

Development of Framework for Eutectic Solidification in OpenFOAM

A THESIS SUBMITTED
FOR THE PARTIAL FULFILMENT TO THE AWARD OF
Bachelor of Science (Research)

BY
Kartik Umate
Undergraduate Programme
Indian Institute of Science

supervised by
Dr. Abhik N. Choudhury
Department of Materials Engineering
Indian Institute of Science



CERTIFICATE

This is to certify that the project report entitled "Development of Framework for Eutectic Solidification in OpenFOAM" submitted by Kartik Umate (Sr No. 14574) to Indian Institute of Science, Bangalore towards partial fulfilment of requirements for the award of degree of Bachelor of Science(Research) in Materials Science is a record of bona fide work carried out by him under my supervision and guidance between February-June 2021.

Dr. Abhik N. Choudhury
Department of Materials Engineering
IISc, Bangalore

Declaration

I certify that

- (a) The work contained in this report has been done by me under the guidance of my supervisor.
- (b) The work has not been submitted to any other Institute for any degree or diploma.
- (c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- (d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

Kartik Umate
Undergraduate Programme (Materials Science and Engineering)
IISc, Bangalore

Acknowledgments

Foremost, I would like to thank my supervisor-Dr. Abhik Choudhury for his invaluable guidance and support. His friendly nature makes his lab an amazing place to work, and regular meetings with him are a constant source of inspiration. Thank you for being patient with me and allowing me to take my time in learning things.

I also express my gratitude to Prof. S Karthikeyan, my faculty advisor, who guided me throughout my undergrad studies.

I also thank the UG Dept. and the Dept. of Materials Engineering for the facilities provided.

I would like to thank Sumeet, without whose guidance, I would probably be still trying to compile my solvers while going through obscure OpenFOAM forums. His constant help made it possible to complete this work. I would also like to thank Swapnil for his help and lively discussions about things we are both passionate about.

I thank all my amazing friends here at IISc, without whom the lockdown period would have been unbearable.

Lastly I would like to thank my family, parents and my sister for everything in my life.

Abstract

In this study, a framework for eutectic solidification for binary system is developed in OpenFOAM. The simulations performed are validated by comparison with the analytical Jackson-Hunt eutectic theory and a good match is obtained. A few 3D simulations are also performed to demonstrate that OpenFOAM could indeed be used for larger simulations. This framework offers advantage over conventional Finite Difference (FD) and Fast Fourier Transform (FFT) based phase field solvers which have limitations with respect to the achievable time step and boundary conditions respectively.

Contents

1	Introduction	1
1.1	The phase field method	1
1.2	OpenFOAM	1
1.3	Aim and scope of this work	2
1.4	Organization of the thesis	2
2	Literature Review	3
2.1	Overview of eutectic solidification	3
2.2	Overview of Jackson-Hunt theory for eutectic solidification	4
2.3	Overview phase field models	6
3	Methods	8
3.1	Model Description	8
3.2	Free energies	9
3.3	Evolution equations	10
3.3.1	Relating γ to surface energy of the interface	12
3.4	2D Simulation Setting	13
3.5	3D Simulation Setting	16
3.5.1	Lamellar morphology	16
3.5.2	Rod morphology	17
4	Results and Discussion	20
4.1	2D simulations with equal volume fractions	20
4.1.1	Phase fields	20
4.1.2	Validating with Jackson-Hunt theory	22
4.2	2D simulations with unequal volume fractions	24
4.3	3D simulations	24
4.3.1	Lamellar Morphology	24
5	Conclusions, Planned work and Future scope	27
5.1	Conclusion	27
5.2	Planned Work	27
5.3	Problems Faced	27
5.4	Future scope of work	28

Appendices	29
-------------------	-----------

List of Figures

2.1	Figure showing various morphologies for binary eutectic systems(a) rod (b) lamellar (c)broken lamellar [7]	3
2.2	Example of complex pattern formation in ternary eutectic system. [8]	4
2.3	Representative geometry of lamellar morphology	5
2.4	Figure showing various instabilities eutectic systems(a) tilt instabilities (b)oscillatory instabilities [10]	6
2.5	Example of complex pattern formation in ternary eutectic system [11]	7
3.1	Common tangent construction	9
3.2	Case setting and initialization for the phase-field simulations of binary eutectic systems. A temperature gradient G is imposed along the growth direction. The solid liquid interface follows the isotherms in the case of stable coupled lamellar growth, in steady state.	13
3.3	Initialization of (a) α phase field, (b) β phase field, (c) liquid phase field, (d) T phase field. Note that the range of values differs in all the fields.	14
3.4	(a)uniform mesh, (b)graded mesh	15
3.5	(a) α , β and liq phase field, (b) Cross section at the solid-liquid interface, (c) T field, (d) representative meshing.	18
3.6	For the initialization of the rod morphology(a) α , β and liq phase field, (b) Cross section at the solid-liquid interface, (c) T field, (d) representative meshing.	19
4.1	Eventual morphology with established lambda spacing. It also shows the triple point and curvature of solid liquid interface	20
4.2	Average interface temperature of the contour $\phi_{liq} = 0.5$	21
4.3	Variation of phase fields (a) α (b) β (c)liquid, at steady state	21
4.4	After 20s of smoothening, Phase fields (a) α (b) β	22
4.5	Variation of μ across the $\alpha - \beta$ interface when it is (a)smoothened (b) not smoothened	23
4.6	Phase fields (a) α (b) β showing the final morphology	23
4.7	Comparison with Analytical JH theory	24
4.8	α phase fields for the case of unequal volume fractions. The images in sequence show the time evolution of the microstructure.	25
4.9	Solid-liquid interface showing eutectic lamellar morphology selected by system. Red is the α phase while green is the β phase. The images in sequence show the time evolution of the microstructure.	26

List of Tables

3.1	The parameters used for 2D lamellar eutectic, $dx = 1$	16
3.2	Parameters used for running simulations with unequal volume fraction . .	16

Chapter 1

Introduction

The availability of software packages for a range of applications, advances in development of simulation methodologies and inexpensive, accessible computing power has enabled employment of computational modeling and simulation of materials for developing new materials and materials processing technology. Computational simulation complements experiments in that they can enable understanding of complex experimental data at relatively low cost through rapid testing of models and theoretical predictions.

1.1 The phase field method

The major aim of Integrated Computational Materials Engineering (ICME) is to use models across multiple length scales to develop new materials, processing techniques and to improve materials properties by a better understanding of the overriding mechanisms driving the microstructural transformations. In the past few years, the phase field method has been used as a powerful computational tool for modeling and predicting morphological and microstructural evolution in materials. The phase field method has been used to simulate microstructural evolutions for a range of phenomena including but not limited to solidification [1], solid-state phase transformations [2], grain growth [3], martensitic phase transformations [4], precipitate growth and coarsening [5].

1.2 OpenFOAM

OpenFOAM stands for Open-source Field Operation And Manipulation. It is a general purpose C++ based toolbox used primarily for Computational Fluid Dynamics(CFD) [6]. It can be used to develop custom numerical solvers for specific physical models by writing partial differential equations in an expressive and versatile syntax.

OpenFOAM comes with numerous standard libraries which address basic physics models, numerical algorithms, mesh handling and discretization strategies. Use of these libraries made it possible to develop an efficient parallelized framework for phase field simulation of eutectic systems.

OpenFOAM has open architecture and open source development, which enables the source code for customization, which isn't possible with commercial software packages.

A case in OpenFOAM is a directory where all the files regarding the meshing, setting parameters, physical constants to be used, field initialization, etc. are stated.

1.3 Aim and scope of this work

Conventional solvers that are used to perform phase field simulations include Finite Difference based solvers, which have a limitation on achievable timestep, and Fast Fourier transform based solvers, which fall short due to limited domain sizes. The major aim of this work is to use OpenFOAM's real space implicit solvers to perform phase field simulations for eutectic solidification, which could be validated with already established theories, and gauge its performance to see if it offers any improvement over the conventional solvers.

1.4 Organization of the thesis

In chapter 2, we look at the relevant background literature needed to understand this study. That involves the phase field method and its exemplary applications in material science problems including eutectic solidification. We also discuss the Jackson-Hunt eutectic theory and the importance of establishing undercooling-spacing curve.

In Chapter 3, the phase field model and the evolution equations used in this study are explained. In addition to that the case setup for the simulations performed is described. The 4th chapter discusses the results and their validation with the analytical theories. And we conclude with the 5th chapter which comprises the conclusions, problems faced and the future scope of work.

Chapter 2

Literature Review

2.1 Overview of eutectic solidification

Eutectic alloys find importance in their low melting properties and mechanical properties provided the structure is uniform at the finest scale. It becomes important to study the effects of solidification conditions on the final microstructure of the alloys. Eutectic phase transformations give rise to microstructure with complex pattern formation like rods, lamellae, labyrinth, etc. for binary alloys which becomes even more complex with increasing number of components and phases. There are a multitude of factors that influence the final morphology including equilibrium volume fraction of phases, surface energy anisotropy, processing parameters among others.

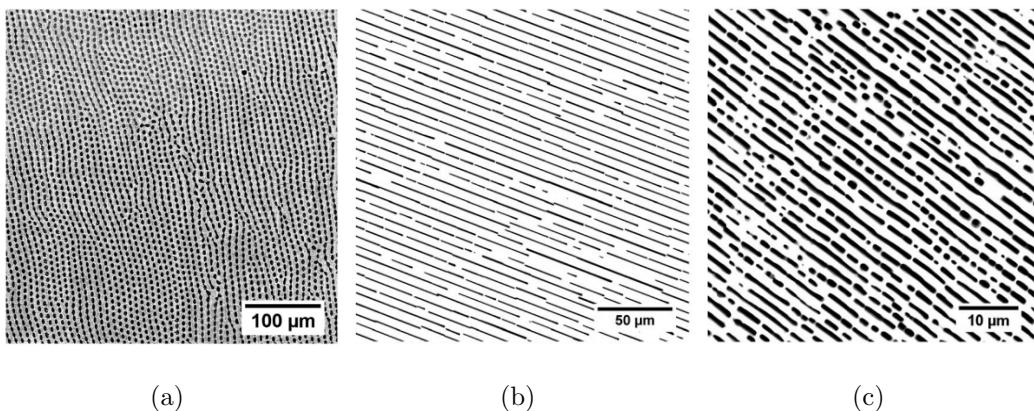


Figure 2.1: Figure showing various morphologies for binary eutectic systems(a) rod (b) lamellar (c)broken lamellar [7]

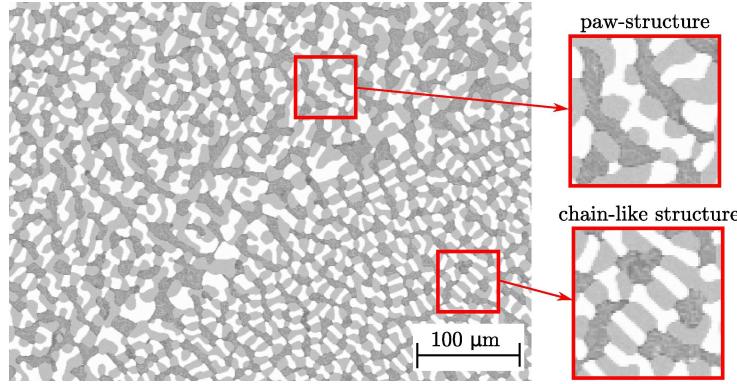


Figure 2.2: Example of complex pattern formation in ternary eutectic system. [8]

2.2 Overview of Jackson-Hunt theory for eutectic solidification

For a binary eutectic system, a eutectic point is characterized by the eutectic temperature T_{eut} and the eutectic composition c_{eut} . At this point in the phase diagram, two solid phases co-exist with the liquid phase at equilibrium. If the far field composition in the virtually infinite melt is c_{eut} , then solidification produces complex patterns consisting of two solid phases. Jackson and Hunt attempted to understand this process of pattern formation in their seminal paper. They calculated approximate solutions for pattern spacing for the rod and lamellar morphology. They state that the average interface undercooling i.e the difference between eutectic temperature and the average interface temperature is given by the relation [9],

$$\Delta T = vK_1\lambda + \frac{K_2}{\lambda} \quad (2.2.1)$$

where v is the imposed velocity of the directional solidification and λ is the spacing which will be width of one lamella pair in the case of lamellar whereas in the case of rods, it is the distance between two rod centers. K_1 is influenced by redistribution of the solute ahead of the front into the liquid and is given by

$$K_1 = \frac{mP(1 + \zeta)^2(C_{eut}^\alpha - C_{eut}^\beta)}{\zeta D} \quad (2.2.2)$$

where ζ is volume fraction S_β/S_α as described in Figure 2.3 and D is diffusivity in liquid, and P is a series,

$$P = \sum_{i=1}^{\infty} (n\pi)^{-3} \sin^2 \left(\frac{n\pi S_\alpha}{S_\alpha + S_\beta} \right) \quad m = \frac{m_\alpha m_\beta}{(m_\beta + m_\alpha)} \quad (2.2.3)$$

while K_2 is influenced by the Gibbs-Thomson coefficient i.e. due to the curvature

$$K_2 = 2m(1 + \zeta) \left(\frac{T_{eut}}{L_\alpha m_\alpha} \sigma_\alpha^{liq} \sin \theta_\alpha^{liq} + \frac{T_{eut}}{L_\beta m_\beta} \sigma_\beta^{liq} \sin \theta_\beta^{liq} \right) \quad (2.2.4)$$

where m_α, m_β are the liquidus slopes of the phases α and β , L is the latent heat of fusion per unit volume of the respective phases, σ is the surface energy of the respective

interphase interfaces and θ is the angle as described in Figure 2.3 finally giving the characteristic shape of the ΔT vs λ curve. This has a minima at

$$\lambda_{min} = \sqrt{\frac{K_2}{vK_1}} \quad (2.2.5)$$

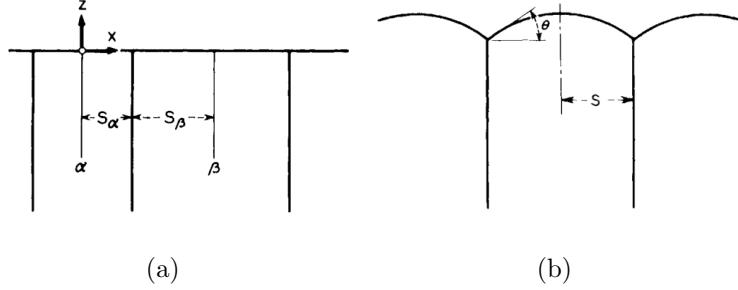


Figure 2.3: Representative geometry of lamellar morphology

[9]

Typically in samples made of eutectic alloys, only a narrow range of spacings around λ_{min} are found. But a range of spacings can be reached in directional solidification experiments by imposing a velocity that could be changed with time; this gives a way to probe stability of the morphologies. Theoretical analysis also shows that the range of spacings, in which steady state growth is stable, is limited by occurrences of instabilities. For spacings much lower than λ_{min} , elimination instability is observed while for higher spacings a multitude of instabilities are seen depending on the geometry of sample including oscillatory instabilities, tilt instabilities, zig-zag instabilities, etc.

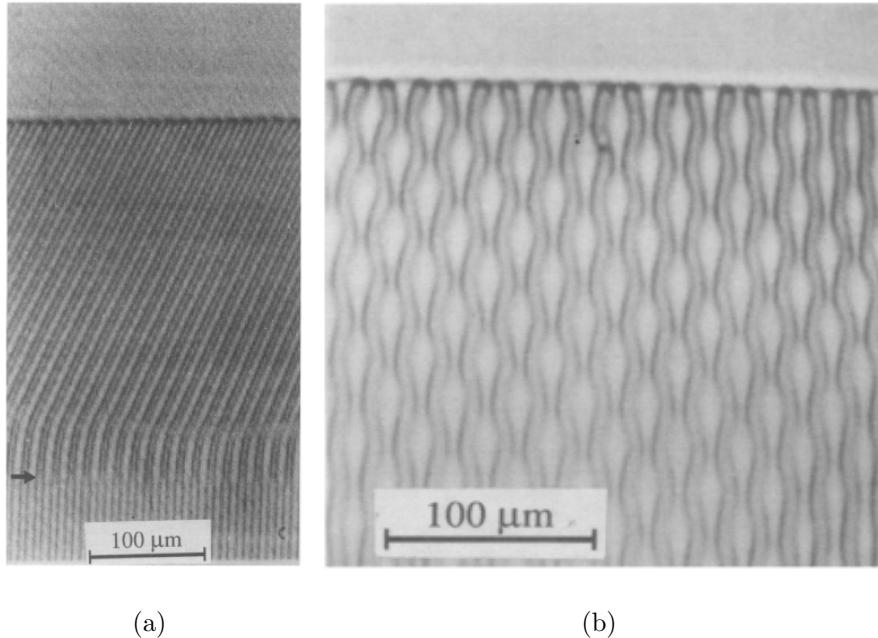


Figure 2.4: Figure showing various instabilities eutectic systems(a) tilt instabilities (b)oscillatory instabilities [10]

2.3 Overview phase field models

One of the reasons why phase field is an attractive option for microstructural evolution simulation is the way it handles interfaces. The interface representing the boundary between two phases is replaced with a continuous and smoothly varying function called order parameter, whose change represents microstructural evolution. This removes the need for tracking the interface and thus makes large scale simulations of microstructural evolution involving complex geometrical changes computationally feasible.

A functional incorporates the energy of the bulk phases as well as surfaces, variational derivative of which with respect to the order parameter chosen gives the driving force for phase change. This driving force in turn depends on the temperature and the compositions of components involved depending on whether the system is pure material or a multi-component system.

The first phase field formulation for solidification was established by Langer [12]. Relating the phase field evolution equation to the sharp interface boundary problem involves taking the sharp interface limit of the phase-field model, in which the interface thickness tends to zero and is replaced by interfacial boundary conditions. This was done by Caginalp and co-workers [13] by asymptotic expansions and they demonstrated that that limiting free boundary problem depends on the particular limit that is taken. A generic framework for treating multi-phase, multi-component systems were proposed by Nestler and co-workers [14] [15][16]. In these models, there is no coupling between interfacial thickness from interfacial energy for parabolic phase free energies. They introduced interpolations functions that prevented the spontaneous appearance of the third phase at a two-phase interface, which are employed in the current work. In [16], phase field model based on grand-potential functional is presented, whereby the interfacial energy is decoupled from interfacial width. Also, an evolution equation for chemical potential of species is used in

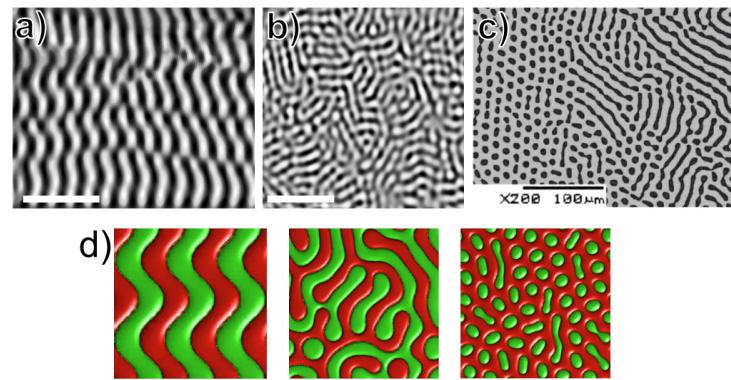


Figure 2.5: Example of complex pattern formation in ternary eutectic system [11]

place of the composition. This model is employed in the current work.

Chapter 3

Methods

3.1 Model Description

The phase field model used in this study is based on the Grand-potential formulation described in [16]. The three phases, alpha, beta and the liquid phase are described by three phase fields ϕ_i , $i = \alpha, \beta, liq$, representing the volume fractions. According to this model, state of the binary eutectic system is described by the order parameter $\boldsymbol{\phi} = \{\phi_\alpha, \phi_\beta, \phi_{liq}\}$, the chemical potential μ , and the temperature T. The constraints on individual order parameters are:

$$\phi_i \in [0, 1] \forall i \quad \sum_i \phi_i = 1 \quad (3.1.1)$$

With ϵ as the diffuse interface width parameter, the Grand-potential functional reads,

$$\Omega(T, \mu, \boldsymbol{\phi}) = \int_V \left[\Psi(T, \mu, \boldsymbol{\phi}) + \epsilon a(\boldsymbol{\phi}, \nabla \boldsymbol{\phi}) + \frac{w(\boldsymbol{\phi})}{\epsilon} \right] dV \quad (3.1.2)$$

where, the grand potential is written as sum of grand potential of phases present in the system,

$$\Psi(T, \mu, \boldsymbol{\phi}) = \sum_i \Psi_i(T, \mu) h_i(\boldsymbol{\phi}) \quad (3.1.3)$$

The partial Legendre Transform of the Grand potential reads,

$$\Psi_i(T, \mu) = f_i(c_i(\mu, T), T) - \mu c_i(\mu, T) \quad (3.1.4)$$

where f_i is the free energy and c_i is the composition of the i^{th} phase and h_i are the interpolation functions.

From equation (3.1.3), we have

$$\frac{\partial \Psi(T, \mu, \boldsymbol{\phi})}{\partial \mu} = \sum_i \frac{\partial \Psi_i(T, \mu)}{\partial \mu} h_i(\boldsymbol{\phi}) \quad (3.1.5)$$

and since,

$$\frac{\partial \Psi(T, \mu, \phi)}{\partial \mu} = -c \quad (3.1.6)$$

we get,

$$c = \sum_i c_i(T, \mu) h_i(\phi) \quad (3.1.7)$$

In the isotropic case, the second term in the functional translates to,

$$a(\phi, \nabla \phi) = \sum_i \gamma |\nabla \phi_i|^2 \quad (3.1.8)$$

where γ is isotropic surface energy parameter.

The third term is taken as a double well potential, it takes the form,

$$w(\phi) = \sum_i 9\gamma \phi_i^2 (1 - \phi_i)^2 \quad (3.1.9)$$

The parameters ϵ and γ together determine the diffuse interface width and the interface energy. This is explained in Section 3.3.

3.2 Free energies

The parabolic approximation for free energies is described in this section.

$$f_i = A_i c_i^2 + B_i(T) c_i + D_i(T) \quad (3.2.1)$$

where the coefficients A_i , B_i and D_i are derived following the consequence of equilibrium between the different phases in the problem described by the common tangent construction in Figure 3.1

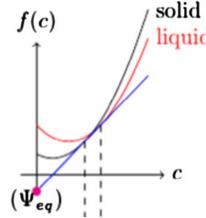


Figure 3.1: Common tangent construction

Taking $B_{liq}(T) = 0$ and $D_{liq}(T) = 0$, and using the following conditions,

$$\mu = \left. \frac{\partial f_\alpha}{\partial c} \right|_{c_\alpha} = \left. \frac{\partial f_{liq}}{\partial c} \right|_{c_{liq}} \quad (3.2.2)$$

$$\mu = \left. \frac{\partial f_\beta}{\partial c} \right|_{c_\beta} = \left. \frac{\partial f_{liq}}{\partial c} \right|_{c_{liq}} \quad (3.2.3)$$

$$f_\alpha - \mu c_\alpha = f_{liq} - \mu c_{liq} \quad (3.2.4)$$

$$f_\beta - \mu c_\beta = f_{liq} - \mu c_{liq} \quad (3.2.5)$$

we arrive at the following expressions for the coefficients,

$$B_\alpha(T) = 2A_{liq}c_{liq}^{eut} - 2A_\alpha c_\alpha^{eut} + (T - T^{eut}) \left[\frac{2A_{liq}}{m_{liq-\alpha}} - \frac{2A_\alpha}{m_{\alpha-liq}} \right] \quad (3.2.6)$$

$$B_\beta(T) = 2A_{liq}c_{liq}^{eut} - 2A_\beta c_\beta^{eut} + (T - T^{eut}) \left[\frac{2A_{liq}}{m_{liq-\beta}} - \frac{2A_\beta}{m_{\beta-liq}} \right] \quad (3.2.7)$$

$$D_\alpha(T) = -A_{liq}(c_{liq}^{eut})^2 + A_\alpha(c_\alpha^{eut})^2 + (T - T^{eut}) \left[\frac{-2A_{liq}c_{liq}^{eut}}{m_{liq-\alpha}} + \frac{2A_\alpha c_\alpha^{eut}}{m_{\alpha-liq}} \right] \quad (3.2.8)$$

$$D_\beta(T) = -A_{liq}(c_{liq}^{eut})^2 + A_\beta(c_\beta^{eut})^2 + (T - T^{eut}) \left[\frac{-2A_{liq}c_{liq}^{eut}}{m_{liq-\beta}} + \frac{2A_\beta c_\beta^{eut}}{m_{\beta-liq}} \right] \quad (3.2.9)$$

where c_β^{eut} , c_α^{eut} and c_{liq}^{eut} are the β , α and the liquid phase compositions respectively at the eutectic temperature, T^{eut} . The liquidus slopes for $\alpha - liq$ phase equilibrium is denoted by $m_{l-\alpha}$ while the corresponding solidus slope by $m_{\alpha-l}$ and similar notation is followed for the $\beta - liq$ phase equilibrium.

For simplicity, we assume $A_{liq} = A_\alpha = A_\beta = A$, which combined with equation (3.1.4) gives,

$$\Psi_\alpha(T, \mu) = -A \left[\frac{\mu - B_\alpha}{2A} \right]^2 + D_\alpha(T) \quad (3.2.10)$$

$$\Psi_\beta(T, \mu) = -A \left[\frac{\mu - B_\beta}{2A} \right]^2 + D_\beta(T) \quad (3.2.11)$$

$$\Psi_{liq}(T, \mu) = -A \left[\frac{\mu}{2A} \right]^2 \quad (3.2.12)$$

The interpolation functions $h_i(\phi)$ [14] are chosen such that

$$h_i = \frac{\phi_i^2}{4} \left\{ 15(1 - \phi_i) [1 + \phi_i - (\phi_k - \phi_j)^2] + \phi_i(9\phi_i^2 - 5) \right\} \quad (3.2.13)$$

where i, j and k are all different and $\epsilon \{ \phi_\alpha, \phi_\beta, \phi_{liq} \}$. These interpolation functions make sure that at the interface of phases i and j, the phase field for phase k remains zero.

3.3 Evolution equations

The evolution equations for the phase fields following minimization of the Grand-potential functional in equation (3.1.2) describes system's approach towards an equilibrium state. The evolution equations are,

$$\frac{\partial \phi_i}{\partial t} = -\frac{1}{\tau} \frac{\delta \Omega}{\delta \phi_i} - \lambda \quad (3.3.1)$$

which translates to,

$$\tau \epsilon \frac{\partial \phi_i}{\partial t} = \epsilon \left(\nabla \cdot \frac{\partial a(\phi, \nabla \phi)}{\partial \nabla \phi_i} - \frac{\partial a(\phi, \nabla \phi)}{\partial \phi_i} \right) - \frac{1}{\epsilon} \frac{\partial w(\phi)}{\partial \phi_i} - \frac{\partial \Psi(T, \mu, \phi)}{\partial \phi_i} - \lambda \tau \quad (3.3.2)$$

where τ is the relaxation constant which influences the kinetics of the phase transformation at the interfaces and if assumed same for all the interfaces, is given by the expression,

$$\tau = \frac{\epsilon(\Delta C)^2}{D \frac{\partial c}{\partial \mu}}(F) \quad (3.3.3)$$

where for the thermodynamics chosen, the constant F takes the value of 0.2.

λ is the Lagrange multiplier used for imposing the constraint $\sum_i \phi_i = 1$. Using equation (3.3.1) one can arrive at,

$$\lambda = -\frac{1}{3\tau} \left(\frac{\delta \Omega}{\delta \phi_\alpha} + \frac{\delta \Omega}{\delta \phi_\beta} + \frac{\delta \Omega}{\delta \phi_{liq}} \right) \quad (3.3.4)$$

These three equations for the phase fields corresponding to the liquid, α and the β phases are solved in a coupled manner along with mass balance equation which takes the form,

$$\frac{\partial c}{\partial t} = \nabla \cdot (M(\phi, c) \nabla \mu) \quad (3.3.5)$$

which when combined with equation (3.1.7),

$$\left[\sum_i \frac{\partial c_i}{\partial \mu} h_i(\phi) \right] \frac{\partial \mu}{\partial t} + \sum_i c_i \left(\sum_n \frac{\partial h_i}{\partial \phi_n} \frac{\partial \phi_n}{\partial t} \right) = \nabla \cdot (M(\phi, c) \nabla \mu) - \nabla \cdot (j_{at}) \quad (3.3.6)$$

where $i, n \in \{\alpha, \beta, liq\}$ and the mobility term M , if the diffusivities $D_\alpha = D_\beta = D_s \neq 0$, takes the form,

$$M(\phi, \mu) = \left(D_{liq} g_{liq}(\phi) \frac{\partial c_{liq}(\mu, T)}{\partial \mu} + D_s [1 - g_s(\phi)] \frac{\partial c_s(\mu, T)}{\partial \mu} \right) \quad (3.3.7)$$

where, in the case since we have, $A_{liq} = A_\alpha = A_\beta$,

$$\frac{\partial c_s(\mu, T)}{\partial \mu} = \frac{\partial c_\alpha(\mu, T)}{\partial \mu} = \frac{\partial c_\beta(\mu, T)}{\partial \mu} \quad (3.3.8)$$

and $g(\phi)$ are interpolation functions later chosen as the value of phase field itself

If $D_\alpha = D_\beta = D_s = 0$ and $g_{liq} = \phi_{liq}$

$$M(\phi, c) = D_{liq} \phi_{liq} \frac{\partial c_{liq}(\mu, T)}{\partial \mu} \quad (3.3.9)$$

In such a case of 1 sided diffusion problem, incorporating asymptotic corrections becomes necessary, which is done by forcing an anti-trapping current, j_{at} from solid to liquid in the direction parallel to solid-liquid interface normal.

$$j_{at} = j_{at}^{\alpha \rightarrow liq} \left(- \frac{\nabla \phi_\alpha}{|\nabla \phi_\alpha|} \cdot \frac{\nabla \phi_{liq}}{|\nabla \phi_{liq}|} \right) + j_{at}^{\beta \rightarrow liq} \left(- \frac{\nabla \phi_\beta}{|\nabla \phi_\beta|} \cdot \frac{\nabla \phi_{liq}}{|\nabla \phi_{liq}|} \right) \quad (3.3.10)$$

where $j_{at}^{i \rightarrow liq}$ is

$$j_{at}^{i \rightarrow liq} = -\frac{\epsilon}{2\sqrt{2}} [c_{liq}(\mu) - c_i(\mu)] \frac{\partial \phi_i}{\partial t} \frac{\nabla \phi_i}{|\nabla \phi_i|} \quad (3.3.11)$$

where, $i \in \{\alpha, \beta\}$

3.3.1 Relating γ to surface energy of the interface

Consider the α -liquid interface. Due to the interpolation functions chosen, the value of ϕ_β will be zero at this interface.

In the absence of driving forces, the interface becomes diffuse over a characteristic length scale defined by the parameters ϵ and γ . The evolution equation (3.3.2) in that case reads,

$$\tau\epsilon \frac{\partial \phi_i}{\partial t} = \epsilon \left(\nabla \cdot \frac{\partial a(\phi, \nabla \phi)}{\partial \nabla \phi_i} - \frac{\partial a(\phi, \nabla \phi)}{\partial \phi_i} \right) - \frac{1}{\epsilon} \frac{\partial w(\phi)}{\partial \phi_i} - \lambda \tau \quad (3.3.12)$$

Expanding each of the terms for the alpha phase evolution equation gives,

$$\begin{aligned} \tau\epsilon \frac{\partial \phi_\alpha}{\partial t} &= -\frac{2}{3} \left[-2\gamma\epsilon \nabla^2 \phi_\alpha + \frac{9\gamma}{\epsilon} 2\phi_\alpha (2\phi_\alpha - 1)(\phi_\alpha - 1) \right] + \\ &\quad \frac{1}{3} \left[-2\gamma\epsilon \nabla^2 \phi_{liq} + \frac{9\gamma}{\epsilon} 2\phi_{liq} (2\phi_{liq} - 1)(\phi_{liq} - 1) \right] \end{aligned} \quad (3.3.13)$$

Now due to the constraint on phase fields we have $\phi_\alpha = 1 - \phi_{liq}$, the above equation becomes,

$$\begin{aligned} \tau\epsilon \frac{\partial \phi_\alpha}{\partial t} &= -\frac{2}{3} \left[-2\gamma\epsilon \nabla^2 \phi_\alpha + \frac{9\gamma}{\epsilon} 2\phi_\alpha (2\phi_\alpha - 1)(\phi_\alpha - 1) \right] - \\ &\quad \frac{1}{3} \left[-2\gamma\epsilon \nabla^2 \phi_\alpha + \frac{9\gamma}{\epsilon} 2\phi_\alpha (2\phi_\alpha - 1)(\phi_\alpha - 1) \right] \end{aligned} \quad (3.3.14)$$

At steady state, $\frac{\partial \phi_\alpha}{\partial t} = 0$, thus,

$$\gamma\epsilon \nabla^2 \phi_\alpha = \frac{9\gamma}{\epsilon} \phi_\alpha (2\phi_\alpha - 1)(\phi_\alpha - 1) \quad (3.3.15)$$

On integrating we get,

$$\gamma\epsilon (\nabla \phi_\alpha)^2 = \frac{9\gamma}{\epsilon} \phi_\alpha^2 (1 - \phi_\alpha)^2 \quad (3.3.16)$$

The interfacial energy, σ , derived from the model will be,

$$\sigma = \int_V \left[\epsilon a(\phi, \nabla \phi) + \frac{w(\phi)}{\epsilon} \right] dV \quad (3.3.17)$$

which when combined with equation (3.3.16) and the constraint $\phi_\alpha = 1 - \phi_{liq}$

$$\sigma = \int_V [4\gamma\epsilon (\nabla \phi_\alpha)^2] dV \quad (3.3.18)$$

$$= \int_\phi [4\gamma\epsilon \nabla \phi_\alpha] d\phi \quad (3.3.19)$$

$$= \int_0^1 \frac{12}{\epsilon} \gamma\epsilon \phi_\alpha (1 - \phi_\alpha) d\phi \quad (3.3.20)$$

$$= 2\gamma \quad (3.3.21)$$

Thus we can see that the interface energy in the problem will have a non-dimensional value of 2γ .

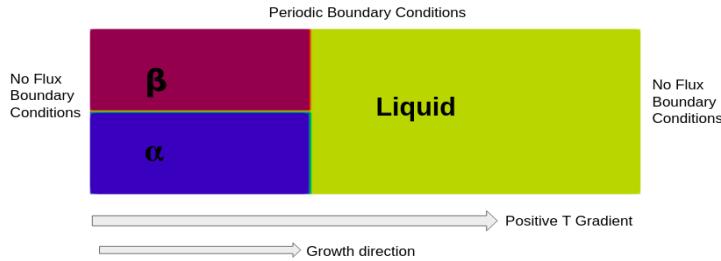


Figure 3.2: Case setting and initialization for the phase-field simulations of binary eutectic systems. A temperature gradient G is imposed along the growth direction. The solid liquid interface follows the isotherms in the case of stable coupled lamellar growth, in steady state.

3.4 2D Simulation Setting

To establish performance of the solver created in OpenFOAM, 2D eutectic growth is simulated to identify the undercooling vs spacing relationship. This is then validated with Jackson-Hunt theory for lamellar morphologies. This section describes the case setup for the 2d-simulations.

Since we want to simulate directional solidification conditions, a moving temperature gradient is imposed in the domain,

$$T(x, t) = T_{initial} + G_T(x - vt) \quad (3.4.1)$$

The thermal diffusion coefficient is typically much larger than the mass diffusion coefficient therefore the temperature profiles adjusts according to the composition profile almost instantly. Thus it can be assumed that the temperature profile is always at steady state.

The case initialization and the boundary conditions of the domain is described in Figure 3.1. The initial values of the phase fields corresponding to the regions shown are set. The chemical potential, μ field is set to the equilibrium value μ_{eq} in the entire domain. And the temperature field is set according to the equation (3.4.1). A representative case is shown in Figure 3.2

Also, the far field composition in the liquid has to correspond to the eutectic composition, which implies that the solute boundary layer has to lie entirely in the domain. The domain size in the growth direction was chosen to be around $3D/v$, which, if the solid-liquid interface is at $(2/5)^{th}$ from the origin as shown in Figure 3.2, gives the liquid region at $1.8D/v$. This makes sure that the boundary layer is contained in the box and thus the far field composition in the liquid remains at the eutectic composition.

The Cartesian meshing of the domain is represented in Figure 3.3(a). The refinement strategy used is such that all the interfaces are in the finest region of mesh, while the bulk of liquid phase, where there are no phase field gradients, is progressively made coarser in the growth direction.

A domain of size 300 in the growth direction, if resolved in 300 grid points, is said to have a dx of 1. If resolved in 150 grid points, it has a dx of 2.

As the simulation progresses, the solid-liquid interface will choose an undercooling and

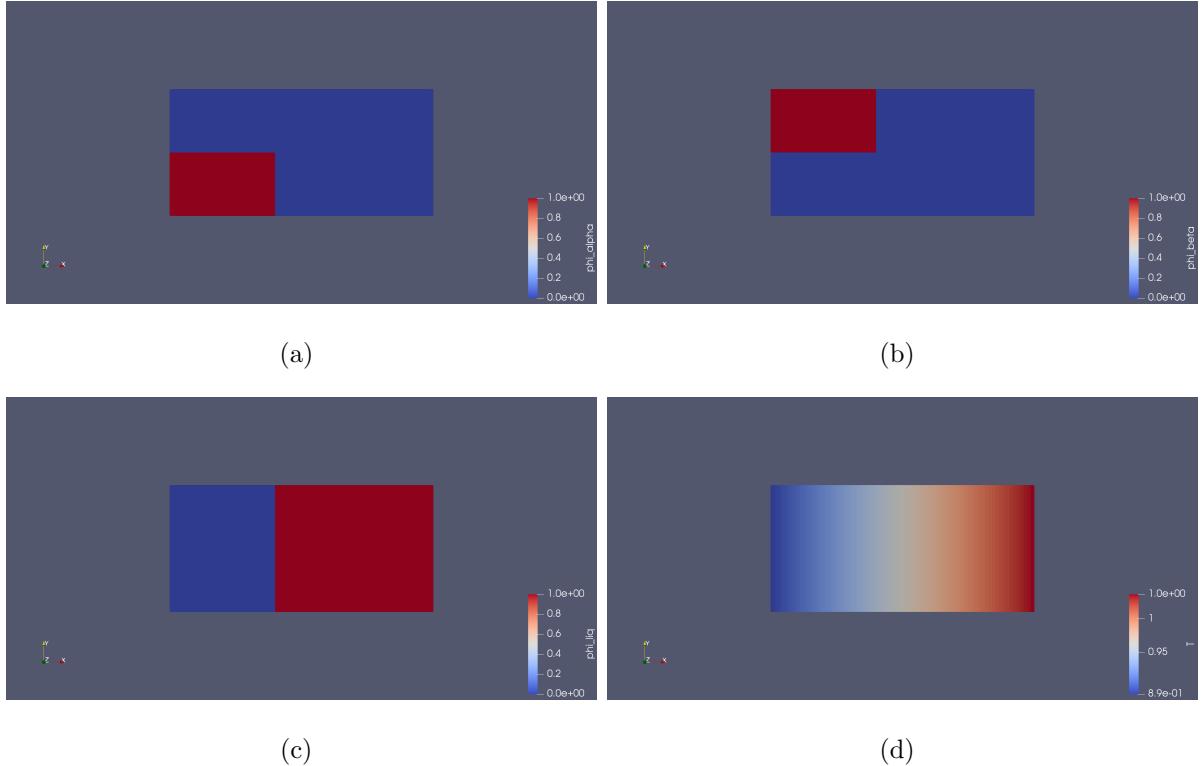


Figure 3.3: Initialization of (a) α phase field, (b) β phase field, (c) liquid phase field, (d) T phase field. Note that the range of values differs in all the fields.

will traverse with the temperature gradient with the imposed velocity. Since we want to simulate solidification from an infinite reservoir of liquid, it becomes necessary to implement shifts in all the fields in the domain to ensure that the far field composition of the liquid remains the same as eutectic composition. This is analogous to having a camera frame at the solid liquid interface which is moving along with it into a virtually infinite liquid in the growth direction.

This is implemented simply by ‘rigidly convecting’ all the fields. This requires adding the flux term $v \cdot \nabla \phi$, where v is the imposed velocity and ϕ is phase field parameter of all phases and also the chemical potential field. Since, v is a constant, all these fields will simply be translated as it as in the entire domain.

This shifts are required to be performed in the growth direction, which is chosen to be X-direction. So the equations each become,

$$\tau \epsilon \frac{\partial \phi_i}{\partial t} - v \frac{\partial \phi_i}{\partial x} = \epsilon \left(\nabla \cdot \frac{\partial a(\phi, \nabla \phi)}{\partial \nabla \phi_i} - \frac{\partial a(\phi, \nabla \phi)}{\partial \phi_i} \right) - \frac{1}{\epsilon} \frac{\partial w(\phi)}{\partial \phi_i} - \frac{\partial \Psi(T, \mu, \phi)}{\partial \phi_i} - \lambda \tau \quad (3.4.2)$$

and the chemical diffusion evolution equation becomes,

$$\left[\sum_i \frac{\partial c_i}{\partial \mu} h_i(\phi) \right] \frac{\partial \mu}{\partial t} + \sum_i c_i \left(\sum_n \frac{\partial h_i}{\partial \phi_n} \frac{\partial \phi_n}{\partial t} \right) - v \frac{\partial c}{\partial x} = \nabla \cdot (M(\phi, c) \nabla \mu) \quad (3.4.3)$$

where $i, n \in \{\alpha, \beta, liq\}$. Since each of the fields is being moved in the direction opposite

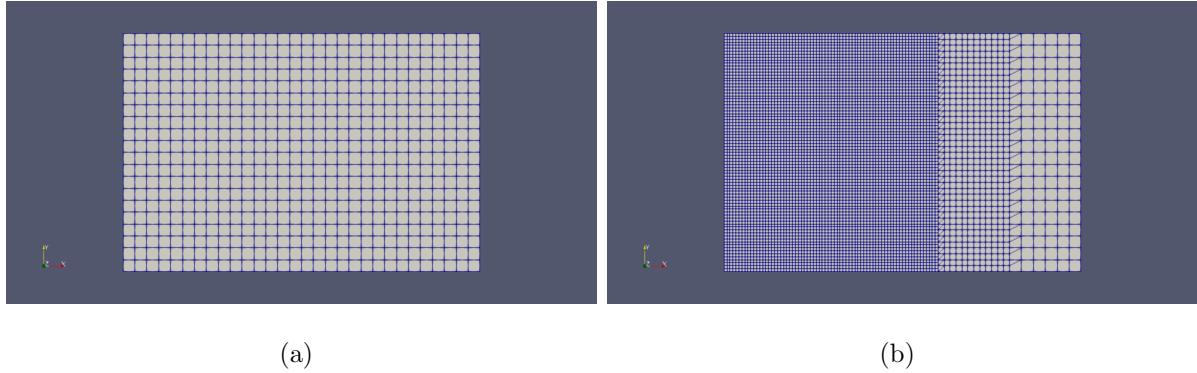


Figure 3.4: (a)uniform mesh, (b)graded mesh

to growth direction, the temperature gradient is kept stationary and solid-liquid interface is expected to find its equilibrium undercooling.

The parameters ϵ and γ are chosen such that all the interfaces involved in the problem are resolved in 10-15 grid points for the chosen dx .

The equilibrium phase diagram compositions are such that the volume fraction of α and β phases is equal. This simulation is first ran for $dx = 1$ with $\epsilon = 4$ and $\tau = 0.144$ and verified with analytical Jackson Hunt theory. In this the interface is resolved in about 13-15 grid points.

Then the same case is ran with $dx = 2$ with $\epsilon = 8$ and $\tau = 0.288$ and verified similarly, to test the numerical consistency for improved speeds. In this too, the interface is resolved in about 13-15 grid points, which is the reason why we doubled the values of τ and ϵ .

Another simulation is ran for identical conditions as described above except the volume fractions are changed to $\alpha = 70\%$ and $\beta = 30\%$. At this volume fraction the minimum lambda spacing increases to a higher value, which requires us to initialize with a higher lambda spacing to obtain a stable microstructure. The only material parameters that are needed to change are the compositions as tabulated below.

This case is initialized with lambda spacing of 256 and the initial phase fractions are kept 50-50 for the α and β phases. It is expected to establish the 30-70 volume fraction itself. In the simulations, the diffusivity in solid is assumed to be zero and only D in liquid is considered. This poses a problem at the $\alpha - \beta$ interface. At the start of the simulation, it is difficult to achieve a well-diffused $\alpha - \beta$ interface because of zero diffusivity.

This makes it essential to run the simulation for the first 20s with diffusivity on both the liquid and solid sides so that the $\alpha - \beta$ interface could achieve a smooth and well diffused profile, after which the simulation could be ran with D non-zero only in the liquid. We refer to this as 'smoothening'.

The interface temperature is extracted by setting the contour for the condition where $\phi_{liq} = 0.5$, and getting the average of temperature of all points on the contour. Steady state is said to be achieved when the interface finds its equilibrium undercooling and stays at that temperature.

ParaView [17] and matplotlib [18] library in Python are used for post-processing and visualization of simulation.

Table 3.1: The parameters used for 2D lamellar eutectic, $dx = 1$

Parameter	Symbol	Value(Non Dimensional)
Eutectic Temperature	T_{eut}	1
Eutectic Composition	c_l^{eut}	0.5
	c_A^{eut}	0.2
	c_B^{eut}	0.8
Liquidus Slopes	$m^{l-\alpha}$	-0.5
	$m^{l-\beta}$	0.5
Solidus Slopes	$m^{\alpha-l}$	-0.5
	$m^{\beta-l}$	0.5
Diffusivity	D	1
Velocity	v	10^{-2}
Interface energy parameter (in the model)	γ	1
Interface width parameter (in the model)	ϵ	4
Interface energy	σ	2
Interface width	δ	14

Table 3.2: Parameters used for running simulations with unequal volume fraction

Parameter	Value(Non Dimensional)
c_l^{eut}	0.5
c_A^{eut}	0.41
c_B^{eut}	0.71

3.5 3D Simulation Setting

To demonstrate that the code written in OpenFOAM could be used to simulate 3-D eutectic growth simulations with accuracy and improvements in speed, the following two 3-D simulations were performed.

3.5.1 Lamellar morphology

The initialization of the case is shown in Fig 3.5. A similar meshing strategy is followed as that of the 2D simulations as shown in Fig 3.5 (d). The chemical potential field is

kept at μ_{eq} all throughout the domain. A temperature gradient in the growth direction (x-direction) is imposed similar to the 2d simulations as shown in Fig. 3.5 (c). The phase fields of α and β phases are initialized randomly in the plane perpendicular to the growth direction as shown. This simulation is ran with $dx = 2$, with the domain size of 160 X 160 X 160. The contours of $\alpha - liq$ interface and $\beta - liq$ interface is followed to get the morphology of the eutectic. This is ran with the same parameters as 2D case setup with equal volume fractions.

3.5.2 Rod morphology

The initialization of the case is shown in Fig 3.6. The meshing strategy and the initialization of μ and the temperature field is identical to that of the 3D lamellar. The phase field of β phase is initialized such that it has a rod morphology in the growth direction as shown. This simulation is ran with $dx = 2$, with the domain size of 184 X 184 X 400. The contours of $\alpha - liq$ interface and $\beta - liq$ interface is followed to get the morphology of the eutectic and this is ran with the same parameters as 2D case setup with unequal volume fractions, 30 % β - 70 % α .

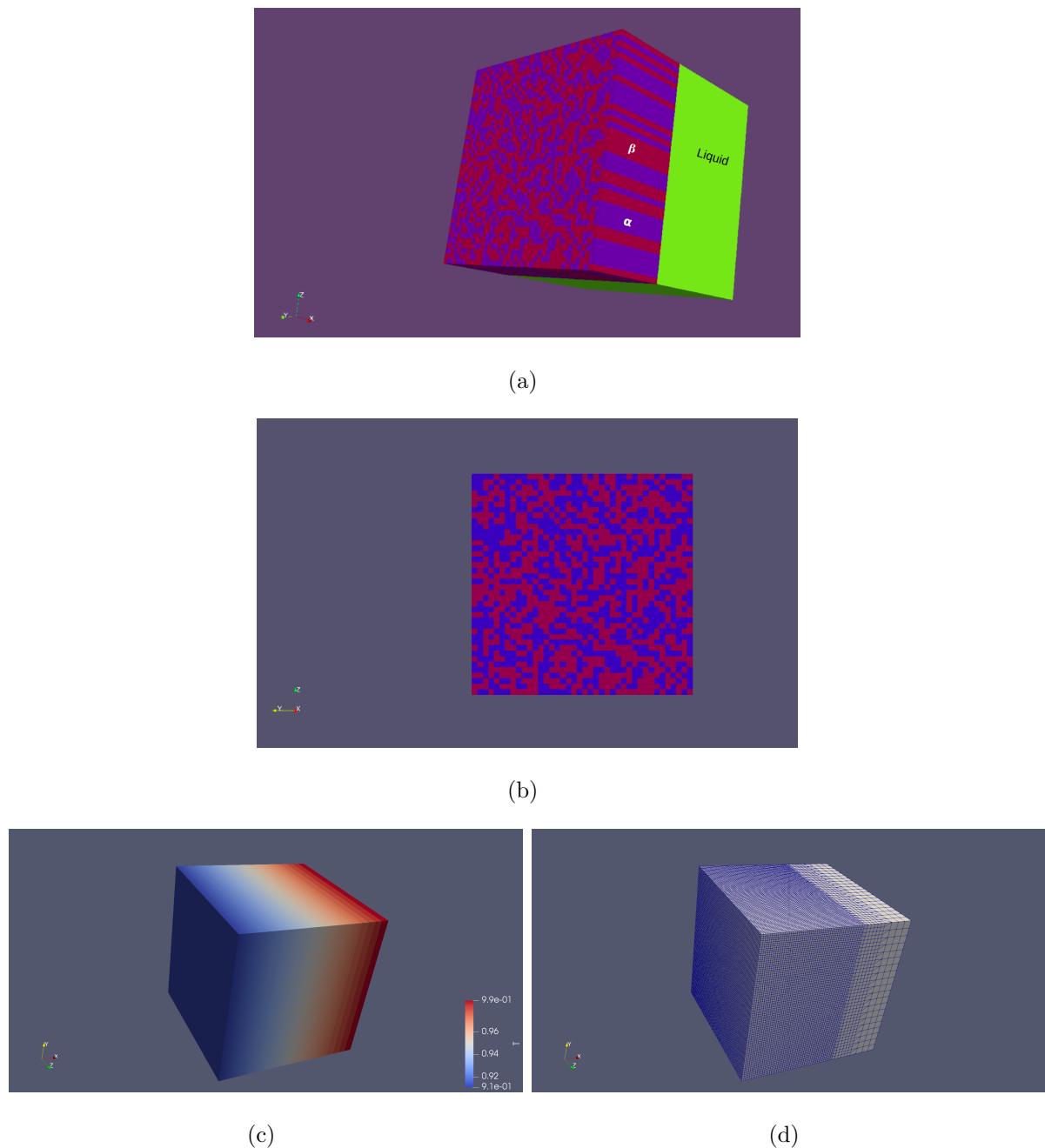


Figure 3.5: (a) α, β and liq phase field, (b) Cross section at the solid-liquid interface, (c) T field, (d) representative meshing.

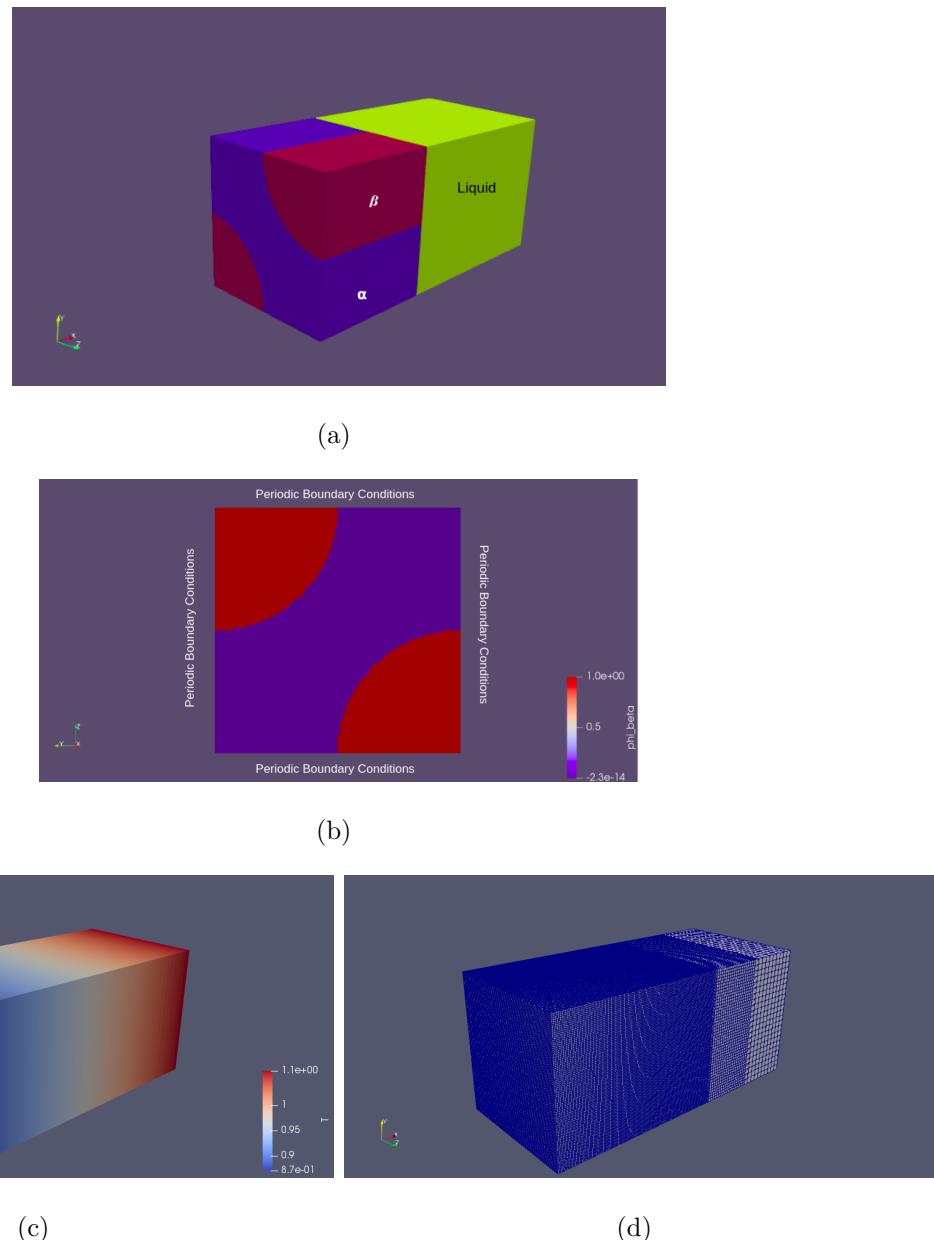


Figure 3.6: For the initialization of the rod morphology (a) α, β and liq phase field, (b) Cross section at the solid-liquid interface, (c) T field, (d) representative meshing.

Chapter 4

Results and Discussion

4.1 2D simulations with equal volume fractions

A representative figure of the final morphology of the 2D lamellar simulations is shown in Figure 4.1. Curvature in the $\alpha - \text{liquid}$ and the $\beta - \text{liquid}$ interface can be clearly seen. This arises due to the triple point where all the three phases meet, where force balance condition in growth direction can only be satisfied if the interfaces are curved.



Figure 4.1: Eventual morphology with established lambda spacing. It also shows the triple point and curvature of solid liquid interface

4.1.1 Phase fields

These 2D phase field simulations are run for varying box sizes in the direction perpendicular to growth direction. In each case, that λ spacing is forced and the system is expected to find the corresponding equilibrium temperature or undercooling.

The Fig 4.3 shows a representative of variation of phase fields in the domain at steady state. We say steady state has been reached when the solid-liquid interface stays at a particular temperature i.e it establishes its undercooling. The interface temperature is tracked as described in Section 3.4 and a representative plot for $\lambda = 80$ is shown in Fig 4.2. We see that with time the temperature becomes constant at a particular value in spite of being initialized at a different value, this way we can be sure that the system has reached steady state.

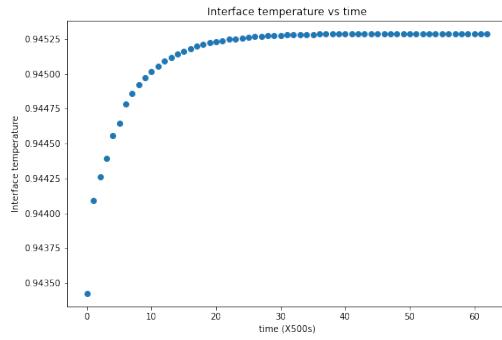


Figure 4.2: Average interface temperature of the contour $\phi_{liq} = 0.5$

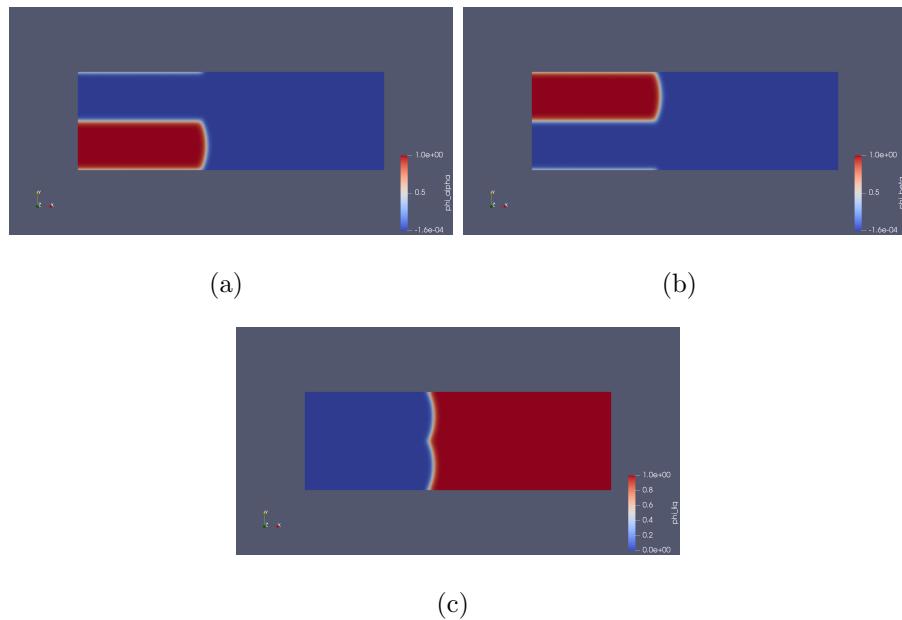


Figure 4.3: Variation of phase fields (a) α (b) β (c)liquid, at steady state

Since for the first 20s, the simulation is run with diffusion in both the liquid and solid to establish well diffused interfaces, the fields at the end of this ‘smoothening’ are shown in Figure 4.4. The μ profile shape and value is consistent with what is expected with diffusion on both sides. Figure 4.5 shows the μ profile with and without smoothening. The need for smoothening becomes clear from Figure 4.5 (b) which clearly shows the non-physical μ variation across the $\alpha - \beta$ interface.

After the 20s, the simulation proceeds with diffusion only in liquid, and all the fields on the bulk solid side are ‘frozen’, due to no diffusion. This can be seen in Figure 4.6. Since we are shifting all the fields opposite to the growth direction, the μ fields take the shape as shown. The part of solid we started with has been shifted so far left that it has left the domain.

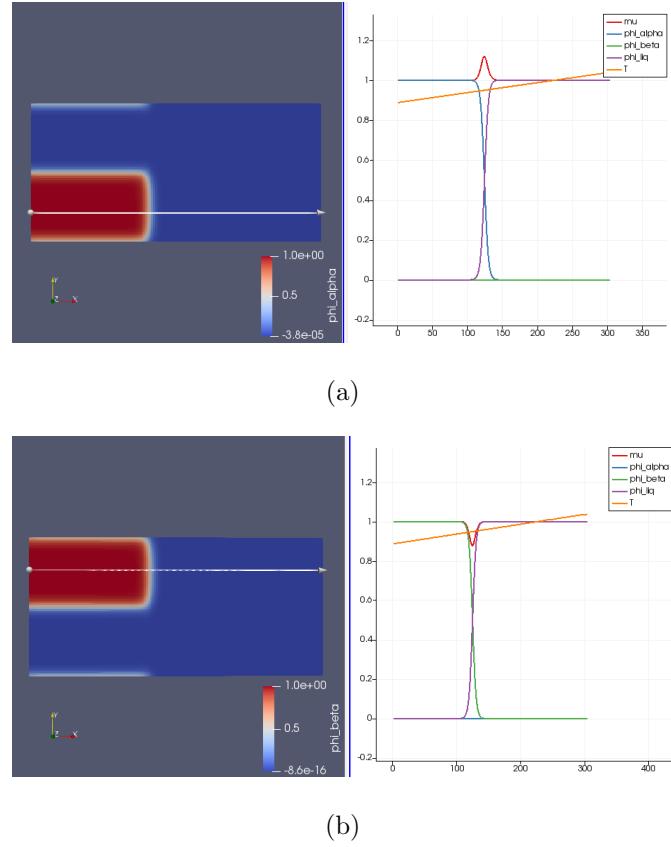


Figure 4.4: After 20s of smoothening, Phase fields (a) α (b) β

4.1.2 Validating with Jackson-Hunt theory

The interface undercooling obtained from the 2d lamella simulations for a range of λ spacing are plotted with the undercoolings obtained by the analytical Jackson-Hunt theory from equation (2.2.1). All the material constants and processing parameters in the analytical expression (2.2.1) must be such that it should match the parameters used in the simulation. The values of liquidus slopes m_α, m_β , the difference in composition $C_{eut}^\alpha - C_{eut}^\beta$, the eutectic temperature T_{eut} are fixed by the thermodynamics used in the phase field simulations and are listed in Table 3.1 Since all the interfacial energies are considered the same, we have $\sigma_\beta = \sigma_\alpha = \sigma$ from the Table 3.1 and also, at the triple point all the angles will have to be of the same value thus $\sin\theta_\beta^{liq} = \sin\theta_\alpha^{liq} = \sin30^\circ = 0.5$ Since in this simulation the volume fraction is chosen to be equal, $S_\alpha = S_\beta = 0.5$ and $\zeta = 1$ The latent heat of fusion is obtained using the relation,

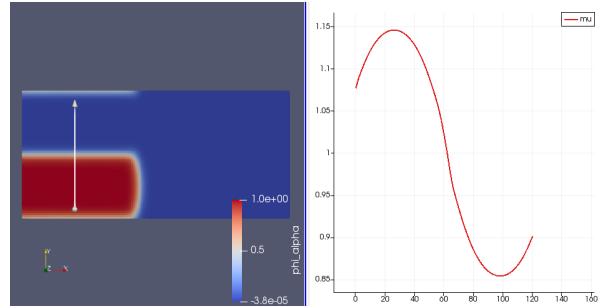
$$\frac{\partial f}{\partial T} = \frac{\partial f}{\partial \mu} \frac{\partial \mu}{\partial T} = -S \quad (4.1.1)$$

where S here, is the entropy of fusion per unit volume, and that

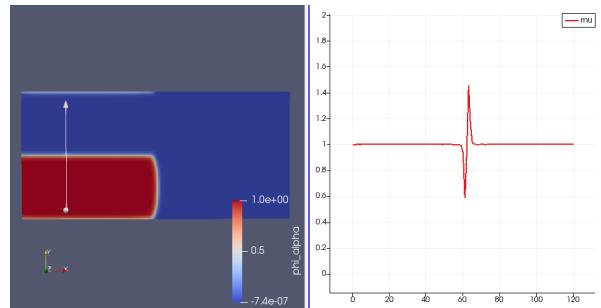
$$L = (\Delta S)T_{eut} \quad (4.1.2)$$

The free energy parabolic coefficients obtained from Equation (3.2.6)-(3.2.9) are used to obtain the value of L for the α and β phases

The comparison is shown in Figure 4.7 for $dx=1$ and $dx=2$. As can be seen the curves show a good match. This match is as good as that obtained from Finite difference based

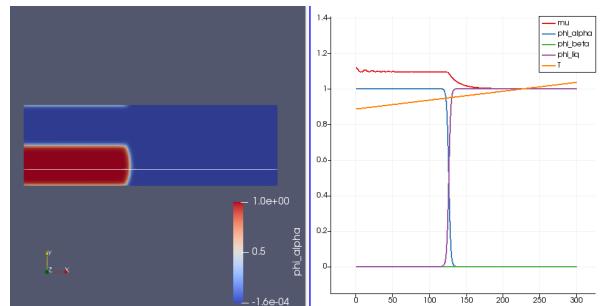


(a)

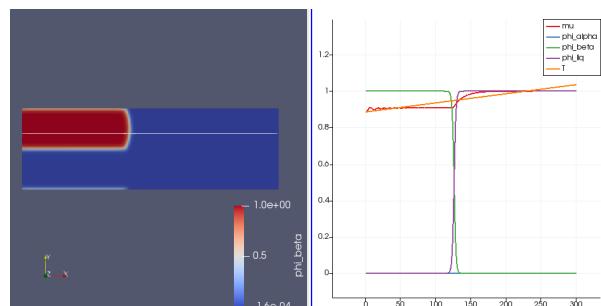


(b)

Figure 4.5: Variation of μ across the $\alpha - \beta$ interface when it is (a)smoothened (b) not smoothened



(a)



(b)

Figure 4.6: Phase fields (a) α (b) β showing the final morphology

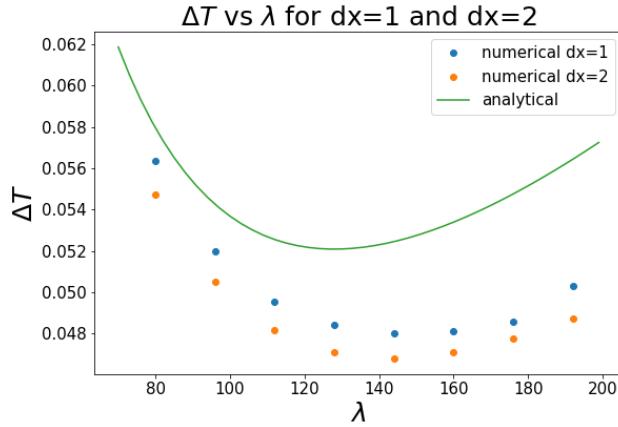


Figure 4.7: Comparison with Analytical JH theory

solvers. The λ_{min} is shifted with respect to analytical curve because JH theory has several assumptions including that of planar interface while considering diffusion gradients in liquid, while there are no assumptions in the phase field models with respect to planar interfaces.

4.2 2D simulations with unequal volume fractions

These simulations were ran with identical parameters with the exception of those listed in Table 3.2. The volume fraction that the equilibrium compositions chosen in Table 3.2 is $\alpha = 70\%$, $\beta = 30\%$. The microstructural evolution as shown in Fig. 4.8 demonstrates that the system tries to achieve its equilibrium volume fraction in spite of starting off with 50-50. The system establishes the 70-30 volume fraction and achieves steady state.

4.3 3D simulations

4.3.1 Lamellar Morphology

The time evolution of the microstructure of the solid-liquid interface is shown in Fig. 4.9. Starting with a random initialization of the phases, the system tries to minimize the $\alpha - \beta$ surface area while keeping volume fraction at 50-50. In doing so, it establishes the lamellar morphology as shown. The λ spacing keeps oscillating as can be seen from Fig. 4.9 (e), (f), (g)

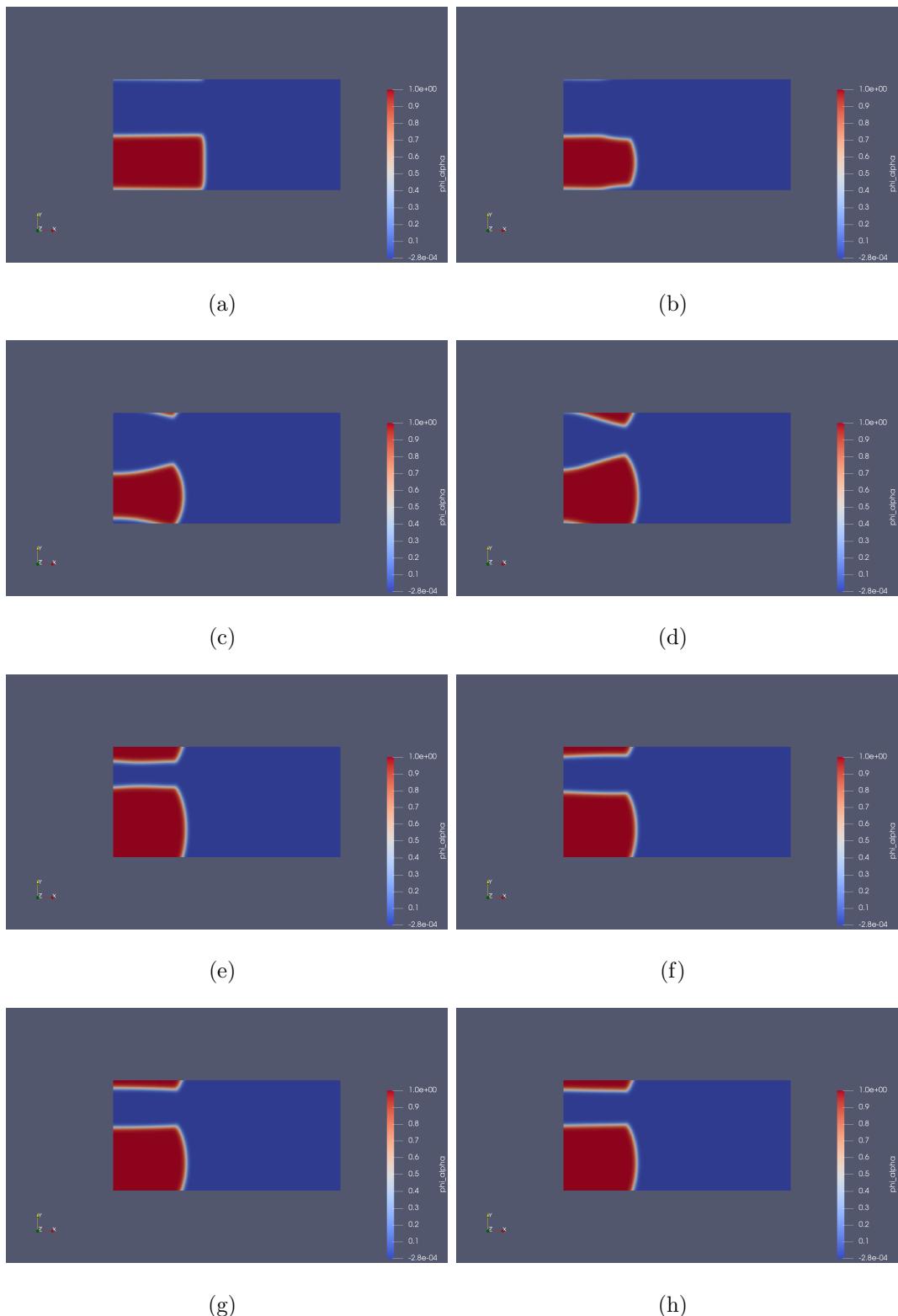


Figure 4.8: α phase fields for the case of unequal volume fractions. The images in sequence show the time evolution of the microstructure.

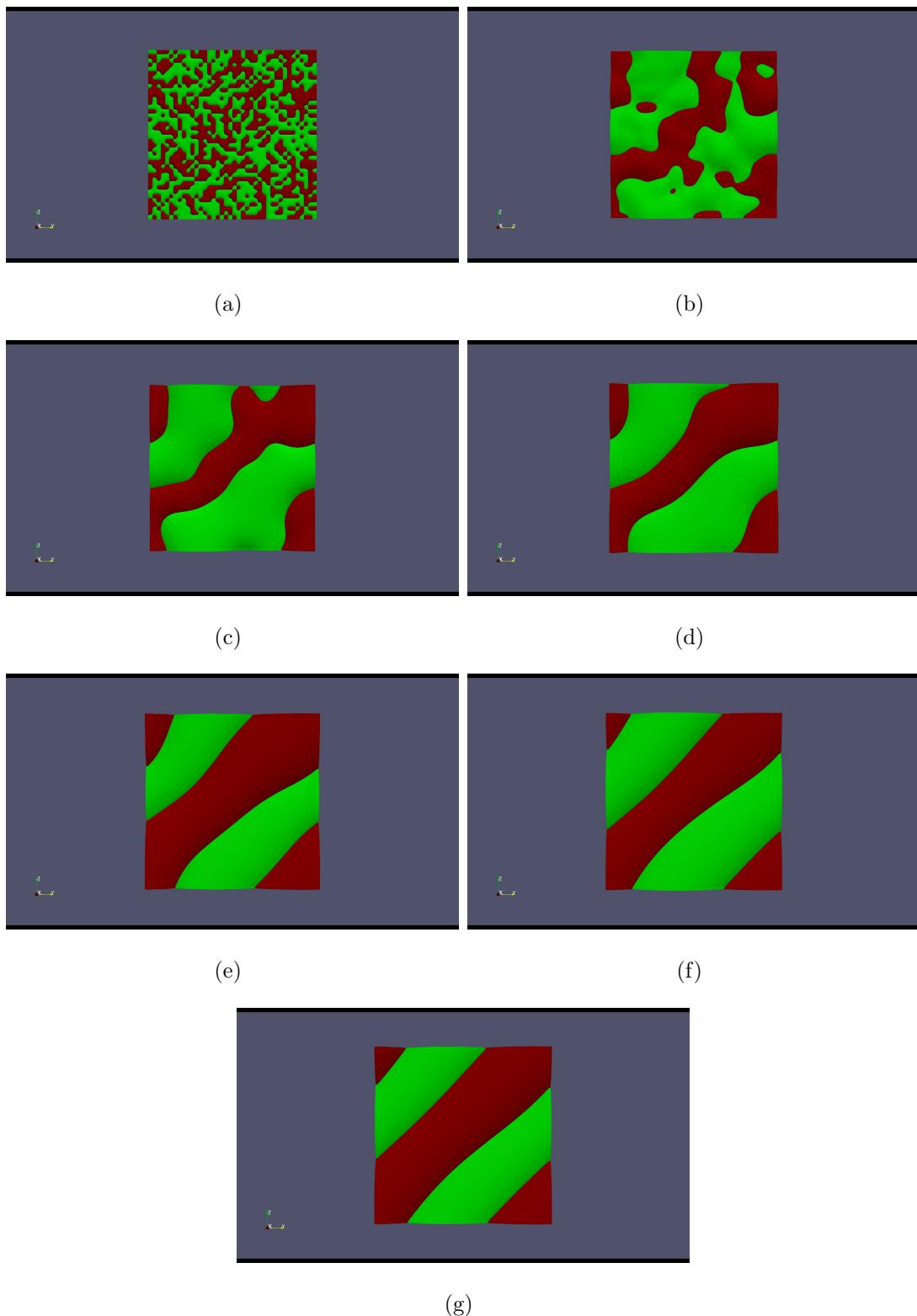


Figure 4.9: Solid-liquid interface showing eutectic lamellar morphology selected by system. Red is the α phase while green is the β phase. The images in sequence show the time evolution of the microstructure.

Chapter 5

Conclusions, Planned work and Future scope

5.1 Conclusion

We conclude that OpenFOAM could be successfully used for running efficient Phase field simulations for eutectic solidification. The implicit Finite Volume solvers in OpenFOAM, although increases cost per iteration, allows for an increased timestep, dt , which is beneficial in running longer simulations. The framework created as a part of this study could be heavily parallelized and thus could be ran on high power computing (HPC) facility such as SAHASRAT Cray XC40 present in SERC, IISc, Bangalore. This enables the use of framework for much larger simulations for studying pattern formation in eutectic systems.

5.2 Planned Work

It was planned to run a variety of 3D simulations to further test the improvements in speed of the framework. Also, running larger simulations on HPCs in SERC could have given a glimpse into study of instabilities in eutectic binary systems.

5.3 Problems Faced

The first problem I faced was the steep learning curve associated with OpenFOAM. With the number of utilities and libraries available, it takes some time to get familiarized with the software. Writing my own solver took fair bit of time since it involved understanding the expression grammar and data types of OpenFOAM. There were also some problems associated with version mismatch.

I became aware of OpenFOAM discussion forums on sites other than the official CFD-online webpage a bit late into the project. These forums on other platforms like Discord

are more active and interactive.

Due to the unprecedented lockdown in city of Bangalore in late April, 2021 work had to be done remotely with the use of remote access software. Though remote access to the lab computers was available, the free remote access software keeps dropping the connection every few minutes. Access to a better commercial remote access software would have proved useful.

5.4 Future scope of work

In addition to performing planned work, the framework developed in OpenFOAM could be extended to multi-component, multi-phase eutectic systems to study the complex pattern formation using phase field model. The phase field model could be made more physically accurate by incorporating thermodynamic data from the available databases, considering and studying the effect of anisotropy in interface energies, etc.

Appendices

Appendix

Phase-Field Solver for Binary Eutectic System

```
dimensionedScalar initial_residual_alpha;
dimensionedScalar initial_residual_beta;
dimensionedScalar initial_residual_liq;

do {
    volVectorField grad_alpha = dimx * fvc::grad(phi_alpha);
    volVectorField grad_beta = dimx * fvc::grad(phi_beta);
    volVectorField grad_liq = dimx * fvc::grad(phi_liq);

    fvScalarMatrix phi_alphaEqn
        (tau * epsilon * dimt * fvm::ddt(phi_alpha) - v * grad_alpha.component(vector::X)
        ==
        (-2.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvm::laplacian(phi_alpha) +
        9.0 * gamma / epsilon * 2.0 * (phi_alpha) * (1.0 - phi_alpha) * (1.0 - 2.0 * phi_alpha)
        (-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * 0.25 * 15 * (
            3.0 * phi_alpha * phi_alpha * phi_alpha * phi_alpha -
            4.0 * phi_alpha * phi_alpha * phi_alpha -
            phi_alpha * phi_alpha + 2.0 * phi_alpha - (phi_beta - phi_liq) * (phi_beta - phi_liq) *
            (2.0 * phi_alpha - 3.0 * phi_alpha * phi_alpha)
        ) +
        (-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * -0.5 * 15 * (
            phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_alpha - phi_liq)
        ) +
        (-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * -0.5 * 15 * (
            phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_alpha - phi_beta)
        )
    ) +
    (1.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_liq) +
```

```

9.0 * gamma / epsilon * 2.0 * (phi_liq) * (1.0 - phi_liq) * (1.0 - 2.0 * phi_liq) +
(-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * -0.5 * 15 * (
(phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_liq - phi_beta)
)

+
(-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * -0.5 * 15 * (
(phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_liq - phi_alpha)
) +
(-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * 0.25 * 15 * (
3.0 * phi_liq * phi_liq * phi_liq * phi_liq -
4.0 * phi_liq * phi_liq * phi_liq - phi_liq * phi_liq +
2.0 * phi_liq - (phi_alpha - phi_beta) * (phi_alpha - phi_beta) *
(2.0 * phi_liq - 3.0 * phi_liq * phi_liq)
))

+
(1.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_beta) +
9.0 * gamma / epsilon * 2.0 * (phi_beta) * (1.0 - phi_beta) * (1.0 - 2.0 * phi_beta) +
(-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * -0.5 * 15 * (
(phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_beta - phi_liq)
) +
(-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * 0.25 * 15 * (
3.0 * phi_beta * phi_beta * phi_beta * phi_beta -
4.0 * phi_beta * phi_beta * phi_beta - phi_beta * phi_beta +
2.0 * phi_beta - (phi_liq - phi_alpha) * (phi_liq - phi_alpha) *
(2.0 * phi_beta - 3.0 * phi_beta * phi_beta)) +
(-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * -0.5 * 15 * (
(phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha)
)
)

);

initial_residual_alpha = phi_alphaEqn.solve().max().initialResidual();

fvScalarMatrix phi_betaEqn
(tau * epsilon * dimt * fvm::ddt(phi_beta) - v * grad_beta.component(vector::X) ==
(-2.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvm::laplacian(phi_beta) +
9.0 * gamma / epsilon * 2.0 * (phi_beta) * (1.0 - phi_beta) * (1.0 - 2.0 * phi_beta) +

```

```

(-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * -0.5 * 15 * (
(phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_beta - phi_liq)
) +
(-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * 0.25 * 15 * (
3.0 * phi_beta * phi_beta * phi_beta * phi_beta -
4.0 * phi_beta * phi_beta * phi_beta - phi_beta * phi_beta +
2.0 * phi_beta - (phi_liq - phi_alpha) * (phi_liq - phi_alpha) *
(2.0 * phi_beta - 3.0 * phi_beta * phi_beta)
) +
(-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * -0.5 * 15 * (
(phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha)
)
) +
(1.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_liq) +
9.0 * gamma / epsilon * 2.0 * (phi_liq) * (1.0 - phi_liq) * (1.0 - 2.0 * phi_liq) +
(-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * -0.5 * 15 * (
(phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_liq - phi_beta)
) +
(-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * -0.5 * 15 * (
(phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_liq - phi_alpha)
) +
(-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * 0.25 * 15 * (
3.0 * phi_liq * phi_liq * phi_liq * phi_liq -
4.0 * phi_liq * phi_liq * phi_liq - phi_liq * phi_liq +
2.0 * phi_liq - (phi_alpha - phi_beta) * (phi_alpha - phi_beta) *
(2.0 * phi_liq - 3.0 * phi_liq * phi_liq)
)
) +
(1.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_alpha) +
9.0 * gamma / epsilon * 2.0 * (phi_alpha) * (1.0 - phi_alpha) * (1.0 - 2.0 * phi_alpha) +
(-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * 0.25 * 15 * (
3.0 * phi_alpha * phi_alpha * phi_alpha * phi_alpha -
4.0 * phi_alpha * phi_alpha * phi_alpha - phi_alpha * phi_alpha +
2.0 * phi_alpha - (phi_beta - phi_liq) * (phi_beta - phi_liq) *
(2.0 * phi_alpha - 3.0 * phi_alpha * phi_alpha)
)
)

```

```

(-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * -0.5 * 15 * (
(phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_alpha - phi_liq)
) +
(-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * -0.5 * 15 * (
(phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_alpha - phi_beta)
));
initial_residual_beta = phi_betaEqn.solve().max().initialResidual();

fvScalarMatrix phi_liqEqn
(tau * epsilon * dimt * fvm::ddt(phi_liq) - v * grad_liq.component(vector::X) ==
(-2.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvm::laplacian(phi_liq) +
9.0 * gamma / epsilon * 2.0 * (phi_liq) * (1.0 - phi_liq) * (1.0 - 2.0 * phi_liq) +
(-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * -0.5 * 15 * (
(phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_liq - phi_beta)
) +
(-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * -0.5 * 15 * (
(phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_liq - phi_alpha)
) +
(-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * 0.25 * 15 * (
3.0 * phi_liq * phi_liq * phi_liq * phi_liq -
4.0 * phi_liq * phi_liq * phi_liq - phi_liq * phi_liq +
2.0 * phi_liq - (phi_alpha - phi_beta) * (phi_alpha - phi_beta) *
(2.0 * phi_liq - 3.0 * phi_liq * phi_liq)
)
) +
(1.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_alpha) +
9.0 * gamma / epsilon * 2.0 * (phi_alpha) * (1.0 - phi_alpha) * (1.0 - 2.0 * phi_alpha) +
(-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * 0.25 * 15 * (
3.0 * phi_alpha * phi_alpha * phi_alpha * phi_alpha -
4.0 * phi_alpha * phi_alpha * phi_alpha -
phi_alpha * phi_alpha + 2.0 * phi_alpha - (phi_beta - phi_liq) * (phi_beta - phi_liq) *
(2.0 * phi_alpha - 3.0 * phi_alpha * phi_alpha)
)
) +
(-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * -0.5 * 15 * (
(phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_alpha - phi_liq)
) +
(-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * -0.5 * 15 * (
(phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_alpha - phi_beta)
)

```

```

)) +
(1.0 / 3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_beta) +
9.0 * gamma / epsilon * 2.0 * (phi_beta) * (1.0 - phi_beta) * (1.0 - 2.0 * phi_beta) +
(-1.0 / 4.0 * (mu - B_alpha) * (mu - B_alpha) / A + D_alpha) * -0.5 * 15 * (
(phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_beta - phi_liq)
) +
(-1.0 / 4.0 * (mu - B_beta) * (mu - B_beta) / A + D_beta) * 0.25 * 15 * (
3.0 * phi_beta * phi_beta * phi_beta * phi_beta -
4.0 * phi_beta * phi_beta * phi_beta - phi_beta * phi_beta +
2.0 * phi_beta - (phi_liq - phi_alpha) * (phi_liq - phi_alpha) *
(2.0 * phi_beta - 3.0 * phi_beta * phi_beta)
) +
(-1.0 / 4.0 * (mu - B_liq) * (mu - B_liq) / A + D_liq) * -0.5 * 15 * (
(phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha)
)
)
);
initial_residual_liq = phi_liqEqn.solve().max().initialResidual();
} while (
initial_residual_alpha.value() > Tol ||
initial_residual_beta.value() > Tol ||
initial_residual_liq.value() > Tol
);
Info << "Tol value = " << Tol << endl;

volVectorField grad_c = dimx * fvc::grad(((mu - B_alpha) / (2 * A)) *
(
0.25 * phi_alpha * phi_alpha *
(
15.0 * (1 - phi_alpha) * (1.0 + phi_alpha - (phi_beta - phi_liq) * (phi_beta - phi_liq)) *
phi_alpha * (9.0 * phi_alpha * phi_alpha - 5.0)
)
)
) +
((mu - B_beta) / (2 * A)) *
(
0.25 * phi_beta * phi_beta *
(
15.0 * (1 - phi_beta) * (1.0 + phi_beta - (phi_alpha - phi_liq) * (phi_alpha - phi_liq)) *
phi_beta * (9.0 * phi_beta * phi_beta - 5.0)
)
)
)
```

```

)
) +
((mu - B_liq) / (2 * A)) *
(
0.25 * phi_liq * phi_liq *
(
15.0 * (1 - phi_liq) * (1.0 + phi_liq - (phi_alpha - phi_beta) * (phi_alpha - phi_beta))
phi_liq * (9.0 * phi_liq * phi_liq - 5.0)
)
));
volVectorField n_alpha = -1 * dimx * fvc::grad(phi_alpha) /
(1E-20 + mag(dimx * fvc::grad(phi_alpha)));
volVectorField n_beta = -1 * dimx * fvc::grad(phi_beta) /
(1E-20 + mag(dimx * fvc::grad(phi_beta)));
volVectorField n_liq = -1 * dimx * fvc::grad(phi_liq) /
(1E-20 + mag(dimx * fvc::grad(phi_liq)));

fvScalarMatrix muEqn
(
0.5 * 1.0 / A * (
0.25 * phi_alpha * phi_alpha *
(
15.0 * (1 - phi_alpha) * (1.0 + phi_alpha - (phi_beta - phi_liq) * (phi_beta - phi_liq))
phi_alpha * (9.0 * phi_alpha * phi_alpha - 5.0)
) +
0.25 * phi_liq * phi_liq *
(
15.0 * (1 - phi_liq) * (1.0 + phi_liq - (phi_alpha - phi_beta) * (phi_alpha - phi_beta))
phi_liq * (9.0 * phi_liq * phi_liq - 5.0)
) +
0.25 * phi_beta * phi_beta *
(
15.0 * (1 - phi_beta) * (1.0 + phi_beta - (phi_alpha - phi_liq) * (phi_alpha - phi_liq))
phi_beta * (9.0 * phi_beta * phi_beta - 5.0)
)
) * dimt * fvm::ddt(mu) - v * grad_c.component(vector::X) +
((mu - B_alpha) / (2 * A)) * (
dimt * fvc::ddt(phi_alpha) * 0.25 * 15 * (

```

```

3.0 * phi_alpha * phi_alpha * phi_alpha * phi_alpha -
4.0 * phi_alpha * phi_alpha * phi_alpha -
phi_alpha * phi_alpha + 2.0 * phi_alpha - (phi_beta - phi_liq) * (phi_beta - phi_liq) *
(2.0 * phi_alpha - 3.0 * phi_alpha * phi_alpha)

) +
dimt * fvc::ddt(phi_beta) * -0.5 * 15 * (
(phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_beta - phi_liq)
) +
dimt * fvc::ddt(phi_liq) * -0.5 * 15 * (
(phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_liq - phi_beta)
)
) +
((mu - B_beta) / (2 * A)) * (
dimt * fvc::ddt(phi_alpha) * -0.5 * 15 * (
(phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_alpha - phi_liq)
) +
dimt * fvc::ddt(phi_beta) * 0.25 * 15 * (
3.0 * phi_beta * phi_beta * phi_beta * phi_beta -
4.0 * phi_beta * phi_beta * phi_beta - phi_beta * phi_beta +
2.0 * phi_beta - (phi_liq - phi_alpha) * (phi_liq - phi_alpha) *
(2.0 * phi_beta - 3.0 * phi_beta * phi_beta)
) +
dimt * fvc::ddt(phi_liq) * -0.5 * 15 * (
(phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_liq - phi_alpha)
)
) +
((mu - B_liq) / (2 * A)) * (
dimt * fvc::ddt(phi_alpha) * -0.5 * 15 * (
(phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_alpha - phi_beta)
) +
dimt * fvc::ddt(phi_beta) * -0.5 * 15 * (
(phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha)
) +
dimt * fvc::ddt(phi_liq) * 0.25 * 15 * (
3.0 * phi_liq * phi_liq * phi_liq * phi_liq -
4.0 * phi_liq * phi_liq * phi_liq - phi_liq * phi_liq +
2.0 * phi_liq - (phi_alpha - phi_beta) * (phi_alpha - phi_beta) *

```

```
(2.0 * phi_liq - 3.0 * phi_liq * phi_liq)
)
) ==
0.5 * dimx * dimx * fvm::laplacian(D * phi_liq, mu) +
0.5 * 0.707 * epsilon * (c_eq_liq - c_eq_alpha) *
dimx * fvc::div(n_alpha & n_liq * dimt * fvc::ddt(phi_alpha) * -1 * n_alpha) +
0.5 * 0.707 * epsilon * (c_eq_liq - c_eq_beta) *
dimx * fvc::div(n_beta & n_liq * dimt * fvc::ddt(phi_beta) * -1 * n_beta)
);

muEqn.solve();
```

Bibliography

- [1] William J Boettinger, James A Warren, Christoph Beckermann, and Alain Karma. Phase-field simulation of solidification. *Annual review of materials research*, 32(1):163–194, 2002.
- [2] Long-Qing Chen and AG Khachaturyan. Computer simulation of structural transformations during precipitation of an ordered intermetallic phase. *Acta metallurgica et materialia*, 39(11):2533–2551, 1991.
- [3] Long-Qing Chen and Wei Yang. Computer simulation of the domain dynamics of a quenched system with a large number of nonconserved order parameters: The grain-growth kinetics. *Physical Review B*, 50(21):15752, 1994.
- [4] Y Wang and AG Khachaturyan. Three-dimensional field model and computer modeling of martensitic transformations. *Acta materialia*, 45(2):759–773, 1997.
- [5] V Vaithyanathan and LQ Chen. Coarsening of ordered intermetallic precipitates with coherency stress. *Acta Materialia*, 50(16):4061–4073, 2002.
- [6] Openfoam v8 user guide.
- [7] Shanmukha Kiran Aramanda, Sumeet Khanna, Sai Kiran Salapaka, Kamanio Chat-topadhyay, and Abhik Choudhury. Crystallographic and morphological evidence of solid–solid interfacial energy anisotropy in the sn-zn eutectic system. *Metallurgical and Materials Transactions A*, 51(12):6387–6405, 2020.
- [8] Philipp Steinmetz, Johannes Hötzer, Michael Kellner, Amber Genau, and Britta Nestler. Study of pattern selection in 3d phase-field simulations during the directional solidification of ternary eutectic al-ag-cu. *Computational Materials Science*, 148:131–140, 2018.

- [9] KA Jackson and JD Hunt. Lamellar and rod eutectic growth. In *Dynamics of Curved Fronts*, pages 363–376. Elsevier, 1988.
- [10] Alain Karma and Armand Sarkissian. Morphological instabilities of lamellar eutectics. *Metallurgical and Materials Transactions A*, 27(3):635–656, 1996.
- [11] Silvère Akamatsu and Mathis Plapp. Eutectic and peritectic solidification patterns. *Current Opinion in Solid State and Materials Science*, 20(1):46–54, 2016.
- [12] James S Langer. Directions in condensed matter physics. *Models of Pattern Formation in First-order Phase Transition*, 165, 1986.
- [13] Gunduz Caginalp and Weiqing Xie. Phase-field and sharp-interface alloy models. *Physical Review E*, 48(3):1897, 1993.
- [14] R Folch and M Plapp. Quantitative phase-field modeling of two-phase growth. *Physical Review E*, 72(1):011602, 2005.
- [15] Bjorn Stinner, Britta Nestler, and Harald Garcke. A diffuse interface model for alloys with multiple components and phases. *SIAM Journal on Applied Mathematics*, 64(3):775–799, 2004.
- [16] Abhik Choudhury and Britta Nestler. Grand-potential formulation for multicomponent phase transformations combined with thin-interface asymptotics of the double-obstacle potential. *Physical Review E*, 85(2):021602, 2012.
- [17] James Ahrens, Berk Geveci, and Charles Law. Paraview: An end-user tool for large data visualization. *The visualization handbook*, 717(8), 2005.
- [18] John D Hunter. Matplotlib: A 2d graphics environment. *IEEE Annals of the History of Computing*, 9(03):90–95, 2007.