INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

NOTE TO USERS

The original manuscript received by UMI contains pages with slanted print. Pages were microfilmed as received.

This reproduction is the best copy available

UMI

A 3D Phase Field Simulation of Solidification

by

Rong Wang, B. Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Engineering

Department of Mechanical and Aerospace Engineering
Ottawa-Carleton Institute
for Mechanical and Aerospace Engineering

Carleton University
Ottawa, Ontario
April, 1998

© 1998

Rong Wang

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-32409-5

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis

A 3D Phase Field Simulation of Solidification

submitted by

Rong Wang, B. Eng.
in partial fulfillment of the requirements
for the degree of Master of Engineering

Thesis Supervisor

Thesis Supervisor

Chair, Department of Mechanical and Aerospace Engineering

Carleton University

April, 1998

Abstract

Solidification is a very important phase transformation. The modeling and simulation of solidification has become essential for many years. This thesis investigates the application of Karma's phase field model for simulation of solidification at a fixed temperature in 3D. In a phase field model, the phase field variable is used to keep track of the solid and liquid phase fractions during solidification.

The finite element method has been used to solve the partial differential phase field equation. 6_Node_Brick finite elements are used. A Gaussian fitting function is chosen to fit the phase field explicitly.

In the present work, 3D finite element phase field simulation and modeling procedures and related computer programs are developed for the analysis of phase field distributions. Spherical and planar growth morphologies are analyzed with supercooling and superheating. A number of case studies for pure metal are carried out to verify the validity of the model.

Acknowledgments

I would like to express my high appreciation to my supervisor, Professor John Goldak, for his knowledge, guidance and supervision during the course of this work.

I wish to thank Professor A. Artemev and the colleagues in MRCO Group for helping me in this work patiently.

Special thanks to Department of Mechanical and Aerospace Engineering, the Faculty of Graduate Studies and Research for their financial support.

Finally, I would like to express my deep gratitude to my husband Jinglei for his love and encouragement, my lovely daughter Carolynn and son Kelvin for their smiling, and my parents for their great help.

Table of Contents

Acceptance Sheet.....	ii
Abstract.....	iii
Acknowledge.....	iv
Table of Contents.....	v
List of Tables.....	vii
List of Figures.....	ix
List of Symbols.....	xi
Chapter 1 Introduction	1
1.1. Importance of Solidification.....	1
1.2. Model and FE Method.....	2
1.3. Scope of Work.....	3
Chapter 2 Solidification Models.....	5
2.1. Literature Review.....	5
2.1.1. Sharp Interface Models.....	6
2.1.2. Smooth Interface Models.....	7
2.2. Summary of Advantages and Disadvantages of Models.....	9

Chapter 3 Karma's Phase Field Model.....	11
3.1. Governing Equations and Assumptions.....	11
3.2. Development of the Model.....	13
3.3. Numerical Algorithm.....	18
3.3.1 Gaussian Function Fitting.....	19
3.3.2. Phase Field Solver.....	22
 Chapter 4 Results of Numerical Studies.....	 28
4.1. Selection of Parameters.....	28
4.2 Three Dimension Results.....	32
4.2.1 Test Case No.1: Plane Geometry.....	32
4.2.2 Test Case No.2: Sphere Geometry.....	40
4.3. Discussions of the Results of Test Cases.....	59
 Chapter 5 Conclusion and Future Work.....	 64
5.1. Conclusion.....	64
5.2. Future work.....	65
 References.....	 66

List of Tables

- 4.1 Properties of the Casting Material, pure Al
- 4.2 Undercooling = 0.001, Test No. 1
- 4.3 Undercooling = -0.001, Test No. 2
- 4.4 Undercooling = 0.0001, Test No. 3
- 4.5 Undercooling = -0.0001, Test No. 4
- 4.6 Undercooling = 0.001, Radius = $6.4 \cdot 10^{-7} m$, Test No. 5
- 4.7 Undercooling = 0.00095, Radius = $6.4 \cdot 10^{-7} m$, Test No. 6
- 4.8 Undercooling = 0.0009, Radius = $6.4 \cdot 10^{-7} m$, Test No. 7
- 4.9 Undercooling = 0.0008, Radius = $6.4 \cdot 10^{-7} m$, Test No. 8
- 4.10 Undercooling = 0.0005, Radius = $6.4 \cdot 10^{-7} m$, Test No. 9
- 4.11 Undercooling = 0.001, Radius = $5.5 \cdot 10^{-7} m$, Test No. 10
- 4.12 Undercooling = 0.001, Radius = $6.0 \cdot 10^{-7} m$, Test No.11
- 4.13 Undercooling = 0.00091, Radius = $6.4 \cdot 10^{-7} m$, Test No.12
- 4.14 Undercooling = 0.00092, Radius = $6.4 \cdot 10^{-7} m$, Test No.13
- 4.15 Undercooling = 0.00093, Radius = $6.4 \cdot 10^{-7} m$, Test No.14

- 4.16 Undercooling = 0.00094, Radius = $6.4 \cdot 10^{-7} m$, Test No.15
- 4.17 Test results for plane from supercooling to superheating
- 4.18 Test results for sphere when radius decreases
- 4.19 Test results for sphere when undercooling decreases
- 4.20 Test results for sphere when undercooling increases a little

List of Figures

- 2.1 Schematic figure for sharp interface
- 2.2 Schematic figure for smooth interface
- 3.1 Phase field element, 6_Nodes_Brick
- 3.2 Flow diagram of the adaptive algorithm
- 4.1(a) big tip radius to element size ratio
- 4.1(b) small tip radius to element size ratio
- 4.2 Curve Distance vs. Time and Curve Velocity vs. Time
for Plane when undercooling = 0.001
- 4.3 Curve Distance vs. Time and Curve Velocity vs. Time
for Plane when undercooling = -0.001
- 4.4 Curve Distance vs. Time and Curve Velocity vs. Time
for Plane when undercooling = 0.0001
- 4.5 Curve Distance vs. Time and Curve Velocity vs. Time
for Plane when undercooling = -0.0001
- 4.6 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.001, Radius=6.4e-7m
- 4.7 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00095, Radius=6.4e-7m
- 4.8 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.0009, Radius=6.4e-7m
- 4.9 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.0008, Radius=6.4e-7m

- 4.10 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.0005, Radius = $6.4 \times 10^{-7} \text{m}$
- 4.11 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.0005, Radius = $5.5 \times 10^{-7} \text{m}$
- 4.12 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.001, Radius = $6.0 \times 10^{-7} \text{m}$
- 4.13 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00091, Radius = $6.4 \times 10^{-7} \text{m}$
- 4.14 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00092, Radius = $6.4 \times 10^{-7} \text{m}$
- 4.15 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00093, Radius = $6.4 \times 10^{-7} \text{m}$
- 4.16 Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00094, Radius = $6.4 \times 10^{-7} \text{m}$
- 4.17 Initially defined sphere as solid (+1) in the center of each cube.
- 4.18 After one step the sphere shrinks a little bit

Chapter 1

Introduction

1.1. Importance of Solidification

Solidification is a very important phase transformation, because all metals undergo the liquid-to-solid transformation involving solidification at some stage. The study of solidification is of immense practical value. The understanding and modeling of solidification have remained a central problem in pattern formation and metallurgy for many years.

It is essential for the metallurgists and engineers that they must deal in a very practical way with the process design for materials in a wide range, from large cast products to relatively small quantities of crystals.

Casting is a major practical application of solidification. It is a very economic method of forming a component [1].

During solidification, crystals are generally formed. The formation of crystals is important because the microstructure dictates many of the resulting material properties. In solidification processing, the most commonly observed microstructures are plane front, cellular, dendrite or eutectic microstructure [2]. From a practical point of view, we need to understand how we can produce a desirable structure.

In recent years, researchers and scientists are very much interested in the studies of solidification process, modeling and numerical simulation.

1.2. Models and FE method

Researchers and scientists have developed some useful models to simulate phase transformation during solidification. The classical approach is tracking the interface between solid phase and liquid phase. The interface thickness is zero. So the interface is very sharp. These models are called sharp interface models.

Another approach is to use smooth interface models. Smooth interface models treat the interface as a thin interface with a smooth transition between liquid and solid. Sharp interface is a limit of smooth interface when the interface thickness is zero.

For many years, the phase field model was implemented for modeling, but did not lead to significant computational results in three dimensions. Kobayashi's work is a demonstration of realistic solid/liquid interfaces by using phase field models of solidification as a computational tool [3].

Recently, the increasing computing power has enabled the computation of quite complex morphologies, such as dendrite growth in two dimensions [4-11]. There has even been some success with computations in three dimensions [12-14].

A. Karma and W. J. Rappel [12] presented mathematical results that dramatically enhance the computational efficiency of the phase-field method for modeling the solidification of a pure material. Dendrite crystal growth is simulated quantitatively in three dimensions using a phase field model. The model assumes that a solid-liquid interface varies smoothly across a finite thickness, so it is a smooth interface model. Three-dimensional computations are accessible through this model.

The emergence of the FE method took place in the early 1960s and since then its use has spread to virtually all fields of engineering. FEM is an important numerical method. We implement the 3D model for solidification analysis using a phase field model in FEM.

We now present some definitions. The finite element method is a method of piecewise approximation in which the approximating function ϕ is formed by a linear combination of simple functions, each defined over a small region (element). A finite element is a region in space in which a function ϕ is interpolated from nodal values of ϕ on the boundary of the region in such a way that interelement continuity of ϕ is maintained

in the assemblage. A finite element analysis typically involves the following steps for the phase field analysis.

1. Divide the structure or continuum into finite elements. Mesh generation programs, called mesh generators, are used.
3. Assemble elements to obtain the finite element global system of equation of the structure.
4. Apply all known values of nodal driving force.
5. Initially, the phase field is known, initialize all known values of nodal phase field.
6. Apply boundary conditions.
7. Solve phase field equation to determine nodal phase field value at a later time. Usually a sequence of time steps is used.
8. Do the phase field interpolation. Output interpretation programs, call postprocessors, help us interpret the output and display it in graphical form.

The power of the finite element method resides principally in its versatility. This method can be applied to various physical problems. The mesh can mix elements of different type, shapes, and physical properties. This versatility can be contained within a single computer program. User-prepared input data, including design file, grid file, etc., control the selection of problem type, geometry, boundary conditions, element selection, and so on.

1.3. Scope of Work

In the present work, a numerical algorithm is developed for a three dimensional phase field analysis and simulation for solidification. Detailed formulations for the phase field model are developed for a partial differential equation which governs the phase field evolution. Use of Gaussian fitting function to minimize the approximation errors is discussed in detail.

The real test case studies take into account the appropriate parameter selection, initial conditions, boundary condition, etc. The analysis focuses on an estimate of phase field distribution.

Software programming has been implemented to accomplish this analysis, and output from the software provides graphical representations of the entire history of phase field distributions.

Chapter 2

Solidification Models

2.1. Literature Review

A good model and good numerical method as well as a powerful computer are needed to accomplish the solidification simulation. Through our study, we know that, it is a challenge to model solidification. Let us review the literature on the existing models.

Up to now, many existing models have been developed, especially in dendrite growth research fields. Basically, all the researchers have to start with the same basic equations, energy equation, phase field equation and solute equation. These basic equations are all partial differential equations. All models have to represent the interface and its motion. Although different models do it in different ways, they choose certain coupling equations and algorithms.

The classical approach to modeling first order phase transformations, involves tracking of the free boundary that separates the new growing phase from the parent phase through a fixed grid. This requires solution of the difficult free boundary problem to implement numerically. One must solve the energy equations for the temperature fields in both solid and liquid subject to boundary conditions for the temperature and the derivatives on the moving solid-liquid interface. Because the solid-liquid interface tracking is involved, and interface thickness is zero, we call them sharp interface models.

An alternative approach is to use smooth interface models, such as phase field model, which have recently become a popular method for modeling complicated problems in the solidification field. Phase field model can be a sharp interface model also. Smooth

interface model tracks the boundary by replacing the PDE for the boundary conditions on the moving boundary by the PDE involving the phase field that keeps track of the phase fractions and makes a transition between these values over a thin transition layer. One needs to couple the equations for the temperature and /or composition field, and the equation for the phase field for this type of model.

Let us review models first, and then briefly consider the strengths and weaknesses of common methods: phase field method, front tracking and level set.

2.1.1 Sharp Interface Models

In this model, the interface is sharp, more precisely speaking, the interface thickness is assumed to be zero. And on the interface between solid and liquid, a jump of temperature gradient or solute concentration must exist during the solidification process.

There are several well-developed numerical methods for curvature dependent motion. J. A. Sethian and John Strain [15] proposed the Level Set Method, and J. Leveque [16] proposed a front tracking method. The energy equation and jump condition equation are used in these two methods. These methods solved the energy equation (2.1), jump function equation (2.2) and undercooling (2.3)

$$\partial_t u = D \nabla^2 u \quad (2.1)$$

$$v_n = D (\partial_n u|_S - \partial_n u|_L) \quad (2.2)$$

$$u_i = -d_0 k - \beta v_n \quad (2.3)$$

where

u is the dimensionless temperature field defined by $(T - T_m) / (L / C_p)$,

T_m is the melting temperature,

L is the latent heat of melting,

C_p is the specific heat at constant pressure,

D is the heat diffusivity coefficient,

v_n is the normal velocity of interface,

$\partial_n T|_s$ is the normal derivative of temperature on the solid side of the interface,

$\partial_n T|_L$ is the normal derivative of temperature on the liquid side of the interface,

u_i is the undercooling on interface,

d_0 is the capilarity length ,

β is the interface kinetics coefficient,

k is the interface curvature

Fig.(2.1) below is schematic, but it shows the basic assumption of a sharp interface approach — the phase field jumps at the interface, and the interface is very sharp because the thickness is zero.

Fig. (2.1) Schematic figure for sharp interface

In level set methods, Sethian represented the interface curve as a level set of a function on the plane, and this function in turn evolves according to a partial differential equation, chosen precisely to produce the proper level set motion. Level set method does not tell the interface is sharp or smooth, it tells the interface position. But level set method is usually used in sharp interface model. This has the advantages of requiring no special treatment of topological changes and a simple algorithm because only finite differences on

a uniform grid are needed. Further, the grid is not artificially fine, so it is reasonable and it extends easily to three dimensions.

In the front tracking methods, the curve is replaced by a string of discrete points, the curvature of interface is computed by finite difference, and the points move in time with the computed velocity. This has the advantage of computational efficiency, and generalizes well to arbitrary motion laws. The main problem to be solved is that the complicated algorithm is needed to track fronts that merge, break, or otherwise change topology. In three dimensions these complications increase severely. J. LeVeque [16] solves the energy equation too with the jump in temperature gradient.

2.1.2 Smooth Interface Models

On the other hand, smooth interface models assume the interface thickness is finite instead of zero and a smooth transition between liquid and solid exists. This means that the phase field varies smoothly across a finite thickness. A phase field model is a good example of smooth interface models. See the schematic figure below.

Fig.(2.2) Schematic figure for smooth interface

The formulation of the phase-field method for an alloy involves both temperature and solute redistribution. By introducing a variable, the phase field variable that defines the physical state (liquid or solid) of the system at each point and the governing equation is as following Eq. (2.4) and (2.5).

An evolution equation for the phase field is often postulated by requiring that the phase field φ evolve so as to minimize a free energy functional F , that is, by setting

$$\tau(n) \frac{\partial \varphi}{\partial t} = - \frac{\delta F}{\delta \varphi} \quad (2.4)$$

where

φ is the phase field,

$\tau(n)$ is the time scale of the phase field,

n is the unit normal of the interface.

The energy equation is as below,

$$\frac{\partial u}{\partial t} = D \nabla^2 u + \frac{1}{2} \frac{\partial \varphi}{\partial t} \quad (2.5)$$

where u is the dimensionless temperature field, and D is the heat diffusivity coefficient.

In the energy equation, the term $\frac{1}{2} \frac{\partial \varphi}{\partial t}$ actually contributes to the heat source.

The phase field variable in solid and liquid is defined as 1 and -1 or 1 and 0. The transition between liquid and solid is represented by a smooth but extremely localized change in the phase field variable. In the limit, the thin transition layer has to approach a sharp interface. The phase field φ is governed by a partial differential equation (2.4). Moreover, the energy equations are modified by the addition of terms that depend on the phase field in (2.5). One then proceeds to solve the coupled equations for the phase field, temperature and/or composition fields without the necessity of explicit tracking of the free boundary.

Phase field models of solidification were invented by Langer, see [17]. For a long time, they have been subjected to development and rigorous mathematical analysis by researchers. Because advanced and sufficient computer technology is developed, it is now possible to numerically integrate the unsteady phase field equations in realistic

configurations. An early computation in one spatial dimension was obtained by Caginalp [18]. In two and three spatial dimensions, a similar model including anisotropy was introduced and proved to be able to describe realistic dendrite patterns numerically by Kobayashi [3] and Karma [7]. Their work clearly showed the feasibility of modelling the evolution of the complicated, realistic solid /liquid interface of a solid dendritic structure into an undercooled melt, and qualitatively demonstrate the possibility of phase field models of solidification as a computational tool.

2.2. Summary of Advantages and Disadvantages of Models

A sharp interface model has the disadvantages of a complicated algorithm to accommodate curves with complicated structure. In three spatial dimension, these complications increase severely. Classically, one must solve the equations for the temperature fields in both solid and liquid subject to boundary conditions for the temperature and its derivatives on the moving solid-liquid interface, the free boundary. It is almost impossible to simulate the dendrite growth with a sharp interface method in three spatial dimensions.

On the other hand, using a smooth interface method, explicit tracking of the solid-liquid interface is avoided although both models are physically correct. This is an advantage in three spatial dimension, because only a simple algorithm is needed. The interface thickness can be chosen to be at the microstructure level. For example, in the phase field model, the phase field variable ϕ , is introduced to keep track of the phase, taking on constant values indicative of each of the bulk phases, e.g. solid -1, liquid 1, and making a transition between these values over a thin transition layer that plays the role of the classically sharp interface.

So, the central change in a smooth interface approach is abandoning the notion of a sharp transition between liquid and solid. By introducing the phase field variable, that defines the physical state (liquid or solid) of the system at each point and its governing equation, one can employ numerical methods that avoid interfacing tracking.

Chapter 3

Phase Field Model

3.1. Governing Equations and Assumptions

As we stated before, phase field models provide a convenient basis for the numerical solution of complicated solidification problems. In a phase field model, the phase field, is introduced to define explicitly the liquid and solid phases. The phase field takes on a constant value in each bulk phase, the solid phase field is assumed to be 1, the liquid phase field is assumed to be -1. The transformation from liquid to solid occurs over a thin transition region where the phase field varies smoothly from -1 to 1.

Recently, Kobayashi [3] introduced a phase field model to compute the dendritic solidification of a pure material from a supercooled melt, in two dimensions. In all computations of dendritic growth based on phase field models, it was necessary to employ high undercooling in order to mitigate the disparity of the thermal length scale and the interface thickness, until A. Karma and W. J. Rappel [12] made it possible to apply to smaller undercooling.

A. Karma and W. J. Rappel [7] have presented an analysis of the smooth interface method of the phase-field model of a pure material which includes temperature variations across the interface thickness. The phase field equation coupled with the transient energy equation have been solved. The mathematical results are reported both in two [7] and three [12] spatial dimensions recently by A. Karma and W. J. Rappel. They made the 3D simulation computations possible by using the phase-field approach.

The two-phase system is described by a phenomenological free-energy, and the free-energy can be written in a general form as:

$$F = \int d\mathbf{r} [W^2(n) |\nabla \varphi|^2 + f(\varphi, u)] \quad (3.1)$$

where,

φ is the phase field,

$n = \frac{\nabla \varphi}{|\nabla \varphi|}$ is the unit normal to the isosurface or level set of phase field,

$W(n)$ is the interface thickness,

u is the dimensionless temperature field.

Dimensionless temperature u is determined by $\frac{T - T_m}{L / C_p}$. The function $f(\varphi, u)$ is chosen as

$$f(\varphi, u) = -\frac{\varphi^2}{2} + \frac{\varphi^4}{4} + \lambda u \left(1 - 2\frac{\varphi^3}{3} + \frac{\varphi^5}{5}\right) \quad (3.2)$$

with $-1 \leq \varphi \leq +1$ that correspond to the liquid and solid phases, respectively. λ is the strength of the coupling constant between u and φ . Generally, the anisotropy of surface free energy or of interface kinetics is thought to play a fundamental role in the dynamics of dendritic growth. The anisotropic surface energy in Eq (3.1) is recovered by choosing interface thickness $W(n)$ as

$$W(n) = W_0 \gamma(n) / \gamma_0 \quad (3.3)$$

$$\gamma(n) = \gamma_0 (1 - 3\varepsilon_4) \left[1 + \frac{4\varepsilon_4}{1 - 3\varepsilon_4} (n_x^4 + n_y^4 + n_z^4) \right] \quad (3.4)$$

with the surface energy for a crystal chosen to have a cubic symmetry. The definition of $W(n)$ and $\gamma(n)$ are both suggested by A. Karma and W. J. Rappel [12].

Since in a phase field formulation the interface is diffuse, the proper incorporation of the surface free-energy anisotropy requires careful consideration. Phase field models with anisotropy have been considered previously for specific choices of gradient energy. Caginalp and Fife [19, 20] have considered using a more general quadratic form with different coefficients in each coordinate direction. In order to obtain more complicated anisotropies, Langer [17] proposed the addition to the gradient energy of terms involving the squares of higher derivatives of the phase field, and gave an example leading to cubic anisotropy.

We assume the phase field, φ , varies smoothly between -1 and +1 across an spatially diffuse interfacial region of finite thickness W_0 , thereby distinguishing between phases without front-tracking. Its dynamics is defined by Eqn (3.5).

$$\tau(n) \frac{\partial \varphi}{\partial t} = - \frac{\delta F}{\delta \varphi} \quad (3.5)$$

And then Eqn (3.5) is coupled to the heat equation

$$\frac{\partial u}{\partial t} = D \nabla^2 u + \frac{1}{2} \frac{\partial \varphi}{\partial t} \quad (3.6)$$

where τ is the time scale of the φ field kinetics and D is the heat diffusivity coefficient. Temperature is allowed to vary across the solid-liquid interface. In this thesis Eqn (3.6) is not solved.

3.2. Development of the Model

The following is the PDE formulation from A. Artemev for the phase field model in 3D [21]. From Eq (3.5), we know that the variation of free energy functional F needs to be developed, and F is defined in Eq (3.1) as

$$F = \int d\mathbf{r} [W^2(n) |\nabla \varphi|^2 + f(\varphi, u)]$$

The contribution from the phase field term into the derivative of $f(\varphi, u)$ can be obtained as:

$$\begin{aligned}\frac{\partial f}{\partial \varphi} &= \varphi^3 - \varphi + \lambda u(1 - 2\varphi^2 + \varphi^4) \\ &= (1 - \varphi^2) [\lambda u(1 - \varphi^2) - \varphi]\end{aligned}\quad (3.7)$$

The gradient derivatives can be written as:

$$\frac{\partial F}{\partial (\partial \varphi_i)} = 2 W^2(n) \nabla \varphi + \left[\frac{\partial W^2(n)}{\partial (\partial \varphi_i)} \right] \cdot |\nabla \varphi|^2 \quad (3.8)$$

The contribution from the first term in Eq. (3.8) into the variational derivative can be written as:

$$-\frac{\partial}{\partial x_i} [2 W^2(n) \cdot \nabla \varphi] = -2 \nabla [W^2(n) \cdot \nabla \varphi] \quad (3.9)$$

The second term in Eq. (3.8) consists of three parts.

$$\left[\frac{\partial W^2(n)}{\partial (\partial \varphi_x)} \right] \cdot |\nabla \varphi|^2 + \left[\frac{\partial W^2(n)}{\partial (\partial \varphi_y)} \right] \cdot |\nabla \varphi|^2 + \left[\frac{\partial W^2(n)}{\partial (\partial \varphi_z)} \right] \cdot |\nabla \varphi|^2$$

Let us have a look at each part.

$$\begin{aligned}& \left[\frac{\partial W^2(n)}{\partial (\partial \varphi_x)} \right] \cdot |\nabla \varphi|^2 \\ &= 2 W(n) \left[\frac{\partial W(n)}{\partial n_x} \frac{\partial n_x}{\partial (\partial \varphi_x)} + \frac{\partial W(n)}{\partial n_y} \frac{\partial n_y}{\partial (\partial \varphi_x)} + \frac{\partial W(n)}{\partial n_z} \frac{\partial n_z}{\partial (\partial \varphi_x)} \right] |\nabla \varphi|^2\end{aligned}$$

$$\begin{aligned}
&= 2 W(n) W_0 (1 - 3\varepsilon_4) \left[\frac{4\varepsilon_4}{1 - 3\varepsilon_4} \cdot 4 \right] \cdot \left[n_x^3 \frac{\partial n_x}{\partial (\partial \varphi_x)} + n_y^3 \frac{\partial n_y}{\partial (\partial \varphi_x)} + n_z^3 \frac{\partial n_z}{\partial (\partial \varphi_x)} \right] \cdot |\nabla \varphi|^2 \\
&= 32 W(n) W_0 \varepsilon_4 \left[n_x^3 \frac{(\partial \varphi_y^2 + \partial \varphi_z^2)}{|\nabla \varphi|} - n_y^3 \frac{\partial \varphi_x \cdot \partial \varphi_y}{|\nabla \varphi|} - n_z^3 \frac{\partial \varphi_x \cdot \partial \varphi_z}{|\nabla \varphi|} \right]
\end{aligned}$$

The other two parts have the similar structure, and can be easily obtained as:

$$\left[\frac{\partial W^2(n)}{\partial (\partial \varphi_y)} \right] \cdot |\nabla \varphi|^2 = 32 W(n) W_0 \varepsilon_4 \left[n_y^3 \frac{(\partial \varphi_x^2 + \partial \varphi_z^2)}{|\nabla \varphi|} - n_x^3 \frac{\partial \varphi_x \cdot \partial \varphi_y}{|\nabla \varphi|} - n_z^3 \frac{\partial \varphi_y \cdot \partial \varphi_z}{|\nabla \varphi|} \right]$$

$$\left[\frac{\partial W^2(n)}{\partial (\partial \varphi_z)} \right] \cdot |\nabla \varphi|^2 = 32 W(n) W_0 \varepsilon_4 \left[n_z^3 \frac{(\partial \varphi_x^2 + \partial \varphi_y^2)}{|\nabla \varphi|} - n_x^3 \frac{\partial \varphi_x \cdot \partial \varphi_z}{|\nabla \varphi|} - n_y^3 \frac{\partial \varphi_y \cdot \partial \varphi_z}{|\nabla \varphi|} \right]$$

Now we can collect all the terms in functional derivatives:

$$\begin{aligned}
&\frac{\delta F}{\delta \varphi} = (1 - \varphi^2) [\lambda u (1 - \varphi^2) - \varphi] - 2 \nabla \cdot [W^2(n) \nabla \varphi] \\
&- \frac{\partial}{\partial x} \left\{ 32 W(n) W_0 \varepsilon_4 \left[n_x^3 \frac{(\partial \varphi_y^2 + \partial \varphi_z^2)}{|\nabla \varphi|} - n_y^3 \frac{\partial \varphi_x \cdot \partial \varphi_y}{|\nabla \varphi|} - n_z^3 \frac{\partial \varphi_x \cdot \partial \varphi_z}{|\nabla \varphi|} \right] \right\} \\
&- \frac{\partial}{\partial y} \left\{ 32 W(n) W_0 \varepsilon_4 \left[n_y^3 \frac{(\partial \varphi_x^2 + \partial \varphi_z^2)}{|\nabla \varphi|} - n_x^3 \frac{\partial \varphi_x \cdot \partial \varphi_y}{|\nabla \varphi|} - n_z^3 \frac{\partial \varphi_y \cdot \partial \varphi_z}{|\nabla \varphi|} \right] \right\} \\
&- \frac{\partial}{\partial z} \left\{ 32 W(n) W_0 \varepsilon_4 \left[n_z^3 \frac{(\partial \varphi_x^2 + \partial \varphi_y^2)}{|\nabla \varphi|} - n_x^3 \frac{\partial \varphi_x \cdot \partial \varphi_z}{|\nabla \varphi|} - n_y^3 \frac{\partial \varphi_y \cdot \partial \varphi_z}{|\nabla \varphi|} \right] \right\} \tag{3.10}
\end{aligned}$$

Then we could use Eq.(3.10) in phase field Eq.(3.5) for phase field evolution coupling with the heat equation Eq.(3.6).

On the right-hand-side of equation Eq.(3.10), there are three derivative terms, let us call them dSx, dSy and dSz. These three derivatives dSx, dSy and dSz, actually are second derivatives. If we rewrite the phase field equation (3.5) as

$$\begin{aligned} \tau(n) \frac{\partial \varphi}{\partial t} = & (1 - \varphi^2) \left[\varphi - \lambda u(1 - \varphi^2) \right] + dSx + dSy + dSz \\ & + 2\nabla [W^2(n) \nabla \varphi]. \end{aligned} \quad (3.11)$$

Let us compare Eq.(3.11) with the transient heat equation

$$C_p \dot{T} + \nabla \cdot (-k \nabla T) + Q = 0 \quad (3.12)$$

where C_p is specific heat, k is heat conductivity, and Q is the heat source. First both equations are diffusion equations. Second these two equations both have transient term $\frac{\partial \varphi}{\partial t}$ and $\dot{T}$, and they both have second derivative term $(\nabla^2 \varphi)$ and $(\nabla^2 T)$. So Eq (3.11) is similar to Eq. (3.12). The three terms $\tau(n)$, $(1 - \varphi^2) \left[\varphi - \lambda u(1 - \varphi^2) \right] + dSx + dSy + dSz$, and $2W(n)^2$ can be thought of, respectively, as analogues of the specific heat, the source term, and conductivity.

Now we are getting close to using FEM to solve this transient equation provided that we define all the parameters $\tau(n)$, λ and $W(n)$ properly, and work out the three derivatives dSx, dSy and dSz. The phase field equation finally will be solved in the form $Kx = b$ in FEM, where K is a stiffness matrix and b is a nodal load vector, and x is the nodal phase field variable that we need.

Function $\tau(n)$ and λ are chosen in such a way that we obtain a Gibbs-Thomson condition without interface kinetics in the sharp-interface limit. And $\tau(n)$, λ are suggested as follows by A. Karma and W. J. Rappel

$$d_0(n) = \frac{I}{J\lambda} [W(n) + W'(n)] \quad (3.13)$$

$$\lambda \approx W(n) / d_0(n)$$

$$\beta(n) = \frac{I}{J\lambda} \frac{\tau(n)}{W(n)} \left[1 - \lambda \frac{W(n)^2}{2 \cdot D \cdot \tau(n)} \frac{K + JF}{I} \right] \quad (3.14)$$

Where d_0 is thermal capillary length, I , J , and K are constant numerical values. The term in square brackets on the right hand side of Eq. (3.14) is the correction to the interface kinetic coefficient arising from the variation of u in the interface thickness. It is the main result of this model which makes it possible to extend the phase field approach. There are four constraints: $|k|W \ll 1$, $W|v_n|/D \ll 1$, $\lambda \cdot |u| \cdot (1 - \phi^2)^2 \ll 1$, and $\tau \cdot |v_n|/W \ll 1$. They originate from the assumptions made in deriving the sharp-interface limit. With the definitions $W(n) = W_0 \cdot f_w(n)$ and $\tau(n) = \tau \cdot f_\tau(n)$, we choose a cubic symmetry anisotropy function as in Eq. (3.6) and (3.4). We can choose $f_\tau(n) = f_w(n)^2$ and choose a kinetic coefficient $\beta(n) = 0$, so that

$$\lambda = \frac{I}{K + JF} \frac{2D\tau}{W^2}. \quad (3.15)$$

To evaluate the three derivatives terms dS_x , dS_y and dS_z efficiently, we use the ADOL-C software [22]. ADOL-C is a package for the automatic differentiation of algorithms written in C/C++ by Andreas Griewank [22]. It can evaluate first and higher

derivatives of vector functions defined by computer programs written in C/C++. Functionally, ADOL-C uses the operator overloading concept to compute in forward and reverse mode. ADOL-C provides a convenient way to evaluate arbitrary order derivatives of functions that are given as a C/C++ source code. The user has to make a few modifications to the evaluation code:

- 1) re-declare type of all variables that are involved in the evaluation code (active variables) to the new type *adouble*.
- 2) mark the evaluation section.
- 3) specify independent and dependent variables and pass their values.
- 4) recompile the code and link the ADOL-C library

In Eq. (3.11), dSx , dSy and dSz are derivatives of functions of x , y and z . They can be rewritten as

$$dSx = \frac{\partial}{\partial \cdot x} \{Function1(x, y, z)\} \quad (3.16)$$

$$dSy = \frac{\partial}{\partial \cdot y} \{Function2(x, y, z)\} \quad (3.17)$$

$$dSz = \frac{\partial}{\partial \cdot z} \{Function3(x, y, z)\} \quad (3.18)$$

Because the phase field function φ is going to be fitted by a Gaussian function, which we are going to discuss in section 3.3.1, so that $Function1(x, y, z)$, $Function2(x, y, z)$ and $Function3(x, y, z)$ are three symbolical functions of x , y and z explicitly according to Eq.(3.10).

For example, to solve dSx , we need to define $Function1(x, y, z)$ as a vector function in the ADOL-C code, and evaluate the first derivatives of $Function1(x, y, z)$ respectively. The value of dSx is the first column of the derivatives of $Function1(x, y, z)$. The value of dSy is the second column of the derivatives of $Function2(x, y, z)$. The value of dSz is the third column of the derivatives of $Function3(x, y, z)$.

We need to write a section of ADOL-C code for differentiation, and this section is recompiled and linked with the library. To differentiate the same function in ADOL-C is

much faster than in Mathematica. Another advantage of ADOL-C is that it is compatible with C/C++ code. It is available on SUN workstations on which our project runs.

3.3. Numerical Algorithm

Through the PDE formulation discussed above, we know that we need 2nd derivatives of the phase fields. What is needed is an approximation of the phase field that is smooth enough to compute second derivatives and still capture the interface without excessive oscillations. In order to solve the phase field as an approximation, we define the phase field by a fitting function to fit the actual behavior as well as we can.

The fitting function has to be chosen so that the phase field can be interpolated from nodal values over the element. To interpolate is to approximate the value of a function between known values. For example, the initial phase field variables have known values, a fitting function is actually a formula for phase field variables, first the function has to operate on the known values of phase field. A fitting function should be able to describe the physical problem, such as the phase field problem.

Since the phase field model defines solid phase as +1, liquid phase as -1, the interface is between +1 and -1, so the whole phase field must be in the range of $[-1, 1]$ over the element.

We also considered the finite element basis functions as fitting functions, but they are not complete second order polynomial. Even if they were, oscillations were a problem, because usually they are not restricted to range of $[-1, 1]$ over the element.

3.3.1 Gaussian Function Fitting

Gaussian fitting function is a possible interpolation function to describe the phase field over the element. A Gaussian function is recommended by Dr. E. Hughes and defined as

$$P(X) = \frac{\sum_i (P_i \cdot w(X_i - X))}{\sum_i w(X_i - X)} \quad (3.16)$$

Where $\{X_i\}$ is supposed to be a set of N points in one, two or three spatial dimensions, and $\{P_i\}$ is a corresponding set of scalar data at the points, for example, phase field values, X is the coordinate for a given point, $P(X)$ is the approximate value for point X , for each i , if X_i and X are vectors, the deviation $X_i - X$ is a vector of the same spatial dimension, and $w(X_i - X)$ is a Gaussian weight function. The Gaussian weight function is defined as

$$W(X, \varepsilon) = \exp(-(X \cdot X) / 2 \cdot \varepsilon^2) \quad (3.17)$$

Where, if X is a vector, $X \cdot X$ is interpreted as the dot product, ε is the Gaussian width parameter.

The Gaussian function $P(X)$ describes a weighted moving average of some values, and it has several important properties: If the data P_i lie in the interval $[-1, 1]$, then so will $P(X)$. And $P(X)$ is a weighted average of the P_i values, with the ones near X having the most weight. These properties are the desired properties for the phase field problem. Because the phase field is defined in the range of $[-1, 1]$ at any point over the element, $P(X)$ can guarantee the phase field between two nodes are in the range of $[-1, 1]$. This is crucial for a phase field model. We tested some other fitting functions, such as the quartic polynomial function which can be written as follows,

$$\varphi = c_0 + c_1x + c_2y + c_3z + c_4x^2 + c_5xy + c_6y^2 + c_7xz + c_8yz + c_9z^2 + \\ c_{10}x^3 + c_{11}x^2y + \dots + c_{32}xz^3 + c_{33}yz^3 + c_{34}z^4$$

the problem from this kind of function is that it need not be in $[-1, 1]$. Because of this, it has heavy oscillations if it is applied to the phase field model. However, the Gaussian function is able to overcome this problem very beautifully.

Furthermore, choosing the fitting function is related to the derivatives of the function φ . In order to understand this, we introduce the following symbolism to define the degree of continuity of a function or a field. A field φ is said to have C^m continuity if all derivatives of the field through order m are continuous. Thus $\varphi = \varphi(x)$ is C^0 continuous if φ is continuous but φ_x is not. In general, it is necessary that derivatives of φ of order m be used as nodal d.o.f. if the field φ produced by a mesh of finite elements is to be C^m continuous.

Now we come back to our Eq (3.10). It has the source terms, the second derivatives of the phase field. These require phase field fitting function to be C^2 continuity or even higher order. The Gaussian fitting function satisfies this.

The Gaussian fitting function method is not an interpolation, it is a smoothing approximation instead, which is what we want for this model.

We are going to use the same N nodes to evaluate the phase field at any specific Gauss point X . So, we can see P_i is the phase field at node N_i in Eq (3.16), X_i is the coordinate for node N_i , $P(X)$ is the phase field at gauss point X or any other point X .

Let us develop step by step how we can apply the Gaussian function to our phase field model.

- (1) Find phase fields for some given points. Let us call $\varphi(i)$ the phase field at point i . These points must have some relationship with one specific element. For example, the element is symmetrically surrounded by these points.
- (2) Use $\varphi(i)$ to interpolate the phase field at Gauss point in the element $\varphi(i_gp)$ by applying the Gaussian function. We could think of $\varphi(i)$ as P_i , and $\varphi(i_gp)$ as $P(X)$.
- (3) use $\varphi(i_gp)$ to get the source value at Gauss points, let us call $gpSource(i_gp)$ the source term at Gauss point i .
- (4) use $gpSource(i_gp)$ to get the source vector over one element.

Now a few questions are raised. Which set of points in step (1) are we going to choose for the Gaussian fitting function? How are they related to a 3D phase field element?

We have to choose some amount of nodes to do the weighting function. In fact, it is quite common, when analyzing data, that we cannot solve the value exactly to the true value and some errors will exist. We need to minimize the errors. It is common sense, if we have fitted more nodes, then, the real value should be more accurately solved.

We are going to choose a 6_Nodes_Brick as our 3D element for phase field analysis. There are six nodes, one node in the center of each face of the 6_Nodes_Brick. It is shown in Fig. (3.1).

Fig.(3.1) phase field element, 6_Nodes_Brick

The 6_Nodes_Brick has fourteen Gauss points inside. They are symmetrically distributed around the center of the 6_Nodes_Brick. To evaluate the phase field at each Gauss point, some points distributed symmetrically around the center of the element need to be chosen.

Based on these reasons, we choose eighty-four nodes for phase field Gaussian fitting analysis. These eighty-four nodes consist of fourteen 6_Nodes_Bricks, eight on the vertex of the central 6_Nodes_Brick (the central one is the element we are going to use), six adjacent to six faces of the central 6_Nodes_Brick.

Since the mesh has a uniform point-density, and ε is fixed, each evaluation of $P(X)$ will involve about the same number of points, and so takes a constant time independent of the total number of points in the mesh. The same is true of the gradient.

3.3.2. Phase Field Solver

After the Gaussian function fitting, the phase field φ is an explicit function of (x, y, z) . dS_x , dS_y and dS_z are also explicit functions of (x, y, z) . Then, for every Gauss point in the element, we could compute the heat source Q according to Eq (3.12) and (3.11) by using ADOL-C. The other two terms, specific heat C_p , and conductivity k are determined by material properties d_0 , D , and the parameters we are going to choose, such as W , λ , and τ . Let us rewrite the phase field equation in the form of the energy equation.

$$C_p \dot{\varphi} + \nabla \cdot (-k \nabla \varphi) + Q = 0 \quad (3.18)$$

Within a 6_Nodes_Brick 3D element, the phase field φ is interpolated from nodal values $\{\varphi_e\} = [\varphi_1, \varphi_2, \dots, \varphi_6]^T$,

$$\varphi = [N] \{\varphi_e\} \quad \text{or} \quad \varphi = \sum N_i \varphi_i \quad (3.19)$$

Differentiation of Eq. (3.19) wrt to time yields

$$\dot{\varphi} = [N] \{\dot{\varphi}_e\} \quad (3.20)$$

To develop the phase field solver formula, we need to introduce the potential energy. As we know, the finite element solution for the energy equation is derived from the potential energy functional Π , and for a single element Π is as follows,

$$\Pi = \frac{1}{2} \{T_e\}^T [k] \{T_e\} + \{T_e\}^T [C_p] \{\dot{T}_e\} - \{T_e\}^T \{Q\} \quad (3.21)$$

Finite element equations are obtained by making Π stationary with respect to variations of nodal temperature,

$$\left\{ \frac{\partial \Pi}{\partial T_e} \right\} = \{0\} \quad \text{yields} \quad [k]\{T_e\} + [C_p]\{\dot{T}_e\} = \{Q\} \quad (3.22)$$

applying to our phase field equation, it is the same solution.

$$\left\{ \frac{\partial \Pi}{\partial \varphi_e} \right\} = \{0\} \quad \text{yields} \quad [k]\{\varphi_e\} + [C_p]\{\dot{\varphi}_e\} = \{Q\} \quad (3.23)$$

Upon assembly of elements, $\{\varphi_e\}$ is replaced by the global vector $\{\varphi\}$, which contains all nodal phase field of the structure. The global equations are therefore

$$(\sum [k])\{\varphi\} + (\sum [C_p])\{\dot{\varphi}\} = \sum \{Q\} \quad (3.24)$$

In integral form, Eq. (3.24) can be written as

$$\int k \nabla \varphi \, d\Omega + \int C_p \dot{\varphi} \, d\Omega = \int Q \, d\Omega \quad (3.25)$$

Substitution in Eq.(3.19) into Eq.(3.20), and applying the shape functions N of a 6_Nodes_Brick, the formula becomes

$$\int \nabla N^T k \nabla N d\Omega \cdot \{\varphi_e\} + \frac{d}{dt} \int N^T C_p N d\Omega \cdot \{\varphi_e\} = \int N^T Q d\Omega \quad (3.26)$$

In Eq.(3.26), the integral term $\int \nabla N^T k \nabla N d\Omega$ is called the element stiffness matrix K , and $\int N^T C_p N d\Omega$ is called the G matrix, $\int N^T Q d\Omega$ is the source term S . So our FEM phase field formula is

$$[G + dt * K] \cdot \varphi^3 = dt * S + G * \varphi^1 \quad (3.27)$$

Where,

K , is element volume stiffness matrix, conductivity involved, unit m^3

G , is element volume G matrix, specific heat involved, unit $m^3 \cdot s$

S , is vector of nodal source values, unit m^3 .

It is the projection of Karma's rhs onto the basis functions.

φ^1 , is the vector of nodal values of phase field $[-1, +1]$ at beginning of time step.

φ^3 , is the vector of nodal values of phase field $[-1, +1]$ at end of time step.

φ^1 and φ^3 are dimensionless, has no unit.

From Eq (3.11) and Eq (3.12), we know that the term $\tau(n)$ is the analog of specific heat, $2W(n)^2$ is the analog of conductivity, $(1 - \varphi^2)[\varphi - \lambda u(1 - \varphi^2)] + dSx + dSy + dSz$ is the analog of source term. $\tau(n)$ has units of second, and $2W(n)^2$ has units of m^2 . So G matrix has units of $m^3 \cdot s$, K matrix has units of m^3 and Q is dimensionless. Let us call the left side $[G + dt * K]$ of Eq (3.27) as VolumeStiffness, the right side $dt * S + G * \varphi^1$ of Eq (3.27) as VolumeResidual, they both have units of $m^3 \cdot s$, so that we could write in the form of linear solver $kx = b$ formulation,

$$\text{VolumeStiffness} \bullet \text{nodalPhaseField} = \text{VolumeResidual}$$

It is easy to see that once we evaluate the VolumeStiffness and VolumeResidual, we will be ready to solve nodalPhaseField for the next step. It means that from given initial conditions, we can acquire the next step condition. The temperature solver formula for the energy equation has a similar form to Eq.(3.28) if we need it in the future,

$$[G + dt * K] T^3 = dt * S + G * T^1 \quad (3.28)$$

Now we can develop a flow diagram for the finite element algorithm, see Fig. (3.2).

The computation program is written in C++. Computations have two major parts: setting the initial conditions and moving the solid/liquid interface. Obviously, the first is executed only once, while the second is iterated until the preset termination conditions are met for each time step.

One thing to be clarified for this flow diagram is that if the energy equation is coupled with the phase field equation, then we need to solve formula (3.28). In this thesis, we skip this step.

Fig.(3.2) Flow diagram of the adaptive algorithm

Chapter 4

Results of Numerical Studies

We implemented a three dimensional version of the numerical algorithm, using the Gaussian fitting function to fit the phase field explicitly, using 6_Node_Brick finite elements for phase field distribution analysis. Spherical and planar growth morphologies were shown. But because of time constraints, we only solved the phase field equation without coupling the energy equation.

As discussed in Chapter 3, we need to select appropriate parameters to satisfy the criteria of the phase field model when we develop the PDE formulation. Selections of parameters is important to a phase field model.

4.1. Selection of Parameters

There are two types of parameters for a phase field analysis. One type is material properties parameters and the other is artificial parameters.

First, the material properties parameters are determined by the type of material. What material properties parameters are we going to use in this model? As we know, the material properties parameters for pure aluminum used in this model are the following data.

Table (4.1) Properties of the Casting Material, pure Al

Capillary length , d_0	$2.4 \cdot 10^{-10} \text{ m}$
Thermal diffusivity, D	$37 \cdot 10^{-6} \text{ m}^2 / \text{s}$
Latent Heat, L	$9.5 \cdot 10^8 \text{ J} / \text{m}^3$
Melting Temperature, T_m	933.6 k
Specific Heat, C_p	$2.79 \cdot 10^6 \text{ J} / \text{m}^3 \text{K}$

These values are chosen from the Appendix of Fundamentals of Solidification by Fisher and Kurz [1].

The other parameters, such as $\beta(n)$, $\tau(n)$, λ and $W(n)$ and time step are artificial parameters, they are chosen by considering the above material properties parameters, and also they have to satisfy the constraints of this phase field model. Let us discuss how to determine them now.

We choose kinetic coefficient $\beta(n) = 0$, because without kinetic undercooling, the moving interface may dissipate faster the energy generated across the interface region to the surrounding area. We want to simulate the important physical limit where the kinetic undercooling $\beta(n)v_n$ is negligibly small compared to the curvature undercooling $d_0 k$ in Eq (2.3).

Interface thickness W_0 is a crucial parameter for Karma's phase field model. We choose it to adjust the interface thickness to the length scale of the dendrite tip radius.

The curvature or Gibbs-Thomson undercooling is defined in Fisher and Kurz's Fundamentals of Solidification [1],

$$\Delta T_r = K \cdot \Gamma \quad (4.1)$$

where,

ΔT_r is the difference (undercooling) between the melting point and the temperature of the melt, $\Delta T_r = T_m - T$,

Γ is the Gibbs-Thomson coefficient,

K is the curvature of interface.

From Eq. (4.1), we can see that if undercooling is small, then the curvature is small, which implies the tip radius is big. So the relationship between undercooling and tip radius is: smaller the undercooling, bigger the tip radius (flat in the limit).

If an element has the same size as the tip radius, for a small size element, the ratio of the tip radius to the element size is big. It needs more cells or more elements to describe the tip. This is easy to understand through Fig.(4.1)a and Fig.(4.1)b. From the computation viewpoint, more elements will reduce efficiency. And if extended to three dimensional computation, it is often not feasible to apply very small size elements. Alternatively, we could enlarge the element size to a practical scale to make the computation feasible. In Karma's model, the ratio of tip radius to interface thickness is adjusted to a smaller number. This, in turn, renders quantitative three-dimensional simulations directly feasible.

Fig. (4.1)a big tip radius to element size ratio

Fig. (4.1)b small tip radius to element size ratio

Parameter τ is similar to a specific heat here in phase field equation. It is not real specific heat parameter. We can choose it according to different cases. From the definition of specific heat of one material, τ actually tells us how fast the diffusions take place. It is a time scale, so it depends on the time step we are going to choose. And it also depends on the undercooling. When the undercooling decreases, power for the interface moving is small, so the interface moves slowly, the heat source generated around interface will increase, then it is necessary to dissipate the heat quickly in order to stabilize the liquid to solid phase transformation. It means, when undercooling decreases, τ will increase. How fast will it increase? This has to meet the restrictions.

Once we have chosen W_0 and, then we could apply Eq. (3.16) to get the parameter $\lambda \sim d_0 / W_0$. Let us start with undercooling $\Delta = -0.39$. When we start to decrease Δ , this makes W_0 increasing gradually from the ratio of $d_0 / W_0 = 0.139$ to 0.130. Using other constraints, we can adjust the λ and τ . From constraint No.3, we find that the product of $(\lambda \cdot u)$ cannot be big, which is related to source distribution. Then we could double check λ and W_0 . This has to be tested repeatedly before we know the right choice of W .

The time step is also a very important parameter to choose. We have to make sure the interface moves at a good velocity. How far can the interface go in a certain time? We may choose from the interface velocity approximated by the kinetic equation $\text{Velocity} = \text{Kinetic Coefficient} \cdot (dT)$. Referring to the article “ Cellular Simulation of the Dendrite Growth in Al-Si Alloys ” by Professor A. Artemev and J. A. Goldak [23], we choose the kinetic coefficient $k = 0.1 \text{ (m/s K)}$. Through the tests, we find that it is good if we choose the time step so that the fastest part of the solidification front travels a distance of almost one element in one time step. Just by simple computation, we will know, for example, when $dT = -1 \text{ K}$, velocities = 0.01 m/s if we want the interface move one element size in one time step. If the velocity is 100 times bigger, then the velocity-dependent kinetics undercooling $\beta(n) v_n$ will not be negligibly small compared to the curvature undercooling $d_0 k$ in Eq (2.3). Then we are not able to solve the important physical limit.

4.2. Three Dimension Results

In order to see the geometry results of phase field interface moving, we are going to run different test cases with different initial interface geometry conditions such as sphere and plane. The design for the casting is the same for different test cases. The design data information is as follows:

Mesh Data:

Number of Elements: 20x20x20

Mesh Size: $8.000 \cdot 10^{-6} \text{ m}$

Casting Parts Information:

Material: AlA380

Initial Temperature: 1100 *K*

Density of the Cast Metal: 2660 *kg/m*³

Simulation Parameters:

Number of Time Step: 10

In the following test cases, because of the time constraint, we only solve the phase field equation using FEM Phase Field Solver formula. The nodal temperature is constant for all nodes and all time. The energy equation is not coupled, which means because the temperature is constant in the phase field, dendrites would not be generated. For example, in the isothermal situation, a sphere could melt, freeze or remain steady, but a dendrite is unable to grow. The following test cases are all isothermal solidification of a pure liquid on an existing solid. Nucleation does not occur.

4.2.1 Test Case No. 1: Plane Geometry

This is a simple test case. The interface between solid and liquid is designed as a plane in the middle of mesh. One half of the mesh is the solid phase field, the other half of the mesh is the liquid phase field. We would like to determine when the plane interface freezes or melts. The plane interface is a perfect plane without any perturbation on it. The reason to choose the perfect plane is that we are not going to solve the energy equation. Dendrites are not expected to grow without solving the energy equation. Then there is no need to put a perturbation on the solid / liquid interface.

The force that drives the interface to freeze or melt is thermal undercooling. There is no curvature undercooling because of the flat interface. If temperature $T < T_m$, then the interface should freeze. If $T > T_m$, it should melt. If $T = T_m$, it should not move.

When undercooling is different, the solidification situation is different, and the interface motion is different. The interface speed is an important parameter to represent the moving interface. To calculate the velocity, we need to know the distance the interface moves in a time period.

The change in volume on melting or freezing is one of the important experimental considerations in solidification [24]. It is possible to calculate the moving distance accurately by knowing the volume of the solid phase field in each time step. The volume of solid phase is the sum of solid phase field value fraction in each cell multiplied by the volume of one element.

$$V_s^t = \left(\sum_{i=1}^N \frac{\phi_i + 1}{2} \right) \cdot (dx \cdot dy \cdot dz) \quad (4.2)$$

where,

V_s^t , is the solid phase volume at time step t ,

$\sum_{i=1}^N \frac{\phi_i + 1}{2}$, is the sum of solid phase field value summed overall all cells,

N is the total number of cells in the element,

dx , dy and dz is the size of element in x , y and z coordinates respectively.

The output of the Phase Field Solver is the nodal phase field ϕ at one time step. The phase field value is between $[-1,1]$, by distinguishing this, $\sum_{i=1}^N \frac{\phi_i + 1}{2}$ is easy to get from the nodal phase field value. Once we have the solid phase volume (V_s^t) at time step t , and (V_s^{t+1}) at time step $t+1$, the distance the interface moves is obvious. In case of a planar interface, the distance the interface moves is determined by the change in solid phase volume on melting or freezing, see Equation (4.3).

$$V_s^{t+1} - V_s^t = L_t^{t+1} * S \quad (4.3)$$

where,

V_s^t , is the solid phase volume at time step (t),

V_s^{t+1} , is the solid phase volume at time step ($t+1$),

L_t^{t+1} , is the distance the interface moves from time step (t) to time step ($t+1$),

S , is the area of planar interface.

The interface speed is determined by L_t^{t+1} and the simulation time at each time step.

Until now, we have discussed choosing interface thickness, time step, etc. Using Karma's phase field model, we could test small undercooling for pure Al, such as $\Delta = 0.001$, which means $dT \approx -0.35$ K is a small amount. The simulation time during one time step is $3.2 \cdot 10^{-6}$ s).

For all the tests, 10 total time steps have been run. It is necessary to point out that in some test cases, the velocity went to zero when all the solid melted entirely before step 10. At this time there is actually no solid/liquid interface moving any more. So that the zero velocity does not mean a steady state between freezing and melting.

In the following tables, the solid volumes are the output values from the C++ programming software, interface moving distances and velocities are the calculated values from solid volumes. The test cases for a plane are undercooling 0.001, -0.001, 0.0001, and -0.0001.

Table (4.2): Undercooling = 0.001, Test No. 1

TimeStep	SolidVolume(m^3)	MovingDistance(m)	Velocity(m/s)
1	3.42E-16	5.35E-06	
2	3.39E-16	5.3E-06	-0.01708
3	3.82E-16	5.97E-06	0.211865
4	4.24E-16	6.62E-06	0.201172
5	4.65E-16	7.27E-06	0.204468
6	5.07E-16	7.92E-06	0.203916
7	5.48E-16	8.56E-06	0.199702
8	5.90E-16	9.21E-06	0.203643
9	6.40E-16	1E-05	0.244502
10	7.10E-16	1.11E-05	0.340991

Fig (4.2) Curve Distance vs. Time and Curve Velocity vs. Time
for Plane when undercooling =0.001

Table (4.3): Undercooling = -0.001, Test No. 2

TimeStep	SolidVolume(m^3)	MovingDistance(m)	Velocity(m/s)
1	3.42E-16	5.35E-06	
2	2.35E-16	3.67E-06	-0.52399
3	1.37E-16	2.15E-06	-0.47724
4	5.14E-17	8.03E-07	-0.4197
5	5.44E-18	8.49E-08	-0.22453
6	0	0	-0.02655
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0

Fig (4.3) Curve Distance vs. Time and Curve Velocity vs. Time
for Plane when undercooling = -0.001

(Note: velocities become zero, when solid melt completely)

Table (4.4): Undercooling = 0.0001, Test No. 3

TimeStep	SolidVolume(m^3)	MovingDistance(m)	Velocity(m/s)
1	3.46E-16	4.35E-06	
2	3.56E-16	4.4E-06	0.012885
3	3.54E-16	4.39E-06	-0.00207
4	3.60E-16	4.42E-06	0.008016
5	3.64E-16	4.43E-06	0.004651
6	3.69E-16	4.45E-06	0.006704
7	3.73E-16	4.47E-06	0.004947
8	3.78E-16	4.49E-06	0.0061
9	3.82E-16	4.5E-06	0.005266
10	3.87E-16	4.52E-06	0.005451

Fig (4.4) Curve Distance vs. Time and Curve Velocity vs. Time
for Plane when undercooling = 0.0001

Table (4.5): Undercooling = -0.0001, Test No. 4

TimeStep	SolidVolume(m^3)	MovingDistance(m)	Velocity(m/s)
1	3.25E-16	4.26E-06	
2	3.22E-16	4.25E-06	-0.00381
3	3.11E-16	4.2E-06	-0.01525
4	3.05E-16	4.18E-06	-0.00832
5	2.97E-16	4.14E-06	-0.01209
6	2.91E-16	4.11E-06	-0.00858
7	2.84E-16	4.08E-06	-0.01052
8	2.77E-16	4.05E-06	-0.01009
9	2.70E-16	4.01E-06	-0.0111
10	2.63E-16	3.98E-06	-0.01059

Fig (4.5) Curve Distance vs. Time and Curve Velocity vs. Time
for Plane when undercooling = -0.0001

4.2.2 Test Case No. 2: Sphere Geometry

In this test case, a solid sphere seed is immersed in an undercooled melt. The seed is placed at the center of the mesh.

Whether the sphere stays steady, melts or freezes depend on the undercooling, surface energy and sphere radius. According to Fisher and Kurz's, Fundamentals of

Solidification [1], the Gibbs-Thomson undercooling for a sphere is given by Equations (4.4) and (4.5),

$$\Delta T_r = K \cdot \Gamma = \frac{2\Gamma}{r^0} \quad (4.4)$$

$$r^0 = \frac{2\Gamma}{\Delta T_r} \quad (4.5)$$

where,

r^0 is the critical radius for nucleation,

Γ is the Gibbs-Thomson coefficient,

ΔT_r is the difference (undercooling) between the melting point and the temperature of the melt, $\Delta T_r = T_m - T$,

K is the curvature of interface.

The Gibbs-Thomson condition can also be written in dimensionless form as Equation (4.6)

$$u = -d_0 k \quad (4.6)$$

where,

u is the dimensionless temperature defined by $\frac{T - T_m}{L / C_p}$,

d_0 is the capillary length defined by $\gamma T_m C_p / L^2$,

k is the curvature of interface.

Let us consider the spherical case. Let the radius of the sphere be R and the undercooling at the infinite place be Δ ; so that $u(\infty) = -\Delta$. It is well known that the curvature of sphere is $k = 2 / R$, so the critical radius could be written as

$$R^* = \frac{2d_0}{\Delta} \quad (4.7)$$

When the sphere radius is equal to R^* , the sphere should stay steady. When it is larger than R^* , the solid sphere should grow. We call this situation as freezing. When it is smaller than R^* , the sphere should melt.

For a specific material, d_0 is a constant because it is a material property. We could adjust R and also Δ to do the tests according to the relationship $R^* \Delta = 2d_0 = \text{constant}$.

In different solidification situations for a sphere, the interface speed is acquired through the similar relationship as plane interface. The solid volume difference from time step (t) to time step (t+1) is as follows,

$$V_s^{t+1} - V_s^t = \frac{4}{3} \pi \cdot R_{t+1}^3 - \frac{4}{3} \pi \cdot R_t^3 \quad (4.8)$$

where,

V_s^t , is the solid phase volume at time step (t),

V_s^{t+1} , is the solid phase volume at time step (t+1),

R_{t+1} is the radius of sphere at time step (t + 1),

R_t is the radius of sphere at time step (t).

Through this relationship, we could compute the radius of next time step. We know the initial sphere radius. So for every time step, the radius is known. Obviously, the difference between the radius of time step (t+1) and time step (t) is the distance the interface moves.

For pure Al, the following tables list all the test results for a sphere under different solidification conditions.

First, we keep the radius constant by changing the undercooling. Then, we keep the same undercooling, and change the radius.

Table (4.6): Undercooling = 0.001, Radius = $6.4 \cdot 10^{-7} m$, Test No. 5

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	3.79E-19	4.49E-07	
2	4.79E-19	4.86E-07	0.011464
3	7.48E-19	5.63E-07	0.024311
4	1.35E-18	6.85E-07	0.038175
5	2.69E-18	8.63E-07	0.05556
6	5.26E-18	1.08E-06	0.067325
7	9.57E-18	1.32E-06	0.074611
8	1.68E-17	1.59E-06	0.084606
9	2.82E-17	1.89E-06	0.094067
10	4.52E-17	2.21E-06	0.1

Fig (4.6) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.001, Radius=6.4e-7m

Table (4.7): Undercooling = 0.00095, Radius = $6.4 \cdot 10^{-7}m$, Test No. 6

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	3.33E-19	4.30E-07	
2	3.28E-19	4.28E-07	-0.00057
3	4.26E-19	4.67E-07	0.0121
4	6.13E-19	5.27E-07	0.018856
5	1.01E-18	6.23E-07	0.030119
6	1.83E-18	7.59E-07	0.04223
7	3.43E-18	9.36E-07	0.055462
8	6.30E-18	1.15E-06	0.065585
9	1.10E-17	1.38E-06	0.07342
10	1.85E-17	1.64E-06	0.081

Fig (4.7) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00095, Radius=6.4e-7m

Table (4.8): Undercooling = 0.0009, Radius = $6.4 \cdot 10^{-7} m$, Test No. 7

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	2.87E-19	4.09E-07	
2	2.09E-19	3.68E-07	-0.01275
3	1.84E-19	3.53E-07	-0.00474
4	1.63E-19	3.39E-07	-0.00438
5	1.40E-19	3.22E-07	-0.00535
6	1.06E-19	2.93E-07	-0.00895
7	5.27E-20	2.33E-07	-0.01897
8	2.67E-20	1.85E-07	-0.01475
9	0	0	-0.05781
10	0	0	0

Fig (4.8) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.0009, Radius=6.4e-7m

Table (4.9): Undercooling = 0.0008, Radius = 6.4 ($\cdot 10^{-7} m$), Test No. 8

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	1.98E-19	3.62E-07	-0.15867
2	5.42E-20	2.35E-07	-0.0397
3	6.40E-21	1.15E-07	-0.03735
4	0	0	-0.03601
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0

Fig (4.9) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.0008, Radius=6.4e-7m

Table (4.10): Undercooling = 0.0005, Radius = $6.4 \cdot 10^{-7} m$, Test No. 9

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	4.57E-20	2.22E-07	
2	0	0	-0.06934
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0

Fig (4.10) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.0005, Radius=6.4e-7m

Table (4.11): Undercooling = 0.001, Radius = $5.5 \cdot 10^{-7}m$, Test No. 10

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	1.51E-19	3.31E-07	
2	6.93E-20	2.55E-07	-0.02365
3	3.70E-20	2.07E-07	-0.01503
4	1.99E-20	1.68E-07	-0.01208
5	0	0	-0.05253
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0

Fig (4.11) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.0005, Radius =5.5e-7m

Table (4.12): Undercooling = 0.001, Radius = $6.0 \cdot 10^{-7} m$, Test No.11

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	3.79E-19	4.49E-07	
2	4.79E-19	4.86E-07	0.011464
3	7.48E-19	5.63E-07	0.024311
4	1.35E-18	6.85E-07	0.038175
5	2.69E-18	8.63E-07	0.05556
6	5.26E-18	1.08E-06	0.067325
7	9.57E-18	1.32E-06	0.074611
8	1.68E-17	1.59E-06	0.084606
9	2.82E-17	1.89E-06	0.094067
10	4.52E-17	2.21E-06	0.1

Fig (4.12) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.001, Radius =6.0e-7m

Table (4.13): Undercooling = 0.00091, Radius = $6.4 \cdot 10^{-7} m$, Test No.12

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	2.96E-19	4.13E-07	
2	2.25E-19	3.77E-07	-0.0113163
3	2.15E-19	3.72E-07	-0.0016932
4	2.18E-19	3.73E-07	0.0005407
5	2.33E-19	3.82E-07	0.0025718
6	2.79E-19	4.05E-07	0.007346
7	3.65E-19	4.43E-07	0.0118776
8	4.95E-19	4.91E-07	0.0149172
9	7.25E-19	5.57E-07	0.0207823
10	1.12E-18	6.44E-07	0.026915

Fig (4.13) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00091, Radius =6.4e-7m

Table (4.14): Undercooling = 0.00092, Radius = $6.4 \cdot 10^{-7} m$, Test No.13

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	3.05E-19	4.18E-07	
2	2.40E-19	3.86E-07	-0.009984
3	2.47E-19	3.89E-07	0.0011578
4	3.02E-19	4.16E-07	0.0083496
5	4.04E-19	4.59E-07	0.0133096
6	5.64E-19	5.13E-07	0.0168803
7	8.79E-19	5.94E-07	0.02553
8	1.47E-18	7.06E-07	0.0349096
9	2.63E-18	8.57E-07	0.0471079
10	4.71E-18	1.04E-06	0.0572247

Undercooling= 0.00092, Radius= $6.4e-7m$, Sphere

Fig (4.14) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00092, Radius =6.4e-7m

Table (4.15): Undercooling = 0.00093, Radius = 6.4 ($\cdot 10^{-7}m$), Test No.14

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	3.14E-19	4.22E-07	
2	2.68E-19	4E-07	-0.0067817
3	3.07E-19	4.18E-07	0.005716
4	4.07E-19	4.6E-07	0.0129756
5	5.76E-19	5.16E-07	0.0176545
6	9.17E-19	6.03E-07	0.0269745
7	1.58E-18	7.22E-07	0.0373891
8	2.88E-18	8.82E-07	0.0500371
9	5.20E-18	1.08E-06	0.0602393
10	8.98E-18	1.29E-06	0.067016

Fig (4.15) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00093, Radius =6.4e-7m

Table (4.16): Undercooling = 0.00094, Radius = $6.4 \cdot 10^{-7} \text{ m}$, Test No.15

TimeStep	SolidVolume(m^3)	Radius(m)	Velocity(m/s)
1	3.23E-19	4.26E-07	
2	2.98E-19	4.15E-07	-0.0035429
3	3.66E-19	4.44E-07	0.0091908
4	5.11E-19	4.96E-07	0.0163
5	7.94E-19	5.75E-07	0.0245394
6	1.34E-18	6.84E-07	0.0340819
7	2.47E-18	8.39E-07	0.0485111
8	4.51E-18	1.03E-06	0.0582487
9	7.92E-18	1.24E-06	0.0660946
10	1.38E-17	1.49E-06	0.0783524

Fig (4.16) Curve Distance vs. Time and Curve Velocity vs. Time
for Sphere when undercooling = 0.00094, Radius = 6.4e-7m

4.3. Discussion of the Results of Test Cases

From the test cases of a plane, it is shown that when a casting part is a supercooled, the liquid phase will gradually solidify. However, when there is superheating, the solid phase field will melt. By comparing Test No.(1), (2) and (3), (4) for a plane, we are able to see the freezing with supercooling, and melting with superheating, freezing or melting quickly with high undercooling, and freezing or melting slowly with low undercooling.

Table (4.17) : Test results for plane from supercooling to superheating

Test No.	Undercooling Value	Undercooling Type	Result
(1)	0.001	supercooling	Freeze quickly
(2)	-0.001	superheating	Melt quickly
(3)	0.0001	supercooling	Freeze slowly
(4)	-0.0001	superheating	Melt slowly

All the test cases of a sphere that we have done indicate that different undercooling or radius will have a different solidification curve. The speed of the solid and liquid interface is determined by the undercooling and radius. Through Equation (4.7), the sphere will have same critical radius with same undercooling. When the initial sphere radius is larger than R^* , the sphere freezes. When the initial sphere radius is smaller than R^* , the sphere should melt. By comparing Test No. 5 and Test No. 10 and 11, we know that when the radius decreases, the sphere will change from freezing to melting.

Table (4.18): Test results for sphere when radius decreases

Test No.	Undercooling Value	Radius ($\cdot 10^{-7}$ m)	Result
(5)	0.001	6.4	Freeze
(11)	0.001	6.0	Freeze
(10)	0.001	5.5	Melt

When the radius of the sphere is kept constant, the test results for different undercooling are shown below in Table (4.19) and Table (4.20) .

Table (4.19): Test results for sphere when undercooling decreases

Test No.	Radius ($\cdot 10^{-7}$ m)	Undercooling	Result
(5)	6.4	0.001	Freeze Quicker
(6)	6.4	0.00095	Freeze
(7)	6.4	0.0009	Melt completely at step 9
(8)	6.4	0.0008	Melt completely at step 4
(9)	6.4	0.0005	Melt completely at step 2

Through Table (4.19) , we know that when undercooling is large, the sphere freezes, and when undercooling become smaller and smaller, the sphere changes from freezing to melting. By comparing Tests No. (7), (8) and (9), we are able to see the sphere melts more quickly when undercooling decreases. The reason is because of the relationship

$R^* \Delta = 2d_0 = \text{constant}$. When the radius is the same, if undercooling decreases, the critical radius increases, so that when the current radius is smaller than the critical radius, the sphere should melt.

Table (4.20): Test results for sphere when undercooling increases a little

Test No.	Radius ($\cdot 10^{-7}$ m)	Undercooling	Average Velocity (m/s)
(7)	6.4	0.0009	-0.01419
(12)	6.4	0.00091	0.007993
(13)	6.4	0.00092	0.021609
(14)	6.4	0.00093	0.030136
(15)	6.4	0.00094	0.036864
(6)	6.4	0.00095	0.042016

Through Table (4.20), we can see that the average interface moving speed for a sphere increases a little when undercooling increases from 0.0009 to 0.00095. We can also find that when undercooling changes from 0.0009 to 0.00091 for a sphere, the velocity changes from negative to positive, and the sphere changes from melt to freeze. It tells us the steady state must exist when undercooling is between 0.0009 and 0.00091. Through the relationship of $R^* \Delta = 2d_0$, we could get the critical radius R^* is between $5.27\text{E-}07 \sim 5.33\text{E-}07$ (m).

Another issue discussed is why the sphere shrinks first and then grows in some test cases. In Test No.5, the initial radius is $6.4 (\cdot 10^{-7}\text{m})$, but the radius becomes $4.49 (\cdot 10^{-7}\text{m})$ at the first step. How do we define the sphere geometry as solid (+1) in the center of the mesh? We use the constraint $(x^2 + y^2 + z^2) \leq R^2$ to define (x, y, z) be inside the sphere and on the sphere. The initial value is either +1 or -1 as value of phase field. In fact, our element is a cube, initially defined solid (+1) or liquid (-1) would fill in the cube. see Fig. (4.12), it would not be a true sphere. After one time step, the FEM Phase Field Solver acquires the true nodal phase field value. Because these values are computed values from C++ software programming, they are between [-1, +1]. The values

are not always equal to -1 or +1, such as 0.8. Then the point would not lie on the sphere. So that the sphere will shrink a little bit. The radius after the first time step is a more accurate measure of the true radius.

Fig (4.17) Initially defined sphere as solid (+1) in the center of each cube.

Fig.(4.18) After one step the sphere shrinks a little bit

From the test cases presented, we know that the three dimensional phase field analysis of a solidification process is quite feasible. We have seen the test cases of simple

shapes of planes and spheres, which are constantly changing under certain growth conditions.

One important solidification problem deals with the stability of simple growth form. We will see that the simple shapes such as planes, spheres, etc., can be unstable under certain growth conditions. The latent heat generated at the interface must be conducted through the liquid in order for the crystal to grow. If the solid seed has a large surface, the latent heat is dissipated more rapidly in such configurations [25]. The interface can break up into dendrites that grow relatively rapidly out from the central seed. The instability occurs under such configurations. Langer [25] mentioned in the instability analysis for sphere growing, when sphere radius R is close to $7 R^*$, the sphere growing becomes instable. He also pointed out that when we deal with the dendrite problem, planar or spherical cases are very important cases to be started with.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Up to now, we have chosen Karma's phase field model and developed the model for the research on the analysis of solidification. We have seen that a three dimensional simulation has been accomplished for the phase field model. Test results for planar and spherical cases have been presented.

By using an FE method, we are able to solve the PDE phase field equation and energy equation. Using the developed Phase Field Solver formula and Gaussian fitting function for the phase field equation, the nodal phase field value is computed in each time step accurately.

The selections of appropriate parameters in the phase field model make it possible to satisfy the criteria of the model, and the real problems can be run for numerical studies.

The software developed provides accurate estimated results for a 20x20x20 mesh and 6_Node_Brick element solidification simulation. The test results of plane and sphere geometry illustrate that the three dimensional simulation for growth of solidification is quite feasible.

5.2 Future Work

This research could be extended, so that the research on solidification is more comprehensive and practical. The following additional works are recommended for the future.

- (1) The result of a phase field model should be extended to dendrite growth, which requires the energy equation to be coupled in three dimension geometry. This could analyze non-isothermal solidification. The dendrite should be generated from a sphere and a plane.
- (2) For the case of a plane, put a perturbation to see dendrite growth. For example, we could put a solid element on the plane interface in the center.
- (3) It should permit more complicated geometry such as paraboloid geometry. Paraboloid geometry is also a very interesting test to do. A sphere is an isotropic seed, but paraboloid is non-isotropic seed.
- (4) Test for alloy not only a pure liquid. The difference is the properties of materials. The model and algorithm have similar ideas, except for alloy, not only the PDE equations for energy and phase field are needed, but also the equation of solute diffusion is needed.

References

- [1] W. Kurz and D. J. Fisher, *Fundamentals of Solidification*, third edition, Trans. Tech. Publications 1989.
- [2] R. Trivedi and W. Kurz, *Solidification microstructures: a conceptual approach*, *Overview No. 110*, Acta Metall. Mater. Vol. 42, No. 1, pp. 15-23 (1994).
- [3] R. Kobayashi, in *Modeling and numerical simulations of dendritic crystal growth*, Physica D **63**, pp. 410-423 (1993).
- [4] A. A. Wheeler, B. T. Murray and R. J. Scharfe, *Computation of dendrites using a phase field model*, Physica D **66**, pp. 243-262 (1993).
- [5] G. B. McFadden, A. A. Wheeler, R. J. Braun, and S. R. Coriell, *Phase-field models for anisotropic interfaces*, Physical Review E, Vol. 48, No. 3, pp. 2016-2024 (September 1993).
- [6] J. A. Warren and W. J. Boettinger, *Prediction of dendritic growth and microsegregation patterns in a binary alloy using the phase-field method*, Acta Metall. Mater. Vol 43, No. 2, pp. 689-703(1995).
- [7] A. Karma and W. J. Rappel, *Phase field model for computationally efficient modeling of solidification with arbitrary interface kinetics*, Physical Review E, Volume 53, No. 4, pp.3017-3020 (April 1996).
- [8] S. L. Wang and R. F. Sekerka, *Algorithms for phase field computation of the dendritic operating state at large supercoolings*, Journal of Computational Physics 127, pp.110-117 (1996)

- [9] A. R. Umantsev, V. V. Vinogradov and V. T. Borisov, *Mathematical model of growth of dendrites in a supercooled melt*, Sov. Phys, Crystallogr. 30(3), pp. 262-265 (May-June 1985).
- [10] Y. Satio, G. Goldbeck-Wood, and H. Müller-Krumbhaar, *Numerical simulation of dendritic growth*, Physical Review A, Vol. 38, No. 4, pp. 2148-2157 (August 1988)
- [11] A. R. Roosen, *Modeling crystal growth in a diffusion field using fully faceted interfaces*, Journal of computational physics, Vol. 114, pp. 113-128 (1994)
- [12] A. Karma and W. J. Rappel, *Numerical simulation of three dimensional dendritic growth*, Physical Review Letters, Vol. 77, No. 19, pp.4050-4053 (November 1996)
- [13] J. C. LaCombe, M.B. Koss, V. E. Fradkov, and M. E. Glicksman, *Three-dimensional dendrite-tip morphology*, Physical Review E, Vol. 52, No. 3, pp. 2778-2786 (September 1995)
- [14] A. Schmidt, *Computation of three dimensional dendrites with finite elements*, Journal of Computational Physics, Vol 125, pp. 293-312 (1996).
- [15] J. A. Sethian and J. Strain, *Crystal growth and dendritic solidification*, Journal of Computational Physics, Vol. 98, pp. 231-253 (1992).
- [16] R. J. LeVeQue and K. M. Shyue, *Two-Dimensional front tracking based on high resolution wave propagation methods*, Journal of Computational Physics, Vol. 123, pp.354-368 (1996).
- [17] J. S. Langer, in *Directions in Condensed Matter Physics*, eds. G. Grinstein and G. Mazenko (World Science) pp. 164-186 (1986)
- [18] G. Caginalp and E. A. Socolovsky, Journal of Computational Physics, 85 (1995).

- [19] G. Caginalp and P. Fife, *Physical Review B* 34, 4940 (1986).
- [20] G. Caginalp, *Ann. Phys. (New York)*, Vol. 172, 136 (1986).
- [21] A. Artemev, *Transformation of variational equation for phase field evolution into PDE*, pp. 1-8 (September 1996).
- [22] ADOL-C: *A Package for the Automatic Differentiation of Algorithms* written in C/C++ Version 1.6, by A. Griewank, D. Juedes and J. Utke. (January 1995), http://www-c.mcs.anl.gov/autodiff/AD_Tools/adolc.html
- [23] A. Artemev and J. A. Goldak, *Cellular Simulation of the Dendrite Growth in Al-Si Alloys*. pp. 1-27
- [24] G. J. Davies, *Solidification and casting*, Applied Science Publishers Ltd., (1973).
- [25] J. S Langer, *Instabilities and pattern formation in crystal growth*, *Reviews of Modern Physics*, Vol. 52. No.1, pp. 1-28 (January 1980).

IMAGE EVALUATION TEST TARGET (QA-3)

APPLIED IMAGE, Inc.
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved