^{(m)})^2 \right]_y \\
&- (1 - \theta) \Delta t \left[ (u^n v^n)_x + (v^{n+1})_y \right] + \theta \Delta t \left[ X_2 \left( \phi^{(m)} \right) + B \left( \phi^{(m)}, T^{(m)} \right) \right] \\
&+ \left( (1 - \theta) \Delta t \right) X_2 \left( \phi^n \right) + B \left( \phi^n, T^n \right) \right).
\end{align*}
\tag{3.45b}
\]

In the above equations, \(D_{0,x}\) and \(D_{0,y}\) are second-order centered-difference operators with respect to \(x\) and \(y\), respectively. \(D_{\mp,x}\) and \(D_{\mp,y}\) are first-order upwinding operators introduced to mitigate the cell-\(Re\) problem. The first-order upwinding operator in the \(x\) direction applied to \(u\) is given as follows:

\[
D_{\mp,x} u = \left\{ \begin{array}{ll}
\frac{u_{i+1,j} - u_{i,j}}{h_x}, & \text{if } u_{i,j} \geq 0, \\
\frac{u_{i,j} - u_{i-1,j}}{h_x}, & \text{if } u_{i,j} < 0,
\end{array} \right.
\tag{3.46}
\]

where \(h_x\) is the spatial step size in the \(x\) direction. Since a staggered grid is implemented, superscripts \(-\) and \(\sim\) are introduced to denote averaged values in \(x\) and \(y\) directions, respectively; \(\bar{u}, \bar{u}, \bar{v}\) and \(\bar{v}\) denote velocities on a (natural) grid point or cell center calculated.
Figure 3.1 Sketch of staggered grid for $u$, $v$ and $p$.

as follows:

$$
\bar{u}_{i,j} = \frac{u_{i,j} + u_{i-1,j}}{2}, \quad \bar{v}_{i,j} = \frac{v_{i,j} + v_{i+1,j}}{2}
$$

These are shown in Figure 3.1.

In the second step of the Douglas-Gunn procedure, we decompose Eq. (3.45) into a two-level split form

$$
(I + A_{1u}) \delta u_1 = S_u^m + X_u^n, \quad (3.47a)
$$

$$
(I + A_{1v}) \delta v_1 = S_v^m + X_v^n, \quad (3.47b)
$$

for the first time-split step, and

$$
(I + A_{2u}) \delta u_2 = \delta u_1, \quad (3.48a)
$$

$$
(I + A_{2v}) \delta v_2 = \delta v_1, \quad (3.48b)
$$

for the second. Here

$$
A_{1u} = -2 \theta \Delta t \left\{ \frac{[D_{0,x} (\mu^{(m)} D_{0,x'})]}{\rho_0} - D_{x,x} (u^{(m)},) \right\},
$$

$$
A_{2u} = -\theta \Delta t \left\{ \frac{[D_{0,y} (\mu^{(m)} D_{0,y'})]}{\rho_0} - D_{x,y} (v^{(m)},) \right\},
$$

48
To advance the numerical solution to time level approximations to Burgers’ equation form of momentum equations, i.e., Eqs. (3.35) and (3.36), carry out the following steps.

1. Set initial values for $u^{(m)}$ and $v^{(m)}$ to $u^n$, $v^n$, respectively;

2. Set quasilinear iteration counter $m = 0$;

3. Solve the first split step, Eqs. (3.47) for $u$ and $v$ to obtain $\delta u_1$ and $\delta v_1$;

4. Solve the second split step, Eqs. (3.48) for $u$ and $v$ to obtain $\delta u_2$ and $\delta v_2$;

5. Update solutions:
   - $u^{(m+1)} = u^{(m)} + \delta u_2$,
   - $v^{(m+1)} = v^{(m)} + \delta v_2$;
6. Compute iteration error for \( u^{(m+1)} \) and \( v^{(m+1)} \):

\[
\text{If error} < \text{convergence tolerance} \ 10^{-10} \\
\quad \text{go to 7}
\]

\[
\text{else if } m > \text{maximum iteration number} \ 300 \\
\quad \text{print} \quad \text{“quasilinear iteration for Burgers’ equations does not converge,” exit}
\]

\[
\text{else } m = m + 1, \quad \text{go to 3}
\]

7. Set \( n = n + 1 \); go to Step 2 in Algorithm 1.

We remark here that splitting errors that are often of concern for linear problems are automatically removed by the nonlinear iterations of this algorithm because it is convergence of the unsplit equations that is tested.

### 3.4.2 Phase-Field Model

In this subsection, we discretize the phase-field and energy equations, i.e., Eqs. (3.30d) and (3.30e), using generalized trapezoidal integration in time and centered differencing in space. Similar to the procedure of Section 3.4.1, we apply a \( \delta \)-form Douglas & Gunn time-splitting scheme.

We first present the detailed \( \delta \)-form quasilinearization of nonlinear terms of the governing equations. We note that the energy equation is linear in \( T \); thus we perform the quasilinearization only on Eq. (3.30d). We denote the nonlinear terms in Eq. (3.30d) by \( N \) and \( N_1 \) such that \( N \) contains derivatives of \( \phi \) as defined by Eq. (3.31a), and \( N_1 \) contains functions of \( \phi \). We express \( N_1 \) as

\[
N_1(\phi, T) = \frac{\rho_0 L_0 (T - T_m)}{MT_m} P'(\phi) - \frac{\rho_0 T}{4aM} \psi'(\phi) + \frac{30(p - p_0)}{M\rho_0} [\rho_L - \rho_S + \beta \rho_L (T - T_m)] \psi(\phi) .
\]

We then expand \( N \) and \( N_1 \) in Fréchet–Taylor series by neglecting higher-order terms as
follows:

\[ N(\phi_x, \phi_y, \phi_{xx}, \phi_{yy}) = N(\phi^{(m)}) + \left( \frac{\partial N}{\partial \phi_x} \right)^{(m)} (\phi_x - \phi^{(m)}) + \left( \frac{\partial N}{\partial \phi_y} \right)^{(m)} (\phi_y - \phi^{(m)}) + \left( \frac{\partial N}{\partial \phi_{xx}} \right)^{(m)} (\phi_{xx} - \phi^{(m)}) + \left( \frac{\partial N}{\partial \phi_{yy}} \right)^{(m)} (\phi_{yy} - \phi^{(m)}) \],
\[ \text{(3.50a)} \]

\[ N_1(\phi, T) = N_1(\phi^{(m)}, T^{(m)}) + \left( \frac{\partial N_1}{\partial \phi} \right)^{(m)} (\delta \phi), \]
\[ \text{(3.50b)} \]

where notation is consistent with that in Section 3.4.1. In the above equations, the corresponding terms are calculated as follows:

\[ \frac{\partial N}{\partial \phi_x} = \left( 2\xi_{1x} + \xi_{2y} \right) \frac{\partial \xi_1}{\partial \phi_x} + \left( 2\frac{\partial \xi_{1x}}{\partial \phi_x} + \frac{\partial \xi_{2y}}{\partial \phi_x} \right) \xi_1 + \xi_{1y} \frac{\partial \xi_2}{\partial \phi_x} + \frac{\partial \xi_{1y}}{\partial \phi_x} \xi_2 \phi_x + 2\xi_1 \xi_{1x} + \xi_2 \xi_{1y} \]
\[ + \xi_1 \xi_{2y} + \left( \frac{\partial \xi_1}{\partial \phi_x} \xi_{2x} + \xi_1 \frac{\partial \xi_{2x}}{\partial \phi_x} + (\xi_{1x} + 2\xi_{2y}) \frac{\partial \xi_2}{\partial \phi_x} + \left( \frac{\partial \xi_{1x}}{\partial \phi_x} + 2\frac{\partial \xi_{2y}}{\partial \phi_x} \right) \xi_2 \right) \phi_y \]
\[ + 2\xi_1 \frac{\partial \xi_1}{\partial \phi_x} \phi_{xx} + 2\xi_2 \frac{\partial \xi_2}{\partial \phi_x} \phi_{yy} + 2 \left( \frac{\partial \xi_1}{\partial \phi_x} \xi_2 + \xi_1 \frac{\partial \xi_2}{\partial \phi_x} \right) \phi_{xy}, \]
\[ \text{(3.51a)} \]

\[ \frac{\partial N}{\partial \phi_y} = \left( 2\xi_{1x} + \xi_{2y} \right) \frac{\partial \xi_1}{\partial \phi_y} + \left( 2\frac{\partial \xi_{1x}}{\partial \phi_y} + \frac{\partial \xi_{2y}}{\partial \phi_y} \right) \xi_1 + \xi_{1y} \frac{\partial \xi_2}{\partial \phi_y} + \frac{\partial \xi_{1y}}{\partial \phi_y} \xi_2 \phi_x + \xi_1 \xi_{2x} + \xi_2 \xi_{1y} \]
\[ + 2\xi_2 \xi_{2y} + \left( \frac{\partial \xi_1}{\partial \phi_y} \xi_{2x} + \xi_1 \frac{\partial \xi_{2x}}{\partial \phi_y} + (\xi_{1x} + 2\xi_{2y}) \frac{\partial \xi_2}{\partial \phi_y} + \left( \frac{\partial \xi_{1x}}{\partial \phi_y} + 2\frac{\partial \xi_{2y}}{\partial \phi_y} \right) \xi_2 \right) \phi_y \]
\[ + 2\xi_1 \frac{\partial \xi_1}{\partial \phi_y} \phi_{xx} + 2\xi_2 \frac{\partial \xi_2}{\partial \phi_y} \phi_{yy} + 2 \left( \frac{\partial \xi_1}{\partial \phi_y} \xi_2 + \xi_1 \frac{\partial \xi_2}{\partial \phi_y} \right) \phi_{xy}, \]
\[ \text{(3.51b)} \]

\[ \frac{\partial N}{\partial \phi_{xx}} = \xi_1^2 + 2\xi_1 \frac{\partial \xi_{1x}}{\partial \phi_{xx}} \phi_x + \left( \xi_1 \frac{\partial \xi_{2x}}{\partial \phi_{xx}} + \xi_2 \frac{\partial \xi_{1x}}{\partial \phi_{xx}} \right) \phi_y, \]
\[ \text{(3.51c)} \]

\[ \frac{\partial N}{\partial \phi_{yy}} = \xi_2^2 + \left( \xi_2 \frac{\partial \xi_{1y}}{\partial \phi_{yy}} + \xi_1 \frac{\partial \xi_{2y}}{\partial \phi_{yy}} \right) \phi_x + 2\xi_2 \frac{\partial \xi_{2y}}{\partial \phi_{yy}} \phi_y, \]
\[ \text{(3.51d)} \]

\[ \frac{\partial N_1}{\partial \phi} = 30 \left\{ \frac{\rho_0 L_0 (T - T_m)}{MT_m} + \frac{(p - p_0)}{M \rho_0} \left[ \rho_L - \rho_S + \beta \rho_L (T - T_m) \right] \right\} \psi'(\phi) - \frac{\rho_0 T}{4aM} \psi''(\phi). \]
\[ \text{(3.51e)} \]
the advanced time level values $\phi^{n+1}, T^{n+1}$ with $\phi^{(m)} + \delta \phi$ and $T^n + \delta T$, respectively, yield

\[
\left\{ I - \frac{\theta \Delta t e^2}{M} \left[ \left( \frac{\partial N}{\partial \phi_x} \right)^{(m)} D_{0,x} + \left( \frac{\partial N}{\partial \phi_y} \right)^{(m)} D_{0,y} + \left( \frac{\partial N}{\partial \phi_{xx}} \right)^{(m)} D_{0,x}^2 + \left( \frac{\partial N}{\partial \phi_{yy}} \right)^{(m)} D_{0,y}^2 \right] \right. \\
\left. + \theta \Delta t \left[ D_{x,x} \left( u^{(m)} \right) + D_{x,y} \left( v^{(m)} \right) \right] - \theta \Delta t \left( \frac{\partial N_1}{\partial \phi} \right)^{(m)} \right\} \delta \phi = \phi^n - \phi^{(m)} \\
+ \theta \Delta t \left[ \frac{\epsilon^2}{M} N \left( \phi^{(m)} \right) - D_{0,x} \left( u^{(m)} \phi^{(m)} \right) - D_{0,y} \left( v^{(m)} \phi^{(m)} \right) + N_1 \left( \phi^{(m)}, T^{(m)} \right) \right] \\
+ (1 - \theta) \Delta t \left[ \frac{\epsilon^2}{M} \left( \phi^n \right) - D_{0,x} \left( u^n \phi^n \right) - D_{0,y} \left( v^n \phi^n \right) + N_1 \left( \phi^n, T^n \right) \right],
\tag{3.52a}
\]

\[
\left\{ I - \frac{\theta \Delta t k}{\rho_0 c_p} \left( D_{0,x}^2 + D_{0,y}^2 \right) + \theta \Delta t \left[ D_{x,x} \left( u^{n+1} \right) + D_{x,y} \left( v^{n+1} \right) \right] \right\} \delta T \\
= \frac{\Delta t k}{\rho_0 c_p} \left( D_{0,x}^2 + D_{0,y}^2 \right) T^n - \theta \Delta t \left[ D_{0,x} \left( u^{n+1} T^n \right) + D_{0,y} \left( v^{n+1} T^n \right) \right] \\
+ \theta \Delta t \left[ \frac{\epsilon^2}{2 \rho_0 c_p} \left( \phi^{n+1} \right) - \frac{30 L_0 \psi \left( \phi^{n+1} \right)}{c_p} \right] \frac{D \phi^{n+1}}{D t} + \theta \Delta t \frac{W \left( u^{n+1}, v^{n+1}, \phi^{n+1} \right)}{\rho_0 c_p} \\
- (1 - \theta) \Delta t \left[ D_{0,x} \left( u^n T^n \right) + D_{0,y} \left( v^n T^n \right) \right] + (1 - \theta) \Delta t \frac{W \left( u^n, v^n, \phi^n \right)}{\rho_0 c_p}.
\tag{3.52b}
\]

Note that the substantial derivative $D\phi/Dt$ in Eq. (3.52b) will be replaced by the right-hand side of Eq. (3.30d) during the computation. Again, since staggered gridding is implemented, $\phi$ and $T$ are computed in the cell center as shown in Figure 3.2. In addition, we have

\[
\tilde{\phi}_{i,j} = \frac{\phi_{i,j} + \phi_{i+1,j}}{2}, \quad \tilde{\phi}_{i,j} = \frac{\phi_{i,j} + \phi_{i,j+1}}{2}, \\
\tilde{T}_{i,j} = \frac{T_{i,j} + T_{i+1,j}}{2}, \quad \tilde{T}_{i,j} = \frac{T_{i,j} + T_{i,j+1}}{2}.
\]

We then split Eq. (3.52) into a two-level form to construct Douglas–Gunn procedure:

\[
(I + A_{1\phi}) \delta \phi_1 = S^n_\phi, \tag{3.53a}
\]

\[
(I + A_{1T}) \delta T_1 = S^n_T + W^n, \tag{3.53b}
\]
for the first split step, and
\[
(I + A_{2\phi}) \delta \phi_2 = \delta \phi_1, \tag{3.54a}
\]
\[
(I + A_{2T}) \delta T_2 = \delta T_1, \tag{3.54b}
\]
for the second. Here we define terms in Eqs. (3.53) and (3.54) by
\[
A_{1\phi} = -\theta \Delta t \left[ \frac{\epsilon^2}{M} \left( \frac{\partial N}{\partial \phi_{xx}} \right)^{(m)} D_{0,x}^2 + \frac{\epsilon^2}{M} \left( \frac{\partial N}{\partial \phi_x} \right)^{(m)} D_{0,x} - D_{0,x} (u^{(m)}.) + \left( \frac{\partial N_1}{\partial \phi} \right)^{(m)} \right],
\]
\[
A_{2\phi} = -\theta \Delta t \left[ \frac{\epsilon^2}{M} \left( \frac{\partial N}{\partial \phi_{yy}} \right)^{(m)} D_{0,y}^2 + \frac{\epsilon^2}{M} \left( \frac{\partial N}{\partial \phi_y} \right)^{(m)} D_{0,y} - D_{0,y} (v^{(m)}.) \right],
\]
\[
A_{1T} = -\theta \Delta t \left[ \frac{k}{\rho_0 c_p} D_{0,x}^2 - D_{0,x} (u^{(m)}.) \right],
\]
\[
A_{2T} = -\theta \Delta t \left[ \frac{k}{\rho_0 c_p} D_{0,y}^2 - D_{0,y} (u^{(m)}.) \right],
\]
\[
S^n_\phi = \phi^n - \phi^{(m)} + \theta \Delta t \left[ \frac{\epsilon^2}{M} N (\phi^{(m)}) - D_{0,x} (u^{(m)} \phi^{(m)}) - D_{0,y} (v^{(m)} \phi^{(m)}) + N_1 (\phi^{(m)}, T^{(m)}) \right]
\]
\[
+ (1 - \theta) \Delta t \left[ \frac{\epsilon^2}{M} N (\phi^n) - D_{0,x} (u^n \phi^n) - D_{0,y} (v^n \phi^n) + N_1 (\phi^n, T^n) \right],
\]
\[ S_T^n = T^n - T^{(m)} + \theta \Delta t \left[ \frac{k}{\rho_0 c_p} (D_{0,x}^2 + D_{0,y}^2) T^{(m)} - D_{0,x} (u^{(m)} T^{(m)}) - D_{0,y} (v^{(m)} T^{(m)}) \right] \]
\[ + \theta \Delta t \left[ \frac{\epsilon^2 N (\phi^{(m)})}{2 \rho_0 c_p} - \frac{30 L_0 \psi (\phi^{(m)})}{c_p} \right] \frac{D\phi^{(m)}}{Dt} + (1 - \theta) \Delta t \left[ \frac{k}{\rho_0 c_p} (D_{0,x}^2 + D_{0,y}^2) T^n \right. \]
\[ - D_{0,x} (u^n T^n) - D_{0,y} (v^n T^n) \left] \right) + (1 - \theta) \Delta t \left[ \frac{\epsilon^2 N (\phi^n)}{2 \rho_0 c_p} - \frac{30 L_0 \psi (\phi^n)}{c_p} \right] \frac{D\phi^n}{Dt}, \]
\[ W_T^n = \frac{\theta \Delta t}{\rho_0 c_p} W (u^{(m)}, v^{(m)}, \phi^{(m)}) + \frac{(1 - \theta) \Delta t}{\rho_0 c_p} W (u^n, v^n, \phi^n). \]

The implementation of these formulas can be summarized as a pseudo-language algorithm presented below.

**Algorithm 3** Suppose \( n \) time steps have been computed using Eqs. (3.53) and (3.54) as approximations to Eqs. (3.30d) and (3.30e). To advance the numerical solution to time level \( n + 1 \), carry out the following steps.

1. Set initial values for \( \phi^{(m)} \) and \( T^{(m)} \) to \( \phi^n, T^n \), respectively;
2. Set quasilinear iteration counter \( m = 0 \);
3. Solve the first time-split step, Eqs. (3.53) for \( \phi \) and \( T \) to obtain \( \delta \phi_1 \) and \( \delta T_1 \);
4. Solve the second time-split step, Eqs. (3.54) for \( \phi \) and \( T \) to obtain \( \delta \phi_2 \) and \( \delta T_2 \);
5. Update solutions:
   \[ \phi^{(m+1)} = \phi^{(m)} + \delta \phi_2, \]
   \[ T^{(m+1)} = T^{(m)} + \delta T_2; \]
6. Compute iteration error for \( \phi^{(m+1)} \) and \( T^{(m+1)} \):
   If error of \( \delta \phi < \) convergence tolerance \( 10^{-8} \)
   AND error of \( \delta T < \) convergence tolerance \( 10^{-6} \)
     go to 7
   else if \( m > \) maximum iteration number 300
     print “quasilinear iteration of phase-field equations does not converge,” exit
   else \( m = m + 1 \), go to 3
7. Set $n = n + 1$; go to 1 to begin next time step.

3.5 Parallel Techniques

We implement parallel techniques, Message Passing Interface (MPI) and OpenMP, to our computations of dendrite growth with and without a flow field. In this section, we will present implementation of the two parallel techniques. Parallel performance comparison between MPI and OpenMP is provided by Xu et al. [81], and we will discuss the parallel performance of MPI algorithm in Section 6.3.

The OpenMP Application Program Interface (API) is a portable, scalable model that gives shared-memory parallel programmers a simple and flexible interface for developing parallel applications [82]. The OpenMP Fortran API uses the fork-join model of parallel computation as shown in Figure 3.3. The OpenMP Fortran API begins execution with a single processor called the master thread. The master thread executes sequentially until it encounters the first parallel construct, and then, it creates new threads to be distributed and executed on different processors. Implementation of the OpenMP Fortran API is quite straightforward: it can be done by automatic parallelization of DO loops. All that is necessary is to share the information required by the parallelization within the DO loop, and this is easily handled within the OpenMP syntax.

![Figure 3.3 Fork-join model for OpenMP Fortran API.](image)

MPI is a standard for inter-process communication on distributed-memory multi-processors [83], [84]. The execution model for MPI is quite different from that for OpenMP. In the simplest MPI programs, a master processor (root) sends off work to other processors. Those
processors receive data, perform computations on them, and send results back to the master processor that then combines results as shown in Figure 3.4. As indicated in this figure, communication between each processor is needed in some cases and will be explained later via an example.

![Figure 3.4](image.png)  
**Figure 3.4** MPI execution model.

The procedure for parallelizing two-step Douglas & Gunn time-splitting method with MPI is to compute different parts of a domain on different processors, i.e., simply a crude form of domain decomposition. In particular, we divide the domain into $n$ equal pieces along separate directions corresponding to each split step and conduct computations of each part in the corresponding processor, where $n$ is the number of processors being used. That is, we first divide the domain in the $y$ direction during the first time-splitting step and then in the $x$ direction during the second step. The sketch of the domain-decomposition procedure is shown in Figure 3.5.

This partitioning technique requires data transformations between each processor during the two steps of Douglas & Gunn time-splitting. Moreover, “boundary” data transformations of the sub-domains are also needed between adjacent processors. We will use two processors as an example to explain the data-block transfer procedure shown in Figure 3.6 and the transformation of boundary data of each processor indicated in Figure 3.7.

Solutions in the pink regions shown in Figure 3.6 will be computed in processor 0, and those in the blue ones will be done in processor 1. Therefore, data in domain 1 as shown in
Figure 3.5  Distribution of processors for two-level Douglas & Gunn time-splitting.

0 & 0: Processor 1; 1 & 1: Processor 2.

Figure 3.6  Example of transferring data block between processors for Douglas & Gunn time-splitting method.
Figure 3.7  Example of transferring data between adjacent processors for Douglas & Gunn time-splitting method.

Figure 3.6(a) have to be transferred from processor 1 to processor 0 after the computation of the first time-splitting step, and data in this domain will be updated by the computation in processor 0 during the second time-splitting step. Moreover, data in domain 1 have to be transferred back to processor 1 to prepare for the computation of the next iteration step. Data in domain 0 shown in Figure 3.6(a) will be transferred between processors 0 to 1 in a similar way. In addition, to perform computation in processor 0 displayed in Figure 3.7, data of line $j_{fp+1}$ will be needed because of the center-differencing discretization on line $j_{fp}$. Hence, data of line $j_{fp+1}$ are required to be transferred from processor 1 to 0. In the same way, data of line $j_{ap-1}$ have to be transferred from processor 0 to 1 to complete the computation in processor 1.
In this chapter, we introduce multiscale methods and present a detailed numerical method to implement a multiscale method for the phase-field model in a flow field via a pseudo-language algorithm. Boundary and initial conditions for a well-posed freezing in lid-driven-cavity-flow problem are first provided followed by the “coarse-grained” algorithm.

4.1 Introduction to Multiscale Methods

Many systems in nature involve multiple scales and become interesting because they exhibit different behaviors on different scales. Such systems range from living organisms, geological and geophysical systems, materials and condensed matter systems, even social structures and hierarchies, etc. The study of systems involving multiple scales in nature can be traced back to hundreds of years ago. The physicist Galileo Galilei had the basic insight that the scaling of area and volume determines how natural phenomena behave differently on different scales. He first reasoned about mechanical structures and later extended his insights to living things, taking the then radical point of view that at the fundamental level, a living organism should follow the same laws of nature as a machine. For example, any biological system has an inherent hierarchy of structures, each being subject to modeling; hence, it can be modeled at different scales. Similar ecological principles can also be applied to urban and urbanizing landscapes. For instance, the urban continuum can be divided into different urban contexts: city, inner suburbs, suburbs, exurban, and hinterlands. Each of these contexts can be divided further into land-use types, neighborhoods, blocks, and so on. Likewise, vegetation in an urban landscape can be delineated by structural characteristics to form tree-covered patches, rural communities, and managed lawns [85]. Each of these nested patch hierarchies is more than a convenient way to organize spatial heterogeneity. Patch hierarchies allow researchers to ask questions related to what factors influence the patterns and processes observed at each nested scale and the functional relationships within and between scales [86]. In geological systems, vortical structures in the atmosphere and ocean range from meters to thousands of kilometers. Materials at different scales—from quantum
to molecular, to mesoscale, to continuum levels—have different behaviors and mechanical properties such as strength, fatigue, and wear behavior. Chemical reactions may take seconds or hours, while the vibration of chemical bonds occurs at the time scale of femtoseconds.

Therefore, two questions become important: what are the rules governing the small scale (microscopic) units in a large (macroscopic) system, and how is the macroscopic behavior influenced by the microscopic behavior? what are the rules governing the large scale behavior, and how do they influence the behavior on the small scale?

The traditional approach to multiscale problems is to obtain, either analytically or empirically, explicit equations for the scale of interest, while eliminating other scales of the problems. Analysis of these systems is conventionally done by construction of approximate solutions using similarity techniques [87], averaging methods [88], renormalization and homogenization theory [89], etc. Recently Eck et al. [90] and Eck [91] [92] derived a two-scale phase-field model for liquid-solid phase transitions of binary mixtures by formal homogenization of a sharp interface model, including the Gibbs–Thomson law and kinetic undercooling. The model consists of a macroscopic energy equation and, for each point of the macroscopic domain, a local cell problem describing the evolution of single equiaxed crystals. They proved the existence, uniqueness, and a partial regularity of the solution, and presented numerical examples via a numerical two-scale method of the model. Turbulence modeling is another example of the traditional approach for multiscale problems. Large-eddy simulation (LES) is one of the typical approaches for turbulence models. A common deduction of Kolmogorov’s (1941) famous theory of self similarity is that large eddies of the flow are dependent on the flow geometry, while smaller eddies are self similar and have a universal character. For this reason, it became a practice to solve only for the large eddies explicitly, and model the effect of the smaller and more universal eddies on the larger ones. Thus, in LES the large scale motions of the flow are calculated, while the effect of the smaller, universal scales (the so called sub-grid scales) are modeled using a sub-grid scale (SGS) model.

Despite some successes, these approaches also force us to introduce empirical closures for other systems that are not always justified or understood. As a result, the success of such phenomenological equations is much less spectacular for a large class of complex systems. Typical examples of such a situation are found in complex fluids, plasticity, fracture
A new approach has emerged in recent years and has quickly attracted attention. The aim of this approach is to model the theoretical input to a coarse-grained model from a more detailed microscopic model, bypassing the necessity of empirical modeling. Car and Parrinello [93] proposed a unified scheme to extend molecular dynamics beyond the usual pair-potential approximation and make possible the simulation of both covalently bonded and metallic systems. In addition, the new approach overcomes the computationally demanding difficulty of density-function calculation, and permitted the application of density-functional theory to much larger systems than previously feasible ones. An outstanding problem of modeling materials from fundamental principles is the role of microstructure in determining material properties. One of the key challenges is the simultaneous operation of multiple scales requiring alternative simulation schemes. Shenoy et al. [94] presented a reformulation of the quasicontinuum method for treating multiple scales and demonstrated its application via two examples: the interaction of lattice dislocations with grain boundaries and the interaction of cracks with grain boundaries. Rudd and Broughton [95] developed a coarse-grained molecular dynamics to simulate mechanics of micron-scale solid systems, which captures the important atomistic effects without the computational cost of the conventional molecular dynamics. Moreover, the coarse-grained molecular dynamics technique agrees with molecular dynamics as the mesh size is reduced to the atomic scale. Broughton et al. [96] illustrated and validated a methodology on seamless coupling of length scale from continuum to statistical to quantum mechanics by the example of crack propagation in silicon. Cai et al. [97] proposed a method for coupling two crystalline domains for which the interactions between domains are linear such that reflection of elastic waves across the boundary is minimized. They demonstrated that the approach gave significantly reduced phonon reflections at the domain boundaries relative to existing coupling method.

Chorin et al. [98] [99] considered problems where good resolution had not been achieved. They assumed that the additional information consisted of explicit information about a measure preserved by the differential equations. Their goal was to calculate averages of the solutions with respect to the constrained measure. Given the constrained measure and the filtered values, the mean and the moments of the solution can be found at all points by...
interpolation, and therefore, the mean derivatives of these quantities at the computational points can be found. The remaining problem is to characterize the evolution of the constrained measure so that the mean solution can be advanced in time. They implemented optical prediction methods to estimate the solution when the solution is too complicated to be fully resolved or when data were missing.

E and Engquist [100] presented a general framework for designing and analyzing numerical methods that deal with problems that can be divided into two types:

A. A macroscopic description is known but ceases to be valid in a localized region in space and/or time, and where the microscopic description has to be used instead. Such problems include defects in crystals where atomistic descriptions have to be used near the defects, continuum theories are valid away from the defects [101], turbulent flame fronts, and chemical systems with localized chemical reactions.

B. A macroscopic model may not be explicitly known, or too expensive to obtain, but is known to exist; i.e., there exists a set of macroscopic variables obeying a closed macroscopic model. Such problems include transport through inhomogeneous media such as porous medium flows, complex fluids, and plasticity.

In addition, there exists a class of problems, say of Type C, which combines the characteristics of A and B, namely that the macroscopic model is not explicitly known and the typical model cease to be valid in some regions.

Their goal was to introduce a framework and analysis that cover many of the existing methods and also derive new techniques from the general formulation. This framework makes efficient use of both the macroscopic and microscopic formulations, even in cases when the macroscopic equations or models are not explicitly known. The main procedure to accurately approximate the macroscopic state of the system is to select a conventional macroscale scheme, and then estimate the necessary data for the macroscale scheme from the microscale model. When the macroscopic model is not fully and explicitly known, a procedure needs to be provided for supplementing the missing data from microscopic models. They called these methods heterogeneous multiscale methods (HMM) to emphasize the fact that different physical models and numerical techniques are used at different scales and on different
grids. Ren and E [102] developed numerical methods based on the framework of the HMM. Their purpose was to study the macroscale dynamics of complex fluids and micro-fluidics in situations where either the constitutive relation or the boundary conditions were not explicitly available and had to be inferred from microscopic models such as molecular dynamics. Yue and E [103] also discussed numerical methods for linear and nonlinear transport equations with multiscale velocity fields using the HMM and applied them to two-phase flow in heterogeneous porous media.

The freezing problem in a flow field can be categorized as type C, i.e., the macroscopic freezing model is not explicitly known, and macroscopic representations cease to be valid on the microscopic level. The reason is that the phase-field-like models are needed to provide the details of dendrite structures on the microscopic levels. However, these cannot be applied to the macroscopic level unless some techniques, e.g., the homogenization theory, are applied. The so-called heterogeneous multiscale method is implemented in our research because of the advantages presented before. In particular, we develop different numerical methods for different physical models at different scales. We make direct use of the microscale model to obtain the information which is required by the macroscale model, e.g., volume fraction of solid phase and energy flux gained or released by the phase-change process. The two-scale phase-field model with flow field will be presented in the next section, followed by details of numerical methods.

### 4.2 Two-Scale Phase-Field Model in Flow Field

We implement the heterogeneous multiscale method for the 2-D governing equations (3.30); we solve the mass conservation and momentum equations on the macroscopic scale and the whole set of governing equations on the microscopic scale. Therefore, we have

\begin{align}
  u_x + v_y &= 0 \tag{4.1a} \\
  u_t + (uu)_x + (vu)_y &= -\frac{p_x}{\rho_0} + \frac{1}{\rho_0} \left[ 2 (\mu u_x)_x + (\mu u_y)_y + (\mu v_x)_y \right] + X_1(\phi), \tag{4.1b} \\
  v_t + (uv)_x + (vv)_y &= -\frac{p_y}{\rho_0} + \frac{1}{\rho_0} \left[ (\mu v_x)_x + 2 (\mu v_y)_y + (\mu u_y)_x \right] + X_2(\phi) + B(\phi, T), \tag{4.1c}
\end{align}
on large scale, and

\[ u_x + v_y = 0 \]  \hspace{1cm} (4.2a)
\[ u_t + (uu)_x + (uv)_y = -\frac{p_x}{\rho} + \frac{1}{\rho} \left[ 2(\mu u_x)_x + (\mu u_y)_y + (\mu v_x)_x \right] + X_1(\phi), \]  \hspace{1cm} (4.2b)
\[ v_t + (uv)_x + (vv)_y = -\frac{p_y}{\rho} + \frac{1}{\rho} \left[ (\mu u)_x + 2(\mu v)_y + (\mu u)_x \right] + X_2(\phi) + B(\phi, T), \]  \hspace{1cm} (4.2c)
\[ \phi_t + (u\phi)_x + (v\phi)_y = \frac{\epsilon^2}{M} N(\phi) + \frac{\rho_0 L_0 P'(\phi)}{M T_m} (T - T_m) - \frac{\rho_0 \psi'(\phi) T}{4a M} + \frac{Y(\phi, T)}{M}, \]  \hspace{1cm} (4.2d)
\[ T_t + (uT)_x + (vT)_y = \alpha \Delta T + \left[ \frac{\epsilon^2 N(\phi)}{2\rho_0 c_p} - \frac{30L_0 \psi(\phi)}{c_p} \right] \frac{D\phi}{Dt} + \frac{W(u, v, \phi)}{\rho_0 c_p}, \]  \hspace{1cm} (4.2e)

on small scale.

The initial and boundary conditions for a well-posed problem on a large-scale domain \( \Omega \equiv [0, L] \times [0, L] \) follow Eq. (3.33), while those conditions on a small-scale domain \( \Omega_s \equiv [0, L_s] \times [0, L_s] \) become

**Initial conditions**

\[ \phi = 0 \quad \text{in} \quad \Omega_0, \]
\[ \phi = 1 \quad \text{in} \quad \Omega_s \setminus \Omega_0, \]
\[ T = T_m \quad \text{in} \quad \Omega_0, \]
\[ T = T_{\text{init}} < T_m \quad \text{in} \quad \Omega_s \setminus \Omega_0, \]  \hspace{1cm} (4.3a)

**Boundary conditions**

\[ u_x = -v_y; \quad v_x = u_y + \omega \quad \text{on} \quad \{ x = 0 \} \cup \{ x = L_s \}, \]
\[ u_y = v_x - \omega; \quad v_y = -u_x \quad \text{on} \quad \{ y = 0 \} \cup \{ y = L_s \}, \]
\[ \frac{\partial \phi}{\partial n} = \frac{\partial T}{\partial n} = \frac{\partial p}{\partial n} = 0 \quad \text{on} \quad \partial \Omega_s. \]  \hspace{1cm} (4.3b)

In the above boundary conditions, vorticity \( \omega \) on boundary \( \partial \Omega_s \) is obtained by solving the vorticity transport equation

\[ \omega_t = -u\omega_x - v\omega_y + \frac{\mu(\phi)}{\rho_0} \Delta \omega. \]

Here, velocity boundary conditions are employed to satisfy the mass conservation law and avoid the effect of the boundary condition to the flow field, as shown by Ziaei et al. [104], especially when moving boundary problems are involved.
For simplicity, it is reasonable to assume that small-scale velocities and pressure are “frozen” within one time step of the large-scale calculation. Hence, velocities and pressure in Eq. (4.2) are unchanged during the small-scale computation. Therefore, the governing equations (4.2) on the small-scale domain \( \Omega_s \equiv [0, L_s] \times [0, L_s] \) are reduced to

\[
\begin{align*}
\phi_t + (u\phi)_x + (v\phi)_y &= \frac{\epsilon^2}{M} N(\phi) + \frac{\rho_0 L_0 P'(\phi)}{MT_m} (T - T_m) - \frac{\rho_0 \psi'(\phi)T}{4aM} + \frac{Y(\phi, T)}{M}, \\
T_t + (uT)_x + (vT)_y &= \alpha \Delta T + \left[ \frac{\epsilon^2 N(\phi)}{2 \rho_0 c_p} - \frac{30L_0 \psi(\phi)}{c_p^2} \right] \frac{D\phi}{Dt} + \frac{W(u, v, \phi)}{\rho_0 c_p},
\end{align*}
\]

with \( u, v \) and \( p \) independent of time. Initial conditions of the above equations again follow Eq. (4.3a), and homogeneous Neumann boundary conditions for \( \phi \) and \( T \) as in Eq. (4.3b) are applied.

### 4.3 Implementation of Multiscale Method

We explain the implementation of a multiscale method via Figure 4.1 and Algorithm 4. In Figure 4.1, we denote a large-scale domain by \( \Omega \) and a small-scale domain inside one grid cell of \( \Omega \) by \( \Omega_s \). We first solve the equations of fluid motion (3.30a–3.30c) on a macroscopic level in \( \Omega \) to obtain \( u, v \) and pressure \( p \) (or pseudo-pressure \( \varphi \)). We then interpolate the large-scale solutions to every grid point of \( \Omega_s \) to prepare for the computation on the small scale. Small-scale computation is conducted to simulate ice growth in the flow field during one time step of the large-scale calculation. We assume that the small-scale computation does not affect the velocity field and pressure on the large scale. Phase field \( \phi \) and temperature \( T \) on the small scale are computed by Eqs. (3.30d) and (3.30e) with Neumann boundary conditions on \( \partial \Omega_s \) to approximate an infinite domain. Moreover, velocity and pressure on the small scale are obtained using Eqs. (3.30a–3.30c) with Dirichlet boundary conditions, which are calculated from the interpolation of large-scale solutions.

**Algorithm 4** Suppose \( n \) time steps have been computed, and \( u^n, v^n, p^n, \phi^n \) and \( T^n \) for the large scale have been obtained. To advance the numerical solution to time level \( n + 1 \), carry out the following steps.

1. **Large-Scale Computation on \( \Omega \):**

   Solve flow field Eqs. (3.30a), (3.30b) and (3.30c) to obtain \( u^{n+1}, v^{n+1} \) and \( p^{n+1} \);
2. Interpolation from Large Scale to Small Scale:

Pick a grid cell $(i_s, j_s)$ in $\Omega$ such that $\Omega_s \subseteq (i_s, j_s)$ as in Figure 4.1. Interpolate large-scale solutions $u^n, v^n$ and $p^n$ to grid points on $\Omega_s$ to obtain initial conditions $u^0_s, v^0_s, p^0_s$ and boundary conditions $u^b_s, v^b_s, p^b_s$ for small-scale computation;

3. Small-Scale Computation on $\Omega_s$:

Suppose $n_s$ time steps have been computed, and $u^n_s, v^n_s, p^n_s, \phi^n_s$ and $T^n_s$ have been obtained. Carry out the following steps to advance to time level $n_s + 1$.

(a) Solve Eqs. (3.30a), (3.30b) and (3.30c) on small scale using boundary conditions $u^b_s, v^b_s, p^b_s$ to obtain $u^{n_s+1}_s, v^{n_s+1}_s$ and $p^{n_s+1}_s$;

(b) Solve Eqs. (3.30d) and (3.30e) on small scale to obtain $\phi^{n_s+1}_s$ and $T^{n_s+1}_s$;

(c) If $|\phi^{n_s+1}_s - 1| < 10^{-12}$ or $|T^{n_s+1}_s - T_m| < 10^{-10}$

$$\Omega_s = 2\Omega_s$$

Interpolate $u^{n_s+1}_s, v^{n_s+1}_s, p^{n_s+1}_s, \phi^{n_s+1}_s$ and $T^{n_s+1}_s$ to new domain $\Omega_s$

Obtain new boundary conditions $u^b_s, v^b_s, p^b_s$ from $u^{n+1}, v^{n+1}$ and $p^{n+1}$ via interpolation

goto (a) for next time step of small scale
else if $n_s + 1 = \text{total number of time steps for small scale}$

$$\phi_{s}^{n+1} = \phi_{s}^{n+1}, T_{s}^{n+1} = T_{s}^{n+1}$$

goto 4;

4. Data transfer from Small Scale to Large Scale:

To determine $\phi^{n+1}$ and $T^{n+1}$ of large scale from $\phi_{s}^{n+1}$ and $T_{s}^{n+1}$ of small scale at large-scale domain grid point $(i_s, j_s)$, apply the following procedure:

Let $\|\phi^{n+1}\| = \|\phi_{s}^{n+1}\|$ and $\|T^{n+1}\| = \|T_{s}^{n+1}\|$ over the whole grid cell $(i_s, j_s)$ of the large-scale domain with

$$\phi^{n+1} = \frac{1}{h_x h_y} \int_{0}^{h_x} \int_{0}^{h_y} \phi_{s}^{n+1} dx dy, T^{n+1} = \frac{1}{h_x h_y} \int_{0}^{h_x} \int_{0}^{h_y} T_{s}^{n+1} dx dy$$

where $\phi_{s}^{n+1} = 1, T_{s}^{n+1} = T_{\text{init}}$ everywhere outside the small-scale domain $\Omega_s$;

Assign $u^{n+1} = v^{n+1} = 0$ in solid phase;

goto 1 for next time step.

Here, $h_x$ and $h_y$ are large-scale spatial step sizes in $x$ and $y$ directions. Note that the detailed descriptions of steps 1 and 3(a) have been provided in Algorithm 3.4.1, and the detailed implementation of step 3(b) are in Algorithm 3.
Chapter 5
SIMULATION OF 2-D DENDRITE GROWTH AND VALIDATION OF THE PHASE-FIELD MODEL

We first introduce the 2-D nondimensional phase-field model with no convection. We then simulate 2-D isotropic dendrite growth from a supercooled melt, pure nickel, for the purpose of conducting numerical convergence tests and validating the 2-D phase-field model without introduction of the flow field. Finally we display simulations of pure nickel dendrites with introduction of four-fold anisotropy on the interface.

5.1 2-D Nondimensional Phase-Field Model

In this section, we first present a mathematically well-posed problem consisting of the governing equations of the phase-field model with boundary and initial conditions, followed by details of the numerical method applied to solve the problem. The purpose of use of a nondimensional model is to make the comparisons between our numerical results and analytical or experimental ones more convenient.

5.1.1 Governing Equations

In this subsection we introduce the equations of the phase-field model along with the scalings employed to render them dimensionless. Boundary and initial conditions required to formulate a well-posed mathematical problem are also prescribed.

The two-dimensional phase-field model (3.30) is reduced to the coupled dimensionless governing equations for phase field and temperature as follows, since no flow field is considered (also shown in Wang et al. [19]):

\[
\phi_t = m_k \Delta \phi + \frac{m_k}{c^2} \phi(1 - \phi) \left[ \phi - 0.5 + 30 \epsilon a_0 \text{St} \phi(1 - \phi) \right], \quad (5.1a)
\]

\[
\theta_t = \Delta \theta - \frac{30 \phi^2 (1 - \phi)^2}{\text{St}} \phi_t, \quad x \in \Omega, \quad t \in (t_0, t_f]. \quad (5.1b)
\]

In the above equations, \( \phi \) is the phase value as was indicated in Section 3.2. By introducing the length scale \( w \) (the geometric size of the domain) and the reference time scale \( w^2/\alpha \) (the thermal diffusion time), dimensionless temperature \( \theta \), dimensionless spatial variable \( \mathbf{x} \), and
dimensionless time variable \( t \) are defined by
\[
\theta = \frac{T - T_m}{T_m - T_{\text{init}}}, \quad x = \frac{\dot{x}}{w}, \quad t = \frac{\dot{t}}{w^2/\alpha}.
\]
Here \( T_{\text{init}} \) is the initial temperature, with \( \alpha \) and \( k \) as defined in Section 3.2.

In addition, four dimensionless parameters of the isotropic phase-field model are defined as
\[
St = \frac{c_p L (T_m - T_{\text{init}})}{L_0}, \quad \epsilon = \frac{\delta}{w}, \quad m_k = \frac{\mu_k \sigma T_m c_p L}{k L_0}, \quad a_0 = \frac{\sqrt{2} \epsilon}{12 d_0}.
\]
Here \( St \) is the Stefan number indicating the intensity of initial supercooling; \( \epsilon \) is the dimensionless interface thickness; \( m_k \) is the dimensionless interface kinetics, and \( a_0 \) relates the length scale in the phase-field model to the capillarity length \( d_0 \); definition of \( d_0 \) was introduced in Section 1.3.3. As shown in [16], as \( \epsilon \to 0 \), the phase-field model approaches the modified Stefan model Eqs. (1.6).

Domain \( \Omega \) will be a rectangle in the present study, and Neumann boundary conditions given by
\[
\frac{\partial \phi}{\partial n} = 0, \quad \frac{\partial \theta}{\partial n} = 0
\]
will be imposed on \( \partial \Omega \). Initially, a small rectangle \( \Omega_0, \frac{1}{40} \) of the domain size, is placed in the center of the domain to serve as a seed for initiating solidification, and initial data are prescribed as follows:
\[
\theta(x, y) = 0, \quad \phi(x, y) = 0, \quad \text{in} \quad \Omega_0, \\
\theta(x, y) = -1, \quad \phi(x, y) = 1, \quad \text{in} \quad \Omega \setminus \Omega_0,
\]
where \( \Omega_0 \equiv \left[ \frac{39}{50} X_0, \frac{41}{50} X_0 \right] \times \left[ \frac{39}{50} Y_0, \frac{41}{50} Y_0 \right], \) and \( \Omega \equiv [0, X_0] \times [0, Y_0] \).

\( 5.1.2 \) Numerical Methods

In this subsection we first implement both explicit and implicit schemes to make comparisons of total arithmetic and computation times between these two approaches. We then provide an outline of the derivations leading to the discrete form of the 2-D coupled nonlinear reaction-diffusion equations (5.1), followed by the pseudo-language algorithm to perform one time step.
Explicit Scheme

We apply forward Euler in time and centered-differencing in space to the governing equations (5.1). Thus, the accuracy is first-order in time and second-order in space. For convenience the nonlinear terms of the phase field and temperature equations are denoted by $F(\theta, \phi)$ and $G(\theta, \phi)$, respectively, defined as follows:

\begin{align}
F(\theta, \phi) &= \phi(1 - \phi) [\phi - 0.5 + C_1 \theta(1 - \phi)] , \\
G(\theta, \phi) &= \frac{30\phi^2(1 - \phi)^2}{St} \phi_t ,
\end{align}

where the time derivative term $\phi_t$ in Eq. (5.2b) is obtained from the right-hand side of the phase field equation (5.1a). Discretization of governing equations (5.1) yields

\begin{align}
\delta \phi &= m_k \Delta t \left( D_{\theta,xx}^2 + D_{\theta,yy}^2 \right) \phi^n + \Delta t C_2 F(\theta^n, \phi^n) , \\
\delta \theta &= \Delta t \left( D_{\theta,xx}^2 + D_{\theta,yy}^2 \right) \theta^n - \Delta t G(\theta^n, \phi^n) .
\end{align}

Here, \( \left( D_{\theta,xx}^2 + D_{\theta,yy}^2 \right) \) is formal notation for a second-order centered discrete Laplacian. Hence, $\phi^{n+1}$ and $\theta^{n+1}$ at the next time step are obtained from $\phi^n + \delta \phi$ and $\theta^n + \delta \theta$, respectively. For convenience, we have introduced the coefficients $C_1$ and $C_2$ defined by

\begin{align}
C_1 &= 30\epsilon a_0 St , \\
C_2 &= \frac{m_k}{\epsilon^2} .
\end{align}

The explicit scheme is straightforward and easy to implement; however, it requires smaller time steps than does the implicit scheme to satisfy stability constraints. Moreover, such requirements typically become more stringent as the spatial dimension is increased.

Implicit Scheme

In this subsection, we describe in detail the implicit numerical procedures employed in this study. We discretize the governing equation using trapezoidal integration in time and centered-differencing in space (hence, a Crank–Nicolson approach) leading to second-order accuracy both in time and in space. In addition, we implement a $\delta$-form Douglas & Gunn time-splitting method [76] because of its advantages as was explained in Section 3.4.1.

Again we define functions $F$ and $G$ as was done in Eqs. (5.2); but now the function $G$ is composed of two parts: one part containing the second derivative terms of $\phi$, the other
having no such terms. Then $G$ can be expressed as

$$G(\theta, \phi) = C_3g_1(\theta, \phi) + C_4g_2(\theta, \phi),$$

where

$$g_1(\theta, \phi) = (\phi^4 - 2\phi^3 + \phi^2)(\phi_{xx} + \phi_{yy}),$$

$$g_2(\theta, \phi) = (\phi^4 - 2\phi^3 + \phi^2)F(\theta, \phi),$$

and $C_3$ and $C_4$ are defined by

$$C_3 = \frac{30m_k}{St}, \quad C_4 = \frac{30m_k}{St^2}.$$

Quasilinearizations of $F$ and $G$ are constructed by Fréchet–Taylor expansion in “$\delta$-form” [105]. Therefore, we have

$$F(\theta, \phi) = F^{(m)} + F^{(m)}_\theta \delta \theta + F^{(m)}_\phi \delta \phi + ..., $$

$$g_1(\theta, \phi) = g_1^{(m)} + g_{1\phi}^{(m)} \delta \phi + g_{1\phi xx}^{(m)} \delta \phi_{xx} + g_{1\phi yy}^{(m)} \delta \phi_{yy} + ..., $$

$$g_2(\theta, \phi) = g_2^{(m)} + g_{2\theta}^{(m)} \delta \theta + g_{2\phi}^{(m)} \delta \phi + ..., $$

where the superscript $(m)$ denotes the previous iteration value, with $m = 0$ as an initial guess; $\delta \theta = \theta^{(m+1)} - \theta^{(m)}$, and $\delta \phi = \phi^{(m+1)} - \phi^{(m)}$. We retain only through first-order terms in the above expansions; the required Fréchet derivatives of $F$, $g_1$ and $g_2$ can be expressed as

$$F_\theta = C_1 \phi^2 (1 - \phi)^2,$$

$$F_\phi = 2C_1 \theta \phi (2\phi^2 - 3\phi + 2) - 3\phi^2 + 3\phi - 0.5,$$

$$g_{1\phi} = (4\phi^3 - 6\phi^2 + 2\phi)(\phi_{xx} + \phi_{yy}),$$

$$g_{1\phi xx} = \phi^2 (1 - \phi)^2,$$

$$g_{1\phi yy} = \phi^2 (1 - \phi)^2,$$

$$g_{2\theta} = C_1 \phi^4 (\phi^4 - 4C_1 \phi^3 + 6C_1 \phi^2 - 4C_1 \phi + 1),$$

$$g_{2\phi} = \phi^2(1 - \phi)^2 \left[2C_1 \theta (4\phi^3 - 6\phi^2 + 2\phi) - 7\phi^2 + 7\phi - 1.5 \right].$$
Applying trapezoidal integration to Eqs. (5.1a) and (5.1b) simultaneously, and replacing the advanced time level values \( \theta^{n+1}, \phi^{n+1} \) with \( \delta \theta + \theta^{(m)} \) and \( \delta \phi + \phi^{(m)} \), respectively, yield

\[
\left[ I - \frac{\Delta t}{2} m (D_{0,x}^2 + D_{0,y}^2) - \frac{\Delta t}{2} C_2 F_\phi^{(m)} I \right] \delta \phi - \frac{\Delta t}{2} C_2 F_\theta^{(m)} I \delta \theta = \\
\left[ I + \frac{\Delta t}{2} m (D_{0,x}^2 + D_{0,y}^2) \right] \phi^{n} - \left[ I - \frac{\Delta t}{2} m (D_{0,x}^2 + D_{0,y}^2) \right] \phi^{(m)}
\nonumber
\]

\[
+ \frac{\Delta t}{2} C_2 (F^{(m)} + F^n) , \tag{5.4a}
\]

\[
\frac{\Delta t}{2} \left( C_3 g_{1,xx}^{(m)} I + C_4 g_{2,xx}^{(m)} I + C_3 g_{1,y}^{(m)} D_{0,x}^2 + C_3 g_{1,yy}^{(m)} D_{0,y}^2 \right) \delta \phi
\nonumber
\]

\[
+ \left[ I - \frac{\Delta t}{2} (D_{0,x}^2 + D_{0,y}^2) + \frac{\Delta t}{2} C_4 g_{2,xx}^{(m)} I \right] \delta \theta = \\
\left[ I + \frac{\Delta t}{2} (D_{0,x}^2 + D_{0,y}^2) \right] \theta^{n} - \left[ I - \frac{\Delta t}{2} (D_{0,x}^2 + D_{0,y}^2) \right] \theta^{(m)}
\nonumber
\]

\[
- \frac{\Delta t}{2} \left[ C_3 \left( g_1^{(m)} + g_1^n \right) + C_4 \left( g_2^{(m)} + g_2^n \right) \right] . \tag{5.4b}
\]

Here \( D_{0,x}^2 \) and \( D_{0,y}^2 \) denote second-order centered differences with respect to \( x \) and \( y \) respectively, and \( I \) is the identity matrix. It should be clear that the right-hand sides of Eqs. (5.4) are the original discrete equations. Hence, these must approach zero as \( m \to \infty \) if the iterations converge, implying that \( \delta \theta, \delta \phi \to 0 \) in this limit.

Note further that in Eq. (5.4a), the matrix multiplying \( \delta \phi \) contains a diagonal matrix \(-\frac{\Delta t}{2} C_2 F_\phi^{(m)} I\). The treatment of this in a two-level Douglas & Gunn time-splitting procedure is accomplished by combining it with the identity matrix and dividing every term in Eq. (5.4a) by \( 1 - \frac{\Delta t}{2} C_2 F_\phi^{(m)} \). The same strategy can be applied to the matrix \( \frac{\Delta t}{2} C_4 g_{2,xx}^{(m)} I \) multiplying \( \delta \theta \) in Eq. (5.4b), i.e., we combine it with the identity matrix and divide every term in Eq. (5.4b) by \( 1 + \frac{\Delta t}{2} C_4 g_{2,xx}^{(m)} \). We guarantee diagonal dominance, and thus stability against rounding errors, for matrices of \( \delta \phi \) in Eq. (5.4a) and \( \delta \theta \) in Eq. (5.4b), by doing so. Moreover, the coefficient matrix of \( \delta \theta, -\frac{\Delta t}{2} C_2 F_\theta^{(m)} I \), in Eq. (5.4a) is split into the same two parts. For the matrices of \( \delta \phi \) in Eq. (5.4b), we add and subtract an identity matrix \( I \) simultaneously, and then merge the remaining diagonal matrices into the matrix \( C_3 g_{1,xx} D_{0,x}^2 \) of the first split.
step. Hence, we obtain the Douglas & Gunn time-splitting form as

\[(I + A_{1\phi_{1}}) \delta \phi_{1} + A_{1\theta_{1}} \delta \theta_{1} = B_{1}^{n} \phi^{n} + S_{1}^{n}, \quad (5.5a)\]
\[(I + A_{2\phi_{1}}) \delta \phi_{1} + (I + A_{2\theta_{1}}) \delta \theta_{1} = B_{2}^{n} \theta^{n} + S_{2}^{n}, \quad (5.5b)\]

for the first split step, and

\[(I + A_{1\phi_{2}}) \delta \phi_{2} + A_{1\theta_{2}} \delta \theta_{2} = \delta \phi_{1}, \quad (5.6a)\]
\[(I + A_{2\phi_{2}}) \delta \phi_{2} + (I + A_{2\theta_{2}}) \delta \theta_{2} = \delta \theta_{1}, \quad (5.6b)\]

for the second split step. In Eqs. (5.5) and (5.6) the various coefficients are defined as follows:

\[A_{1\phi_{1}} = -\frac{m\Delta t}{2 - \Delta t C_{2} F_{\phi}^{(m)}} D_{0,x}^{2}, \]
\[A_{2\phi_{1}} = \frac{\Delta t}{2 \Delta t C_{4} g_{2\phi}^{(m)}} \left[ \Delta t \left( C_{3} g_{1\phi}^{(m)} + C_{4} g_{2\phi}^{(m)} - C_{4} g_{2\theta}^{(m)} \right) - 2 \right] I + \Delta t C_{3} g_{1\phi_{xx}}^{(m)} D_{0,x}^{2}, \]
\[A_{1\theta_{1}} = A_{1\theta_{2}} = -\frac{\Delta t C_{2} F_{\theta}^{(m)}}{4 - 2 \Delta t C_{2} F_{\phi}^{(m)}} I, \]
\[A_{2\theta_{1}} = -\frac{\Delta t}{2 + \Delta t C_{4} g_{2\theta}^{(m)}} D_{0,x}^{2}, \]
\[A_{1\phi_{2}} = -\frac{m\Delta t}{2 - \Delta t C_{2} F_{\phi}^{(m)}} D_{0,y}^{2}, \]
\[A_{2\phi_{2}} = \frac{\Delta t C_{3} g_{1\phi_{yy}}^{(m)}}{2 + \Delta t C_{4} g_{2\theta}^{(m)}} D_{0,y}^{2}, \]
\[A_{2\theta_{2}} = -\frac{\Delta t}{2 + \Delta t C_{4} g_{2\theta}^{(m)}} D_{0,y}^{2}, \]
\[B_{1}^{n} = \frac{2I + m \Delta t \left( D_{0,x}^{2} + D_{0,y}^{2} \right)}{2 - \Delta t C_{2} F_{\phi}^{(m)}}, \]
\[B_{2}^{n} = \frac{2I + \Delta t \left( D_{0,x}^{2} + D_{0,y}^{2} \right)}{2 + \Delta t C_{4} g_{2\theta}^{(m)}}, \]
\[S_{1}^{n} = -\frac{\left[ 2I - m \Delta t \left( D_{0,x}^{2} + D_{0,y}^{2} \right) \right] \phi^{(m)} + \Delta t C_{2} \left( F^{(m)} + F^{n} \right)}{2 - \Delta t C_{2} F_{\phi}^{(m)}}, \]
5.2 Numerical Convergence Tests

In this section we first provide specific details of physical and numerical parameters employed in the computations. We then present data showing convergence of physical energy of the system as truncation error is reduced and numerical precision is increased, and we display data associated with convergence rate of quasilinear iterations for two different versions of the overall solution procedure. Finally, we supply specific information associated with grid function convergence.

5.2.1 Physical and Numerical Run Conditions

Computations were performed on a HP N-4000 SMP, and later on a HP SuperDome, operated by the University of Kentucky Computing Center. The Fortran 90 computer programming language is used with 64-bit arithmetic unless otherwise noted. In the present study, the (nondimensional) square domain is chosen to be $\Omega \equiv [0,4] \times [0,4]$ except when specified otherwise, to represent “infinite” domain in the computational sense. Pure nickel is used as the material, but no anisotropy of the interface is assumed. There are two main reasons to choose pure nickel instead of other materials (e.g., SCN, water): one is that it was used in the previous numerical simulations using phase-field model as in [22]; the other is to make convenient comparison with the experimental data provided in [38]. The thermophysical parameters of nickel can be found in Brandes [106] and are listed in Table 5.1; the parameters of water are also listed for later use.

For simplicity, we assume that density, specific heat and thermal conductivity are constant, and values in the liquid are equal to those in the solid. Since there is no general strategy for determining the kinetic coefficient $\mu_k$ for the reason described in Section 2.3, we estimate the value of the kinetic coefficient by several numerical tests to obtain the dendrite growth rate based on the kinetic coefficient. We found that the kinetic coefficient can be assumed of $O(1)$ for pure metal, then we choose $\mu_k = 3.08 \, m/(K \cdot s)$ in the current computa-


Table 5.1 Thermo-physical parameters of Nickel and Water

<table>
<thead>
<tr>
<th></th>
<th>Nickel</th>
<th>Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melting Temperature $T_m$ (K)</td>
<td>1728</td>
<td>273</td>
</tr>
<tr>
<td>Heat of Fusion $L_0$ (J/kg)</td>
<td>292333.9</td>
<td>333605.69</td>
</tr>
<tr>
<td>Thermal Conductivity $k$ (J/m·K·s)</td>
<td>88.5</td>
<td>0.561</td>
</tr>
<tr>
<td>Surface Tension $\sigma$ (J/m²)</td>
<td>0.37</td>
<td>0.07564</td>
</tr>
<tr>
<td>Density of Solid $\rho_S$ (kg/m³)</td>
<td>7905</td>
<td>917.0</td>
</tr>
<tr>
<td>Density of Liquid $\rho_L$ (kg/m³)</td>
<td>8900</td>
<td>999.84</td>
</tr>
<tr>
<td>Specific Heat in Solid $c_{pS}$ (J/kg·K)</td>
<td>649.9</td>
<td>2092.7</td>
</tr>
<tr>
<td>Specific Heat in Liquid $c_{pL}$ (J/kg·K)</td>
<td>656.3</td>
<td>4186.8</td>
</tr>
<tr>
<td>Liquid Kinetic Viscosity $\nu_L$ (m²/s)</td>
<td>$4.901 \times 10^{-3}$</td>
<td>$9.9984 \times 10^{-4}$</td>
</tr>
<tr>
<td>Thermal Volumetric Expansion Coefficient $\beta$ (1/K)</td>
<td>$-1.73 \times 10^{-3}$</td>
<td>$-4.5 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

It is clear from Eqs. (5.1) that $\epsilon \rightarrow 0$ represents an extreme singularity of this model, and in this case neither initial nor boundary conditions can be satisfied. Indeed, the semilinear terms of Eq. (5.1a) dominate. In light of this, one should expect that the unconditional stability of trapezoidal integration for linear constant-coefficient problems will be lost for $\epsilon > 0$ sufficiently small. Moreover, $\epsilon$ is required to be in the appropriate range according to values of $\Delta x$ and $\Delta y$. On the one hand, we will lose resolution of the interface, and interface propagation is no longer smooth if $\epsilon$ is too small (approximately 10 times smaller than the space step). On the other hand, the interface region becomes mushy if $\epsilon$ is too large (see Figure 5.1 for details). Figure 5.1 presents the choice of $\epsilon$ for different space steps. The two dashed curves in Figure 5.1 are open upper and lower bounds of $\epsilon$ corresponding to the spatial step sizes. We use the terminology “open” for the reason that $\epsilon$ has to be chosen within the range of two curves but never reaches the curves. For example, for $\Delta x = \Delta y = 5 \times 10^{-3}$, it is required that $0.0008 < \epsilon < 0.0047$; in addition, $0.0009 \leq \epsilon \leq 0.0046$ is satisfied. Therefore, the numerical error of $\epsilon$ is $10^{-4}$. The minimum time step sizes $\Delta t$ to satisfy stability requirements are $6 \times 10^{-6}$, $10^{-5}$, $2 \times 10^{-5}$, $2.2 \times 10^{-5}$, $4 \times 10^{-5}$ and $6 \times 10^{-5}$ corresponding to $\Delta x = \Delta y = 1.25 \times 10^{-3}$, $2.5 \times 10^{-3}$, $5 \times 10^{-3}$, 0.01, 0.02 and 0.04,
respectively. In the current computations, a fixed $\epsilon = 0.004$ is used to maintain consistency with other authors’ work except in Section 5.3.3, where the effect of $\epsilon$ on the interface will be investigated. The length scale is $w = 2.1 \times 10^{-6}$ m, and $a_0 = 400$. The Neumann boundary condition for temperature implies that the domain $\Omega$ is adiabatic, i.e., it is an adiabatic and thus for the present case also an isenthalpic system.

![Figure 5.1](image.png)

**Figure 5.1** Semilog plot of $\epsilon$ versus $\Delta x$ and $\Delta y$.

### 5.2.2 Comparison between Explicit and Implicit Schemes

We have conducted computations using two different types of time integration schemes as explained in Section 5.1.2. We compare the total arithmetic for one time step and time step size required for stability. We find that, although the ratio of total arithmetic between the implicit and explicit schemes is $O(10)$, the time step size required for stability of the implicit scheme is 10 times greater than that of the explicit scheme with typical space steps $\Delta x = \Delta y = 5 \times 10^{-3}$. Moreover, the time step size must be decreased quadratically with linear decrease of spatial step size to maintain stability of the explicit scheme, while the required time step size for the implicit scheme decreases only linearly. We conclude that
the explicit scheme is simpler, and it is efficient if the spatial step size is not required to be small. However, since we are solving highly nonlinear equations (Eq. (5.1a)), and grid sizes should be small enough to capture the details of dendrite structure, the implicit method is superior for the current problem. Moreover, since we will calculate dendrite growth in a flow field (and in three space dimensions) in the future, leading to a need for even finer spatial resolution (and thus more stringent requirements for stability), we choose to emphasize the implicit scheme for our computations.

5.2.3 Energy Conservation and Quasilinear Iteration Convergence

In this subsection we present results for two specific basic tests performed on the code employing implementation of our implicit method. The first of these checks is basically a physical one: that energy (in the sense of the $L^2$ norm) is conserved in the discrete calculations. The second involves monitoring convergence rates of the nonlinear iterations required within each time step, thus providing information to allow quantitative assessment of the efficiency of the computations.

We compare computed total energy of the system for a fine grid ($\Delta x = \Delta y = 0.005$, $\Delta t = 5 \times 10^{-5}$), a coarse grid ($\Delta x = \Delta y = 0.01$, $\Delta t = 1 \times 10^{-4}$), and using single-precision arithmetic for the coarse grid to verify that energy is conserved in this system. We found that energy of the single-precision coarse grid calculation is decreasing in time, while energies of the coarse and fine grids of the double-precision computations are increasing. Moreover, the rate of energy increase is lower on the fine grid than on the coarse grid for double-precision arithmetic, and the rate of decrease of the single-precision calculation for the coarse grid is greater than the rate of increase of the double-precision calculation on the same grid. We conclude that the energy decrease of single precision for the coarse grid is primarily caused by truncation error, while the energy increase in this system is caused by rounding error; so the rate of energy increase is decreased when we refine the grid. We display the error in total energy (absolute value of computed energy difference between current and initial time) caused by energy increasing as a function of spatial step size in Figure 5.2, which shows that the total energy error is decreasing and approaching zero as we refine the grid. Therefore, the system energy is conserved to within controllable arithmetic errors.
5.2 also shows that the energy error is decreased by a factor of four as we refine the space and time step sizes by a factor of two, which indicates a second-order accuracy in a weak sense.

![Figure 5.2](image)

**Figure 5.2** Total energy error as function of spatial step size $\Delta x$ ($\Delta t = 0.2$).

To analyze convergence of quasilinear iterations, we compare the computations using a fully-coupled technique (i.e., solving Eq. (5.5a) coupled with Eq. (5.5b) followed by solving Eq. (5.6a) coupled with Eq. (5.6b)) and a sequential technique (i.e., solving Eq. (5.5a), (5.6a), (5.5b) and (5.6b) successively). The comparison of $L^\infty$ residual as a function of number of iterations per time step shown in Figure 5.3 indicates that the Newton–Kantorovich (quasilinearization) convergence rate with convergence tolerance $10^{-8}$ of the coupled method is greater than that of the sequential method, as might be expected. In addition, we observe that the CPU time of the sequential method is approximately 10% greater than that of the coupled method. However, storage needed for the coupled method is much greater than that for a sequential method because we must solve a $n \times n$ block-banded matrix system with each block being a $2 \times 2$ matrix for the coupled method rather than a simple $n \times n$ matrix for the sequential method. In the current computations, since the spatial grid spacing should be very small to perform accurate simulations at the interface, large storage is needed. Hence, the sequential method is chosen to reduce this as much as possible.

We also note that the quasilinear iteration does not converge quadratically but only almost log-linearly (i.e., one decimal place of accuracy is gained with each iteration), as
shown in Figure 5.3. We conduct several further numerical tests and find that the expected quadratic convergence rate can be achieved if we use smaller time steps. However, computations with a smaller time step cause an increase of CPU time despite the need for fewer iterations per time step. Therefore, we choose to compute with the larger time step which proves to be sufficiently accurate and stable but leads to more iterations per time step. Figure 5.3 displays results at three different times in parts (a)–(c). It is clear that over the observed time interval the convergence rate of nonlinear iterations is quite insensitive to the (physical) time at which this is monitored.

5.2.4 Grid Function Convergence Tests

Grid function convergence tests are conducted using four different combinations of spatial step and time step sizes as shown in Table 5.2. We start from uniform $h \equiv \Delta x = \Delta y = 0.01$ and $k \equiv \Delta t = 5 \times 10^{-5}$. We then successively decrease $h$ and $k$ simultaneously by a factor of two and compute the $L^2$ norm of solution differences on successive grids for $\phi$ and $\theta$ using four different sets of discrete step sizes: $h$ and $k$, $\frac{h}{2}$ and $\frac{k}{2}$, $\frac{h}{4}$ and $\frac{k}{4}$, and $\frac{h}{8}$ and $\frac{k}{8}$, which are denoted by $\| \phi^h - \phi^{h/2} \|$ and $\| \theta^h - \theta^{h/2} \|$, etc., in Table 5.2. We then compute the ratios $\frac{\| \phi^h - \phi^{h/2} \|}{\| \phi^{h/2} - \phi^{h/4} \|}$ and $\frac{\| \phi^h - \phi^{h/2} \|}{\| \phi^{h/2} - \phi^{h/4} \|}$ to show convergence in the strong $L^2$ sense. We note that the computed order of accuracy is increasing monotonically as we refine the grid, and we are able to obtain essentially second-order accuracy at some fine grid since the theoretical value of four for the error-norm ratios is between the computed values 2.73 and
9.60 for $\phi$, and 3.51 and 10.56 for $\theta$.

<table>
<thead>
<tr>
<th>Table 5.2 Grid function convergence tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>$k$</td>
</tr>
<tr>
<td>$|\phi^h - \phi^{h/2}|$</td>
</tr>
<tr>
<td>$|\phi^{h/2} - \phi^{h/4}|$</td>
</tr>
<tr>
<td>$|\phi^{h/4} - \phi^{h/8}|$</td>
</tr>
<tr>
<td>$|\theta^h - \theta^{h/2}|$</td>
</tr>
<tr>
<td>$|\theta^{h/2} - \theta^{h/4}|$</td>
</tr>
<tr>
<td>$|\theta^{h/4} - \theta^{h/8}|$</td>
</tr>
</tbody>
</table>

To explain this apparent “superconvergence” rate of fine grids, in Figure 5.4 we compare a small part of the dendrite surface for four different grid spacings with tic marks indicating actual dendrite surface locations. We display only $\phi$ because the temperature profile $\theta$ is similar. Recall that the leading truncation error for $\phi$ using centered differencing in space and trapezoidal integration in time is $1/12 (\phi_{xxxx} h^4 + \phi_{ttt} k^2)$, where $\phi_{xxxx}$ and $\phi_{ttt}$ represent the fourth derivative in space and the third derivative in time, respectively. We note that the number of dendrite surface points shown in Figure 5.4, together with the fact that the number of grid points in the mushy zone increases approximately two times faster than the number of grid points of the total domain when we refine the grid, implies both a smoother solution and a somewhat under estimation of actual grid refinement in precisely the region where most of the truncation error occurs. Hence, the high-order derivatives of $\phi$ remain bounded and become small as we refine the grid, and ratios of the $L^2$ norm continue to increase as we refine the grid. Therefore, the formal second-order accuracy should be obtained, even pointwise for a sufficiently fine grid. In any case, we have demonstrated that solutions to the discrete equations do, in fact, converge to those of the phase-field differential system as discretization step sizes are refined.
Figure 5.4  Dendrite surface locations for four different combinations of space and time step sizes with $St = 0.5$ at $t = 0.05$: (a) $\Delta x = \Delta y = 0.01$, $\Delta t = 5 \times 10^{-5}$; (b) $\Delta x = \Delta y = 0.005$, $\Delta t = 2.5 \times 10^{-5}$; (c) $\Delta x = \Delta y = 0.0025$, $\Delta t = 1.25 \times 10^{-5}$; (d) $\Delta x = \Delta y = 0.00125$, $\Delta t = 6.25 \times 10^{-6}$.

5.3 Computed Results of Solidification with Isotropic Surface Energy

In this section, we present the dendrite shapes and temperature fields at three different times with initial Stefan number $St = 0.5$ to demonstrate physical realism of the computations. Nucleation conditions and the physics of crystal growth rates are then discussed. Effects of the free parameter $\epsilon$ on the interface growth rate are also investigated, followed by the effects of the kinetic coefficient $\mu_k$ on the interface growth rate. We then compare the computed interface growth rate with the experimental one. Finally, we describe the effect of the shape of the initial seed on the final computed dendrite shape.

5.3.1  Dendrite Structures at Different Times

Figures 5.5 and 5.6 present the interface locations and temperature fields at three different times $t = 0.05$, 0.1 and 0.2 for the Stefan number $St = 0.5$. In Figure 5.5, dendrite structures at different times display the end-splitting property during dendrite growth and exhibit different growth rates of the various interfacial points. In Figure 5.6, we note that the shape
of isothermal lines does not follow the interfacial shape. The reason is that the interfacial
temperature determined by Eq. (1.6c) is a function of local curvature and interfacial speed.
Therefore, interfacial temperature is not a constant.

![Diagram](image)

**Figure 5.5** Dendrite structures for $St = 0.5$ at time $t = 0.05$, $t = 0.1$ and $t = 0.2$ computed with grid spacing $\Delta x = \Delta y = 0.01$, and $\Delta t = 5 \times 10^{-5}$.

![Temperature Fields](image)

**Figure 5.6** Temperature fields at (a) $t = 0.05$, (b) $t = 0.1$ and (c) $t = 0.2$.

We describe the method used to determine the location of interfacial points via Figure 5.7.
The intersections of grid lines and the fictitious surface $\phi = 0.5$ are considered as points on
the interface. The criterion that an interface point must be on the cell wall, where the phase-
field value $\phi$ is changing from less than 0.5 to greater than 0.5, is applied to determine these
intersections. Locations of intersections on a cell wall are computed by linear interpolation
of phase field $\phi$ at two adjacent grid points. Two flags are introduced: one is the counter for
the number of interface points in one cell; the other is the label of the cell boundary with or without an interface point. For example, 1 denotes a cell wall containing an interface point; 2 marks a cell wall without an interface point, or with an interface point for which computations are complete. Finally, 0 corresponds to a new cell wall at which no analysis has yet been done.

Initially all the cell boundaries are labeled as 0. The first step is to locate the interface points in the current cell. For example, we start from grid point \((i - 1, j - 1)\) in cell \((i, j)\) and choose the bottom boundary as the initial cell wall to be examined; then we proceed in a clockwise direction around the cell boundary. Phase-field values \(\phi\) on grid points \((i - 1, j - 1)\), \((i - 1, j)\) and \((i, j)\) are \(\phi > 0.5\), \(\phi < 0.5\) and \(\phi > 0.5\), respectively. Therefore, there exist two interface points on the boundary of cell \((i, j)\): one is between grid points \((i - 1, j - 1)\) and \((i - 1, j)\), and the other is between grid points \((i - 1, j)\) and \((i, j)\). Hence, the left and upper cell boundaries that have interface points on them are labeled as 1; the others (on the same
cell) are labeled as 2. Moreover, for cell \((i, j)\), the number of interface points, 2, is stored in an interface-points counter. The second step is to compute the location of interface points \(\text{1}\) and \(\text{2}\) and the interfacial velocity of each point in the current cell. After the computations, the labels “1” of cell boundaries are changed to “2”, preventing interface point \(\text{1}\) from being counted again in cell \((i - 1, j)\) and interface point \(\text{2}\) from being counted a second time in cell \((i, j + 1)\). In Figure 5.7 “1 \(\rightarrow\) 2” indicates that the cell wall is labeled as 1 in the first step, then changed to 2 in the second step of the calculations. We now move to the next cell, \((i, j + 1)\). This cell is selected based on the last interface point location in the previous cell. For example, the last point on the bottom, left, top or right cell boundary leads to moving to the lower, left, upper or right cell, respectively. It should be mentioned that if the interface is smooth enough, there must be two interface points in each cell through which it passes. Failure of this implies that the grid being employed is not sufficiently refined. Our algorithm prints an error message and moves to the next time step if this is detected. These steps are summarized in the following algorithm.

**Algorithm 5** Suppose \(\phi\) and \(\theta\) have been computed using Eqs. (5.5) and (5.6) as approximations to Eqs. (5.1) at time step \(n\). Label all cell boundaries as 0, and carry out the following steps to determine the location of the interface:

1. Start from a grid point (e.g., center point of the domain), check the value of \(\phi_{i,j}\) in positive \(x\) direction until \(\phi_{i-1,j} < 0.5\) and \(\phi_{i,j} > 0.5\). Then cell \((i, j)\) is the starting cell.

2. Set point-counter = 0

3. If \((\phi_{i-1,j} < 0.5 \text{ and } \phi_{i,j} > 0.5)\) or \((\phi_{i-1,j} > 0.5 \text{ and } \phi_{i,j} < 0.5)\)

   \[\text{LabelI}(i - 1, j) = 1,\]

   \[\text{point-counter} = \text{point-counter} + 1\]

   else

   \[\text{LabelI}(i - 1, j) = 2\]

4. If \((\phi_{i-1,j} < 0.5 \text{ and } \phi_{i-1,j+1} > 0.5)\) or \((\phi_{i-1,j} > 0.5 \text{ and } \phi_{i-1,j+1} < 0.5)\)

   \[\text{LabelJ}(i - 1, j) = 1,\]

   \[\text{point-counter} = \text{point-counter} + 1\]
else

LabelJ(i - 1, j) = 2

5. If (\(\phi_{i-1,j+1} < 0.5\) and \(\phi_{i,j+1} > 0.5\)) or (\(\phi_{i-1,j+1} > 0.5\) and \(\phi_{i,j+1} < 0.5\))

LabelI(i - 1, j + 1) = 1,

point-counter = point-counter + 1

else

LabelI(i - 1, j + 1) = 2

6. If (\(\phi_{i,j} < 0.5\) and \(\phi_{i,j+1} > 0.5\)) or (\(\phi_{i,j} > 0.5\) and \(\phi_{i,j+1} < 0.5\))

LabelJ(i, j) = 1,

point-counter = point-counter + 1

else

LabelJ(i, j) = 2

7. Check smoothness:

If point-counter > 2

print “no smooth interface,”

go to 12

8. If LabelI(i - 1, j) = 1

compute interface-point location

LabelI(i - 1, j) = 2,

point-counter = point-counter - 1

If point-counter = 0

i = i,

j = j - 1,

go to 2

9. If LabelJ(i - 1, j) = 1

compute interface-point location

LabelJ(i - 1, j) = 2,

point-counter = point-counter - 1
If point-counter = 0
    \[ i = i - 1, \]
    \[ j = j, \]
    go to 2

10. If \( LabelI(i-1, j+1) = 1 \)
    compute interface-point location
    \( LabelI(i-1, j+1) = 2, \)
    point-counter = point-counter - 1
    If point-counter = 0
    \[ i = i, \]
    \[ j = j + 1, \]
    go to 2

11. If \( LabelJ(i, j) = 1 \)
    compute interface-point location
    \( LabelJ(i, j) = 2, \)
    point-counter = point-counter - 1
    If point-counter = 0
    \[ i = i + 1, \]
    \[ j = j, \]
    go to 2

12. Continue to solve Eqs. (5.5) and (5.6) at time level \( n+1 \)

This algorithm works well for a simple interface separating solid and liquid phases but lacks the sophistication needed to handle cases for which the solid region is not simply connected. We note that in this algorithm only one interface point is allowed on each cell boundary; and only two interface points are considered to compose a smooth interface in one grid cell. On the one hand, multiple interface points on a cell boundary cannot be computed from the phase-field values obtained by the currently used grid spacing. We explain this phenomena via an example of interface point on one cell boundary as in Figure 5.8. We can
find one interface point where $\phi = 0.5$ if the phase-field values of two grid points on this cell boundary are $\phi > 0.5$ and $\phi < 0.5$, respectively, as shown in Figure 5.8(a), no interface point if the phase-field values are both $\phi > 0.5$ as in Figure 5.8(b). However, the situation in (b) might be altered if we refine the grid and find the phase-field value of the mid-point to be $\phi < 0.5$ as indicated in Figure 5.8(c). Therefore, a situation like Figure 5.8(b) has two possibilities: one is no interface point and the other is multiple interface points. We can obtain more accurate details of the dendrite structure if we further refine the grid. In addition, more than two interface points in one grid cell requires the refinement of the grids. On the other hand, the current algorithm considers only one interface separating solid and liquid phases. It does not consider the existence of multiple interface in the solid phase.

![Sketch of interface point on one cell boundary](image)

**Figure 5.8** Sketch of interface point on one cell boundary: (a) one interface point; (b) no interface point; (c) two interface points for finer grid.

### 5.3.2 Growth Rates

In this subsection the behaviors of dendrites under various supercoolings are studied, and a comparison of growth rate at different Stefan numbers is presented. In addition, the relationship between growth rate and Stefan number is discussed. The domain size is enlarged to $\Omega \equiv [0,8] \times [0,8]$ to accommodate the larger dendrites that occur when the Stefan number is large. The grid spacing is uniform $\Delta x = \Delta y = 0.01$, and $\Delta t = 5 \times 10^{-5}$. We conduct the simulation until dimensionless time $t = 0.1$. Computed growth rates, $V_n$, are the averaged ones; i.e., $V_n = d\sqrt{A}/dt$, where $A$ is the solid-phase area. Glicksman and
Schaefer [107] concluded that the interface is in dendrite form in a moderately supercooled melt \((St < 1)\), and “scallops” form in a hypercooled melt \((St > 1)\); then a quasi-planar form (a form of interface with only small curvature) occurs if supercooling is increased further. Figure 5.9 illustrates that the computed interface shapes produced by the phase-field model follow these same patterns: interface shapes at time \(t = 0.1\) with three different Stefan numbers 0.5, 1.0 and 1.4 are plotted. This provides convincing evidence of the transition from dendritic to “scalloped,” and then to the quasi-planar configuration. We could expect that as supercooling continues to increase, the interface becomes almost indistinguishable from a planar interface, from a morphological standpoint.

![Figure 5.9](image.png)

**Figure 5.9** Interface shapes at \(t = 0.1\) for three different Stefan numbers: (a) \(St = 0.5\) (dendritic), (b) \(St = 1.0\) (scalloped) and (c) \(St = 1.4\) (quasi-planar).

The similarity solution of a classical Stefan problem provided in Alexiades and Solomon [5] and Davis [7] indicates that for one-dimensional freezing into a supercooled melt \((St < 1)\), interface location as a function of time is \(X \sim t^{\frac{1}{2}}\), i.e., \(V_n \sim t^{-\frac{1}{2}}\). Moreover, there exists no solution for \(St > 1\) because without kinetic supercooling, the heat required to raise the temperature to \(T_m\) is higher than the latent heat provided, violating the energy balance. However, in the presence of kinetic supercooling, there are solutions for one-dimensional freezing into a supercooled melt for all \(St\): \(X \sim t^{\frac{1}{2}}\) corresponding to \(V_n \sim t^{-\frac{1}{2}}\) for \(St < 1\), \(X \sim t^{\frac{3}{2}}\) corresponding to \(V_n \sim t^{-\frac{3}{2}}\) for \(St = 1\), and \(X \sim t\), and thus \(V_n \sim constant\) for \(St > 1\) [7]. We observe in the scalings relating \(V_n\) and \(t\) that the power of \(t\) tends to increase with increasing \(St\).

Absence of analytical solutions for two-dimensional freezing into a supercooled melt on a
rectangular domain implies that no analytical expression for interfacial speed as a function of time has been obtained. But the computational solutions shown in Figure 5.10 provide this relation as follows: for $St < 1$, $0.3 t^{-2/5}$ and $2t^{-2/5}$ are used to represent the interfacial speeds for $St = 0.1$ and $St = 0.5$, respectively, to a good approximation over the entire time interval being considered; for $St = 1$, the curve $16t^{-1/5}$ can be used to approximate the computed results; and for $St > 1$ (only results for $St = 1.4$ are shown in Figure 5.10), the growth rate is essentially constant at late times. It is clear that the trend is the same as that seen in the 1-D analytical solution of Davis [7]; that is, we observe an increasing power of $t$ for $V_n$ with the increase of supercooling $St$, which follows the analytical solution trend.

![Figure 5.10](image)

**Figure 5.10** Computed interfacial speed as function of time for different Stefan numbers: (a) $St = 0.1$; (b) $St = 0.5$; (c) $St = 1.0$; (d) $St = 1.4$.

We note that the power of $t$ is essentially fixed if $St < 0.5$, but increases with $St$ if $0.5 < St < 1.0$ (not shown in Figure 5.10). This observation can be explained by the relation between the averaged interfacial speed and the Stefan number displayed in Figure 5.11. The
existence of a quadratic relationship between \( V_n \) and \( St \) is related to the lateral mechanism for dendrite growth with \( St < 1 \) and the continuous mechanism with \( St > 1 \) by applying Eqs. (2.13) and (2.14). This implies that these growth mechanisms are determined by the physical properties of pure nickel, and they would be somewhat different for various other materials. The quadratic rather than linear relationship for \( St < 1 \) also explains why the power of \( t \) is increasing as \( St \) increases. We also observe that there is no apparent discontinuity in slope in the neighborhood of \( St = 1 \) in Figure 5.11, where the system is transitioning from supercooled to hypercooled. This observation is coincident with the experimental evidence of Glicksman [107] and in conflict with expectations arising from elementary analyses of the classical Stefan problem. Wang and Sekerka [108] investigated the relations between the tip velocity and Stefan number using a phase-field model at large supercooling. Their results are very similar to our results shown in Figure 5.11 for large \( St \).

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure5.11.png}
\caption{Computed interfacial speed versus Stefan number at \( t = 0.1 \).}
\end{figure}

5.3.3 Influence of \( \epsilon \) on Interface

As we mentioned before, a sharp interface model for phase-transition problems can be smoothed using the phase-field model. Moreover, the interfacial thickness parameter \( \epsilon \) is a free parameter, i.e., it can be changed considerably without causing a significant difference in the development of the interface as noted by Caginalp and Socolovsky [109]. To some extent this is rather counterintuitive; therefore, in this subsection we conduct several numerical
tests to investigate the influence of $\epsilon$ on details of the interface, including interfacial speed and interface structure. Space and time step sizes $\Delta x = \Delta y = 0.01$ and $\Delta t = 5 \times 10^{-5}$, respectively, are used. We found that the interface becomes mushy if $\epsilon \geq 0.006$, and the numerical scheme is no longer stable if $\epsilon \leq 0.001$ for these discrete step sizes. Hence, we display the numerical results for four different values of $\epsilon$: 0.002, 0.003, 0.004 and 0.005. In Figure 5.12 we compare interfacial speed versus time for these values of $\epsilon$. The results shown here indicate that the change of interfacial thickness will not significantly alter the interface from the point of view of mean propagation speed. Nevertheless, the interface structure changes with $\epsilon$ as shown in Figure 5.13. We note, however, that since the interface structure is difficult to predict quantitatively, and it changes significantly with perturbations introduced to the system, it does not seem to provide a very suitable property for comparison of the numerical results.

![Figure 5.12](image.png)

**Figure 5.12** Computed interfacial speed for different interfacial thickness parameter $\epsilon$ at $St = 0.5$.

### 5.3.4 Effect of Kinetic Coefficient $\mu_k$

We present the interface growth rate and dendrite structure at different values of kinetic coefficient $\mu_k$ in this subsection to demonstrate the effect of $\mu_k$. The nondimensional parameter corresponding to $\mu_k$ is $m_k$. Hence, we display the averaged interface growth rate $V_n$.
and the dendrite structure for three different cases: $m_k = 0.025, 0.05, 0.075$—corresponding to $\mu_k = 1.54, 3.08, 6.16$ $J/kg \cdot K$—in Figures 5.14 and 5.15. Figure 5.14 shows that the interfacial speed increases with an increasing kinetic coefficient, which can be deduced from the physics of $\mu_k$. We have explained in Section 2.3 how the kinetic coefficient reflects the interface propagation rate. The higher the kinetic coefficient, the faster the rate. However, differences in $\mu_k$ do not produce too much difference on $V_n$. Moreover, we observe that as the kinetic coefficient becomes larger, the dendrite structure becomes more complicated as shown in Figure 5.15. This phenomena indicates that the kinetic coefficient acts as a perturbation parameter. We have also found that the interface region becomes mushy if $m_k \geq 0.075$. The mushy interface makes the computation of the interface growth rate inaccurate. Therefore, we observe in Figure 5.14 that the behavior of $V_n$ versus time for $m_k = 0.075$ is not similar to those for $m_k = 0.025$ and 0.05.

5.3.5 Comparison of Interface Growth Rates between Computed Results, Asymptotic Solutions and Experimental Data

Brener [37] analytically investigated the effect of anisotropy of surface energy and kinetics on the velocity and the direction of two-dimensional dendrite growth. He derived expressions for velocity of dendrite growth in different limiting cases of surface energy and kinetics. The asymptotic behavior for dendrite growth rate in the limit of small surface energy, i.e., the
capillary length $d_0$ is small, given in [37] obeys

$$V_n = \frac{Pe \epsilon'^{5/4} \mu_k}{\gamma_0},$$  \hspace{1cm} (5.7)

if the degree of supercooling is moderate. Here $Pe$ is the Peclet number; $\epsilon'$ is the degree of anisotropy, and $\gamma_0$ is a constant which can be determined from analytical asymptotics. The dependence of the Stefan number on the Peclet number $Pe$ can be obtained from the
Ivantsov parabola solution:

\[
St(Pe) = 2\sqrt{Pe} \int_{\sqrt{Pe}}^{\infty} e^{-x^2} dx \quad \text{in 2D},
\]

\[
St(Pe) = Pe \int_{Pe}^{\infty} \frac{e^{-x}}{x} dx \quad \text{in 3D}.
\]

Therefore, the relation between dendrite growth rate \(V_n\) and degree of supercooling \(St\) can be determined by computing \(Pe\) for a given \(St\) from Eqs. (5.8), then substituting \(Pe\) into Eq. (5.7) to obtain \(V_n\). We remark here that the asymptotic behavior of \(V_n\) in Eq. (5.7) is derived in 2D, and we could extend this relation to 3D using the Peclet number computed from Eq. (5.8b).

We note that the constant \(\gamma_0\) in Eq. (5.7) does not affect the qualitative behavior of \(V_n\). Hence, we determine \(\gamma_0\) via minimizing the difference between \(V_n\) predicted by Brener [37] in 2D and that computed from the phase-field model in 2D, and minimizing the difference of \(V_n\) predicted by Brener in 3D and that obtained from the experiments in 3D, respectively. In other words, we minimize the \(L^2\)-norm \(\|V_n(\text{Brener}) - V_n(\text{phase-field})\|\) in 2D, and the \(L^2\)-norm \(\|V_n(\text{Brener}) - V_n(\text{experiment})\|\) in 3D. We then obtain \(\gamma_0 = 7 \times 10^{-4}\) for 2D and \(\gamma_0 = 1 \times 10^{-3}\) for 3D.

We dimensionalize the interfacial speed obtained from the phase-field model (5.1) to make an equivalent comparison with the asymptotic solutions and the experimental data. The computed results and the analytical solutions are calculated in the range of 0.05 \(\leq St \leq 0.8\), and the experimental data for pure nickel are provided by Herlach et al. [38] within 0.08 < \(St < 0.6\). The computed \(V_n\) obtained from the phase-field model are 2-D results, while the experiments to determine the growth rate for pure nickel are conducted in 3D. Therefore, we cannot make direct comparison between the computed \(V_n\) and the experimental one. We thus compare the simulated results with the analytical ones in 2D, and the experimental data with the analytical results in 3D to make qualitative comparison.

The computed, asymptotic and experimental interfacial propagation speeds \(V_n\) are plotted versus the Stefan number \(St\) as displayed in Figure 5.16. The results in Figure 5.16 show that \(V_n\) computed from the 2-D phase-field model qualitatively follows the asymptotic solution. We also observe that the difference between the computed and analytical \(V_n\) is increasing for \(St > 0.6\) or if \(St <\sim 0.2\). Recall that the derivation of \(V_n\) in Eq. (5.7) is based
Experimental data
Phase-field simulation
Brener theory (2D)
Brener theory (3D)

Figure 5.16 Interfacial propagation speed versus Stefan number: comparison between computed results from the phase-field model, solutions obtained from Brener’s theory in 2D and 3D, and experimental data.

on the assumption that the degree of supercooling is moderate. Therefore, Eq. (5.7) is not appropriate for predicting \( V_n \) if \( St > 0.6 \). In addition, we draw the same conclusion in 3D as in 2D from comparison between the experimental data and the 3-D analytical results. Comparison between the experimental data and the analytical results in 3D shows that the experimental data are qualitatively consistent with the analytical ones when \( St \) is between 0.2 and 0.6. The disagreement for \( St < 0.2 \) is attributed to the forced convective flow in the droplets and impurities in the “nominally pure” nickel samples during the experiments [38]. It is worth mentioning that comparison between the computed solutions obtained from the phase-field model by Nestler et al. [35], the analytical predictions by Brener and the experimental data in [38] presents results similar to ours.

5.3.6 Effect of Initial Shape

Figure 5.17 presents a sketch of approximating a disk with a union of squares as is necessary on a Cartesian grid. To apply the disk-shaped initial condition generally used, the phase-field value \( \phi \) of the squares inside the circle is assigned a value zero, corresponding to the solid phase. If the distance between a grid point and the disk’s center is in the range
of \( r_d - \left( r_d + \frac{\sqrt{2}}{2} \Delta x \right) \), the phase-field value \( \phi \) of this grid point is also prescribed to be zero. Therefore, the shaded area in Figure 5.17 represents the solid phase. The representation of the union of squares has \( \pm 5\% \) error of the area.

![Figure 5.17 Approximation to round initial shape.](image)

A comparison of Figure 5.5 and Figure 5.18 shows the large difference of the dendrite structure between two different initial shapes, square and round, at all times displayed. The differences are produced by the local curvature of the initial shape. The larger the curvature, the higher the interface velocity. For an ideal disk, the same curvature at every point causes the interface to propagate at the same speed initially, uniformly over its extent. However, since a union of squares is used to approximate the sphere in our case, there exists a difference of curvature from point to point, hence, producing dendrites. Moreover, the varieties of rounding error introduced by a square initial shape and a union of squares in the case of round initial shape introduce different fluctuations, and fluctuations are one of the (physical) necessities for dendrite formation, which explains the existence of various physical dendrite structures. Thus, the phase-field model is able to mimic physics even at this level of detail.

### 5.4 Computed Results of Solidification with Anisotropic Surface Energy

We have conducted numerical simulations of solidification of nickel with isotropic interfacial energy in Section 5.3 to make comparisons with analytical solutions and experimental
Figure 5.18  Dendrite structures for approximately round initial seed at time \( t = 0.05 \), \( t = 0.1 \) and \( t = 0.2 \) (outward direction) with grid size \( \Delta x = \Delta y = 0.01 \) and \( \Delta t = 5 \times 10^{-5} \) for \( St = 0.5 \).

Data. Here, we first demonstrate a qualitative consistency for critical radius of solidification between 2-D theoretical results and an implementation of the phase-field model to complete the validation of this model. We will introduce anisotropic interfacial energy in the following part of this section, and emphasize the influence of parameters (e.g., degree of anisotropy \( \epsilon' \) and interfacial thickness \( \delta \)) and the initial conditions on the dendrite structures. Solidification is simulated for supercooled nickel with four-fold anisotropy. Two-dimensional governing equations (3.30) will be applied in this section with the assignment \( u = v = 0 \) in domain \( \Omega \).

5.4.1 Critical Initial Area

In this subsection we report comparisons of the preceding theoretical results with results obtained from numerical simulations using the phase-field model of solidification. Since the comparisons are made based on different \( \Delta c_p \) (\( \Delta c_p = 0 \), constant and \( f(T) \)) as in Section 2.1, the specific heat \( c_p \) is computed by Eq. (3.14)

\[
c_p = [1 - w(\phi)] c_{pS} + w(\phi)c_{pL}.
\]  \hspace{1cm} (5.9)

where \( w(\phi) \) is the mass fraction of liquid.

In the present study, we have conducted numerical simulations on pure substances, nickel (metal) and water (non-metal), to demonstrate the phase-field model is able to predict the
existence of critical radius and to obtain a quantitative relation between critical radius and degree of supercooling. Physical parameters of nickel and water were provided in Table 5.1. Since there is no general strategy for determining the kinetic coefficient \( \mu_k \), we assume that it is of \( O(1) \) for pure metal and \( O(10^{-2}) \) for water, and choose \( \mu_k = 3.08 m/s \cdot K \) for nickel and \( \mu_k = 0.05 m/s \cdot K \) for water such that a thin interface can be observed. The interfacial thickness \( \delta \) is chosen based on the spatial step size \( \Delta x \) and \( \Delta y \), and the relation between \( \delta \) and \( \Delta x (\Delta y) \) can be found in Section 5.2.1.

We first present simulations for the pure metal, nickel. The computational domain \( \Omega \equiv [0, 5.25 \times 10^{-7} m] \times [0, 5.25 \times 10^{-7} m] \) is chosen to be an insulated square region representing an “infinite” domain with respect to the size of the growing solid region, and anisotropy of surface tension and kinetic coefficient is introduced.

Initially, a small square with area \( A \) and temperature \( T_m \) is placed at the center of the domain to represent a nucleus. The nucleus grows if \( A \) is greater than the critical area \( A^* \) and shrinks otherwise. Therefore, area of the nucleus as a function of time is an indicator of growth or shrinkage. The critical area is given by Eq. (2.9) if \( \Delta c_p = 0 \). In addition, the critical area can be computed by Eq. (2.4) with \( \Delta g_V \) given by Eq. (2.10) if \( c_{pL} \) and \( c_{pS} \) are constants. The critical areas for nickel are listed in Table 5.3 at different degrees of supercooling \( \Delta T \).

<table>
<thead>
<tr>
<th>( \Delta T = T_m - T(K) )</th>
<th>( A^*(m^2) (\Delta c_p = 0) )</th>
<th>( A^*(m^2) (\Delta c_p = 6.4) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 2.4048 \times 10^{-13} )</td>
<td>( 2.4047 \times 10^{-13} )</td>
</tr>
<tr>
<td>2</td>
<td>( 6.0120 \times 10^{-14} )</td>
<td>( 6.0118 \times 10^{-14} )</td>
</tr>
<tr>
<td>3</td>
<td>( 2.6720 \times 10^{-14} )</td>
<td>( 2.6718 \times 10^{-14} )</td>
</tr>
<tr>
<td>4</td>
<td>( 1.5030 \times 10^{-14} )</td>
<td>( 1.5029 \times 10^{-14} )</td>
</tr>
<tr>
<td>5</td>
<td>( 9.6193 \times 10^{-15} )</td>
<td>( 9.6182 \times 10^{-15} )</td>
</tr>
<tr>
<td>6</td>
<td>( 6.6800 \times 10^{-15} )</td>
<td>( 6.6792 \times 10^{-15} )</td>
</tr>
<tr>
<td>7</td>
<td>( 4.9078 \times 10^{-15} )</td>
<td>( 4.9070 \times 10^{-15} )</td>
</tr>
<tr>
<td>8</td>
<td>( 3.7575 \times 10^{-15} )</td>
<td>( 3.7569 \times 10^{-15} )</td>
</tr>
<tr>
<td>9</td>
<td>( 2.9689 \times 10^{-15} )</td>
<td>( 2.9683 \times 10^{-15} )</td>
</tr>
<tr>
<td>10</td>
<td>( 2.4048 \times 10^{-15} )</td>
<td>( 2.4043 \times 10^{-15} )</td>
</tr>
</tbody>
</table>

We note that the difference between the analytical critical areas for nickel when \( \Delta c_p = 0 \) and \( \Delta c_p = 6.4 J/kg \cdot K \) is negligible if \( \Delta T \) is small (the highest \( \Delta T \) in Table 5.3 corresponds
to \( St = 0.022 \), where \( St = c_p L \Delta T / L_0 \) is the Stefan number). Therefore, it is reasonable to make the assumption that \( \Delta c_p = 0 \) for metal as in Eq. (2.7) when \( \Delta T \) is not too high.

We present the variations of solid-phase area with time to show the growth or shrinkage caused by different initial areas in Figures 5.19, 5.20 and 5.21 with three different degrees of supercooling: \( \Delta T = 3, 6 \) and 10, for the case \( \Delta c_p = 0 \). Spatial step size \( \Delta x = \Delta y = 5.25 \times 10^{-9} m \) and time step \( \Delta t = 5 \times 10^{-13} s \) are employed. We remark that all computations employed an algorithm reported by Xu et al. [110] that has demonstrated second-order accuracy in both space and time; hence, spatial errors are \( O(10^{-18} m) \), which is three to four orders of magnitude smaller than the computed areas. Similarly, the temporal errors are \( O(10^{-26} s) \), permitting accurate resolution of growth or shrinkage of the seed during early times of its evolution.

We find that if the initial area \( A_0 \) is smaller than the critical area \( A^* \) listed in Table 5.3, the solid-phase area will decrease with time, indicating a shrinking seed; on the contrary, \( A_0 \) larger than \( A^* \) will lead to increase of the solid area, implying a growing seed. Therefore, a computed critical area exists between the initial area in Figures 5.19(a), 5.20(a) and 5.21(a) and that in Figures 5.19(b), 5.20(b) and 5.21(b), respectively.

![Figure 5.19](image.png)

**Figure 5.19** Solid phase area versus time for nickel, \( \Delta T = 3 K \): (a) Initial area \( A_0 = 2.1412 \times 10^{-14} m^2 \); (b) Initial area \( A_0 = 2.7952 \times 10^{-14} m^2 \).

To verify the existence of computational critical area, and that the relation between critical area and the degree of supercooling obeys Eq. (2.4), we conduct simulations at degrees of supercooling \( \Delta T = 3 \rightarrow 10 \) under two different conditions: one is \( c_p S = c_p L \), the
other is that $c_{ps}$ and $c_{pL}$ are unequal constants. We observe that the computed critical areas are qualitatively very similar to the theoretical ones in both cases as shown in Figure 5.22. The two points connected by the vertical line for each value of $\Delta T$ in Figure 5.22 are two computational critical areas that cause the nucleus to shrink (the point below the analytical line) or grow (the point above the analytical line). It should be mentioned that the vertical lines in Figure 5.22 do not indicate continuous change of computational critical areas from one value to another, since the values of computational critical areas are discrete, and the computational critical areas shown in those figures are the closest values to the analytical ones at the chosen grid sizes. Results in Figure 5.22 also indicate that the critical area is

**Figure 5.20** Solid phase area versus time for nickel, $\Delta T = 6K$: (a) Initial area $A_0 = 5.2312 \times 10^{-15}m^2$; (b) Initial area $A_0 = 6.8904 \times 10^{-15}m^2$.

**Figure 5.21** Solid phase area versus time for nickel, $\Delta T = 10K$: (a) Initial area $A_0 = 1.7489 \times 10^{-15}m^2$; (b) Initial area $A_0 = 2.7411 \times 10^{-15}m^2$. 
Figure 5.22 Comparison of analytical and computational critical areas at different $\Delta T$ for nickel: (a) $\Delta c_p = 0$, (b) $\Delta c_p \neq 0$.

decreasing as $\Delta T$ increases, as predicted by Eq. (2.4).

We have also conducted simulations for water to investigate the relation between critical area and degree of supercooling for a non-metal material. The computational domain for this case is $\Omega = [0, 0.1\mu m] \times [0, 0.1\mu m]$. Again, physical and thermodynamical parameters are listed in Table 5.1. We provide the critical areas for water in Table 5.4 at different $\Delta T$ in three cases: $c_{pS} = c_{pL}$, $c_{pS}$ and $c_{pL}$ unequal constants, and $c_{pL}$ a function of temperature with $c_{pS}$ constant. In the third case, we choose $c_{pL} = 1.133 \times 10^4 + 84.53T - 0.177T^2 + 1.373 \times 10^{-4}T^3 + 2.139 \times 10^8/T^2$ [111] to represent the specific heat in water.
Table 5.4 Analytical critical area of water

<table>
<thead>
<tr>
<th>ΔT = Tm − T (K)</th>
<th>A* (∆cp = 0) (m²)</th>
<th>A* (∆cp = 2094.1) (m²)</th>
<th>A* (∆cp = f(T)) (m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.2041 × 10⁻¹⁴</td>
<td>1.1965 × 10⁻¹⁴</td>
<td>1.1965 × 10⁻¹⁴</td>
</tr>
<tr>
<td>2</td>
<td>3.0102 × 10⁻¹⁵</td>
<td>2.9726 × 10⁻¹⁵</td>
<td>2.9726 × 10⁻¹⁵</td>
</tr>
<tr>
<td>3</td>
<td>1.3379 × 10⁻¹⁵</td>
<td>1.3129 × 10⁻¹⁵</td>
<td>1.3129 × 10⁻¹⁵</td>
</tr>
<tr>
<td>4</td>
<td>7.5254 × 10⁻¹⁶</td>
<td>7.3391 × 10⁻¹⁶</td>
<td>7.3388 × 10⁻¹⁶</td>
</tr>
<tr>
<td>5</td>
<td>4.8163 × 10⁻¹⁶</td>
<td>4.6677 × 10⁻¹⁶</td>
<td>4.6674 × 10⁻¹⁶</td>
</tr>
<tr>
<td>6</td>
<td>3.3446 × 10⁻¹⁶</td>
<td>3.2213 × 10⁻¹⁶</td>
<td>3.2209 × 10⁻¹⁶</td>
</tr>
<tr>
<td>7</td>
<td>2.4573 × 10⁻¹⁶</td>
<td>2.3519 × 10⁻¹⁶</td>
<td>2.3516 × 10⁻¹⁶</td>
</tr>
<tr>
<td>8</td>
<td>1.8814 × 10⁻¹⁶</td>
<td>1.7895 × 10⁻¹⁶</td>
<td>1.7891 × 10⁻¹⁶</td>
</tr>
<tr>
<td>9</td>
<td>1.4865 × 10⁻¹⁶</td>
<td>1.4051 × 10⁻¹⁶</td>
<td>1.4048 × 10⁻¹⁶</td>
</tr>
<tr>
<td>10</td>
<td>1.2041 × 10⁻¹⁶</td>
<td>1.1311 × 10⁻¹⁶</td>
<td>1.1307 × 10⁻¹⁶</td>
</tr>
</tbody>
</table>

The data in Table 5.4 indicate that the critical area for Δcp = 0 is different from that for constant Δcp; however, the difference of the critical area between constant Δcp and Δcp as function of T can be neglected for small ΔT (in the current research, the highest ΔT corresponds to St = 0.126). As a consequence, it is evident that constant Δcp would be sufficient to conduct the simulations for water in this work.

Figure 5.23 Solid phase area versus time for water, ΔT = 3K: (a) Initial area A₀ = 1.2933 × 10⁻¹⁵ m²; (b) Initial area A₀ = 1.5465 × 10⁻¹⁵ m².

Similar to the computations for nickel, we first exhibit the growth or shrinkage of seeds having different initial area in Figures 5.23, 5.24 and 5.25 for three different degrees of supercooling: ΔT = 3, 6 and 10. We consider the constant Δcp case in this part. Grid
Figure 5.24  Solid phase area versus time for water, $\Delta T = 6K$: (a) Initial area $A_0 = 3.1550 \times 10^{-16} m^2$; (b) Initial area $A_0 = 3.7332 \times 10^{-16} m^2$.

Figure 5.25  Solid phase area versus time for water, $\Delta T = 10K$: (a) Initial area $A_0 = 1.1150 \times 10^{-16} m^2$; (b) Initial area $A_0 = 1.4780 \times 10^{-16} m^2$.

spacings $\Delta x = \Delta y = 1 \times 10^{-9} m$ and time step $\Delta t = 4 \times 10^{-13} s$ are employed. We again conclude that if the initial area $A_0$ is smaller than the critical area $A^*$ listed in Table 5.4, the decreasing solid area indicates a shrinking seed (as shown in Figures 5.23(a), 5.24(a) and 5.25(a)), while $A_0 > A^*$ leads to the increase of solid area, i.e., a growing seed as indicated in Figs 5.23(b), 5.24(b) and 5.25(b).

Comparisons of analytical and computational critical areas for all three cases are provided in Figure 5.26. As in the case for nickel, simulations are conducted for a range of $\Delta T$ from 3 to 10. We note that the computed critical areas qualitatively follow the theoretical ones in all cases shown in Figure 5.26, which indicates existence of a computational critical area,
and its relation with $\Delta T$ obeys Eq. (2.4).

Simulations of the critical area for nickel and water indicate that critical areas computed from the phase-field model are qualitatively the same as the theoretical ones, which demon-

![Figure 5.26](image)

**Figure 5.26** Comparison of analytical and computational critical areas at different $\Delta T$ for water: (a) $\Delta c_p = 0$, (b) $\Delta c_p$ constant, (c) $\Delta c_p$ a function of temperature.
strates that the phase-field model is able to capture the physics of solidification associated with existence of a critical radius of seeds. It is also shown that the often-used assumptions $\Delta c_p = 0$ for metal and $\Delta c_p = \text{constant}$ for water are reasonable.

We further observe that the scales for interface thickness and time step size $\Delta t$ are sufficiently small as to correspond to scales of atomic absorption and attachment of atoms at the interface [25]. In particular, such small scales make it possible for the phase-field model to simulate the nucleation of the first stage of solidification. The consistency shown here between theoretical critical areas and those computed via the phase-field model indicate that the phase-field model is able to capture the effects of basic small-scale physics of solidification despite the fact that it is a macroscopic model; this is made possible by its formulation which permits it to exhibit behaviors akin to renormalization discussed in Karma and Rappel [112] and elsewhere.

5.4.2 Effects of Degree of Anisotropy $\epsilon'\!$

The computational domain $\Omega \equiv [0, L] \times [0, L] = [0, 2.52 \times 10^{-5}m] \times [0, 2.52 \times 10^{-5}m]$ is chosen. Grid sizes $\Delta x = \Delta y = 3.15 \times 10^{-8}m$ and $\Delta t = 10^{-12}s$ are applied to all computations. The kinetic coefficient $\mu_k = 3.08J/kg\cdot K$ is used. The interfacial thickness $\delta = 1.05 \times 10^{-8}m$ is chosen to guarantee stability of the numerical scheme and a smooth interface. However, we will use $\delta = 1.26 \times 10^{-8}m$ in Section 5.4.3 to compare the dendrite structure at different interfacial thickness. Four-fold anisotropy is typical for pure metal, nickel. Hence, the number of folds in Eq. (3.32) is $m = 4$. The initial Stefan number $St = 0.5$ will be applied for all the computations. The shape of the initial solid seed in this section is diamond as shown in Figure 5.27, except in Section 5.4.4 where a round shape is implemented to study the effect of the initial condition. The diagonal length for the initial diamond seed is $1.89 \times 10^{-7}m$, and the radius $r_d$ for the initial round seed is $9.45 \times 10^{-8}m$, which is half of the diagonal length of the diamond shape. The initial seed will be placed in the center of the domain to initiate growth of the dendrite.

We display the dendrite structure contours in Figure 5.28 for five different degrees of anisotropy: $\epsilon' = 0.01, 0.02, 0.03, 0.04$ and $0.05$. We start from the initial time $t = 0$, and finish at $t = 3.4 \times 10^{-7}s$. The time difference between each contour is $\delta t = 2 \times 10^{-8}s$. We
observe in Figure 5.28 that the main branches of dendrite grow faster with the increase of $\epsilon'$, and the side branches develop better with the decrease of $\epsilon'$. Moreover, we do not observe any side branches if $\epsilon' = 0.05$. The branches in the direction of angle $0$, $\pi/2$, $\pi$ and $3\pi/2$ have more chances to grow than in other directions with the introduction of anisotropy. Therefore, we should observe four main branches in those four directions with side branches developing perpendicular to the main branches. However, the side branches are not perpendicular to the main branches because of the existence of the grid anisotropy as shown in Figure 5.28. Moreover, the larger the degree of anisotropy, the higher the growth rate of the branches in the four directions. Hence, propagation speed of the main branches is increasing from (a) to (e) in Figure 5.28.

We also compare the interfacial speed $V_n$ versus time for the five different $\epsilon'$ to demonstrate that different degrees of anisotropy only produce different dendrite structures; however, they do not change the interfacial speed. The results in Figure 5.29 verify that the interfacial speed is not affected by the degree of anisotropy. Indeed $\epsilon'$ behaves as a physical perturbation parameter as shown in Figure 5.28. Different degrees of anisotropy introduce different amount of perturbation to the system, hence, produce different dendrite structures. We emphasize here that the interfacial speed $V_n$ is the averaged ones as in Section 5.3.2, i.e., $V_n = d\sqrt{A}/dt$, where $A$ is the solid-phase area. Therefore, the averaged growth rates for different $\epsilon'$ are essentially the same as shown in Figure 5.29, although the growth rate of the main branches increase with increasing $\epsilon'$ as in Figure 5.28.
Figure 5.28  Dendrite structure contours for different degrees of anisotropy with grid sizes $\Delta x = \Delta y = 3.15 \times 10^{-8} m$ and $\Delta t = 10^{-12} s$ for Stefan number $St = 0.5$: (a) $\epsilon' = 0.01$; (b) $\epsilon' = 0.02$; (c) $\epsilon' = 0.03$; (d) $\epsilon' = 0.04$; (e) $\epsilon' = 0.05$. 
5.4.3 Effects of Interfacial Thickness $\delta$

We have discussed the influence of nondimensional interfacial thickness parameter $\epsilon$ on the interface in Section 5.3.3. We will study the influence of the interfacial thickness $\delta$ on the dendrite structure in the case of anisotropic interfacial energy in this subsection. We conduct the simulation for $\epsilon' = 0.03$ and $\delta = 1.26 \times 10^{-8}$, and compare the dendrite structure in Figure 5.30 with that in Figure 5.28(c). As in Section 5.4.2, The time difference between each contour is $\delta t = 2 \times 10^{-8}s$, and the final time is $t = 3.4 \times 10^{-7}s$. We observe more side branches on the main branches and the secondary side branches on the side branches. We can draw the same conclusion as in Section 5.3.3 that

We compare the phase field for two different $\delta$: $1.05 \times 10^{-8}$ and $1.26 \times 10^{-8}$ at time $t = 3.4 \times 10^{-7}s$ in Figure 5.31. We note that the structure with the large interfacial thickness is more complicated than that with the small one. The larger interfacial thickness produces not only complex dendrite structures such as side branches and secondary side branches, but also multiple interface in the solid region (cavities and gaps in the solid region in Figure 5.31(b)). Furthermore, we find that the interface region becomes mushy if $\delta \geq 1.26 \times 10^{-8}m$ (not shown).
Figure 5.30  Dendrite structure contours with grid sizes $\Delta x = \Delta y = 3.15 \times 10^{-8} m$ and $\Delta t = 10^{-12}s$ for Stefan number $St = 0.5$, degree of anisotropy $\epsilon' = 0.03$ and interfacial thickness $\delta = 1.26 \times 10^{-8}$.

Figure 5.31  Comparison of dendrite structures for different interfacial thickness with grid sizes $\Delta x = \Delta y = 3.15 \times 10^{-8} m$ and $\Delta t = 10^{-12}s$ for Stefan number $St = 0.5$ and $\epsilon' = 0.03$: (a) $\delta = 1.05 \times 10^{-8}$; (b) $\delta = 1.26 \times 10^{-8}$. 
5.4.4 Effects of Initial Shape

We have compared the dendritic structures between two different initial shapes, square and round, in Section 5.3.6. We will further study the effect of initial shape on dendrite structure and the interface speed in this section. We conduct the simulation for $\epsilon' = 0.03$ and $\delta = 1.26 \times 10^{-8}$, and compare the dendrite contours of initial round shape in Figure 5.32 with those of initial diamond shape in Figure 5.28(c). We observe more side branches with the initial diamond shape than the round one. Thus we make the same conclusion as in Section 5.3.6 that the different dendrite structures are produced by the different local curvature of the initial shape. Moreover, the initial diamond shape introduces more perturbation to the dendrite than the round one, thus we observe more side branches. The comparison of the interface speed in Figure 5.33 reinforces the conclusion in Section 5.4.2 that the perturbation parameter, initial shape, does not affect the interface speed.

![Figure 5.32](image)

**Figure 5.32** Dendrite structure contours for approximately round initial seed with grid sizes $\Delta x = \Delta y = 3.15 \times 10^{-8} m$ and $\Delta t = 10^{-12} s$ for Stefan number $St = 0.5$, $\epsilon' = 0.03$ and $\delta = 1.05 \times 10^{-8}$. 
Figure 5.33 Interface propagation speed for different initial shape: diamond and round.
Chapter 6
SIMULATION OF 2-D DENDRITE GROWTH IN LID-DRIVEN-CAVITY FLOW

In this chapter we will solve a freezing problem in a lid-driven-cavity flow. We implement a multiscale method to solve this problem using Algorithm 4 of Chapter 4 following procedures described below. We first introduce lid-driven-cavity flow on the large scale until stationarity is achieved, and then initiate freezing in the flow field. We begin with a detailed analysis of lid-driven-cavity flow on the large scale, providing streamlines, phase portraits and power spectra corresponding to the time series. We then compare dendrite growth in the flow field under the assumption that $u$, $v$ and $p$ are independent of time on small scales and without this assumption. After that, we present the small-scale streamlines of the flow field, the dendrite contours, and the large-scale streamlines at different times. Finally, we will discuss the parallel performance achieved when the code was executed on the HP SuperDome using the MPI paradigm.

6.1 Lid-Driven-Cavity Flow

Water, instead of nickel, is used for computations of this chapter to avoid introduction of radiation in the energy equation. The physical and thermodynamical parameters of water were listed in Table 5.1. Lid-driven-cavity flow is employed on the large scale for the reason that it has served over and over again as a model problem for testing and evaluating numerical techniques in CFD codes. The large-scale domain is the square $\Omega = [0, 0.1m] \times [0, 0.1m]$; we discretize the momentum equation on $\Omega$ using backward Euler in time and centered differencing in space. Details of numerical methods and algorithms employed to solve the momentum equations were provided in Section 3.4.1. We will present the computed results at $Re = 100$, 400 and 5000, and compare them with those provided by Ghia et al. [113] in the following subsections. In addition, we will implement a second-order and a fourth-order Shuman filter to treat aliasing, and compare the results obtained with the two Shuman filters using different filter parameters.
6.1.1 Comparisons between Computed Results and Published DNS Results at Low $Re$

We first perform computations at low Reynolds number, $Re = 100$ and 400, on grids of 200 × 200 cells, and make comparisons with the results provided by Ghia et al. [113]. The corresponding $u$-velocities of the lid movement are $U = 10^{-3} m/s$ and $U = 4 \times 10^{-3} m/s$ for $Re = 100$ and 400, respectively. The grid sizes are $\Delta x = \Delta y = 5 \times 10^{-4} m$, and the time step is $\Delta t = 5 \times 10^{-4} s$. The $u$-velocity component along the vertical centerline ($x = 0.05 m$) and $v$-velocity component along the horizontal centerline ($y = 0.05 m$) with $Re = 100$ and 400 are provided in Figures 6.1 and 6.2 compared with results in [113]. Velocity profiles shown in Figures 6.1 and 6.2 are steady-state solutions, and they display excellent agreement with the published DNS results.

The streamlines at $Re = 100$ and 400 are displayed in Figure 6.3 with the same set of seeds plotted via Tecplot [114] to make equivalent comparison. We observe that the primary vortex center moves toward the domain center, and the secondary vortices at the right-bottom and left-bottom corners become larger as the Reynolds number increases. The phenomena are consistent with those in [113] and earlier publications referenced therein.

We conduct grid function convergence tests at $Re = 100$ and 400 on grid sizes $\Delta x = \Delta y = 5 \times 10^{-4} m$ and $\Delta t = 5 \times 10^{-4} s$, $\Delta x = \Delta y = 10^{-3} m$ and $\Delta t = 10^{-3} s$, and $\Delta x = \Delta y = 10^{-4} m$.
Figure 6.2  Computed velocity profiles for $Re = 400$ compared with results of Ghia et al.: (a) $u$-velocity component at $x = 0.05m$; (b) $v$-velocity component at $y = 0.05m$.

Figure 6.3  Streamlines of lid-driven-cavity flow at steady state with grid sizes $\Delta x = \Delta y = 5 \times 10^{-4}m$ and $\Delta t = 5 \times 10^{-4}s$. Left: $Re = 100$; Right: $Re = 400$. 
2 \times 10^{-3} m \text{ and } \Delta t = 2 \times 10^{-3} s. \text{ The degree of accuracy we obtained is essentially first order instead of second order that we might expect. We conclude that the first-order convergence rate is caused by two singular points at the top corners. The divergence-free condition is no longer satisfied at these points, and the magnitudes of pressure are high there. Moreover, those points where velocities switch signs at the neighboring points, i.e., velocities change from positive to negative or vice versa, are not appropriate for the computation of order of accuracy. Hence, the convergence rate, even computed via the } L^2 \text{-norm, will be decreased.}

6.1.2 Implementation of Shuman Filters

We will demonstrate how we implement Shuman filters and the strategy to obtain a higher-order one in this subsection. It is necessary to implement a filter, e.g., the Shuman filter as in the form of Eq. (3.42b), for application to solutions of the N.-S. equations at high Reynolds number to treat aliasing. Moreover, choice of filter parameter is important since this parameter is inversely proportional to the filter width. The smaller the filter width, the more the solution is passed; i.e., more wavenumbers in a Fourier representation have been retained in the solution. In other words, for small filter parameter the high-wavenumber content of the Fourier representation is almost completely removed—exactly what is needed to treat aliasing, and as the filter parameter increases fewer Fourier coefficients are removed by the filter.

We recall here that the Shuman filter introduced in Eq. (3.42b) of Section 3.4.1 is mainly of second-order accuracy, as can be demonstrated via a simple truncation error analysis which we now carry out. Here we analyze the } u \text{-velocity component as an example. For simplicity, we assume } \beta = \beta_u = \beta_v \text{ and } h = h_x = h_y. \text{ We first expand grid-function values } u_{i+1,j}, u_{i-1,j}, u_{i,j+1} \text{ and } u_{i,j-1} \text{ in Taylor series:}

\begin{align*}
    u_{i+1,j} &= u_{i,j} + h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2} \frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{h^3}{6} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{24} \frac{\partial^4 u_{i,j}}{\partial x^4} + \cdots, \\
    u_{i-1,j} &= u_{i,j} - h \frac{\partial u_{i,j}}{\partial x} + \frac{h^2}{2} \frac{\partial^2 u_{i,j}}{\partial x^2} - \frac{h^3}{6} \frac{\partial^3 u_{i,j}}{\partial x^3} + \frac{h^4}{24} \frac{\partial^4 u_{i,j}}{\partial x^4} + \cdots, \\
    u_{i,j+1} &= u_{i,j} + h \frac{\partial u_{i,j}}{\partial y} + \frac{h^2}{2} \frac{\partial^2 u_{i,j}}{\partial y^2} + \frac{h^3}{6} \frac{\partial^3 u_{i,j}}{\partial y^3} + \frac{h^4}{24} \frac{\partial^4 u_{i,j}}{\partial y^4} + \cdots, \\
    u_{i,j-1} &= u_{i,j} - h \frac{\partial u_{i,j}}{\partial y} + \frac{h^2}{2} \frac{\partial^2 u_{i,j}}{\partial y^2} - \frac{h^3}{6} \frac{\partial^3 u_{i,j}}{\partial y^3} + \frac{h^4}{24} \frac{\partial^4 u_{i,j}}{\partial y^4} + \cdots. 
\end{align*}
The substitution of these into Eq. (3.42b) yields:

\[ \hat{u}_{i,j} = u_{i,j} + \frac{h^2}{4 + \beta} \left( \frac{\partial^2 u_{i,j}}{\partial x^2} + \frac{\partial^2 u_{i,j}}{\partial y^2} \right) + \frac{h^4}{12(4 + \beta)} \left( \frac{\partial^4 u_{i,j}}{\partial x^4} + \frac{\partial^4 u_{i,j}}{\partial y^4} \right) + O(h^6). \]  

(6.2)

Here we suppress one “\(^\wedge\)” of \( u \) for convenience. It is clear from Eq. (6.2) that \( \hat{u}_{i,j} \to u_{i,j} \) as \( \beta \to \infty \). We also note that the dominant truncation error in Eq. (6.2) is diffusive and second-order accurate in space, and the \( O(h^4) \) term is anti-diffusive. In particular, the added diffusion is one of the main requirements for successful treatment of aliasing. At the same time, the actual amount of added diffusion is controllable through the filter parameter \( \beta \). However, truncation error of the filter will accumulate and can lead to inaccurate results in time integrations.

We apply this second-order Shuman filter to mollify computed solutions. It turns out that solutions obtained with this filter might not satisfy desired accuracy requirement if the grid is coarse, and therefore, too much aliasing is introduced in the basic calculation. In this case, the filter with large filter parameter is not able to remove enough aliasing, while the filter with small filter parameter will eliminate some important features of the solution, leading to inaccurate results. We hence build a higher-order accurate Shuman-like filter in which a nine-point stencil shown in Figure 6.4 is employed to avoid introduction of more than three grid points in any one direction. As observed by Shapiro [115], symmetry of the Shuman filter formula with respect to the discrete point at which it is applied leads to zero phase error in the filtered result—a desirable property. Therefore, one of the rules in building a higher-order filter should be to maintain symmetry of the formula; another rule is to guarantee that the \( O(h^4) \) term represents dominant truncation error and is diffusive.

As a result, we obtain

\[ \hat{u}_{i,j} = u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} + \beta u_{i,j} - \frac{1}{2 + \beta} \left( u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1} \right). \]  

(6.3)

Taylor expansion of the above filter leads to

\[ \hat{u}_{i,j} = u_{i,j} - \frac{h^4}{2(2 + \beta)} \frac{\partial^4 u_{i,j}}{\partial x^2 \partial y^2} + \cdots. \]  

(6.4)

Hence, the dominant truncation error of the filter in the form of Eq. (6.3) is diffusive and fourth-order accurate in space; also the formula (6.3) is symmetric.
6.1.3 Solutions at High \( \text{Re} \)

In this section we display solutions at \( \text{Re} = 5000 \) using different filters to exhibit behavior of lid-driven-cavity flow and to make comparisons between the higher-order and low-order filters. All of the computations are conducted on grids of \( 200 \times 200 \) cells. The corresponding grid sizes are \( \Delta x = \Delta y = 5 \times 10^{-4} \text{m} \) and \( \Delta t = 5 \times 10^{-4} \text{s} \); the lid-movement velocity is \( U = 0.05 \text{m/s} \). The higher-order Shuman-like filter in Eq. (6.3) and the low-order filter in Eq. (3.42b) are utilized to treat aliasing introduced by the coarse grid. In addition, the filter parameter \( \beta \) is chosen to be 2000 for both filters. By doing so, we compare solutions for both filters under the condition that all physical and numerical parameters are the same.

We first compare in Figure 6.5 the computed steady-state \( u \)-velocity component along the vertical centerline and \( v \)-velocity component along the horizontal centerline at \( \text{Re} = 5000 \) with those given in [113] computed on a \( 257 \times 257 \) grid. We point out that the small difference between our computation and that by Ghia et al. is mainly caused by the coarser grid we used. As we refine the grid, computed results will be more accurate. We note that there is no apparent difference between the results from the fourth-order Shuman-like filter and the second-order Shuman filter in Figure 6.5. This is caused by the high-value of filter parameter \( \beta \) we employed. Recall that in Sec. 6.1.2, we mentioned that more original Fourier coefficients are retained if \( \beta \) increases. Therefore, at high \( \beta \) results are less dependent on the filter than at low \( \beta \). Moreover, leading truncation errors from both the higher-order filter and the
low-order one are very small if \( \beta \) is large. We thus do not observe a significant difference between computed results from the two filters in Figure 6.5. However, we emphasize that as we zoom Figure 6.5 to some extent (not shown), the higher-order filter solutions are closer to the results in [113] than are the low-order filter ones.

The streamlines at \( Re = 5000 \) are displayed in Figure 6.6 with the same set of seeds as those at \( Re = 100 \) and 400. By comparing Figures 6.3 and 6.6, we note that again the largest vortex center moves towards the domain center. In addition, the vortices at the right-bottom and left-bottom corners become larger as the Reynolds number increases. Moreover, a vortex at the left-upper corner and a tertiary vortex at the right-bottom corner appear when \( Re = 5000 \). We remark that these observations are consistent with the published results. The phase portrait and power spectrum to identify the fundamental frequency are presented in Figure 6.7. These figures indicate a steady-state flow at \( Re = 5000 \). According to the results by Peng et al. [116], the first Hopf bifurcation (to periodicity and thus time dependence) of a 2-D square lid-driven-cavity flow occurs around \( Re = 7600 \). Therefore, we expect a steady-state flow at \( Re = 5000 \).
Figure 6.6  Streamlines of lid-driven-cavity flow at steady state with grid sizes $\Delta x = \Delta y = 5 \times 10^{-4} m$ and $\Delta t = 5 \times 10^{-4} s$ at $Re = 5000$.

Figure 6.7  Lid-driven-cavity flow at $Re = 5000$ with $\Delta x = \Delta y = 5 \times 10^{-4} m$ and $\Delta t = 5 \times 10^{-3} s$. (a): Phase portrait of $u$ versus $v$; (b): Power spectral density of $u$ velocity.
6.1.4 Effects of Filter Parameter $\beta$

We have analyzed the dominant error of the Shuman filter (3.42b), and extended the Shuman filter to a fourth-order form (6.4) in Section 6.1.2. In this section, we will compare solutions obtained by applying the two filters using different filter parameter values $\beta$. As we mentioned in Section 6.1.3, the solution obtained from the fourth-order filter is more accurate than that from the second-order filter if $\beta = 2000$. This conclusion is again valid if $\beta = 200$ (as one might expect), as shown in Figure 6.8, although the difference is still very small. We note that results for $\beta = 200$ are more accurate than those for $\beta = 2000$ away from wall boundaries. Recall that for large filter parameter less high-wavenumber content of the Fourier representation is removed. In other words, it is possible that aliasing remains in the solution if the filter parameter is not small enough. Therefore, the difference of solutions between the two filter parameters is mainly caused by the aliasing still present when $\beta = 2000$. In addition, we found that solutions from the second-order filter with $\beta = 200$ are less accurate than those from the fourth-order filter near the boundaries. We realize that high resolution is required in the boundary layer to obtain accurate solutions since velocities vary rapidly in this region. Moreover, the higher-order filter provides a more accurate result than does the low-order filter if the necessary high resolution (to avoid aliasing) cannot be satisfied, as in the current case.

6.2 Dendrite Growth in Lid-Driven-Cavity Flow

In this section, we first determine the values for some physical parameters used in the computations, e.g., the degree of anisotropy, the kinetic coefficient and the dynamic viscosity. Validation of the assumption that velocity and pressure are independent of time during one large-scale time step is also conducted via comparison of the $L^2$-norm between the results with and without the assumption. We also investigate behavior of the lid-driven-cavity flow during freezing for two different cases: freezing from two discrete points and from three successive points near the bottom of the cavity. In addition, the small-scale dendrite structures are displayed at different locations of the large-scale lid-driven-cavity flow.
Figure 6.8  Computed velocity profiles at $Re = 5000$ for different filters (fourth-order filter with $\beta = 2000$, fourth-order and second-order filters with $\beta = 200$) compared with data by Ghia et al.: (a) $u$-velocity component at $x = 0.05m$; (b) $v$-velocity component at $y = 0.05m$.

6.2.1 Choice of Physical Parameters

The square small-scale domain $\Omega_s \equiv [0, L_s] \times [0, L_s] = [0, 4 \times 10^{-5}m] \times [0, 4 \times 10^{-5}m]$ is chosen to conduct the simulation of ice crystal growth at the bottom of the cavity. Six-fold anisotropy of the interface is introduced, and the degree of anisotropy $\epsilon' = 0.03$ is assumed.

The kinetic coefficient $\mu_k$ is $3.08m/s \cdot K$ and of $\mathcal{O}(1)$ for pure metal, nickel. Furthermore, since the kinetic coefficient of water should be at least one order smaller than that of pure metal, as was explained in Section 2.3 and Section 5.2.1, $\mu_k = 0.05m/s \cdot K$ for water is chosen (based on several numerical tests) to produce a thin interface as occurs physically.

Since in the phase-field model both bulk phases are determined implicitly via values of the phase field, neither the interface position between each phase nor its shape is explicitly known. Therefore, it is difficult to apply the no-slip condition on the interface. Furthermore, since the momentum equation contains a forcing term which is equivalent to the forcing term in the immersed boundary method as was described in Section 3.1, it is reasonable to solve the momentum equation on the entire domain using different values for the dynamic viscosity.

By doing so, we obtain solutions in the liquid and the mushy zone, as well as that in the solid.

We will discuss here how we treat the dynamic viscosity $\mu_S$ in the solid. Recall that in
Section 3.1 we expressed the dynamic viscosity in Eq. (3.15) as

\[ \mu(\phi) = [1 - w(\phi)] \mu_S + w(\phi)\mu_L. \] (6.5)

We note that the dynamic viscosity \( \mu_S \) in a solid should be infinity, which can be replaced by a large number, e.g., \( 10^{12} \), during computation. However, \( \mu_S \to \infty \) in the above equation does not provide a reasonable value for \( \mu \) in the mushy zone. We therefore estimate the dynamic viscosity in the mushy zone via a method provided by Kafanov [117]. Hence, the dynamic viscosity will be represented by:

\[ \mu(\phi) = \begin{cases} 
\mu_S, & \text{if } \phi < \phi_c \quad \text{(solid),} \\
3.5 [1 - w(\phi)] \mu_L + w(\phi)\mu_L, & \text{if } \phi \geq \phi_c \quad \text{(mushy and liquid).}
\end{cases} \] (6.6)

Here, \( \phi_c \) is a value to denote solid during computation, i.e., \( \phi < \phi_c \) is solid. We choose \( \phi_c = 0.1 \) in the current computations, and expect that magnitude of the computed velocity in the solid is negligible since the dynamic viscosity in the solid is very large.

### 6.2.2 Validation of Time-Independent Velocity and Pressure Assumption

The most important work before simulating freezing in lid-driven-cavity flow is to verify the reasonability of the assumption proposed in Section 4.2 that small-scale velocity and pressure are independent of time within one time step of the large-scale calculation. To accomplish this, we compare the simulations on small scales using Eqs. (4.2) and (4.4), i.e., with and without this assumption. Spatial step sizes are \( \Delta x_s = \Delta y_s = 10^{-7}m \), and time step size is \( \Delta t_s = 2 \times 10^{-14}s \) to maintain stability while using the governing equations (4.2). The interfacial thickness parameter is chosen to be \( \delta = 2 \times 10^{-8}m \) for the requirements of maintaining stability and producing a thin but smooth interface. The relation between the interfacial thickness parameter \( \delta \) and the spatial step size \( \Delta x (\Delta y) \) in this case should also follow the rule discussed in Section 5.2.1.

Comparisons of the velocity vector field and streamlines for the two cases are presented in Figure 6.9. We observe a complicated flow over the solid body with moving boundary in Figure 6.9(a) as predicted in Section 3.1. In particular, the immersed boundary method introduces additional normal and shear forces on the moving boundary. The normal force pushes the flow near the interface outward, while the shear force makes the flow parallel to
the interface. Moreover, magnitude of the velocity near the interface is higher than elsewhere for the reason that the forcing term (3.27a) in the momentum equation (3.26) is dominant.

We observe from the velocity vectors and streamlines in Figure 6.9(a) that the flow toward the vertical walls of the solid body does not behave like a flow near a solid wall; instead it goes through the solid body. The reason is that we employ a “pseudo” solid; i.e., the N.-S. equation is solved in this area using a large dynamic viscosity, rather than applying the no-slip condition on its boundary, as indicated earlier. Therefore, the corresponding velocity in the solid is nonzero, but very small, and streamlines would exist in the solid. However,
we do not observe any streamlines in the solid in the figures since we assign zero to the velocity in solid after computation. The velocity vector field and streamlines of Figure 6.9(b) correspond to a steady-state flow. Although details of the flows in Figures 6.9(a) and (b) are quite different, we found that the $L^2$-norm of phase field and temperature values for both cases are the same. This provides a mechanism for transferring information between large and small scales, and at the same time indicates that it is reasonable to assume that the large-scale flow is steady during small-scale computation.

6.2.3 Lid-Driven-Cavity Flow with Freezing from Two Discrete Points Near Bottom

We conduct simulations of freezing from the bottom of steady-state lid-driven-cavity flow in the current and the following sections to make a comparison between freezing from different positions near the bottom. The large-scale solutions have been presented in Section 6.1.3, and implementation of the multiscale method follows Algorithm 4 in Section 4.3. The large-scale spatial step sizes $\Delta x$ and $\Delta y$ are the same as those in Section 6.1.3, while the time step size is reduced to $\Delta t = 4 \times 10^{-5}s$ such that it is the same as the small-scale time interval. Degree of supercooling is $\Delta T = 40K$, which corresponds to $St = 0.5$.

We first consider the case that crystal growth initiates from two discrete points at the bottom center of the large-scale domain as in Figure 6.10. We specify two small-scale domains $\Omega_s = [0, 4 \times 10^{-5}m] \times [0, 4 \times 10^{-5}m]$ with the same size in the grid cells (100, 2) and (102, 2) of the large-scale domain, and the initial seeds are squares with side length $1.2 \times 10^{-6}m$. The grid spacings for small-scale computation are $\Delta x_s = \Delta y_s = 10^{-7}m$ and $\Delta t_s = 2 \times 10^{-9}s$. The time interval used to conduct the small-scale computation is $t_{ns}^s \in [0, \Delta t] = [0, 4 \times 10^{-5}s]$, where $n_s$ is numbers of the small-scale time step. Velocities in the corresponding large-scale grid cells are $u_{99,2} = -1.85 \times 10^{-2}m/s$, $u_{100,2} = -1.88 \times 10^{-2}m/s$, $u_{101,2} = -1.91 \times 10^{-2}m/s$, $u_{102,2} = -1.93 \times 10^{-2}m/s$, and $v_{100,2} = 4.49 \times 10^{-5}m/s$, $v_{101,2} = 3.90 \times 10^{-5}m/s$ and $v_{102,2} = 3.26 \times 10^{-5}m/s$. These velocities are the results of a steady-state lid-driven-cavity flow at $Re = 5000$ as were given in Section 6.1.3.

Velocity vector fields of small-scale computations at $t = 4 \times 10^{-5}s$ and $t = 8 \times 10^{-5}s$ are displayed in Figures 6.11 and 6.12, and dendrite contours are labeled with purple lines.
We note that since $u$, $v$, and $p$ are assumed to be independent of time on such a small time interval, velocity fields remain steady during one time interval, i.e., one large-scale time step. Therefore, the main purpose of Figures 6.11 and 6.12 is to demonstrate influence of the flow field on dendrite structure. We hardly observe any difference between dendrite structures in grid cells (100, 2) and (102, 2) from Figures 6.11 and 6.12. Since the difference of velocity magnitudes between these two points is very small, their effects on dendrite growth are very similar. We therefore compare dendrite shapes at these locations with flow, and dendrite structure without flow, in Figures 6.13(a) at $t = 4 \times 10^{-5} s$ and 6.13(b) at $t = 8 \times 10^{-5} s$ for further study.

By comparing Figures 6.11 and 6.13(a), we find that the difference between dendrite structures in cells (100, 2) and (102, 2) is unobservable. However, a difference is observed at a longer time $t = 8 \times 10^{-5} s$ as shown in Figures 6.12 and 6.13(b). (We will compare dendrite structures at other positions in the lid-driven-cavity flow to further demonstrate the effect of the flow field on dendrite structure in Section 6.2.5.) Moreover, we found that the dendrite structures without flow are essentially symmetric in both $x$ and $y$ directions, while introduction of the flow field breaks this symmetry: the two main branches in the $x$-direction grow at different rates, causing unequal length of the two main branches; sizes of the two side branches in the $y$-direction are also no longer the same. Furthermore, competition between two side branches makes the short branch grow slower than the long branch. In addition, some details of the side branches are lost with introduction of the flow field as shown in Figures 6.12 and 6.13(b). We note that the flow is mainly in the negative $x$-direction with very
Figure 6.11  Velocity vector field and dendrite shape of small-scale computation in: (a) cell $(100,2)$ and (b) cell $(102,2)$ as shown in Figure 6.10 on large-scale domain at $t = 4 \times 10^{-5}\, s$.

Figure 6.12  Velocity vector field and dendrite shape of small-scale computation in: (a) cell $(100,2)$ and (b) cell $(102,2)$ as shown in Figure 6.10 on large-scale domain at $t = 8 \times 10^{-5}\, s$. 
small positive $v$-velocity component (of $\mathcal{O}(10^{-5})$), and the dendrite grows faster downstream than upstream. Recall that dendrite growth rate is directly related to the temperature gradient: the higher the temperature gradient, the faster the dendrite grows. In the present case, the dendrite leftmost end has the highest temperature gradient (not shown); therefore, it grows faster than do other branches. Although dendrites display different structures with and without flow, their solid phase areas are the same, as shown in Figure 6.14; i.e., the mean growth rates for the three cases shown are approximately equal.

Streamlines of the lid-driven-cavity flow at two different times are presented in Figure 6.15, and they are obtained from the same seeds as those plotted in Figure 6.6. We do not observe any difference by introducing the freezing at the bottom center via comparison between Figures 6.15 and 6.6. We note that area of the solid phase at $t = 8 \times 10^{-5}$ s is $1.7 \times 10^{-10} m^2$, which is only $0.017\%$ of the area of one large-scale grid cell (and hence, far smaller than truncation error in the large-scale computation). Furthermore, we recall that the phase-field and temperature values on the large scale are obtained from the $L^2$-norm of the phase-field and temperature values on the small scale. Therefore, the phase-field and temperature values on the large scale at $t = 8 \times 10^{-5}$ s are too close to those in the liquid to produce changes on the large-scale flow field.
Figure 6.14  Comparison of solid phase area for three cases: dendrites grow in grid cells (100, 2) and (102, 2) of large scale, and dendrite grows without flow at $t = 8 \times 10^{-5}$ s.

Figure 6.15  Streamlines of lid-driven-cavity flow with freezing from two discrete points on bottom center on small scale at: Left $t = 4 \times 10^{-5}$ s, Right $t = 8 \times 10^{-5}$ s.
To further study the effect of freezing on the lid-driven-cavity flow, we consider the case that \( \phi = 0 \) and \( T = T_m = 273K \) in grid cells (100, 2) and (102, 2) on the large-scale domain, which is equivalent to the case that dendrite area is the same as the area of one large-scale grid cell. To accomplish this, we first estimate the time when \( \phi = 0 \) and \( T = T_m = 273K \) on the large scale from a solid-phase area vs. time relationship. Recall from Section 5.3.2 that computed dendrite growth rate obeys \( V_n \sim t^{-3/2} \) for \( St = 0.5 \). Hence, computed area of the solid phase \( A \) should follow \( A \sim t^{3/2} \). We therefore compare the computed solid area in the large-scale grid cell (100, 2) with a function close to the relation above to obtain the best fit. To better describe variation of the solid area versus time, we use a dimensionless function \( 86.5t^{1.34} + 9.0 \times 10^{-4} \) to fit the dimensionless computed solid area as shown in Figure 6.16. Here the constant \( 9.0 \times 10^{-4} \) is the dimensionless area of the initial seed; the length and time scales to nondimensionalize the area and time are \( L_s \) and \( L_s^2/\alpha \), respectively, where \( \alpha = k/\rho_L c_p L \) is the thermal diffusivity. Therefore, we estimate from the relation above that when \( t = 0.05s \), the approximate solid area becomes \( 10^{-6}m^2 \), which is exactly the same as the area of one large-scale grid cell. In other words, the phase field and temperature on the large scale are 0 and 273K, respectively, if \( t = 0.05s \).

From the estimation above, we can draw the conclusion that it takes approximately 1250 large-scale time intervals of the small-scale computations to make a significant change to the

![Figure 6.16](image_url)
large-scale computation, and this requires an extremely long run time: $\sim 1250$ days. Hence, it is not realistic for current computations using the available computational hardware. We note that the incredibly long run time is mainly caused by the number of time steps, $2.5 \times 10^7$, required for the small-scale computation at $t = 0.05s$, considering the fact that the small-scale time step is only two nano-seconds. Therefore, a better multiscale method is required to be implemented to current computation such that the information obtained from small-scale computations is more efficient.

Despite the fact that the small-scale computation is extremely time consuming, it is feasible to estimate the phase-field and temperature values on the large scale from the relation above to investigate the effect of freezing on the lid-driven-cavity flow. We therefore assign $\phi_{100,2} = \phi_{102,2} = 0$ and $T_{100,2} = T_{102,2} = 273K$, and conduct the large-scale simulation on one large-scale time step. In Figure 6.17, comparison of the streamlines for such a flow and the flow without freezing shows the difference between these two flows. It should be pointed out that there exist only slight differences between the two flows. For example, sizes of the vortices at the left-upper corner are different: bottom of the vortex is near $y = 0.073m$ for the flow field with freezing, while it is near $y = 0.072m$ for the case without freezing; the streamlines near the bottom are also different (in particular, location of the dividing streamline between primary and secondary vortices), and a small peak of the streamlines can be found at the bottom near $x = 0.05m$; observably-different secondary and tertiary vortices at the right-bottom corner can be found. We will provide detailed comparisons via velocity vector fields in Figure 6.23 in Section 6.2.4.

6.2.4 Lid-Driven-Cavity Flow with Freezing of Three Successive Points Near Bottom

Computations have also been conducted from three successive points at the bottom center of the large-scale domain as displayed in Figure 6.18. Three small-scale domains are specified in grid cells $(100, 2)$, $(101, 2)$ and $(102, 2)$ of the large-scale domain, each with the same size as those in Section 6.2.3. Grid sizes, time interval and other parameters for small-scale computations are also the same as those given in Section 6.2.3.

We present velocity vector fields of the small-scale computation at $t = 4 \times 10^{-5}s$ and
Figure 6.17  Streamlines of lid-driven-cavity flow at $Re = 5000$: Left, with freezing from two discrete points at bottom center on small scale; Right, without freezing.

Figure 6.18  Position of freezing in large domain from three successive points near the bottom.

t = $8 \times 10^{-5}$s in Figures 6.19 and 6.20 with dendrite structures labeled by purple lines. Similar to the results in Section 6.2.3, we do not observe any apparent difference between dendrite structures in grid cells (100,2), (101,2) and (102,2) from the velocity vector fields in Figures 6.19 and 6.20. We emphasize again that no apparent difference of dendrite structures is caused by the small difference between velocities of these grid points as mentioned in Section 6.2.3; e.g., variations of $u$ and $v$ velocity components are $8 \times 10^{-4}m/s$ and $1.23 \times 10^{-5}m/s$, respectively.

From the analyses in Section 6.2.3, we learned that freezing for such a short time, $t =$
Figure 6.19  Velocity vector field and dendrite shape of small-scale computation in: (a) cell (100, 2), (b) cell (101, 2) and (c) cell (102, 2) as shown in Figure 6.18 on large-scale domain at $t = 4 \times 10^{-5} \text{s}$. 
Figure 6.20 Velocity vector field and dendrite shape of small-scale computation in: (a) cell (100, 2), (b) cell (101, 2) and (c) cell (102, 2) as shown in Figure 6.18 on large-scale domain at \( t = 8 \times 10^{-5} \) s.

\( 8 \times 10^{-5} \) s, does not have an obvious effect on the large-scale lid-driven-cavity flow (as should be expected based on simple scaling analyses—the motivation for a multiscale approach). As a consequence, the streamlines at \( t = 4 \times 10^{-5} \) s are essentially the same as those at \( t = 8 \times 10^{-5} \) s as shown in Figure 6.21. Indeed, such effects can be observed only if the dendrite growth time is around, or larger than, \( t = 0.05 \) s. Moreover, there is no observable difference between streamlines in Figure 6.21 and those in Figure 6.6 without freezing for the same reason explained in Section 6.2.3.

We apply the same procedure as that in Section 6.2.3 to investigate the effect of freezing
Figure 6.21  Streamlines of lid-driven-cavity flow with freezing from three successive points at bottom center on small scale at: Left $t = 4 \times 10^{-5}\text{s}$, Right $t = 8 \times 10^{-5}\text{s}$.

on the large-scale flow by assigning the phase-field and temperature values on the large scale; i.e., $\phi_{100,2} = \phi_{101,2} = \phi_{102,2} = 0$ and $T_{100,2} = T_{101,2} = T_{102,2} = T_m = 273K$. We note that there exist slight differences between streamlines in the two parts of Figure 6.22, and these differences are very similar to those in Figure 6.17 of Section 6.2.3. A small peak of the streamlines is observed near the bottom at $x = 0.05\text{m}$ in the case of flow with freezing, and structures of the secondary vortices at the right-bottom corner are different. In addition, sizes of the vortices at the left-upper corner are changed: bottom of the vortex is near the position $y = 0.073\text{m}$ with freezing, while the bottom is near $y = 0.072\text{m}$ for the case without freezing.

We present velocity vector fields in Figure 6.23 for three different cases: the flow with freezing from two discrete points as shown in Figure 6.10, the flow with freezing from three successive points displayed in Figure 6.18, and the flow without freezing, to demonstrate the different effects of freezing. In the case of flow without freezing, the flow near the bottom is almost parallel to the boundary with very small positive $v$-velocity. However, these vectors change directions if there exists freezing in the flow, as shown in Figure 6.23. The black dots in the left figure of Figure 6.23 are grid cells at which freezing is initiated (in the two-point case, dots 1 and 3; in the three-point case, dots 1, 2 and 3).
Figure 6.22 Streamlines of lid-driven-cavity flow at $Re = 5000$: Left, with freezing from three successive points at bottom center on small scale; Right, without freezing.

Figure 6.23 Comparison of velocity vector fields at $Re = 5000$ for: Left: freezing from two discrete points (blue vector), three successive points (red vector) around $t = 0.05s$; Right: without freezing.

Considering the fact that three successive points can be regarded as a small solid body because of no (computational) interface exists in between, and two discrete points are two small solid bodies, the flow over these points is actually flow over solid bodies with moving boundary. Therefore, the flow will firstly be pushed in the direction of dendrite growth at the
front of the solid body, and then will reattach at the rear part. We thus observe vectors near the bottom that are in the positive $y$-direction at $x = 0.0505m$ and $0.051m$, and turn to the negative $y$-direction at $x = 0.0485m$ and $0.049m$. Magnitude of the $v$-velocity component for freezing from two points is less than that from three points since the two points are discrete; hence, no freezing occurs at point 2. Therefore, the “push” and “reattach” forces are smaller than those in the three-point case, which makes magnitude of the $v$-velocity component smaller. It should be pointed out that magnitudes of the velocities at $y = 0.0005m$ for the cases with freezing near the bottom are apparently smaller than the case without freezing. Recall that the dynamic viscosity is determined by Eq. (6.6), and its value in the mushy zone is higher than that in the liquid. In addition, grid points at $y = 0.0005m$ near points 1, 2 and 3 are located in the mushy zone. Hence, velocities at these points are smaller than those computed by $\mu_L$ for the flow without freezing. Moreover, despite the fact that the freezing time is relatively short, $t = 0.05s$, the effects of freezing extend at least to the fourth row of grid cells as displayed in Figure 6.23.

### 6.2.5 Freezing at Different Positions of Lid-Driven-Cavity Flow

We conduct small-scale computations at different positions of the lid-driven-cavity flow to further investigate the effect of flow field on dendrite structure. Simulations are conducted in grid cells (191,161), (101,191), (31,31) and (161,31) on the large scale as displayed in Figure 6.24. Velocity components in these grid cells at $t = 4 \times 10^{-5}s$ are listed as follows:

$$
\begin{align*}
&u_{190,161} = 8.70 \times 10^{-4}m/s, \quad u_{191,161} = 5.86 \times 10^{-4}m/s, \\
v_{191,160} = -1.98 \times 10^{-2}m/s, \quad v_{191,161} = -1.95 \times 10^{-2}m/s, \quad \text{in grid cell (191,161)}; \\
u_{100,191} = 2.41 \times 10^{-2}m/s, \quad u_{101,191} = 2.42 \times 10^{-2}m/s, \\
v_{101,190} = 1.08 \times 10^{-3}m/s, \quad v_{101,191} = 1.03 \times 10^{-3}m/s, \quad \text{in grid cell (101,191)}; \\
u_{30,31} = -6.59 \times 10^{-3}m/s, \quad u_{31,31} = -7.37 \times 10^{-3}m/s, \\
v_{31,30} = 6.32 \times 10^{-3}m/s, \quad v_{31,31} = 7.10 \times 10^{-3}m/s, \quad \text{in grid cell (31,31)};
\end{align*}
$$

$$
\begin{align*}
u_{160,31} &= -1.24 \times 10^{-2}m/s, \quad u_{161,31} = -1.13 \times 10^{-2}m/s, \\
v_{161,30} &= -0.96 \times 10^{-2}m/s, \quad v_{161,31} = -1.07 \times 10^{-2}m/s, \quad \text{in grid cell (161,31)}.
\end{align*}
$$
Figure 6.24  Locations of grid cells (191,161), (101,191), (31,31) and (161,31) on large-scale domain.

We found that dendrite contours shown in Figure 6.25 display quite different structures from position to position. In these figures, the phenomenon that dendrite growth rate in the downstream direction is higher than that in the upstream is obvious. In addition, downstream side branches develop better than those in the upstream direction. Moreover, we observe development of secondary side branches in the downstream in Figure 6.25(a).

We further compare the solid phase areas at these locations with and without the flow to demonstrate the difference in mean interface propagation rate with introduction of the flow field. The comparison in Figure 6.26 indicates that dendrite growth rate with the flow field is faster than that without the flow. In particular, dendrite growth rates listed from the highest to the lowest are those in grid cells (101,191), (191,161), (161,31) and (31,31); different dendrite growth rates are caused by different magnitudes of the velocity. We note that magnitude of the velocity in the grid cell (101,191) is the highest, followed by that at (191,101), while the velocity magnitude at (31,31) is the lowest. Therefore, we conclude that the larger the velocity magnitude, the faster the dendrite grows in flow.

To further study the influence of the large-scale velocity magnitude on dendrite growth, we conduct simulations at points with different magnitudes of velocity, and compare the solid phase areas corresponding to different magnitudes of velocity in Figure 6.27. We found
Figure 6.25 Velocity vector field and dendrite shape of small-scale computation in grid cells: (a) (191,161), (b) (101,191), (c) (31,31) and (d) (161,31) on large-scale domain at $t = 4 \times 10^{-5}$s.

that the solid phase area increases monotonically with increase of velocity magnitude. We also note that the function $0.018|u|^{2.8} + 0.0428125$ can be employed to approximately fit the dimensionless computed solid-area curve as displayed in Figure 6.27, where the constant 0.0428125 is the dimensionless solid-phase area at zero velocity, i.e., solid area of dendrite without a flow. Here we use the length scale $L_s$ and the lid-movement velocity $U$ to nondimensionalize the solid area and velocity magnitude for the purpose of describing variation of the solid area with velocity magnitude of the flow field in a proper way. Although the increase of the solid area might not exactly obey such a power law, this empirical relation
Figure 6.26  Comparison of solid phase area for dendrite growth in grid cells (191,161), (101,191), (31,31) and (161,31) on large scale, and dendrite growth without flow at $t = 4 \times 10^{-5}$s. provides a basic idea of how and to what extent the solid-phase area changes with the velocity magnitude of the large-scale lid-driven-cavity flow.

Figure 6.27  Solid phase area of dendrite variation with velocity magnitude in large-scale lid-driven-cavity flow.
6.3 Parallel Performance

We discussed in Section 3.5 how we implement the parallel techniques, OpenMP and MPI, in our computations. We remark here that the main purpose to introduce parallel computing is to save wall-clock time, namely, obtain the results faster. Recall that simulations with the phase-field model are extremely time consuming, which makes the parallelization necessary. From the discussion in Section 3.5, we learned that the OpenMP algorithm is easily implemented but cannot satisfy high-memory requirement of the program, while the implementation of the MPI paradigm is more complicated. Comparison of the performance between OpenMP and MPI has been provided for the computations without implementation of the multiscale method by Xu et al. [81]. Therefore, we will analyze the performance of MPI algorithm of the computations using multiscale method in this section.

Parallelization of the numerical solution procedure is based on the MPI paradigm using the HP Fortran 90 HP-UX compiler and executed on the HP SuperDome at the University of Kentucky Computing Center. The maximum number of threads available on a single hypernode of the HP SuperDome is 64, and in the current study each processor is used to compute one part of the whole domain. The procedure to parallelize two-step Douglas & Gunn time splitting and line SOR with MPI is to compute different parts of the domain on different processors, which is equivalent to distributing the computations on a 2-D domain line by line to each processor, i.e., simply a crude form of domain decomposition. In particular, we divide the domain into $n$ equal pieces along the separate directions corresponding to each split step as was shown in Figure 3.5, where $n$ is the number of processors being used.

The shortcoming of this decomposition from the speedup point of view is that it increases communication time as number of processors is increased. We explain this via an example of data transfer between adjacent processors as was shown in Figure 3.7. For example, if the communications between adjacent processors are $2 \times 3 = 6$ units for four processors, then the communication for eight processors is 14 units in this case. As a result, the overall performance, as quantified by speedup will be sublinear. In parallel computing, speedup refers to how much a parallel algorithm is faster than a corresponding sequential algorithm. It is defined by a formula speedup $= t_1/t_n$, where $t_1$ is the execution time of the sequential algorithm, and $t_n$ is the execution time of the corresponding parallel algorithm with $n$
To study speed-up of parallelization, different numbers $n$ of processors ($n = 2, 4, 8, 16, 32$) are used to execute the algorithm. Figure 6.28 presents the speed-up factor versus the number of processors for the procedure being used here. It shows that, as the number of processors increases, the speed-up factor increases only sub-linearly, suggesting that the demonstrated speedups are not extremely good for $n \geq 32$. We compared the parallel performance of OpenMP and MPI in [81]. It shows that, as the number of processors increases, the speed-up factor also increases sub-linearly both OpenMP and MPI. Moreover, the speed-up performance of MPI is better than that of OpenMP.

Better performance of MPI could be achieved if further optimization is applied within the context of the current algorithm. Such optimization might include use of nonblocking communication, sending noncontiguous data using pack/unpack functions, decreasing unnecessary blocking, and optimizing the number of Douglas & Gunn split lines sent to each processor. The last of these can significantly alter the communication-time to compute-time trade off.
Chapter 7
SUMMARY AND CONCLUSIONS

Numerous efforts have contributed to the study of phase-change problems for over a century—both analytical and numerical. The Stefan problem is a mathematical model to determine the temperature distribution of solid and liquid phases and the position of the interface between them. Analytical approximations have been applied to Stefan problems for the purpose of obtaining explicit or approximate solutions to phase-change problems. However, these methods are not able to solve all problems because of their inherit restrictions: geometrical limitations, assumptions and simplifications that are difficult to relax, exclusion of accounts of complicated physical processes, etc. Therefore, numerical approximations have become important, especially with the dramatic improvement of computer hardware and numerical techniques. In this dissertation, we have emphasized computations of solidification using the phase-field model, studies of which were initiated in late 1970s and which are receiving increased recent attention. In the research of this dissertation, it has been used to simulate 2-D dendrite growth of pure nickel without a flow, and 2-D ice crystal growth in a high-Reynolds-number lid-driven-cavity flow. To accomplish the latter simulations, a heterogeneous multiscale method was implemented to solve equations of the phase-field model with convection. In this final chapter, we summarize the main portions of this work, draw conclusions based on the results that were obtained and offer suggestions for continued studies.

7.1 Summary
In Chapter 1, we first emphasized the importance of solidification processes in different areas. We then introduced the classical and modified Stefan problems, and their underlying assumptions, to solve the phase-change problems. Analytical solutions of some simple problems were also provided for both the classical and modified Stefan problems. Explicit solutions exist only for simple problems in 1D, as was mentioned in Section 1.2. Therefore, numerical methods, e.g., front-tracking methods, enthalpy methods and phase-field models, have been introduced to simulate more complicated, and thus realistic, problems. Among
these numerical approximations, we chose the phase-field model because it not only captures two important effects, surface tension and supercooling, but also enables explicitly labeling the solid and liquid phases and the position of the interface. We summarized the current state of the art of phase-field models in this chapter, i.e., what has been done in this area recently and the important problems that have not yet been solved. We also briefly outlined what we have done in the current research and how this will help further understanding and use of this approach.

It is important to understand the physics of solidification before one attempts to simulate solidification processes. Therefore, in Chapter 2, we provided analyses of the physics that the phase-field model must capture, such as conditions for onset of nucleation, dendrite growth rate, and an important physical parameter related to the growth rate, the kinetic coefficient $\mu_k$.

In Chapter 3, we provided a detailed derivation of a thermodynamically-consistent phase-field model with convection in the melt, and with anisotropic interfacial energy, based on the works of Anderson et al. [42]; such a phase-field model obeys the first and second laws of thermodynamics. This phase-field model contains the mass conservation equation (divergence-free condition), the momentum equations to solve for flow over the immersed moving boundary problem, the phase-field equation to label the bulk phases and the interface, and the energy equation to determine the temperature profile. Boundary and initial conditions to construct a well-posed problem have been presented for these governing equations.

In this chapter, we also described in detail the numerical methods and algorithms employed in the current study. We discretized the governing equations using a generalized trapezoidal formula in time (forward Euler, backward Euler or Crank–Nicolson) and centered differencing in space. We used a staggered grid for the purpose of maintaining stability in connection with pressure-velocity decoupling. Therefore, $u$ and $v$ velocity components are computed at the center of the vertical and horizontal grid-cell boundaries, respectively; pressure $p$, phase field $\phi$ and temperature $T$ are calculated at the grid-cell centers. In addition, we applied a $\delta$-form Douglas & Gunn time-splitting method to solve the governing equations, and a $\delta$-form quasilinearization, i.e., Fréchet–Taylor expansion, to linearize the nonlinear terms. A projection method was implemented to solve the complete system of
equations of motion, and the Shuman filter was employed to mollify the solutions as needed to eliminate aliasing caused by under resolution. Finally we described parallelization of the numerical solution procedure based on both the MPI paradigm and OpenMP, e.g., distribution of processors for the Douglas–Gunn time-splitting method and transferring data between processors.

A major achievement of this dissertation research is the application of a multiscale method to the phase-field model in the presence of a flow field. In order to obtain the details of dendrite structure as well as the flow field behavior during freezing, it is necessary to solve the phase-field model and the equations of motion on two different scales. To accomplish this, we simulate the phase-field model on a microscopic scale and the equations of motion on a macroscopic scale. Among those approaches applied to the multiple scale problems, we chose a heterogeneous multiscale method for the advantages presented in Chapter 4. In this context, the governing equations are decomposed into two parts on two different scales: the momentum equation to solve the flow over ice crystals on a macroscopic scale, and the phase-field model with the flow field on a microscopic scale. Boundary and initial conditions for the well-posed problems on large and small scales are also provided in this chapter. It is worth mentioning that we employ the vorticity transport equation and the divergence free condition to obtain an “artificial” boundary condition for velocity on the small scale such that the boundary condition will not affect the flow field around the dendrite. The detailed pseudo-language algorithm to implement the multiscale method was also presented at the end of this chapter.

Results of the current research consist of two parts: simulations of 2-D dendrite growth in Chapter 5 and those of 2-D ice crystal growth in a high-$Re$ lid-driven-cavity flow in Chapter 6. One of the purposes of the first part is the validation of the phase-field model and the study of dendrite growth under different conditions. We first conducted the numerical convergence tests via the choice of physical and numerical run conditions, grid function convergence tests, determination of quasilinear iteration convergence rates for both fully-coupled and sequential techniques, and comparison between forward and backward Euler schemes. We also proved that the energy is conserved for the adiabatic system.

In the remaining part of Chapter 5, we displayed the computed results of solidification
of pure metal, nickel, with isotropic surface energy. The dendrite structures and temperature profiles at different times are presented, followed by the dendrite structures at different degrees of supercooling (indicated by different Stefan numbers). We also provided the interfacial growth rate $V_n$ varying with time at different Stefan numbers, and the steady-state interfacial growth rate as a function of the Stefan number, and made qualitative comparisons with analytical approximations. In addition, we studied the effects of interfacial thickness parameter $\epsilon$ and kinetic coefficient $\mu_k$ on the dendrite structure and the interfacial growth rate. The influence of different shapes of the seed to initiate solidification was investigated. Validation of the phase-field model was finally accomplished by comparisons between the computed results, approximate analytical solutions and experimental data.

Furthermore, the computed results of solidification with anisotropic surface energy have been studied in a similar way as those used with the isotropic one. We investigated the effects of interfacial thickness $\delta$, initial shape and a new parameter introduced by the anisotropic surface energy, degree of anisotropy $\epsilon'$, on the dendrite structures and the interfacial growth rate. Existence of the critical radius (area) predicted by the phase-field model was verified in this section for both nickel and water for different cases: $\Delta c_p = 0$, $\Delta c_p$ a constant and $\Delta c_p$ a function of temperature. Moreover, the computational results were also compared with the analytical ones.

In Chapter 6, we conducted simulations of freezing from the bottom center in a lid-driven-cavity flow using a heterogeneous multiscale method, i.e., simulating ice crystal growth on the small scale to obtain the details of dendrite structure while solving the lid-driven-cavity flow with freezing on the large scale to study the flow behavior with the existence of phase change. Computations of two different cases are considered: freezing from two discrete points and from three successive points at the bottom center. We also compared dendrite growth at different positions in the large-scale flow field, and dendrite growth without the flow field to demonstrate the effect of the flow field on the dendrite structure. Moreover, we introduced a fourth-order Shuman-like filter in this chapter and compared the results from the fourth-order and the second-order filters with different filter parameter $\beta$. 

145
7.2 Conclusions

In this section, we will draw the conclusions from three different aspects of the current research: validation of the 2-D phase-field model via numerical tests, dendrite growth without introduction of the flow field, and dendrite growth in the lid-driven-cavity flow at $Re = 5000$.

From the numerical tests conducted on the 2-D phase-field model without a flow field, we learn that:

1. Stability of the numerical method employed in the phase-field model is satisfied by the choice of the interfacial thickness parameter $\epsilon$, since the unconditional stability of trapezoidal integration for linear constant-coefficient problems will be lost for $\epsilon > 0$ sufficiently small in the present nonlinear case. The interfacial thickness parameter $\epsilon$ should be in an appropriate range according to the values of spatial step size $\Delta x$ and $\Delta y$ as shown in Section 5.2.1.

2. Comparison between the explicit and implicit schemes shows that although the ratio of total arithmetic between the implicit and explicit schemes is of $\mathcal{O}(10)$, the time step size required for stability of the implicit scheme is 10 times larger than that of the explicit one.

3. Comparison between the fully-coupled and sequential solution techniques applied to the discretized governing equations indicates that the CPU time of the sequential method is approximately 10% larger than that of the coupled method. However, the storage needed for the coupled method is much greater than that for the sequential one. Therefore, we choose the sequential method because of the high demand of storage in the current computations.

4. The grid function convergence tests imply a primary second-order accuracy at some fine grids. These numerical tests proved the correctness of the numerical methods employed, and they also showed that the solutions obtained from the discrete equations converge to the solutions of the differential equations, i.e., the governing equations.

Simulations of 2-D dendrite growth without a flow field as was conducted in Chapter 5 show that:
1. Dendrite exhibits different structures at different Stefan numbers: from dendritic to “scalloped”, then quasi-planner. We can also expect that as supercooling continues to increase, the interface becomes almost indistinguishable from a planar interface. The above phenomena are also predicted analytically in the literature.

2. The computed solutions provide the relation between the interfacial growth rate \( V_n \) and time \( t \) at different Stefan numbers \( St \) as follows: \( V_n \sim t^{-\frac{2}{3}} \) for \( St < 1 \), \( V_n \sim t^{-\frac{1}{5}} \) for \( St = 1 \), and \( V_n = \text{constant} \) for \( St > 1 \). It is clear that this trend is the same as that seen in the 1-D analytical solution (1-D analytical solution is used for comparison because of the absence of a 2-D one).

3. Comparisons of \( V_n \) between the computed results, approximate solutions and experimental data show that \( V_n \) computed from the 2-D phase-field model qualitatively follows the analytical solution. In addition, we predict that solutions of the 3-D phase-field model should also follow this rule.

4. We determine the lateral mechanism for dendrite growth with \( St < 1 \) and the continuous mechanism with \( St > 1 \) for pure nickel.

5. The interfacial thickness parameter \( \epsilon \) is a free parameter, i.e., it can be changed considerably without causing a significant difference in the development of the interface. Therefore, the change of \( \epsilon \) will not alter the interface mean propagation speed \( V_n \). We note, however, that dendrite structure changes with \( \epsilon \); and it is caused by the perturbation introduced to the system by \( \epsilon \).

6. We observe that the interfacial speed increases with the increase of the kinetic coefficient \( \mu_k \). Moreover, dendrite structure becomes more complicated as the kinetic coefficient becomes larger, since the kinetic coefficient acts as a perturbation parameter.

7. Different shapes of the initial seed will cause the dendrite to grow into different structures. This phenomenon can be explained via the difference produced by the local curvature of the initial shape. The larger the curvature, the higher the interface
speed—thus the more complicated the dendrite structure. Therefore, various speeds on the interface cause the difference in dendrite structure. However, the interfacial mean propagation speed will not be affected by the initial shape.

8. The phase-field model is able to capture the physics, e.g., the critical area (radius) exhibited in solidification processes. Comparison between the computed critical areas at different degrees of supercooling $\Delta T$ and the analytical ones indicates that the critical area is decreasing as $\Delta T$ increases, and the computed critical areas are qualitatively very similar to the theoretical ones. Moreover, the results for nickel show that the assumption $\Delta c_p = 0$ is good enough for the computation, while those of water indicate that constant $\Delta c_p(\neq 0)$ should be assumed.

9. Effects of the degree of anisotropy $\epsilon'$ must be studied with introduction of anisotropy in surface energy. We found that the main branches of a dendrite grow faster with increase of $\epsilon'$, while the side branches develop better with the decrease of $\epsilon'$. Moreover, we do not observe any side branches if $\epsilon'$ is large enough, e.g., $\epsilon' = 0.05$. However, different degrees of anisotropy only produce different dendrite structures; they do not change the interfacial speed.

10. Simulations with different interfacial thickness and initial shapes with introduction of anisotropy reinforce our previous conclusions (for isotropic dendrite growth); i.e., dendrite structure computed from the large interfacial thickness is more complicated than that obtained from the small one, and multiple interfaces may exist in the solid region if the interfacial thickness is large enough; different initial shapes produce different dendrite structures. However, dendrite growth rate remains unchanged no matter how we choose the interfacial thickness and the initial shape.

In Chapter 6, we conduct simulations of ice crystal growth in the lid-driven-cavity flow at $Re = 5000$ with implementation of a multiscale method. We found that:

1. Comparison of the results from Shuman filters with different filter parameter $\beta$ indicates that more aliasing can be removed for $\beta = 200$ than $\beta = 2000$. In addition, solution
from the fourth-order filter is more accurate than that from the second-order filter, especially in the boundary-layer region.

2. The flow field has great effect on dendrite structure; indeed it increases dendrite growth rate more in the downstream direction than in the upstream. In addition, development of the side branches is influenced by the flow field such that side branches develop better in downstream than upstream.

3. The mean interface propagation speed increases monotonically with increase of velocity magnitude on the large scale.

4. The flow field behavior on the large scale is altered by freezing inside it. Not only the flow near where the freezing results in flow over the solid body, but also the flow structures (e.g., sizes and structures of the vortices in the lid-driven-cavity flow) far away from the freezing location are changed.

5. Study of the parallel performance shows that as the number of processors increases, the speedup factor also increases only sub-linearly for both OpenMP and MPI algorithms. However, the speedup performance of MPI is better than that of OpenMP.

7.3 Future Work

Although numerous works have been done on simulations using phase-field models (e.g., freezing with introduction of a flow field as in the current research, and previous works such as solidification processes of binary alloy by peers), there are many problems that have not been solved and plenty of areas in which phase-field models could be employed but have not yet to be. These problems include simulations of ice flows, casting processes, melting and premelting problems, de-icing problems, etc. In addition, the current phase-field model can be extended to 3D to solve more realistic problems, and to different fields such as electromagnetic field, and flows considering elastic forcing. Moreover, they can also be applied to those problems with a sharp front, e.g., problems solving flame front and shock capturing problems. In the near future, we are planning to simulate ice flows and melting
processes under different conditions. Other interesting problems and applications will be considered later.

We note that simulations of the lid-driven-cavity flow with freezing in the current research can be easily extended to an arbitrary 3-D flow with ice formation using the phase-field model (3.26) provided in Chapter 3 and the multiscale method discussed in Chapter 4. Very little effort has been contributed to simulations of melting problems because of limited knowledge in this area. Nevertheless, the phase-field model can be applied to solve such problems via exchanging the description of bulk phases, determining the direction of heat flux on the interface, replacing the Stefan condition (1.6c) by a relation that includes the surface tension and other effects on the interface during melting processes. As long as melting processes can be simulated successfully, it is possible to solve premelting problems using a similar method.

It should be mentioned that Du et al. [118] applied a phase-field approach to compute the equilibrium configurations of a vesicle membrane under elastic bending energy. In their research, they introduced an elastic energy replacing the internal energy (3.2) as in the current phase-field model. Therefore, the structure of the interface (vesicle membrane) is determined by the elastic energy rather than the interfacial energy as in phase-change problems. This study ensures that phase-field approaches can be applied to problems with an interface in different areas. We also note that the forcing term (3.27a) in the momentum equations (3.26a) has a similar form as that in problems to solve for different flows. For example, in the approximation of liquid crystal flows by Liu and Walkington [119], we note that the governing equations are very similar to the phase-field model with flow field in the current research, and the so-called director fields of a liquid crystal have the same properties as the phase field of a melt.

Considering the fact that introduction of the entropy functional (3.3) in Chapter 3 is very similar to that in shock capturing problems, the phase-field model has the possibility to solve shock capturing problems, especially with its advantage of labeling the front implicitly.

Finally, we summarize that phase-field approaches can be applied to a wide range of problems and applications, especially to those problems with different phases, in different areas. We foresee that these approaches will cause more and more interest with the improvement of the algorithm efficiency and the performance of computer systems.
BIBLIOGRAPHY


114. Copyright ©1988-2006 Tecplot, Inc.


PERSONAL INFORMATION

Date of Birth: August 13, 1974
Place of Birth: Jiangxi, China

EDUCATION

Aug. 2006: Ph.D., Mechanical Engineering, University of Kentucky, Lexington, KY, GPA 3.951/4.0
May 2005: M.A., Mathematics, University of Kentucky, Lexington, KY
Jan. 1998: M.S., Thermal Engineering, Tongji University, Shanghai, China
July 1995: B.S., Thermal Engineering, Tongji University, Shanghai, China

WORK EXPERIENCE

Graduate Assistant, Department of Mechanical Engineering and Mathematics, University of Kentucky, Lexington, KY, Aug. 2000 – present:

- Performed research on thermal sciences and computational fluid dynamics
- Served as Teaching Assistant and taught courses

Engineer, Research and Development Center, Shanghai Hitachi Electronic Appliance Co., Ltd., Shanghai, China, Feb. 1998 – July 2000:

- Developed and designed compressor for air conditioner with new refrigerant using SolidWorks
- Chose lubricant corresponding to new refrigerant
- Conducted life tests for new compressor
• Communicated with Production Department and Quality Assurance Department

RESEARCH INTERESTS/EXPERIENCE

• Multiscale scheme (macroscale computation on flow field, microscale/nanoscale computation on crystal growth)
• Numerical analysis and methods
• Computational fluid dynamics
• Fluid dynamics and heat transfer
• Turbulence models: Direct Numerical Simulation (DNS), Reynolds-averaged Navier-Stokes (RANS) and Large-Eddy Simulation (LES), etc.
• Phase-change processes in turbulent flows
• Multiphase flows, liquid crystal flows and viscoelastic flows
• Radiation-Turbulence Interactive (RTI) flames and combustion

AWARDS

• Fellowship, Center for Computational Sciences at the University of Kentucky (2005)
• Fellowship, Center for Computational Sciences at the University of Kentucky (2004)
• Nominated as Outstanding Mechanical Engineering Graduate Student at the University of Kentucky (2004)

PROFESSIONAL ACTIVITIES

• Member: the American society of Mechanical Engineering (ASME)
• Member: the American Physical Society (APS)
PUBLICATIONS

• Archival Journal Papers


• Refereed Conference Papers


CONFERENCE PRESENTATIONS/ABSTRACTS (UNREFEREED)


Ying Xu, J.M. McDonough and K.A. Tagavi, “Two-Dimensional Simulation of Solidification in Flow Field Using Phase-Field Model: Multi-Scale Scheme Implemented,” Parallel
INVITED TALKS


Ying Xu, J.M. McDonough and K.A. Tagavi, “Simulations of Phase-change process with Phase-Field Model in Flow Field,” *CCS Brownbag Seminar*, Center for Computational Sciences, University of Kentucky, April 12, 2005.

Ying Xu, J.M. McDonough and K.A. Tagavi, “Two-Dimensional Simulation of Solidification in Flow Field Using Phase-Field Model: Multi-Scale Scheme Implemented,” *CCS Brownbag Seminar*, Center for Computational Sciences, University of Kentucky, April 18, 2006.