

OpenFOAM for solving material science problems

A Thesis

Submitted for the Degree of
Master of Engineering
in the **Faculty of Engineering**

by

Ravi Kumar Singh



Department of Materials Engineering
Indian Institute of Science
Bangalore – 560 012 (INDIA)

Jun 2019

DEDICATED TO

my Parents and Teachers

Signature of the Author:

Ravi Kumar Singh
Dept. of Materials Engineering
Indian Institute of Science, Bangalore

Signature of the Thesis Supervisor:

Dr. Abhik Choudhury
Assistant Professor
Dept. of Materials Engineering
Indian Institute of Science, Bangalore

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance. I would first like to thank my supervisor, Dr. Abhik N Choudhury, whose expertise was invaluable in the formulating of the research topic and methodology in particular.

I would also like to thank Prof. T.A. Abinandanan the Chairman of Department of Materials Engineering, for allowing me to access all the available facilities. I would like to thank SERC, IISc for providing computational resources.

I would like to acknowledge my colleagues for their extension of support. You supported me greatly and were always willing to help me. I would particularly like to thank Kiran for helping me in conducting the experiments and Bhalachandra , Fiyanshu , Sumeet for helping in developing codes, I want to thank you for your excellent cooperation and for all of the opportunities I was given to conduct my research and further my dissertation at IISc Bangalore.

I would also like to thank Yadvendra, Nihal and Jitin for their valuable guidance. You provided me with the tools that I needed to choose the right direction and successfully complete my dissertation.I would like to thank all my department friends with whom I could enjoy activities like Cultural fests, Holi and Diwali celebration and institute open days.

In addition, I would like to thank my parents for their wise counsel and sympathetic ear. You are always there for me. Finally, there are my friends, who were of great support in deliberating over our problems and findings, as well as providing happy distraction to rest my mind outside of my research.

Contents

Acknowledgements	i
Contents	ii
List of Figures	v
List of Tables	viii
1 Directional solidification	1
1.1 Background	1
1.2 Research goal	2
1.3 Outline	2
1.4 Physical phenomenon in Bridgman furnace	2
1.5 Governing equations and implementation in OpenFOAM	3
1.5.1 Equations in fluid region	3
1.5.2 Equations in solid region	4
1.6 Boundary conditions	5
1.6.1 Velocity	5
1.6.2 Temperature	5
1.7 Furnace design in OpenFOAM	6
1.8 Experimental validation	7
2 Additive manufacturing	10
2.1 Background	10
2.2 Research goal	11
2.3 Outline	11
2.4 Physical model	11
2.5 Solid-Liquid model	12

2.5.1	Governing equations	12
2.5.2	Momentum source and sink	12
2.5.3	Energy source term	13
2.5.4	Surface heat source	13
2.5.5	Stress analysis equations	13
2.6	Implementation in OpenFOAM	14
2.7	Numerical test cases	18
2.7.1	Problem description	18
2.7.2	Simulation setting	19
2.7.3	Boundary conditions	20
2.7.4	Mesh sensitivity analysis	21
2.7.5	Laser source and melt pool geometry	21
2.7.6	Test case for stress analysis	27
3	Phase-Field modeling of equilibrium precipitate shapes under the influence of coherency stresses	37
3.1	Background	37
3.2	Research goal	38
3.3	Outline	38
3.4	Physical model	39
3.4.1	Elastic energy interpolation	40
3.4.2	Conservation of volume	41
3.4.3	Mechanical equilibrium	42
3.5	Implementation in OpenFoam	42
3.6	Numerical test case	47
3.6.1	Model parameters	47
3.6.2	Cubic anisotropy in elastic energy with dilatational misfit	48
3.6.3	Anisotropy in misfit strain	50
4	Dendrite tip selection:LGK theory and phase-field simulations	52
4.1	Background	52
4.2	Research goals	54
4.3	Outline	54
4.4	Physical model	54
4.5	Implementation in OpenFOAM	58

4.5.1	Analytical code in Matlab	60
4.6	Numerical test case based on Phase-field model	61
5	Conclusions	70
	Appendices	72
A		73
	Bibliography	79

List of Figures

1.1	(a)Furnace drawing (b)Sample holder.	2
1.2	Furnace design in OpenFOAM	6
1.3	Temperature Profile inside Furnace without Quartz tube	7
1.4	Furnace thermal profile andtemperature profile inside fluid along shown line.	8
1.5	SEM images at different pulling velocities longitudinal section (upper) and transverse section (lower)	9
1.6	SEM images(a) Velocity 0 to $1.2\mu\text{m/sec}$ (b)Velocity 1.2 to $1.4\mu\text{m/sec}$	9
2.1	2D section of melt pool in solid-liquid as well as solid-liquid-gas model.[4]	11
2.2	Initial grid distribution.	19
2.3	Final grid distribution.	20
2.4	Liquid interface resolution at grid size 50x10x20	22
2.5	Liquid interface resolution at grid size 75x15x30	22
2.6	Liquid interface resolution at grid size 100x20x40	22
2.7	Liquid interface resolution at grid size 125x25x50	23
2.8	Heat source profile at initial stage with temperature distribution along line.	23
2.9	Heat source profile at later stage with temperature distribution along line	24
2.10	(a)A cut section showing heat affected zone near the laser source . (b)Topview of velocity profile inside melt pool. (c) A cut section showing melt pool shape. .	24
2.11	Liquid phase contour corresponding to liquid fraction of 0.5(a)X normal (b)Y normal (c) Z normal (d) X normal surface	25
2.12	Velocity profile in melt pool (a)Zero surafce tension gradient (b)Velocity streamlines.	26
2.13	Velocity profile in melt pool (a)Negative surafce tension gradient (b)Velocity streamlines.	26
2.14	Velocity profile in melt pool (a)Positive surafce tension gradient (b)Velocity streamlines.	27

2.15	Temperature profile along laser scan direction.	29
2.16	Total displacement field magnitude.	29
2.17	Displacement field x component.	30
2.18	Displacement field y component.	30
2.19	Displacement field z component.	31
2.20	sqrtJ2 magnitude.	31
2.21	Longitudinal stress (σ_{xx}).	32
2.22	Transverse stress (σ_{yy}).	32
2.23	Through thickness stress (σ_{zz}).	33
2.24	Variation of σ_{xx} in z direction.	33
2.25	Variation of σ_{yy} in z direction.	34
2.26	Variation of σ_{zz} in z direction.	34
2.27	Shear stress (σ_{xy}).	35
2.28	Shear stress (σ_{xz}).	35
2.29	Shear stress (σ_{yz}).	36
3.1	Az=3.0, R=30, 2D section normal to X axis(a)Initial configuration (b) Final configuration (c)3D equilibrium shape.	49
3.2	Az=3.0, R=50, 2D section normal to X axis(a)Initial configuration (b) Final configuration (c) 3D equilibrium shape	50
3.3	2D section normal to X axis, R=55(a)Initial configuration(b) Final configuration.	51
3.4	2D section normal to Z axis,R=55(a)Initial configuration(b) Final configuration.(c) 3D equilibrium shape	51
4.1	Final profile for $\Delta T=0.065$ (a)Order parameter, α (b) Chemical potential, μ	62
4.2	Final profile for $\Delta T=0.065$ (a) α contour at 0.5 (b) Parabolic fit at dendrite tip.	62
4.3	Final profile for $\Delta T=0.065$ (a)Order parameter, α (b) Chemical potential, μ	63
4.4	Final profile for $\Delta T=0.07$ (a) α contour at 0.5 (b) Parabolic fit at dendrite tip.	63
4.5	Final profile for $\Delta T=0.075$ (a)Order parameter, α (b) Chemical potential, μ	64
4.6	Final profile for $\Delta T=0.075$ (a) α contour at 0.5 (b) Parabolic fit at dendrite tip.	64
4.7	Profile at T=1 (a)Order parameter, α (b) Chemical potential, μ	65
4.8	Profile at T=3 (a)Order parameter, α (b) Chemical potential, μ	66
4.9	Profile at T=5 (a)Order parameter, α (b) Chemical potential, μ	66
4.10	Profile at T=10 (a)Order parameter, α (b) Chemical potential, μ	67
4.11	Profile at T=15 (a)Order parameter, α (b) Chemical potential, μ	67
4.12	Profile at T=20 (a)Order parameter, α (b) Chemical potential, μ	68

4.13 Profile at T=25 (a)Order parameter, α (b) Chemical potential, μ .	68
---	----

List of Tables

2.1	Thermophysical properties used in test case	19
2.2	Thermophysical Properties of Ti-6Al-4V. [13]	27
2.3	Temperature dependent mechanical properties of Ti-6Al-4V. [13]	28
2.4	Temperature dependent yield strength of Ti-6Al-4V. [13]	28
4.1	Parameters used in this simulation	61
4.2	Comparison of dendrite tip radius using different methods	65

Chapter 1

Directional solidification

1.1 Background

Directional Solidification (DS) is one of the most sought after technologies used in automotive, aircraft, aerospace, energy industries. Most of the DS experimental work is carried out in Bridgeman type furnace. The factors which controls the solidification process in this furnace are rate of withdrawal or in simple term ,the pulling velocity of furnace or sample and thermal gradient. Both of these factors are responsible for the shape and size distribution of grains and the final properties of products. As the interface velocity matches, that of the imposed temperature gradient which is positive going from the solid to the liquid, the mean relative position of the interface with respect to the furnace does not change. This setup is normally referred to as a Bridgeman furnace [1]. Too high withdrawal velocity will lead to formation of dendrites while too low velocity gives rise to coarser grains and also results in lower productivity. A lot of work has been carried out both experimental as well as numerical model to simulate the process of Bridgeman DS process [2]. For experimental work, a modified Bridgeman type DS furnace has been used which comprises of a hot zone, a chill zone separated by an gradient zone and water cooling facility at sample holder. The imposed thermal gradient can be varied by varying the heater and chiller temperature. The pulling velocity is controlled by stepper motor. In this work heat transfer model is built to study the solidification process. The simulation is done using OpenFOAM (Open Source Field Operation and Manipulation) and experimental validation has been done on Sn-Zn eutectic.

1.2 Research goal

The main aim of this project is to find out the response of imposed thermal gradient on the temperature gradient G at the solidification front and thus the solidification rate above which unstable growth (i.e: dendrite) is observed.

1.3 Outline

In section 1.4, the physical model describing the fluid flow and heat transfer in a Bridgman furnace is presented. In section 1.8 the results of the validation cases for the solver and experimental results are discussed.

1.4 Physical phenomenon in Bridgman furnace

It is a multiregion multi physics modelling. Figure 1.1(a) gives an overview of problem modelled in this research. The quartz tube sits on the sample holder. The sample holder has a provision of inlet and outlet for flow of water to cool the quartz tube from below. In multiregion multi physics separate governing equations for each continuum/region (solid,liquid) are solved. The boundary condition includes multiregion coupling at the interface between two regions.

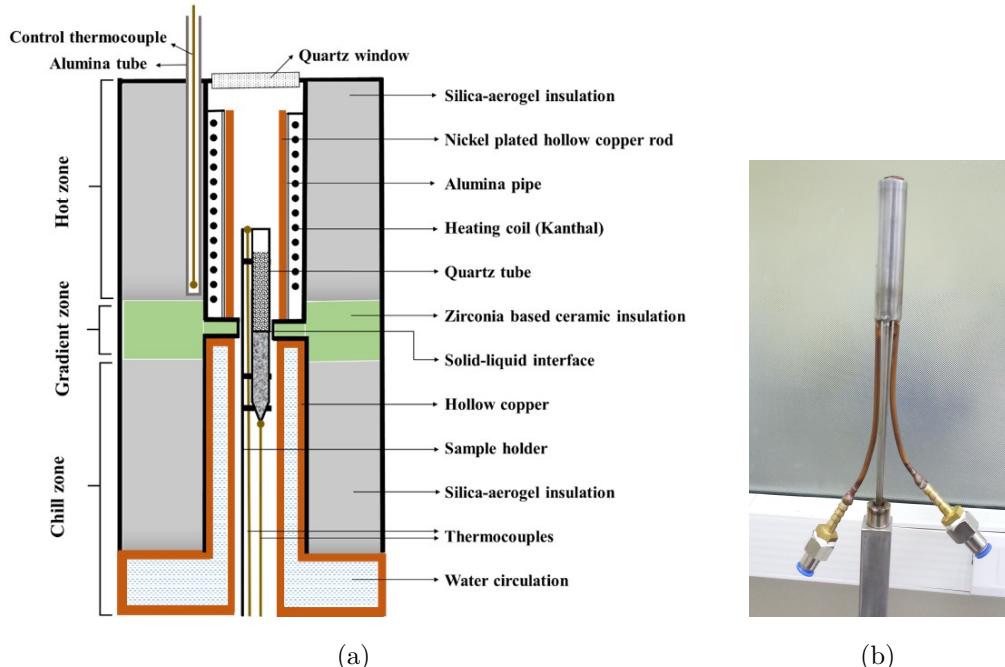


Figure 1.1: (a)Furnace drawing (b)Sample holder.

1.5 Governing equations and implementation in Open-FOAM

Depending upon the regions whether it is a solid or fluid the according equations for the solid or fluid is solved.

1.5.1 Equations in fluid region

For each fluid region the following continuum equations of the conservation of mass, momentum and energy is solved.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0 \quad (1.1)$$

For an incompressible fluid ρ is constant. Thus, the above equation changes to

$$\nabla \cdot (U) = 0 \quad (1.2)$$

The momentum equation

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) = -\nabla p + \nabla \cdot (\mu \nabla U) \quad (1.3)$$

is implemented in OpenFOAM as

```
tmp<fvVectorMatrix>tUEqn
(
    fvm::ddt(rho, U) + fvm::div(phi, U)
    + turbulence.divDevRhoReff(U)
    ==
    fvOptions(rho, U)
);
fvVectorMatrix& UEqn = tUEqn.ref();

UEqn.relax();

fvOptions.constrain(UEqn);
```

```

if (momentumPredictor)
{
    solve
    (
        UEqn
        ==
        fvc::reconstruct
        (
            (
                - ghf*fvc::snGrad(rho)
                - fvc::snGrad(p_rgh)
            )*mesh.magSf()
        ),
        mesh.solver(U.select(finalIter))
    );
    fvOptions.correct(U);
}

```

The energy equation

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho U h) = \nabla \cdot (\alpha \nabla h) \quad (1.4)$$

is implemented in OpenFOAM as

```

fvScalarMatrix EEqn
(
    fvm::ddt(rho, he) + fvm::div(phi, he)
    -fvm::laplacian(turbulence->alphaEff(), he)
    ==
    fvOptions(rho, he)
);

```

1.5.2 Equations in solid region

For solid region only energy equation is solved. Steady-state conduction of heat in solid is defined by the Fourier's Heat equation. The energy equation states that the temporal change of enthalpy

of the solid is equal to the divergence of the heat conducted through the solid.

$$\frac{\partial \rho h}{\partial t} = \nabla \cdot (\alpha \nabla h) \quad (1.5)$$

$h=c_p T$ is the specific enthalpy, ρ the density and $\alpha = \kappa/c_p$ is the thermal diffusivity which is defined as the ratio between the thermal conductivity κ and the specific heat capacity c_p , μ is dynamic viscosity, U is fluid velocity, p is pressure.

OpenFOAM Code:

```
tmp<fvScalarMatrix>hEqn
(
    fvm::ddt(rho, h)
    - (
        thermo.isotropic()
        ? fvm::laplacian(thermo.alpha(), h "laplacian(alpha,h)")
        : fvm::laplacian(taniAlpha(), h, "laplacian(alpha,h)")
    )
    ==
    fvOptions(rho, h)
);
```

1.6 Boundary conditions

To specify the boundary conditions, Figure 1.1(a) is used. All the boundaries in this model are solid boundaries (top,bottom,left and right sides).Velocity and temperature are defined on these boundaries.

1.6.1 Velocity

No slip boundary condition is imposed on all the boundaries.

1.6.2 Temperature

The temperature boundary condition can be of fixed value type or of gradient type. Adiabatic boundary condition is imposed on top and bottom sides as well as left and right sides in gradient zone of furnace. In the hot zone fixed temperature boundary condition is imposed. The chill

zone has a gradient boundary condition given by following equation:

$$\kappa \frac{\partial T}{\partial x} = h_{conv}(T - T_{ref}) \quad (1.6)$$

h_{conv} is heat transfer coefficient, κ is thermal conductivity and T_{ref} is outside temperature.

1.7 Furnace design in OpenFOAM

The main aim for modeling this furnace (Figure 1.2) is to obtain the thermal gradient at solid-liquid interface. The material under investigation is put inside a closed quartz tube which in turn is kept inside furnace. However, it is not possible to put a thermocouple inside quartz tube to get the temperature profile. Thus, we need a model which can best represent the furnace and obtain the gradient at solidification front. In this model we have not used latent heat of melting,

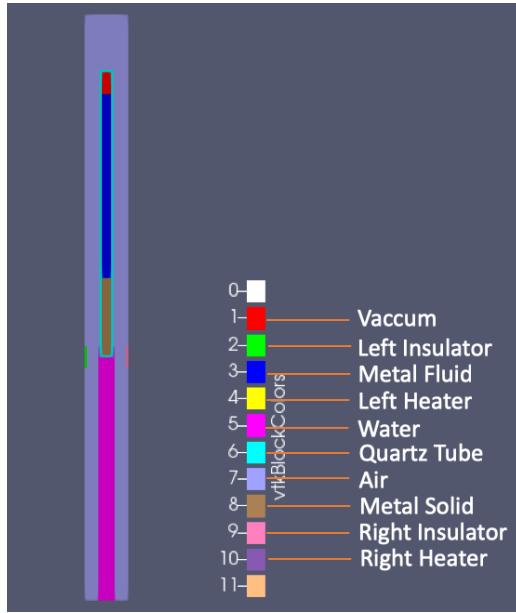


Figure 1.2: Furnace design in OpenFOAM

but we have iterated the solution with respect to the melting point of solid. This is done by changing the length of solid level inside quartz tube and then checking the temperature at that level after each simulation. If it matches with the melting point of solid, then that length is used for analysis. Otherwise, the solid length inside quartz tube is increased or decreased, if the temperature is lower or higher respectively with respect to melting temperature.

1.8 Experimental validation

An initial thermal gradient without quartz tube is obtained by passing thermocouple in the furnace. The data obtained is then used to verify the model. Figure 1.3 shows the thermal profile obtained in experiment with respect to the modelled one. As we can clearly see that the maximum gradient matches approximately in both the cases.

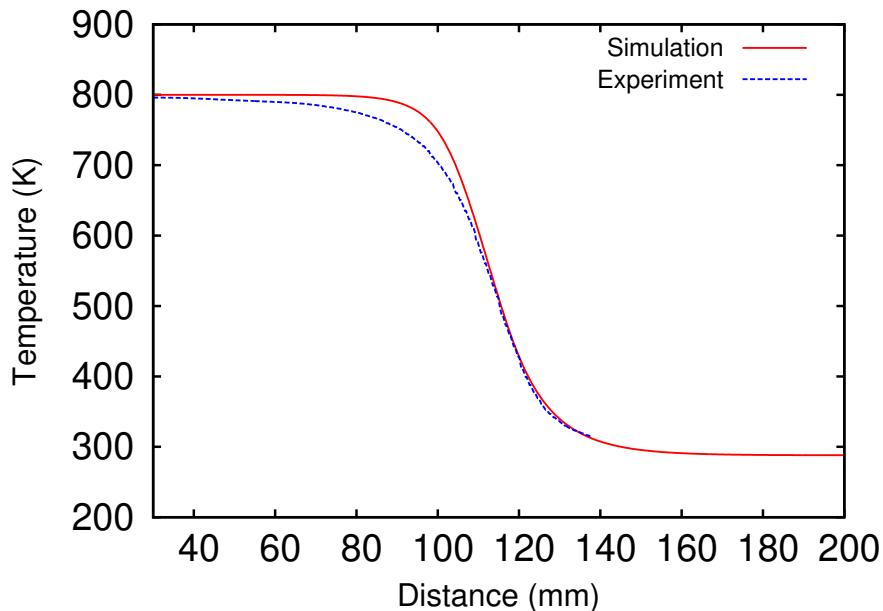


Figure 1.3: Temperature Profile inside Furnace without Quartz tube

The next set of tests includes the quartz tube filled with Sn-Zn (4% off eutectic) alloy. The simulation ran till it reached steady state .The thermal profile in different regions (solid,fluid) of furnace is observed using Paraview(a visualisation software). The thermal gradient just ahead of solidification front is measured (Figure) and the value obtained is 5.8K/mm.

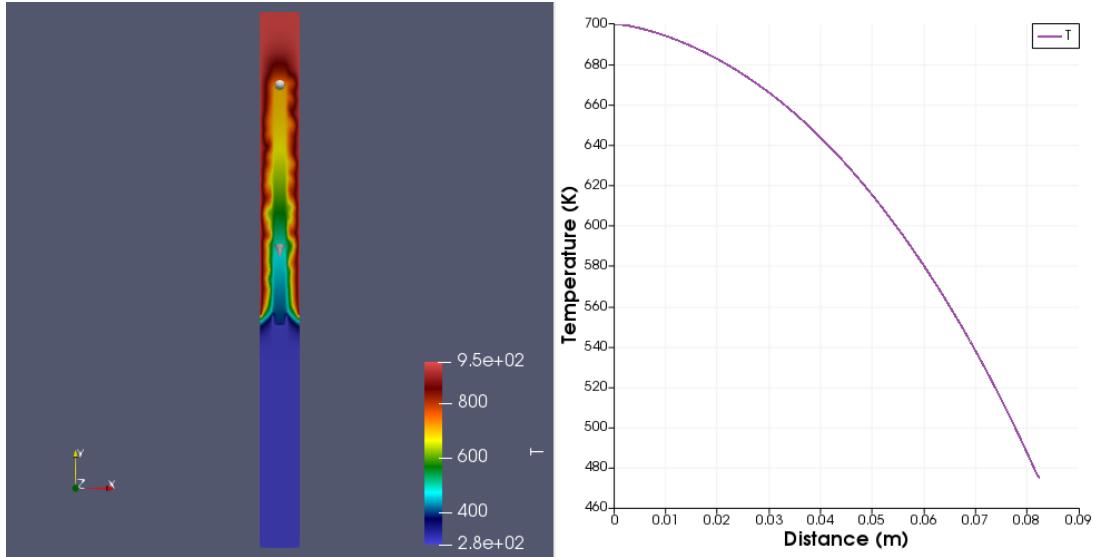


Figure 1.4: Furnace thermal profile and temperature profile inside fluid along shown line.

The thermal gradient obtained is then used for calculating the critical pulling velocity according to given equation [3]:

$$\frac{v(C_e - C_o)(1 - k)}{-DG} < \frac{1}{m} \quad (1.7)$$

where v is sample pulling velocity, C_e is eutectic composition, C_o is initial composition in melt, $k = \frac{C_s}{C_l}$ is partition coefficient, C_s and C_l are the compositions in the solid and the liquid, D is diffusivity, G is thermal gradient ahead of solidification front and m is liquidus slope.

The critical velocity obtained after putting all relevant values in above equation is $1.3\mu\text{m/sec}$. Experiments are carried out at a velocity of $1\mu\text{m/sec}$ and $2\mu\text{m/sec}$. SEM images are taken for analysis (Figure 1.5). It is clearly evident from the SEM images that there is no interface instability at $1\mu\text{m/sec}$, but instability is observed at $2\mu\text{m/sec}$ in form of dendrites.

To obtain more accurate results further experiments are carried out at $1.2\mu\text{m/sec}$ and $1.4\mu\text{m/sec}$. In this case too, a stable interface is observed in SEM image of former while an unstable interface in case of later Figure 1.6(a) and 1.6(b). Thus, we can clearly infer that the model is able to predict the thermal gradient at solidification front and thus the critical velocity with huge accuracy.

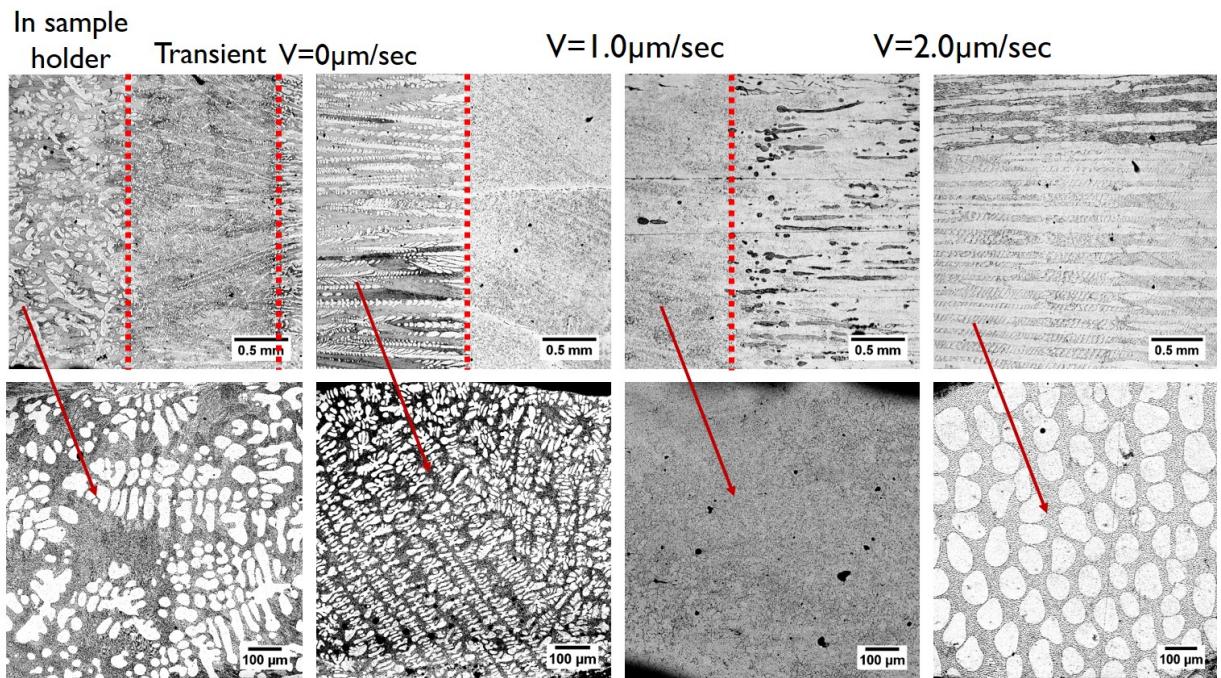


Figure 1.5: SEM images at different pulling velocities longitudinal section (upper) and transverse section (lower)

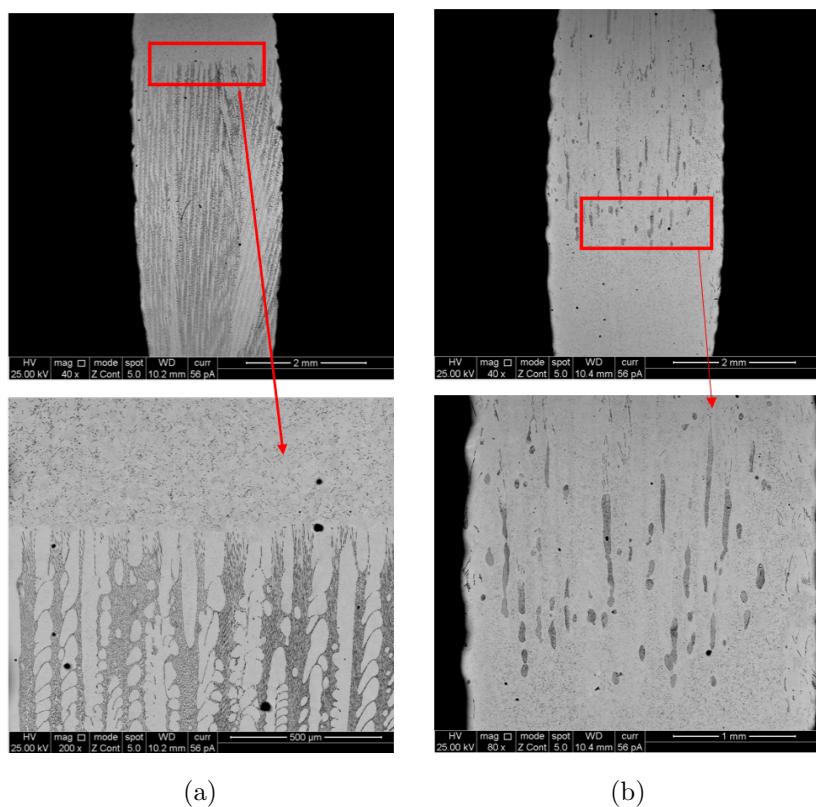


Figure 1.6: SEM images(a) Velocity 0 to $1.2\mu\text{m/sec}$ (b)Velocity 1.2 to $1.4\mu\text{m/sec}$.

Chapter 2

Additive manufacturing

2.1 Background

Additive manufacturing as its name suggest is an additive process rather than subtracting. There is no or very less wastage of materials in this process as compared to milling, forming, rolling etc. In its initial phase of development this was known as rapid prototyping, to check form, fit and functionality. With passage of time plastic processing technology known as Stereolithography attracted designers and engineers. In this process UV light sensitive liquid polymers were solidified using Lasers. This was the first Additive Manufacturing systems commercialisation in the world. In early 1990s Selective Laser Sintering (SLS) was introduced , a process that fuses powder materials using a laser.In early 2000s, Metal Deposition (DMD) laser-cladding systems was introduced, a process that produces and repairs parts using metal powder.

Additive Manufacturing Systems can be classified by the energy source or the way the material is being joined, for example using a binder, laser, heated nozzle etc. Classification is also possible by the group of materials being processed, such as plastics, metals or ceramics. In order to meet the requirements in modern era, there is a continuous need to develop improved techniques. These traits can only be guaranteed through continuous improvement of quality, mainly by advancing the methodologies that allow for tailoring of the melt pool composition, structure, and properties.The properties of melt pool is generally dependent on hydrodynamics, which is predominantly driven by Marangoni convection due to surface tension gradients. It plays an important role in the energy transfer to the surrounding HAZ(Heat Affected Zone) .The energy transfer affects the final melt pool shape, residual stresses and solidified microstructure which in turn affect the mechanical strength, resistance to corrosion, fatigue life, etc. Therefore, it is better to know how melt pool hydrodynamics influences heat and mass transfer in order to

yield high-quality products.

A large number of experiments as well as numerical simulations have been carried out over last decades. In general, all researchers face three major hurdles: high temperatures, the small dimensions and the lack of transparency of the melt pool. These obstacles can be overcome by numerical studies using Computational Fluid Dynamics (CFD). CFD methods serve the purpose of detailed analysis for heat transfer and fluid flow in the melt pool and the HAZ.

2.2 Research goal

The main aim of this project is to find out the melt pool geometry, thermal profile, residual stress and distortion in sample with a moving heat source. Analysis is to be carried out with varying laser power, scan speed, hatch patterns and heat source geometry.

2.3 Outline

In section 3.4, the physical model describing the fluid flow, heat transfer and stress analysis along with its implementation in OpenFOAM is discussed. In section 2.7, the simulation results of the solver is discussed.

2.4 Physical model

The problem of liquid metal dynamics involves solid and liquid phase. The solid-liquid interface is captured by solving energy conservation equation coupled with fluid fraction field.

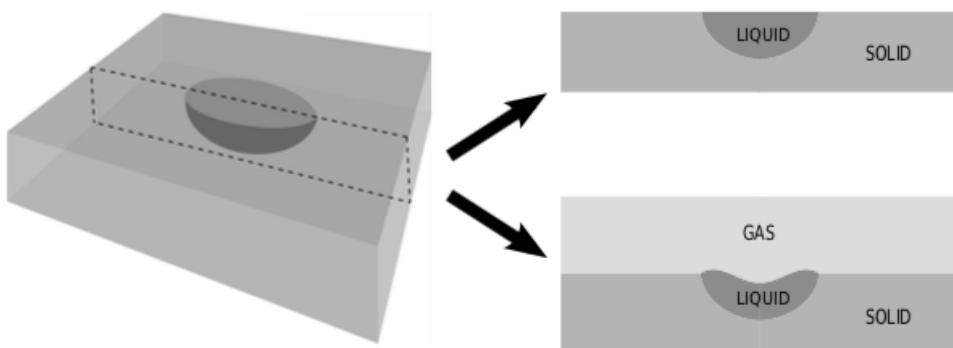


Figure 2.1: 2D section of melt pool in solid-liquid as well as solid-liquid-gas model.[4]

2.5 Solid-Liquid model

2.5.1 Governing equations

The model is based on the following continuum equations of the conservation of mass, momentum, and energy [5]. The governing equations are following

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0 \quad (2.1)$$

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) = -\nabla p + \nabla \cdot (\mu \nabla U) + S_u \quad (2.2)$$

$$\frac{\partial \rho C_p T}{\partial t} + \nabla \cdot (\rho U C_p T) = \nabla \cdot (k \nabla T) + S_T \quad (2.3)$$

Here, ρ is the fluid density, U the velocity vector, t time, p pressure, μ dynamic viscosity, C_p specific heat constant, T temperature, and k thermal conductivity. Additionally, S_u and S_T are momentum and energy source terms, respectively.

2.5.2 Momentum source and sink

S_u caters for two effects [6]. Firstly, a density difference due to temperature difference, giving rise to buoyancy force in the direction of the gravity vector. Secondly, the velocity in the solid has to be 0, thus requires a momentum sink which decelerates the flow in the solid. Accordingly, S_u reads

$$S_u = S_b + S_d = \rho g \beta (T - T_{ref}) + A U \quad (2.4)$$

where

$$A = -\frac{C (1 - f_l)}{f_l^3 + e_0} \quad (2.5)$$

The first term (the thermal buoyancy) uses Boussinesq approximation, where g is gravity, and β the thermal expansion coefficient. The second term in the right-hand side represents the frictional dissipation of momentum in the mushy zone according to the Carman-Kozeny equation for flow through porous media [7] [8], f_l is the liquid fraction. The multiplier C is a constant and represents mushy zone morphology, which is chosen to be large enough ($= 10^6$), to force the velocity in the solid zone to be zero whereas e_0 is a small constant ($= 10^{-3}$), to avoid the division by zero when the liquid fraction f_l is zero in the solid.

2.5.3 Energy source term

The energy source S_T takes into account, the evolution of the latent heat during phase change. Latent heat changes from zero in the solid to the maximum value that equals the phase change latent heat L in the liquid. Implementation of S_T makes use of the source term linearization technique [9], where the non-linearities are accounted for through iterations.

$$S_T = - \left[\frac{\partial \rho f_l L}{\partial t} + \nabla \cdot (\rho \mathbf{U} f_l L) \right] \quad (2.6)$$

where f_l is the liquid fraction and L the phase change latent heat.

2.5.4 Surface heat source

The heat influx is given by a moving Gaussian laser source:

$$q_{in} = \frac{2P\eta}{\pi b^2} \exp - \left[\frac{2(x - x' - vt)^2}{b^2} + \frac{2(z - z')^2}{b^2} \right] \quad (2.7)$$

where P is laser power, η is process efficiency, v is laser scan velocity, b is heat source radius.

2.5.5 Stress analysis equations

The temperature field obtained from solving above equations is then used for thermo-mechanical analysis. In this case the simplest perfect plasticity is assumed. Perfect plasticity is defined as below a maximum level of stress, deformation will be elastic in nature. The solid will deform irreversibly without any increase in stresses or loads once maximum stress is reached. Considering, a solid body, the momentum balance states following equation:

$$\frac{\partial^2(\rho u)}{\partial t^2} - \nabla \cdot \sigma = 0 \quad (2.8)$$

where u is the solid displacement vector, ρ is the density, and σ is the stress tensor.

$$\sigma = 2\mu\varepsilon_e + \lambda tr(\varepsilon_e)I \quad (2.9)$$

where I is the unit tensor, μ and λ are material properties called Lame's parameters. And, the strain tensor ε is defined in terms of u :

$$\varepsilon = 1/2[\nabla u + (\nabla u)^T] \quad (2.10)$$

$$\varepsilon = \varepsilon_e + \varepsilon_p + \varepsilon_{th} \quad (2.11)$$

where ε_e is elastic, ε_p is plastic and $\varepsilon_{th} = \beta\Delta T$ is thermal strain. β is thermal expansion coefficient.

Most of the materials undergo non-linear deformation. After adding plastic and thermal features, we have solved steady state governing equations in incremental form.

$$\nabla \cdot \{\mu \nabla(du) + \mu [\nabla(du)]^T + \lambda I tr[\nabla(du)] - [2\mu(d\varepsilon_p) + \lambda I tr(d\varepsilon_p)] - \varepsilon_{th}\} = 0 \quad (2.12)$$

where du is the incremental displacement vector, and $d\varepsilon_p$ stands for the incremental plastic strain tensor, which is stress-level dependent: when the stress is below the limit level, only elasticity occurs, $d\varepsilon_p = 0$; while the limit stress level is reached, plasticity occurs, $d\varepsilon_p$ shall then be computed from the stress σ .

This equation is split into two terms explicit and implicit.

$$\begin{aligned} \nabla \cdot [(2\mu + \lambda) \nabla(du)] &= - \nabla \cdot \{\mu [\nabla(du)]^T + \lambda I tr[\nabla(du)] - (\mu + \lambda) \nabla(du)\} \\ &\quad + \nabla \cdot [2\mu(d\varepsilon_p) + \lambda I tr(d\varepsilon_p)] + \nabla \cdot \varepsilon_{th} \end{aligned} \quad (2.13)$$

2.6 Implementation in OpenFOAM

The momentum conservation equation using Eqn. 2.2 2.4 , mathematically expressed as:

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) = - \nabla p + \nabla \cdot (\mu \nabla U) + \rho g \beta (T - T_{ref}) + AU \quad (2.14)$$

is implemented in OpenFOAM as

```

surfaceScalarField muf = twoPhaseProperties.muf();
fvVectorMatrix UEqn
(
    fvm::ddt(rho, U) + fvm::div(rhoPhi, U)
    - fvm::laplacian(muf, U)
    - (fvc::grad(U) & fvc::grad(muf))
    + fvm::Sp(C*pow(1-g1, 2)/(pow(g1, 3)+q), U)
    ==
    fvOptions(rho, U)
);
UEqn.relax();
fvOptions.constrain(UEqn);
if (pimple.momentumPredictor())
{
    solve
    (
        UEqn
        ==
        fvc::reconstruct
        (
            fvc::interpolate(rho*(1- beta*(T-TRef)))*(g&mesh.Sf())
            +
            -fvc::snGrad(p)
            )*mesh.magSf()
        )
    );
    fvOptions.correct(U);
}

```

The energy conservation equation using Eqn. 2.3–2.6 is mathematically expressed as:

$$\frac{\partial \rho C_p T}{\partial t} + \nabla \cdot (\rho U C_p T) = \nabla \cdot (k \nabla T) - \left[\frac{\partial \rho f_l L}{\partial t} + \nabla \cdot (\rho \mathbf{U} f_l L) \right] \quad (2.15)$$

is implemented in OpenFOAM as

```

for (int i=0; i<10; i++)
{
surfaceScalarField kappaaf = mixture.kappaaf();
i++;
fvScalarMatrix TEqn
(
    fvm::ddt(rhoCp, T)
    +fvm::div(rhoCpPhi, T)
    -fvm::laplacian(kappaaf, T)
    ==
    -(L*rho)*(fvc::div(phi, gl)+(fvc::ddt(gl)))
);
TEqn.relax();
fvOptions.constrain(TEqn);
TEqn.solve();
fvOptions.correct(T);
volScalarField glNew = gl + glRelax*(T-((Tl-Ts)*gl+Ts))*Cp/L;
gl = max(scalar(0), min(glNew, scalar(1)));
}
Info<< "Min/maxT:" << min(T()).value() << ''
<<max(T()).value()<< endl;

```

The momentum balance equation for solid body without any body force, mathematically expressed as:

$$\nabla \cdot [(2\mu + \lambda) \nabla (du)] = - \nabla \cdot \{ \mu [\nabla (du)]^T + \lambda I \operatorname{tr}[\nabla (du)] - (\mu + \lambda) \nabla (du) \} + \nabla \cdot [2\mu(d\varepsilon_p) + \lambda I \operatorname{tr}(d\varepsilon_p)] + \nabla \cdot \varepsilon_{th} \quad (2.16)$$

is implemented in OpenFOAM as

```

muE=muE1*Foam::max((69.5-0.0535*T*sig3)*1e9,4e9);
lambdaE=lambdaE1*Foam::max((104-0.08*T*sig3)*1e9,6e9);

```

```

depsEigen=(1-alpha)*fvc::ddt(T)*runtime.time().deltaT()
           *Foam::min((4.22e-9*T*sig3+7.65e-6),12.5e-6)*threeKalpha;

do
(
{
fvVectorMatrix dDEqn
(
fvm::laplacian((2*muE +lambdaE), dD, "laplacian(dDdD,dD)")
==
-(1-alpha)*fvc::div
(
muE*graddD.T() + lambdaE*(I*tr(graddD))
-(muE + lambdaE)*graddD, "div(sigmaExp)"
)
+fvc::div
(
(1-alpha)*(2*muE*depsP+lambdaE(I*tr(depsP))), "div(sigmaP)"
)
+fvc::div
(
2*muE*depsEigen + lambdaE*I*tr(depsEigen), "div(sigmaEigen)"
)
);
initialResidual = dDEqn.solve().max().initialResidual();
graddD = fvc::grad(dD);
)while(initialResidual>convergenceTolerance && ++iCorr<nCorr);

D = D*(1-alpha) + dD;
scalar constant1=1e-12;
graddD = fvc::grad(dD);
volSymmTensorField sigmaOld = (1-alpha)*sigmaD;
sigmaD=sigmaOld+(1-alpha)*(2*muE*(symm(graddD)-depsEigen-depsP)
+lambdaE*I*(tr(graddD)-tr(depsEigen)-tr(depsP)));
sqrtJ2=sqrt ((1.0/2.0)*magSqr(dev(sigmaD)));

```

```

volScalarField ky4=Foam::max(ky*(-0.5843*T*sig3+1160)*1e6 ,ky3);
volScalarField fac=sqrtJ2/(ky2);
forAll(fac,cellI)
(
    if((fac[cellI])>1.0)
    (
        sigmaD[cellI]=1/3*I*tr(sigmaD[cellI])
            +dev(sigmaD[cellI])/fac[cellI];
        symmTensor dsigma=sigmaD[cellI]-sigmaOld[cellI];
        symmTensor depsE=1/3*I*tr(dsigma)
            /(3*lambdaE[cellI]+2*muE[cellI]+constant1)
            +dev(dsigma)/(2*muE[cellI]+constant1);
        symmTensor deps=symm(graddD[cellI]);
        depsP[cellI] +=deps - depsE - depsEigen[cellI];
    )
    else
    (
        depsP[cellI] =symmTensor::zero;
    )
)
)

```

2.7 Numerical test cases

The OpenFOAM comes with all basic packages. It is very flexible to apply different physical models depending upon the requirements. In the first case, only Eqn 2.1 2.3 and 2.3 has been solved. This test case deals only with solid-liquid region (flat surface assumption). The simulations initially are used to check the heat source geometry and mesh sensitivity on temperature distribution. Later on solid-liquid flat surface is taken into account.

2.7.1 Problem description

The sample geometry is a cuboid of 5mm x 1mm x 2mm dimension. The sample was subjected to a laser beam of $150 \mu\text{m}$ radius and a moving velocity of 0.95m/sec with a laser powers of 290W. The absorptivity of the sample surface to the laser is 0.2. The following thermophysical property has been used for the test case.

Properties	Values	Units
Density(Solid and Liquid phase)	8100	kg/m^3
Melting Temperature	1723	K
Kinematic Viscosity of Solid	2.97E-07	$\text{kg}/(\text{m.s})$
Kinematic Viscosity of Liquid	2.97E-07	$\text{kg}/(\text{m.s})$
Thermal Conductivity of solid	20	$\text{J}/(\text{m.s.K})$
Thermal Conductivity of liquid	30	$\text{J}/(\text{m.s.K})$
Latent heat of melting	2.6E+05	$\text{J}/(\text{kg})$
Specific heat capacity of solid	450	$\text{J}/(\text{kg.K})$
Specific heat capacity of liquid	800	$\text{J}/(\text{kg.K})$

Table 2.1: Thermophysical properties used in test case

2.7.2 Simulation setting

A mesh of resolution (50x10x20) is initially generated as shown in Figure 2.2 giving a cell size of 0.1mm in all the directions. To avoid computational misuses a very high resolution is not used from initial step. Static refining of mesh is done along the laser movement direction. This method generates 3 stages of refining. The first or the outermost region is the coarsest while the innermost (near laser source) region is finest. A comparison of initial and final resolution can be visualised from Figure 2.2 and 2.3.

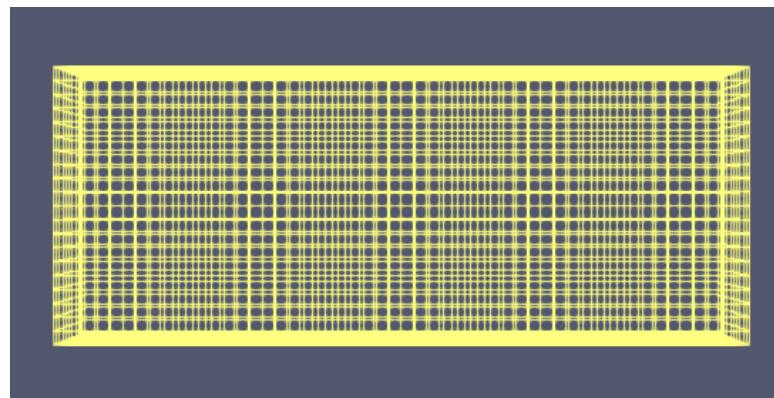


Figure 2.2: Initial grid distribution.

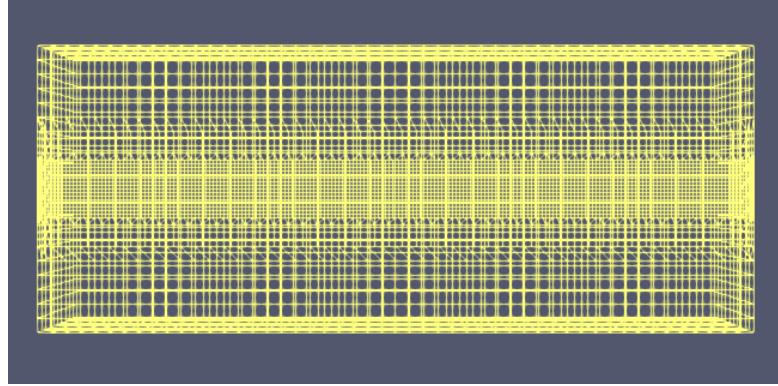


Figure 2.3: Final grid distribution.

2.7.3 Boundary conditions

Temperature

The heat influx is incorporated as a boundary condition. The flux distribution is assumed to be Gaussian. Thus, the thermal boundary condition can be written at top surface is given as:

$$\kappa \frac{\partial T}{\partial \eta} = q_{in} - q_{out} \quad (2.17)$$

where κ is thermal conductivity, q_{in} is influx due to laser source (Eqn 2.7) and q_{out} is heat loss due to radiation, convection and evaporation. At other boundaries adiabatic boundary condition is imposed.

Velocity

Marangoni force [10] [11] is responsible for convective flow in melt pool. Surface tension varies on the top surface of the melt pool resulting from the spatial gradient of temperature. The surface tension gradient can be of two types. If surface tension varies due to concentration , the Marangoni effect is referred to as the solutocapillary effect. In cases where temperature drives the variation in surface tension, the Marangoni effect is referred to as the thermocapillary effect. The Marangoni number is the nondimensional number that gives the ratio between the thermocapillary effect and the viscous forces. The tangential force driven by the surface tension gradient is balanced by the viscous force of the flow due to shear, i.e.

$$\mu \frac{\partial u}{\partial \eta} = \frac{\partial \sigma}{\partial T} \frac{\partial T}{\partial \tau} \quad (2.18)$$

where η is normal and τ the tangential vectors to the boundary plane. The temperature gradient of surface tension $\frac{\partial \sigma}{\partial T}$ is a function of temperature and surface active element concentration. In

this study dependence on solute concentration is neglected.

The tractionDisplacement Boudary [12]

Fixed traction force (T) can be expressed as : $T=\sigma \cdot n$, where n is surface normal at boundary. For an incremental displacement,

$$dT = d\sigma \cdot n. \quad (2.19)$$

Also we have incremental displacement(du) as a variable, thus we need a boundary condition on du .

$$\begin{aligned} d\sigma = & [(2\mu + \lambda)\nabla(du)] + \{\mu[\nabla(du)]^T + \lambda Itr[\nabla(du)] - (\mu + \lambda)\nabla(du)\} \\ & - [2\mu(d\varepsilon_p) + \lambda Itr(d\varepsilon_p)] - [2\mu(d\varepsilon_{th}) + \lambda Itr(d\varepsilon_{th})] \end{aligned} \quad (2.20)$$

Using Eqn 2.19 and 2.20

$$\begin{aligned} (\nabla du).n = & \frac{1}{2\mu + \lambda}(dT - \{\mu[\nabla(du)]^T + \lambda Itr[\nabla(du)] - (\mu + \lambda)\nabla(du)\} \\ & - [2\mu(d\varepsilon_p) + \lambda Itr(d\varepsilon_p)] - [2\mu(d\varepsilon_{th}) + \lambda Itr(d\varepsilon_{th})]).n \end{aligned} \quad (2.21)$$

2.7.4 Mesh sensitivity analysis

Four different mesh resolutions are studied to find the sensitivity of grid size on simulation results. The maximum temperature and interface resolution in all these cases are observed by keeping all other laser and sample properties same. It is clearly evident that increasing the mesh resolution beyond (100x20x40) is not giving significant change in interface resolution and maximum temperature value. Thus, we concluded that a (100x20x40) initial mesh gives a mesh independent solution and this mesh resolution is used for further simulations.

2.7.5 Laser source and melt pool geometry

A laser heat source given by Eqn 2.7 is used in simulation. Thermophysical properties given in Table 2.7.1 is used. After running the simulation, the following results are obtained. The result presented is just initial phase of this project. A detailed analysis as well as validation is still going to be carried out. This test case is just to optimise the solver.

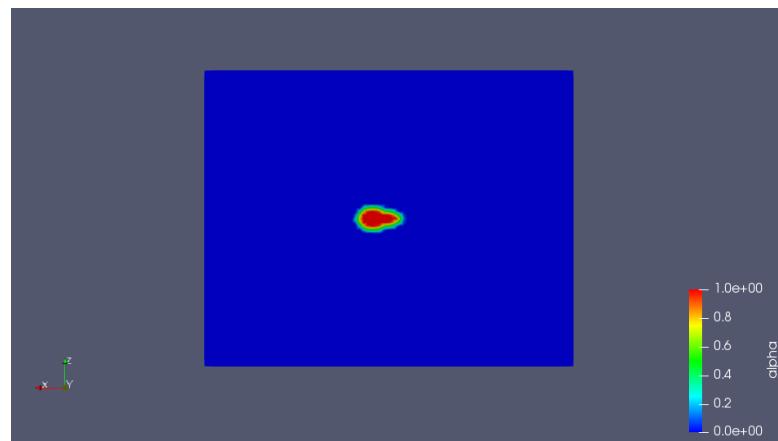


Figure 2.4: Liquid interface resolution at grid size $50 \times 10 \times 20$.

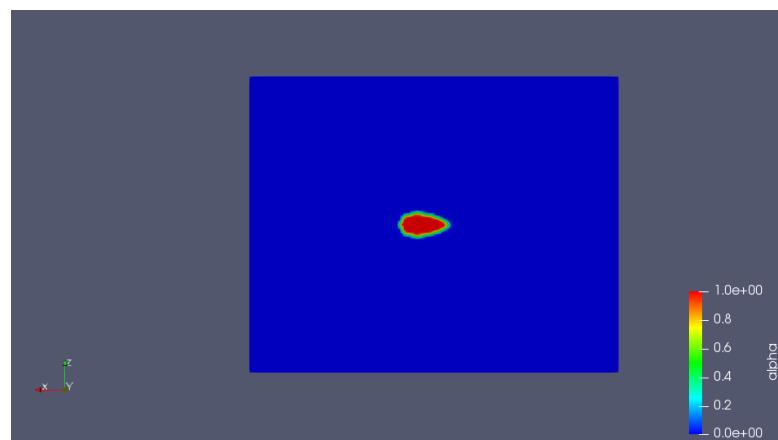


Figure 2.5: Liquid interface resolution at grid size $75 \times 15 \times 30$.

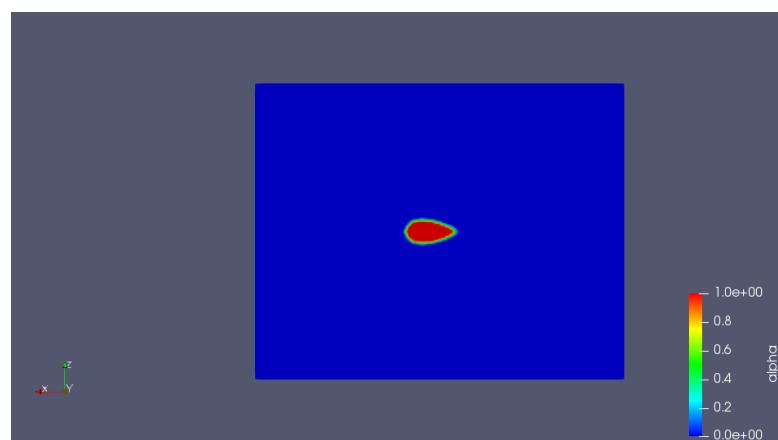


Figure 2.6: Liquid interface resolution at grid size $100 \times 20 \times 40$.

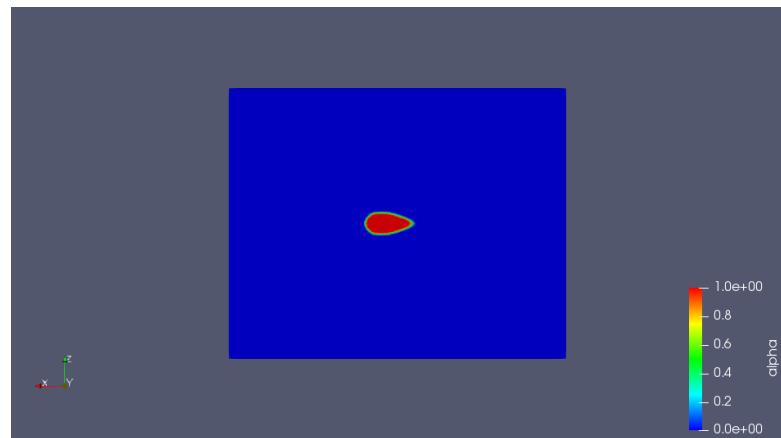


Figure 2.7: Liquid interface resolution at grid size $125 \times 25 \times 50$.

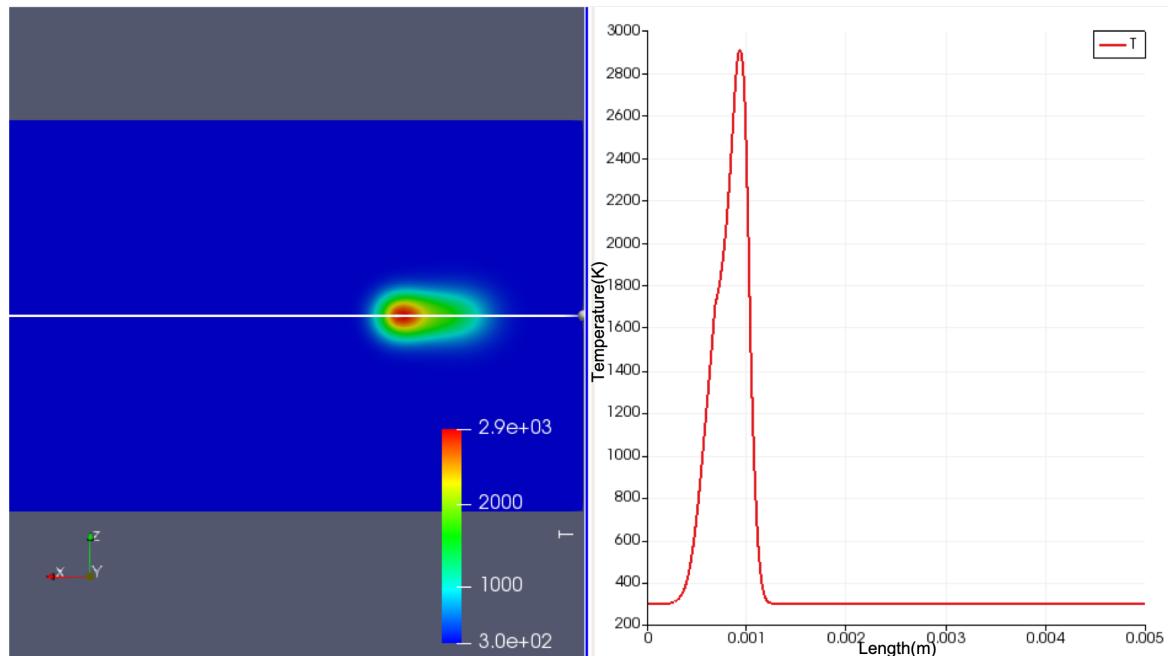


Figure 2.8: Heat source profile at initial stage with temperature distribution along line.

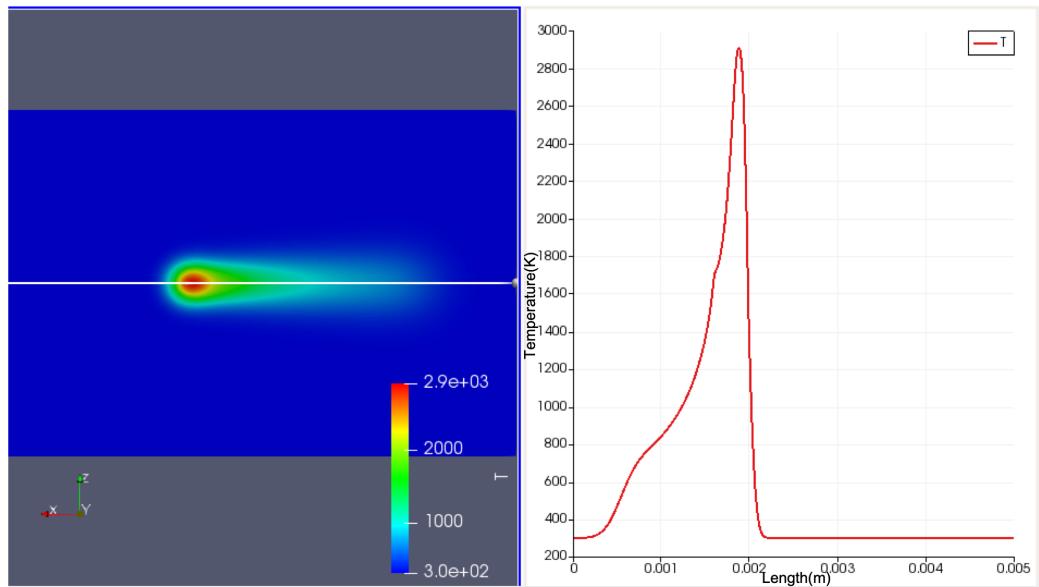


Figure 2.9: Heat source profile at later stage with temperature distribution along line .

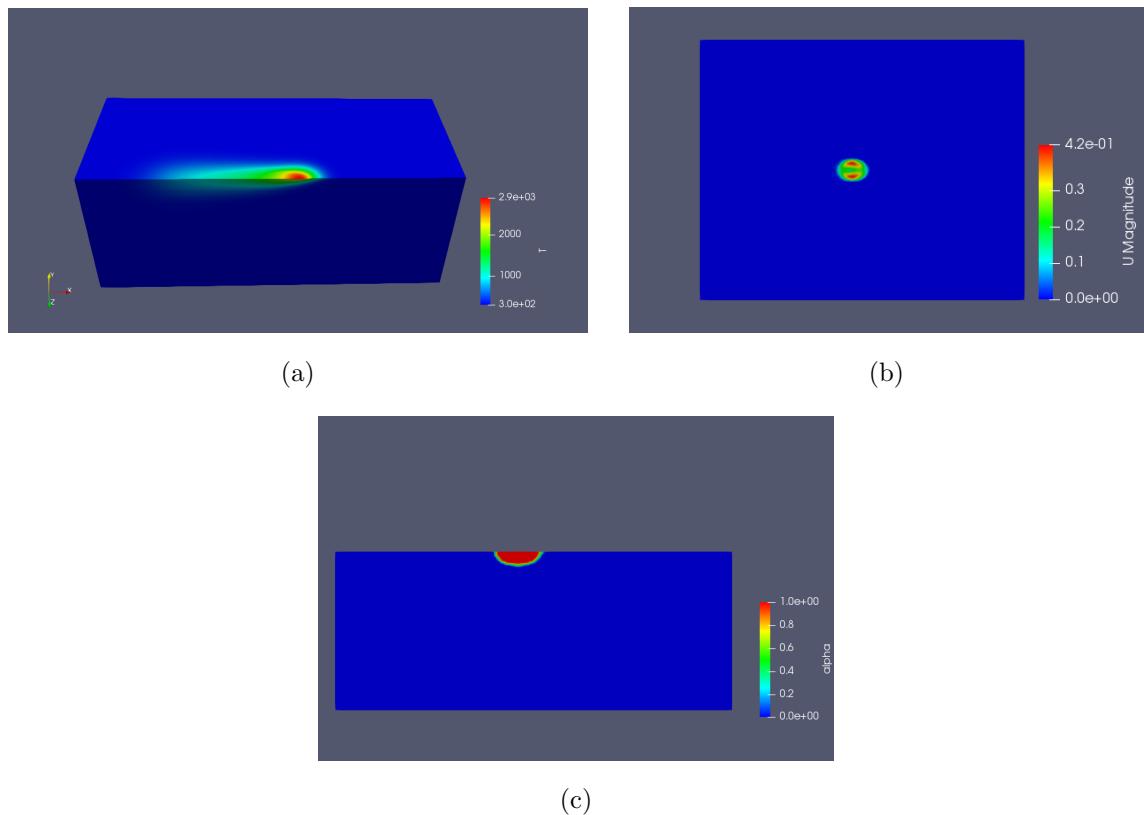


Figure 2.10: (a)A cut section showing heat affected zone near the laser source . (b)Topview of velocity profile inside melt pool. (c) A cut section showing melt pool shape.

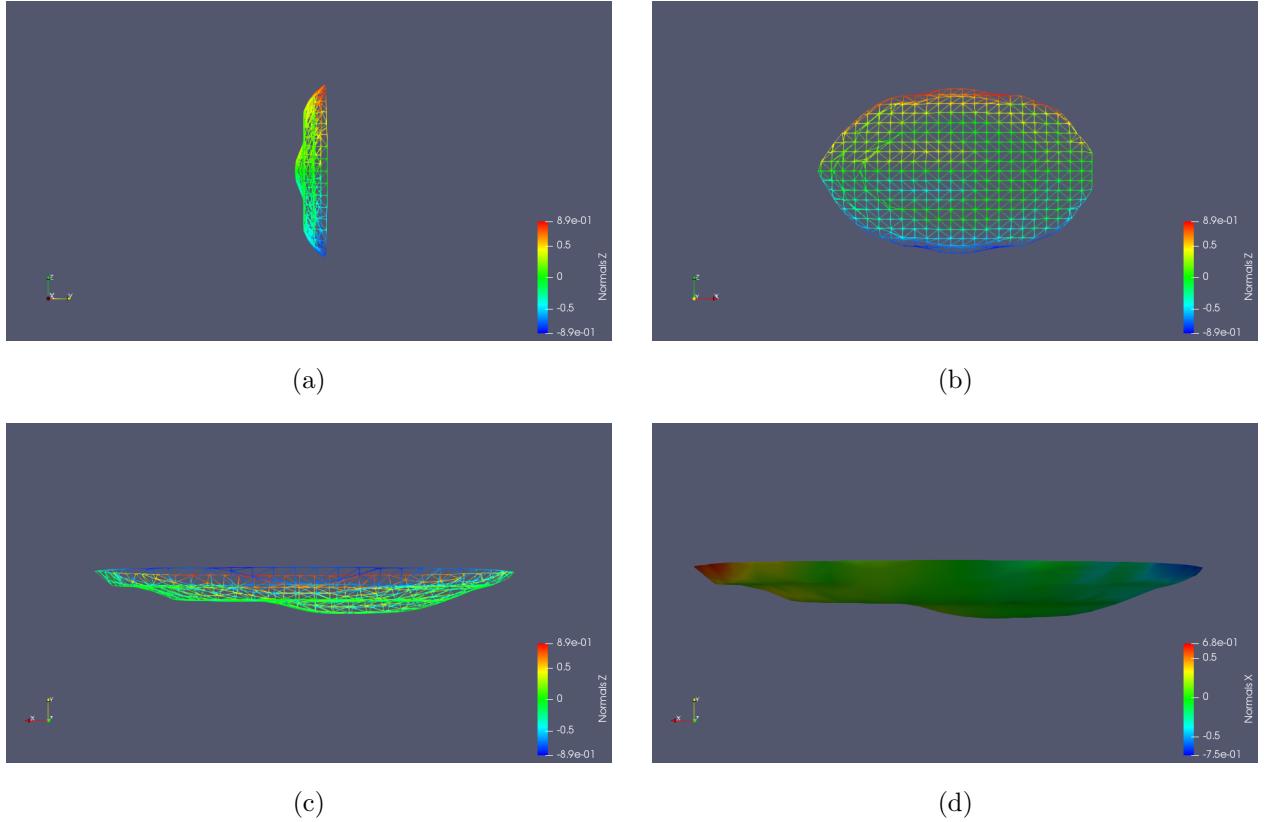


Figure 2.11: Liquid phase contour corresponding to liquid fraction of 0.5(a)X normal (b)Y normal (c) Z normal (d) X normal surface

A preliminary test is run to check the effect of surface tension gradient on the melt pool flow velocity profile. In our case surface tension is only dependent on temperature. In case of $\frac{\partial\sigma}{\partial T} = 0$, the flow obtained is due to natural convection or buoyancy. In case of positive $\frac{\partial\sigma}{\partial T}$, which is possible by changing the surface properties (ex: addition of surfactants), there is larger surface tension in hotter region than in colder region. Thus, there is fluid flow from cold to hot region at free surface. While negative surface tension gradient gives rise to high surface tension in colder region and low surface tension in hot region. This gives rise to flow from hot region to cold region at free surface.

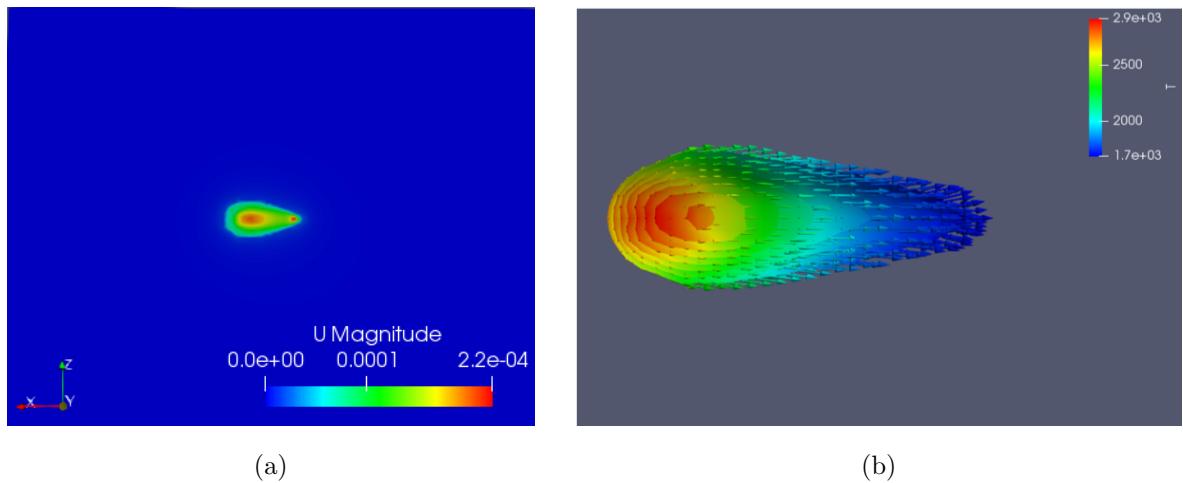


Figure 2.12: Velocity profile in melt pool (a)Zero surface tension gradient (b)Velocity streamlines.

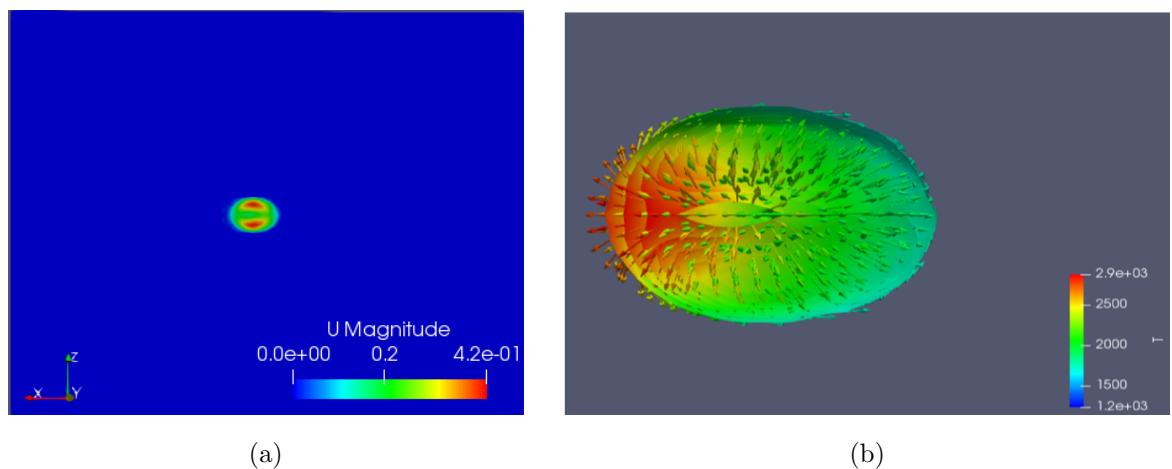


Figure 2.13: Velocity profile in melt pool (a)Negative surface tension gradient (b)Velocity streamlines.

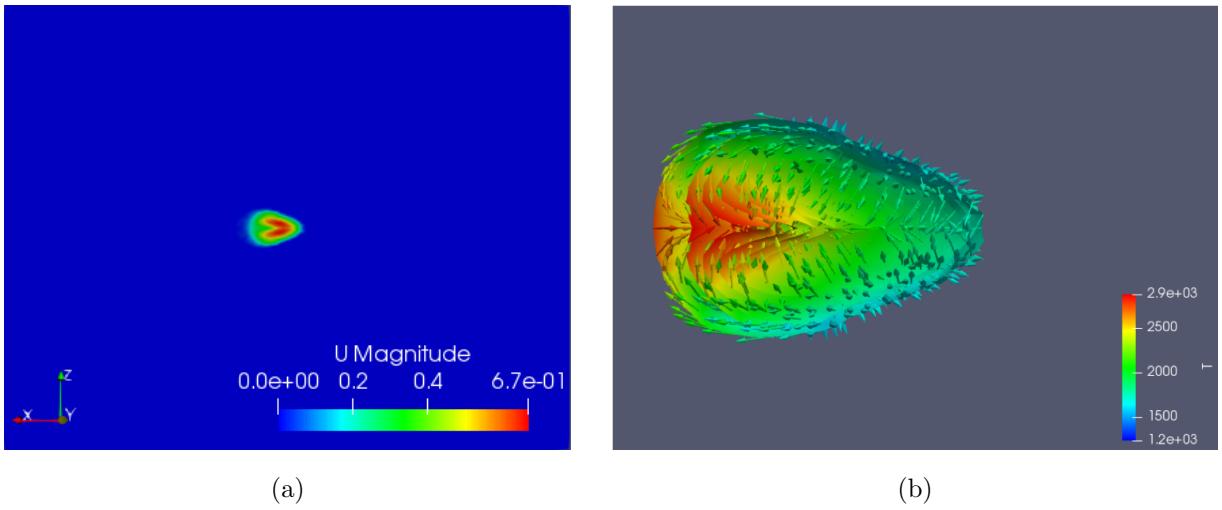


Figure 2.14: Velocity profile in melt pool (a)Positive surface tension gradient (b)Velocity streamlines.

2.7.6 Test case for stress analysis

The sample geometry is a cuboid of 15mm x 5mm x 2mm dimension. The sample was subjected to a laser beam of $100 \mu\text{m}$ radius and a moving velocity of 0.95m/sec with a laser powers of 150W. The absorptivity of the sample surface to the laser is 0.2. The following thermophysical property and temperature dependent mechanical properties of Ti-6Al-4V has been used for the test case.

Properties	Values	Units
Density(Solid and Liquid phase)	4000	kg/m^3
Solidus Temperature	1878	K
Liquidus Temperature	1928	K
Kinematic Viscosity of Solid	2.97E-07	$\text{kg}/(\text{m.s})$
Kinematic Viscosity of Liquid	2.97E-07	$\text{kg}/(\text{m.s})$
Thermal Conductivity of solid	$1.57+0.016 \times T-1E-06 \times T^2$	$\text{J}/(\text{m.s.K})$
Thermal Conductivity of liquid	30	$\text{J}/(\text{m.s.K})$
Latent heat of melting	2.6E+05	$\text{J}/(\text{kg})$
Specific heat capacity of solid	$492.4+0.025 \times T-4.18E-06 \times T^2$	$\text{J}/(\text{kg.K})$
Viscosity	4.0E-03	$\text{kg}/(\text{m.s})$
$\frac{d\gamma}{dT}$	-0.37E-03	$\text{N}/(\text{m.K})$
Specific heat capacity of liquid	800	$\text{J}/(\text{kg.K})$

Table 2.2: Thermophysical Properties of Ti-6Al-4V. [13]

Temperature (K)	Young's Modulus (GPa)	Volumetric Expansion Coefficient (/K)
300	125	8.78E-06
533	110	9.83E-06
589	100	1.00E-05
700	93	1.07E-05
755	80	1.11E-05
811	74	1.12E-05
923	55	1.17E-05
1073	27	1.22E-05
1098	22	1.23E-05
1123	18	1.24E-05
1573	12	1.30E-05
1873	9	1.63E-05

Table 2.3: Temperature dependent mechanical properties of Ti-6Al-4V. [13]

Temperature (K)	Yield Stress (MPa)
300	955
573	836
773	732
1023	581
1073	547
1173	480
1273	405
1373	330

Table 2.4: Temperature dependent yield strength of Ti-6Al-4V. [13]

A transient solver is developed which solves for the temperature profile and stress analysis in the sample. In the first step thermal profile is calculated which serves as input for the stress analysis. A steady state mechanical equilibrium equation is solved for stress analysis. We have taken a case of elastic perfectly plastic properties in order to find out the residual stress and final distortion in the material post cooling of the material to room temperature. The boundary condition used is similar to what is described in subsection 2.7.3. The incremental displacement boundary condition is used. The bottom part is kept at fixed value incremental displacement boundary condition having a value of zero. All other sides is kept at zero increment traction.

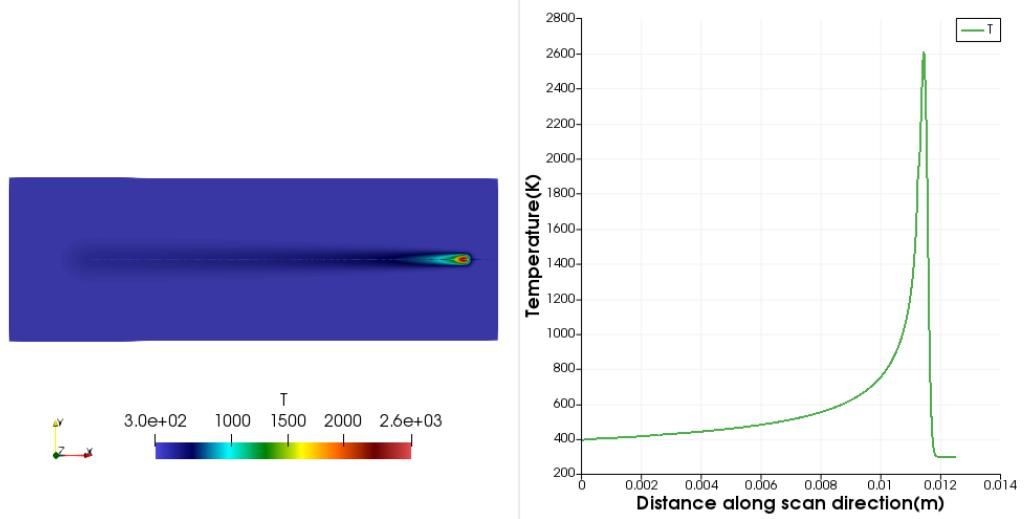


Figure 2.15: Temperature profile along laser scan direction.

Figure 2.15 show the temperature profile along the laser scan direction. The peak temperature is obtained just below the laser tip i.e. at the center of the weld pool[14]. Temperature drops down abruptly ahead of the weld pool while it decays smoothly in the scanned region behind weld pool. This is due to rapid scanning of laser beam, the temperature contours are elongated behind the heat source and compressed in front of the beam. This transient and spatially non-uniform temperature field distribution, is responsible for the generation of the stress and strain fields.

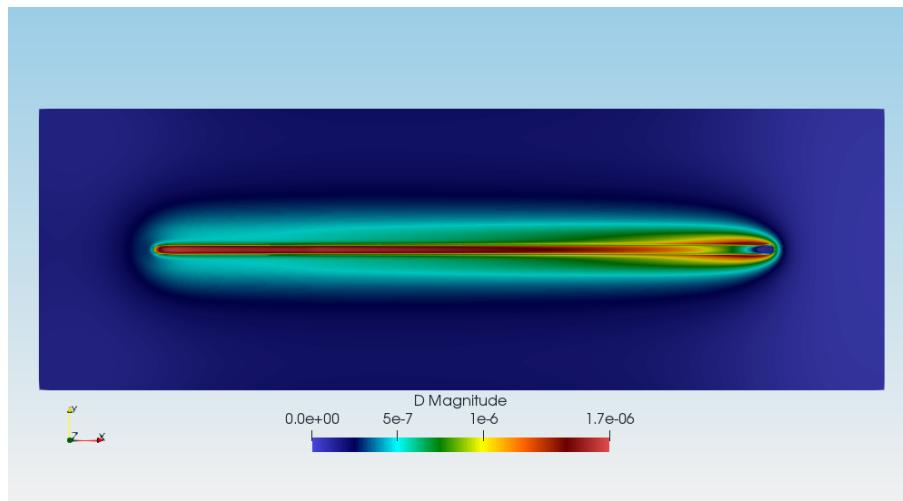


Figure 2.16: Total displacement field magnitude.

Figure 2.16 shows the magnitude of total displacement which is calculted as sum of incremental displacement obtained from solution of equation 2.16 .

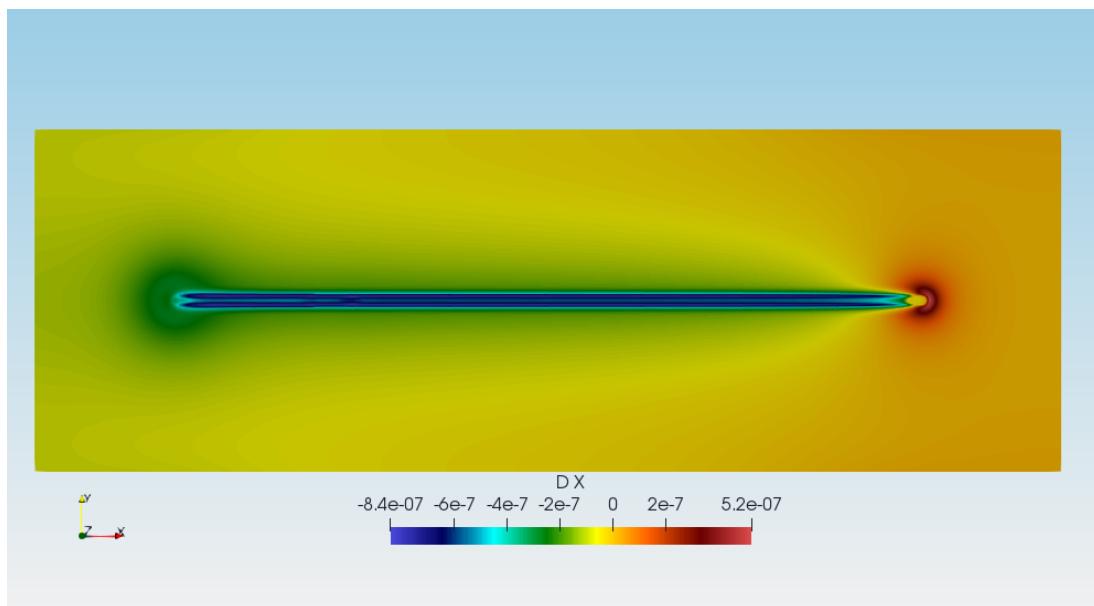


Figure 2.17: Displacement field x component.

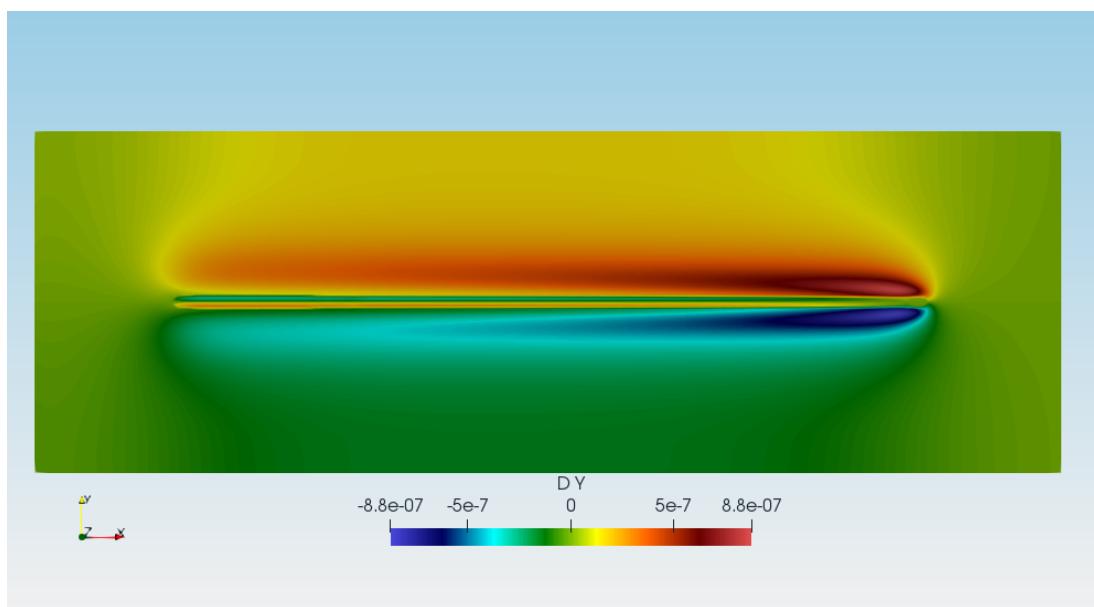


Figure 2.18: Displacement field y component.

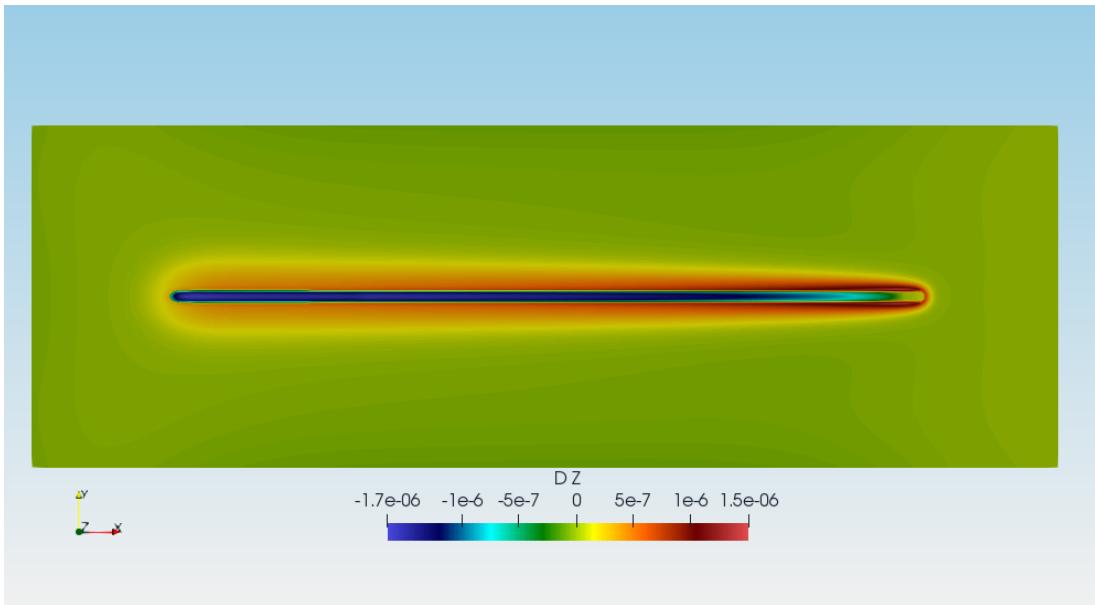
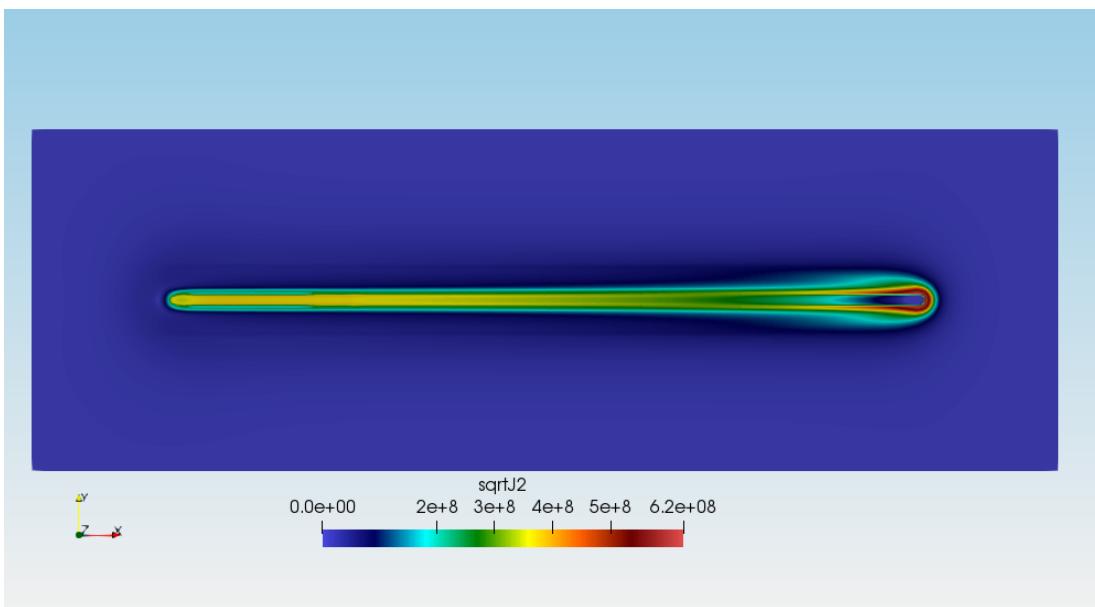


Figure 2.19: Displacement field z component.

Figure 2.20: sqrtJ2 magnitude.

Stress evolution depends on the transient temperature distribution mostly during cooling of the material. Fig. 2.15 shows the temperature distribution along longitudinal direction during cooling of the build. Fig. 2.21 represents the corresponding longitudinal stress profile developed during first scan. The high temperature near the weld pool softens the material in that region. Therefore, the magnitude of the stress is relatively low as it is limited by the yield strength at

the high temperature. The stress field evolves as the material continues to cool down . The temperature of the material almost cools down to the room temperature away from location of laser source 2.15.The longitudinal stress thus developed is highly tensile, as shown in Fig. 2.21.

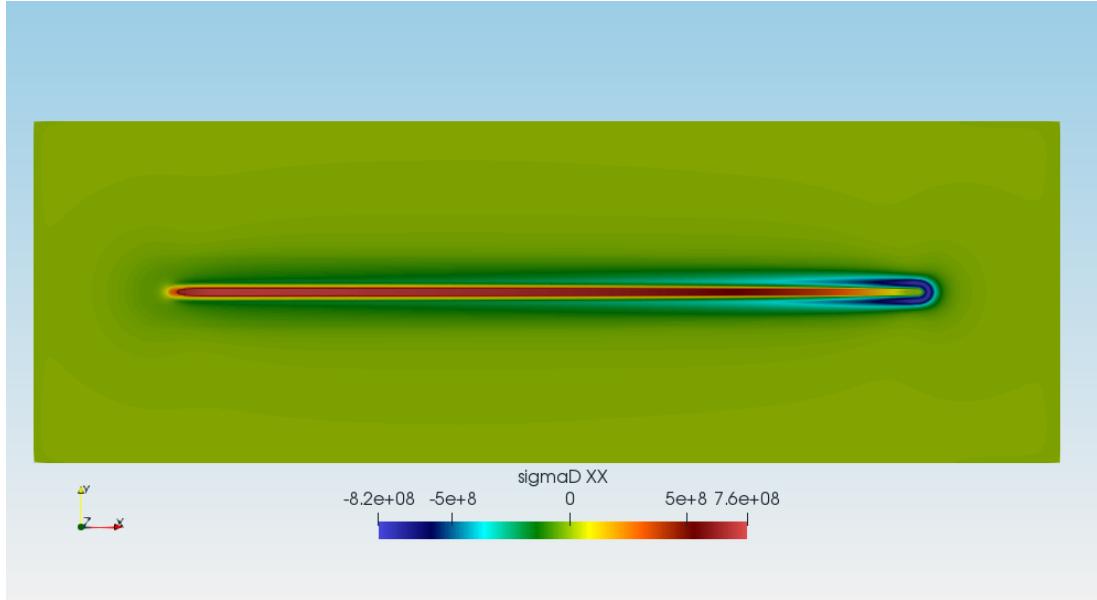


Figure 2.21: Longitudinal stress (σ_{xx}).

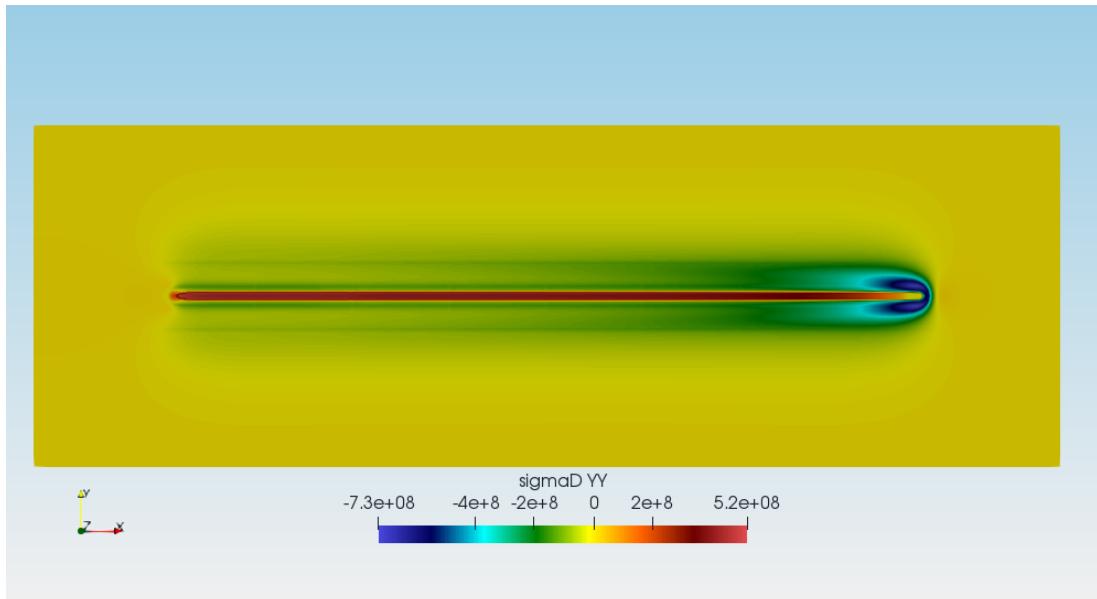


Figure 2.22: Transverse stress (σ_{yy}).

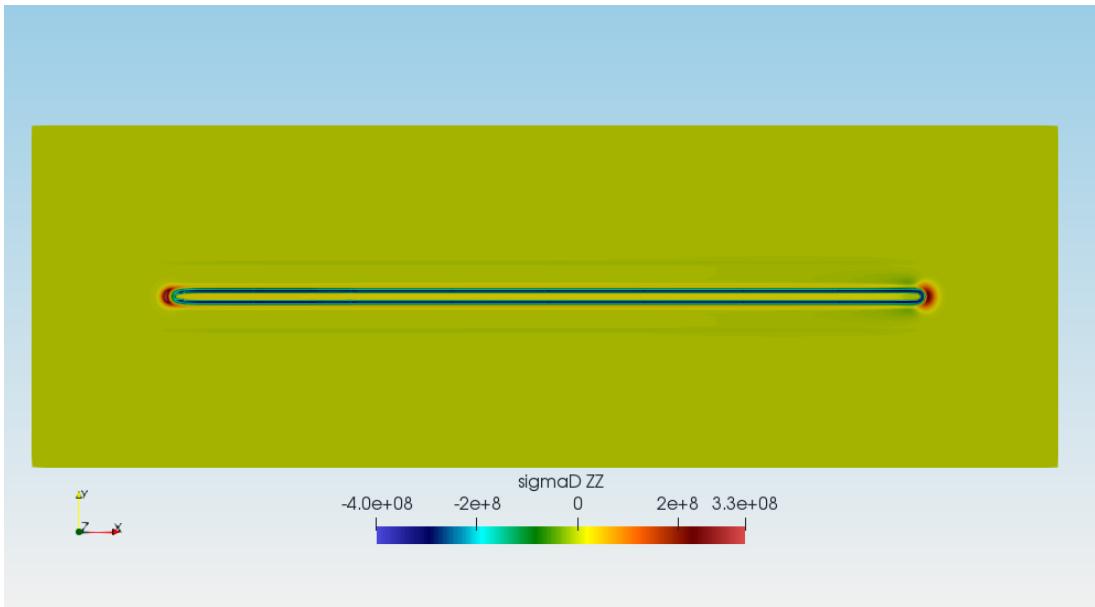


Figure 2.23: Through thickness stress (σ_{zz}).

Figs. 2.24 2.25 and 2.26 shows the distribution of the residual stresses along x (longitudinal), y (transverse) and z (through-thickness) directions, respectively. The following major observations can be made from these results. The location of the maximum longitudinal and transverse (tensile) residual stresses are near the top of the deposit.

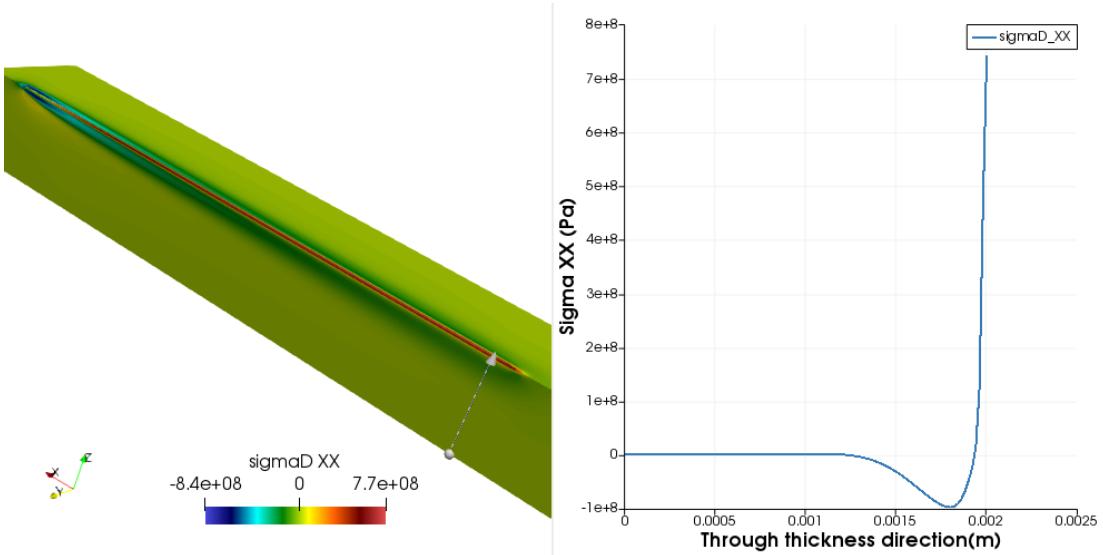
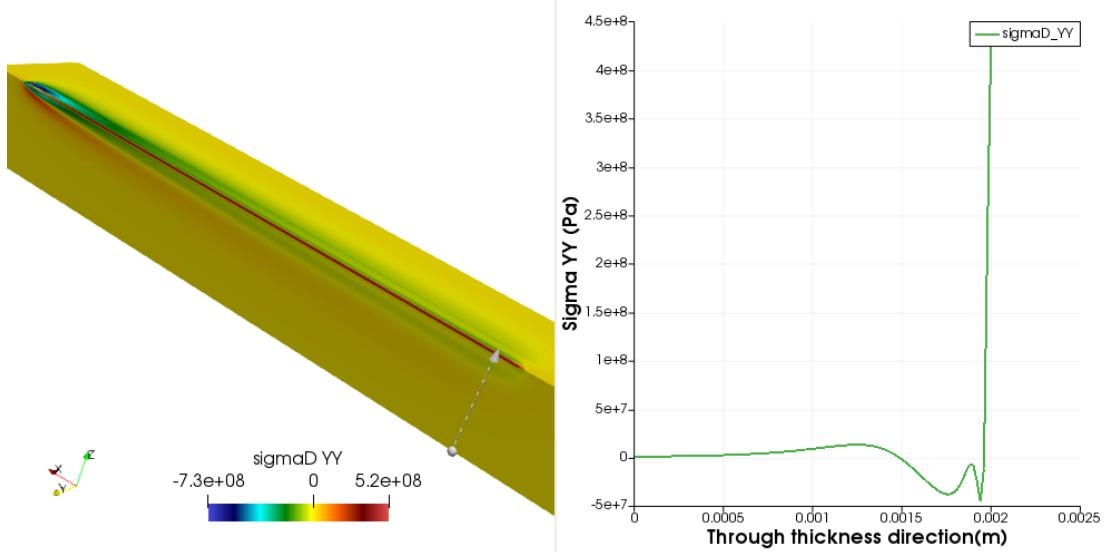
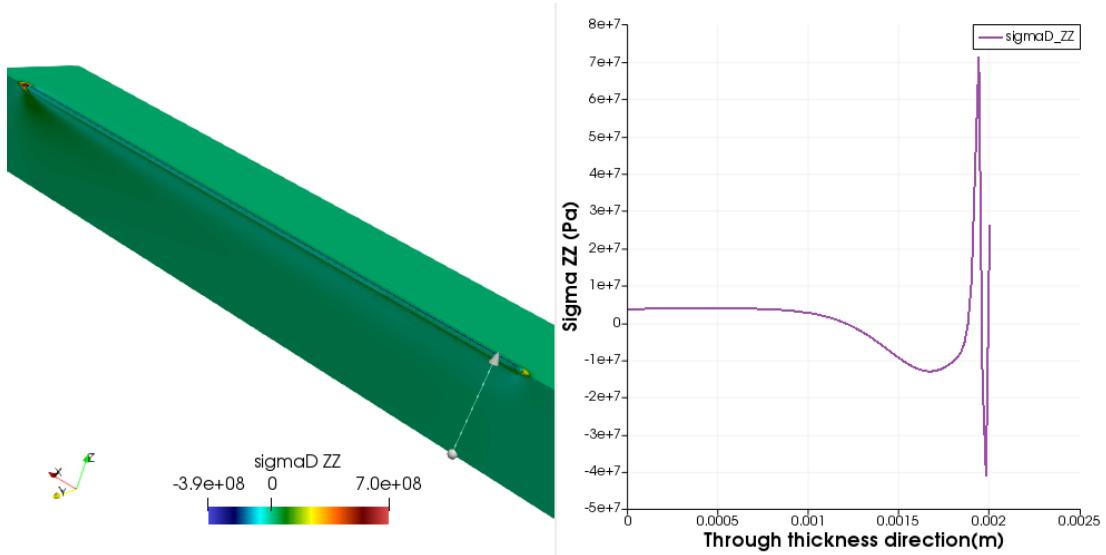


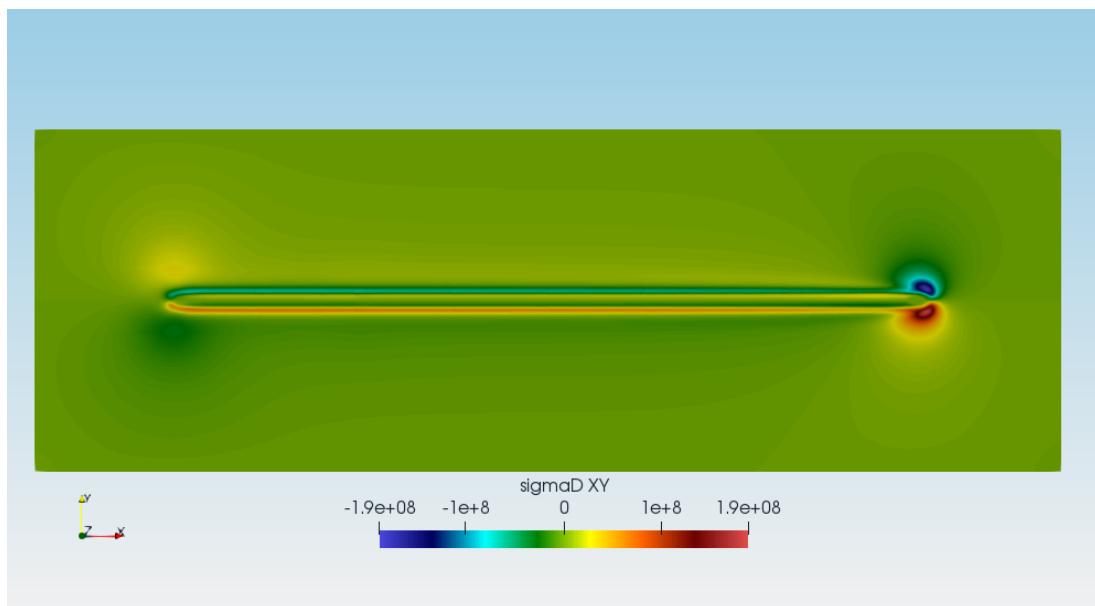
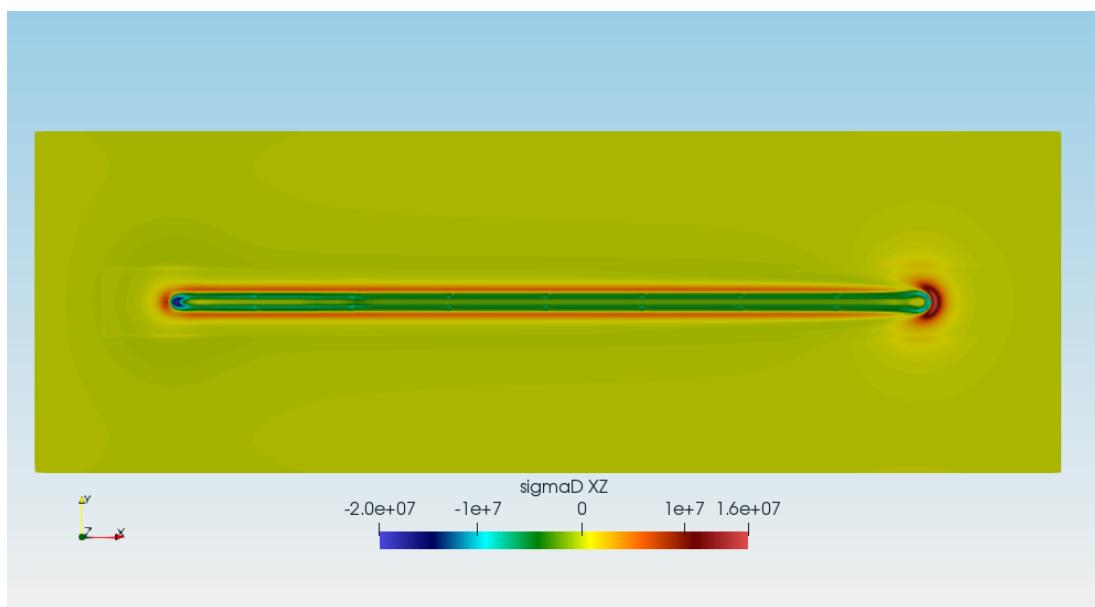
Figure 2.24: Variation of σ_{xx} in z direction.

Figure 2.25: Variation of σ_{yy} in z direction.

Secondly, in both Figs. 2.24 and 2.25, the residual stresses change from tensile to compressive. Finally, the through-thickness stress in Fig. 2.26 is compressive in the center of the deposit and tensile near the top and bottom of the material.

Figure 2.26: Variation of σ_{zz} in z direction.

The longitudinal residual stress exhibits a steep gradient that makes the parts susceptible to buckling and warping. These computed residual stress results can play a significant role in the study the delamination of deposit from substrate, warping and separation of layers.

Figure 2.27: Shear stress (σ_{xy}).Figure 2.28: Shear stress (σ_{xz}).

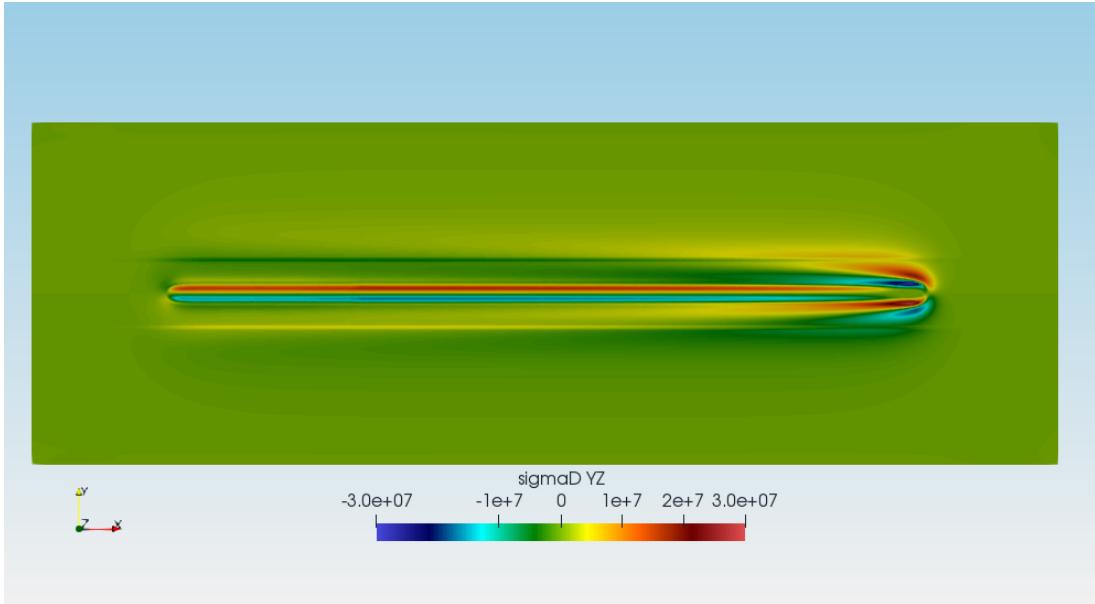


Figure 2.29: Shear stress (σ_{yz}).

However, susceptibility to warping and delamination depends not only on the magnitude of the residual stresses but also the yield strength of the alloy. Therefore, a normalized residual stress expressed as the ratio of the longitudinal residual stress along longitudinal direction to the room temperature yield strength of the alloy is used for assessment of such problems [13]. The through-thickness stress is responsible for the possible delamination of a component. It should be noted that the through-thickness stress is always lower than the yield strength of the alloy. In this case maximum through-thickness stress for Ti-6Al-4V is 70Mpa. The formation of tensile residual stresses has significant effects on the mechanical properties of the product. Fatigue cracks in additively manufactured Ti-6Al-4V parts grow faster in the presence of a higher tensile residual stress in the component. The residual stress changes from tensile to compressive at the layer interfaces. In extreme cases, this behavior may result in the separation of layers.

Chapter 3

Phase-Field modeling of equilibrium precipitate shapes under the influence of coherency stresses

3.1 Background

Phase field models are used to simulate evolution of microstructures. The microstructure is described by a system of continuous variables, where the microstructure interfaces have a finite width over which the variables transition between values. The evolution of the microstructure is defined in terms of the free energy of the system, and can be coupled to other physics to provide a complete view of the material behavior. The phase-field method has become a powerful tool for simulating the microstructural evolution in a wide variety of material processes, such as solidification, solid-state phase transformations, precipitate growth and coarsening, martensitic transformations and grain growth, domain evolution in ferro-electric and ferromagnetic materials [15] [16]. The phase-field models include formulations for pure substances, for multicomponent systems, and for polycrystalline structure and solidification in eutectic, peritectic, and monotectic systems. Other simulation techniques for dendrite growth are cellular automata and hybrid methods such as the multiscale diffusion Monte Carlo (DMC). The phase-field method requires previous knowledge of the material properties of the system in study. The input includes bulk properties such as density, heat capacity and latent heat, and others such as interfacial and kinetic growth coefficients, being the latter properties which are hardly accessible in experiments. It has multiple advantages that include:(1) no prior assumption on microstructure morphology, (2) no need to explicitly track the location of interfaces, (3) multi-dimensional (two-dimensional (2-D), 3-D), computationally efficient representation of

multiple material processes with short- and long-range interactions, and (4) multi-dimensional (2-D, 3-D), computationally efficient representation of inhomogeneous and anisotropic properties of defects and materials.

Precipitate strengthened alloys finds its application in high-temperature environment, where, the strengthening is achieved through interaction between precipitate and dislocation. Mostly precipitate size, morphologies, and their distribution drives the mechanical properties of these high temperature alloys. Thus, experimental studies focussing on the precipitate morphology, growth and coarsening, strengthening have been carried out in the past. Basically, there properties are achieved through two mechanisms. In one case the precipitates are large enough such that there is does not exist any coherency between the precipitate and the matrix while in second case precipitates are small such that coherency exists between the matrix and the precipitate. In the former case, the interaction is purely physical, in the latter case, coherency stresses around the precipitate also influence the interaction with the precipitate. In both the cases, shape of precipitate plays a vital role in interaction with dislocations.

The first theory in this field was given by Johnson and Cahn [17], who predicted an equilibrium shape transition of an elastically isotropic misfitting precipitate in a stiffer matrix. The equilibrium shape of a precipitate is determined by minimization of total energy i.e: sum of elastic and interfacial energies under the constraint of volume conservation of the precipitate. The theory proposes the shape transition with size, similar to a second-order phase transition with the shape of the precipitate as an order parameter. The theory analytically predicts the equilibrium shape order parameters as a function of precipitate size whereby for certain conditions, below a critical size there is an unique order parameter describing the shape of the particle and a bifurcation into two or more variants beyond it [18].

3.2 Research goal

Our objective in this chapter will be related to finding the shapes of coherent precipitates, more importantly the understanding of equilibrium morphology of precipitates as a function of elastic and misfits anisotropy.

3.3 Outline

In the following sections, we firstly describe the model, followed by a discussion on the results. We then continue with other combinations of tetragonal misfits and elastic anisotropies which gives rise to different morphologies of equilibrium shape of precipitates.

3.4 Physical model

Solid-state phase transformations mostly involves a difference of the lattice parameter between the precipitate and the matrix. This gives rise to misfit strains/stresses for a coherent interface. This in turn contributes to the system energy in terms of an elastic contribution which scales with the volume of the precipitate. Similarly, the interfacial energy which is the other component of the energy in the system, varies with the interfacial area. In this context, the equilibrium shape of the precipitate is the one which minimizes the sum total of the contributions from both the elastic energy and interfacial components, which given the scaling of the two energy components is a function of the size of the precipitate.

In this section, we formulate a phase-field model, where the functional consists of both the elastic and the interfacial energy contributions. Since the equilibrium precipitate shape depends on the size of the precipitate, we formulate a model which minimizes the system energy while preserving the volume of the precipitate, and thereby allows the computation of the equilibrium shape of precipitates. This allows the determination of the precipitate shapes as a function of the different precipitate sizes as has been done previously using sharp-interface methods. This constrained minimization is achieved through the technique of volume preservation which is also described elsewhere [19] that is essentially the coupling of the Allen–Cahn type equation with a correction term using a Lagrange parameter that ensures the conservation of the precipitate volume during evolution.

In the following, we discuss the details of the phase-field model. We begin with the free energy functional that reads

$$\mathcal{F}(\phi) = \int_v [\gamma W a^2(\mathbf{n}) |\nabla \phi|^2 + \frac{1}{W} \omega(\phi)] dV + \int_v f_{el}(\mathbf{u}, \phi) dV + \lambda_\beta \int_v h(\phi) dV \quad (3.1)$$

where V is the volume of system. ϕ is the phase field order parameter, $\phi=1$ is precipitate phase and $\phi=0$ is matrix phase. The first term on the right hand side of equation represents the interfacial energy which is sum total of gradient energy and potential contributions. γ controls the interfacial energy in the system. W influences the width of the diffuse interface between precipitate and matrix phases. $a(\mathbf{n})$ represents interfacial energy anisotropy between matrix/precipitate phase. It is a function of interface normal given by, $\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$. The second term in first integral is double obstacle potential given as,

$$w(\phi) = \begin{cases} \frac{16}{\pi^2} \gamma \phi (1 - \phi) & \phi \in [0, 1] \\ \infty & otherwise \end{cases} \quad (3.2)$$

The second integral is elastic energy contribution to the free energy density of the system which is a function of the order parameter ϕ that is also used to interpolate between the phase properties and the misfit. $\int_v h(\phi) dV$ represents the volume of precipitate, where $h(\phi) = \phi^2(1-2\phi)$ is an interpolation function which varies from 0 to 1. λ_β is the Lagrange parameter that is added for volume conservation of the precipitate. The evolution equation of ϕ as given by Allen-Cahn is

$$\tau W \frac{\partial \phi}{\partial t} = -\frac{\partial F}{\partial \phi} \quad (3.3)$$

and elaborated as

$$\tau W \frac{\partial \phi}{\partial t} = 2\gamma W \nabla \cdot \left[a(\mathbf{n}) \left[\frac{\partial a(\mathbf{n})}{\partial \nabla \phi} |\nabla \phi|^2 + a(\mathbf{n}) \nabla \phi \right] \right] - \frac{16}{\pi^2} \frac{\gamma}{W} (1-2\phi) - \frac{\partial f_{el}(\mathbf{u}, \phi)}{\partial \phi} - \lambda_\beta h'(\phi) \quad (3.4)$$

where τ is the interface relaxation constant, which in the present modeling context is chosen as the smallest value that allows for a stable explicit temporal evolution using a simple finite difference implementation of the forward Euler-scheme. Note, ' denotes differentiation of the function with respect to its argument. In order to complete the energetic description, it is important to elaborate the elastic energy density $f_{el}(\mathbf{u}, \phi)$ terms of the physical properties of the matrix and the precipitate phases that are the stiffness matrices, as well as the misfit. The elastic energy of each phase can be written as

$$f_{el} = \frac{1}{2} \sigma_{ij} (\varepsilon_{ij} - \varepsilon_{ij}^*) \quad (3.5)$$

where ε_{ij}^* is the eigenstrain at the interface produced due to a lattice mismatch between the precipitate and matrix. We will be assuming that the eigenstrain is accommodated in the precipitate phase, such that the eigenstrain in the matrix phase is zero.

3.4.1 Elastic energy interpolation

The stiffness tensor interpolation can be written as :

$$C(\phi) = \sum C_{ijkl} h(\phi) \quad (3.6)$$

where $h(\phi) = \phi^2(3 - 2\phi)$ is an interpolation function. The elastic energy density can be written in form of stiffness matrices and strain.

$$f_{el} = \frac{1}{2} (\varepsilon_{ij} - \varepsilon_{ij}^*(\phi)) C_{ijkl}(\phi) (\varepsilon_{kl} - \varepsilon_{kl}^*(\phi)) \quad (3.7)$$

After taking the derivative with respect to order parameter , we will obtain the following equation

$$\begin{aligned} \frac{\partial f_{el}}{\partial \phi} &= \frac{1}{2} (\varepsilon_{ij} - \varepsilon_{ij}^*(\phi)) \frac{\partial C_{ijkl}(\phi)}{\partial \phi} (\varepsilon_{ij} - \varepsilon_{ij}^*(\phi)) - \frac{1}{2} (\varepsilon_{ij} - \varepsilon_{ij}^*(\phi)) C_{ijkl}(\phi) \frac{\partial^* \varepsilon_{kl}(\phi)}{\partial \phi} \\ &\quad - \frac{1}{2} (\varepsilon_{kl} - \varepsilon_{kl}^*(\phi)) C_{ijkl}(\phi) \frac{\partial^* \varepsilon_{ij}(\phi)}{\partial \phi} \end{aligned} \quad (3.8)$$

where $\frac{\partial C_{ijkl}(\phi)}{\partial \phi} = C_{ijkl} \frac{\partial h(\phi)}{\partial \phi} = \delta C_{ijkl}^*$ and $\frac{\partial \varepsilon^*}{\partial \phi} = \epsilon^* \frac{\partial h(\phi)}{\partial \phi}$
Thus, $\delta C_{ijkl}^* (\varepsilon_{kl} - \varepsilon_{kl}^*(\phi)) = \delta \sigma_{ij}$

Putting all these values in above equation , we obtain the final elastic energy relation as:

$$\frac{\partial f_{el}}{\partial \phi} = 6\phi(1-\phi) [0.5 (\delta \sigma : (\epsilon - \phi^2(3 - 2\phi)\epsilon^*)) - \sigma : \epsilon^*] \quad (3.9)$$

3.4.2 Conservation of volume

The remaining part of the evolution equation 3.4 that is yet to be determined is the Lagrange parameter λ_β which would conserve the volume during interface evolution. Volume conservation is affected through the constraint,

$$\int h(\phi) dx = constor \int \delta h(\phi) dx = 0 \quad (3.10)$$

$$\int \delta h(\phi) dx = 0 \quad (3.11)$$

where $\delta h(\phi)$ is the change in the value of at a given spatial location. Reformulation of this condition in discrete terms is performed in the following manner. From Eq. 3.4, we have the rate of change of the order parameter at a given location, i.e.,

$$\tau W \frac{\partial \phi}{\partial t} = rhs_\alpha - \lambda_\beta h'(\phi) \quad (3.12)$$

where the term rhs_α constitutes all the terms in the evolution equation of the order parameter in Eq. 3.4 leaving out the Lagrange parameter. In order to affect the volume constraint as given by Eq. 3.10, the Lagrange parameter λ_β is computed as

$$\lambda_\beta = \frac{\sum_v rhs_\alpha}{\sum_v h'(\phi)} \quad (3.13)$$

where the summation \sum_v is over the entire volume. This essentially ensures that the summation of all the changes in the order parameter over the entire volume returns zero, thus affecting the volume constraint in the discrete framework.

3.4.3 Mechanical equilibrium

As a final aspect, what remains is the computation of the displacement fields as a function of the spatial distribution of the order parameter. This is done iteratively by solving the damped wave equation written as

$$\rho \frac{d^2 \mathbf{u}}{dt^2} + b \frac{d\mathbf{u}}{dt} = \nabla \cdot \boldsymbol{\sigma} \quad (3.14)$$

that is solved until the equilibrium is reached, i.e., $\nabla \cdot \boldsymbol{\sigma} = 0$. The terms a and b are chosen such that the convergence is achieved in the fastest possible time.

The optimization procedure for finding the equilibrium shape is performed by solving the evolution of the order parameter ϕ in Eq. [4] along with the equation of mechanical equilibrium at each time step, until a converged shape is reached.

3.5 Implementation in OpenFoam

Calculating displacement field which is given by momentum conservation of solid body which includes the stress developed due to phase transformation.

$$\frac{\partial^2 (\rho u)}{\partial t^2} - \nabla \cdot \boldsymbol{\sigma} = 0 \quad (3.15)$$

which is implemented in OpenFoam as:

```
#include "fvCFD.H"
#include "Switch.H"

// * * * * *

```

```

int main(int argc, char *argv[])
{
    #include "postProcess.H"

    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createControls.H"
    #include "createFields.H"

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

Info<< "Calculating displacement field" << endl;
while (runTime.loop())
{
    Info<< "Iteration:" << runTime.value() << nl << endl;
    #include "readSolidDisplacementFoamControls.H"
    int iCorr = 0;
    scalar initialResidual = 0;
    TOld=T;
    gradD = fvc::grad(D);
    #include "TEqn.H"

do
{
{
    fvVectorMatrix DEqn
    (
        fvm::d2dt2(D)
        ==
        sig1*fvm::laplacian(2*(mu1*T*T*(3-2*T)
        + mu2*(1-T)*(1-T)*(1+2*T))
        + lambda1*T*T*(3-2*T)
        + lambda2*(1-T)*(1-T)*(1+2*T), D, "laplacian(DD,D)")
        + (sig1/sig2)*divSigmaExp
        - (sig1)*fvc::div((2*mu1*T*T*(3-2*T)

```

```

+ 2*mu2*(1-T)*(1-T)*(1+2*T))*T*T*(3-2*T)*cEigenStrain
+(lambda1*T*T*(3-2*T)
+ (1-T)*(1-T)*(1+2*T)*lambda2)*I*tr(T*T*(3-2*T)
*cEigenStrain));
);

initialResidual = DEqn.solve().max().initialResidual();

if (!compactNormalStress)
{
    divSigmaExp = fvc::div(DEqn.flux());
}
}

{
gradD = fvc::grad(D);

strain=((gradD - T*T*(3-2*T)*cEigenStrain)
&&symmTensor(1 0 0 0 0 0))*symmTensor(1 0 0 0 0 0)
+((gradD - T*T*(3-2*T)*cEigenStrain)
&&symmTensor(0 0 0 1 0 0))*symmTensor(0 0 0 1 0 0)
+((gradD - T*T*(3-2*T)*cEigenStrain)
&&symmTensor(0 0 0 0 1 0))*symmTensor(0 0 0 0 0 1);

sigmaD=(mu1*T*T*(3-2*T)
+mu2*(1-T)*(1-T)*(1+2*T))*twoSymm(gradD)
+(lambda1*T*T*(3-2*T)
+lambda2*(1-T)*(1-T)*(1+2*T))*(I*tr(gradD))
+(mu1_*T*T*(3-2*T)+mu2_*(1-T)*(1-T)*(1+2*T))*strain;

if (compactNormalStress)
{
    divSigmaExp = sig2*fvc::div
(
    sigmaD - (2*mu1*T*T*(3-2*T) + 2*mu2*(1-T)*(1-T)*(1+2*T)

```

```

+lambda1*T*T*(3-2*T) + (1-T)*(1-T)*(1+2*T)*lambda2)*gradD ,
"div(sigmaD)"
);
}
else
{
    divSigmaExp += sig2*fvc::div(sigmaD);
}
}

}while(initialResidual>convergenceTolerance && ++iCorr<nCorr);

#include "calculateStress.H"

Info<< "ExecutionTime=" << runTime.elapsedCpuTime() << "s"
      << "ClockTime=" << runTime.elapsedClockTime() << "s"
      << nl << endl;
dimensionedScalar totalEnergy = 0.0;
dimensionedScalar elasticEnergy = 0.0;
dimensionedScalar surfaceEnergy = 0.0;
volScalarField consta(Sigma && (symm(gradD)-T*T*(3-2*T)
                                     *cEigenStrain));

volVectorField gradT(fvc::grad(T));
forAll(consta, cellI) {
    elasticEnergy += 0.5*consta[cellI]*4.0;
    surfaceEnergy += 2.0*Gamma*Epsilon*(magSqr(gradT[cellI]))*4.0;
    totalEnergy += elasticEnergy + surfaceEnergy;
}
Info<< "Min/maxT:" << min(T()).value() << ''
      << max(T()).value() << endl;
}
Info<< "End\n" << endl;
return 0;
}

```

Allen Cahn dynamics evolution for ϕ is given by following equation

$$\tau W \frac{\partial \phi}{\partial t} = 2\gamma W \nabla \cdot \left[a(\mathbf{n}) \left[\frac{\partial a(\mathbf{n})}{\partial \nabla \phi} |\nabla \phi|^2 + a(\mathbf{n}) \nabla \phi \right] \right] - \frac{16}{\pi^2} \frac{\gamma}{W} (1 - 2\phi) - \frac{\partial f_{el}(\mathbf{u}, \phi)}{\partial \phi} - \lambda_\beta h'(\phi) \quad (3.16)$$

In this case we assume that surface energy is isotropic i.e: $a(\mathbf{n})$ is constant. Thus $\frac{\partial a(\mathbf{n})}{\partial \nabla \phi} = 0$.

```

scalar sumT = 0.0;
scalar sum2T = 0.0;
scalar sum3T = 0.0;

strain=((gradD - T*T*(3-2*T)*cEigenStrain)
    &&symmTensor(1 0 0 0 0))*symmTensor(1 0 0 0 0)
    +((gradD - T*T*(3-2*T)*cEigenStrain)
    &&symmTensor(0 0 0 1 0))*symmTensor(0 0 0 1 0)
    +((gradD - T*T*(3-2*T)*cEigenStrain)
    &&symmTensor(0 0 0 0 1))*symmTensor(0 0 0 0 1);

Sigma=(2*(mu1*T*T*(3-2*T) + mu2*(1-T)*(1-T)*(1+2*T))
    *(symm(fvc::grad(D)) - T*T*(3-2*T)*cEigenStrain)
    +(lambda1*T*T*(3-2*T) + lambda2*(1-T)*(1-T)*(1+2*T))
    *(I*tr(fvc::grad(D) - T*T*(3-2*T)*cEigenStrain)))
    +(mu1_*T*T*(3-2*T) + mu2_*(1-T)*(1-T)*(1+2*T))*strain;

deltaSigmaD=((mu1-mu2)*twoSymm(fvc::grad(D))
    +(lambda1-lambda2)*(I*tr(fvc::grad(D)))
    -2*(mu1-mu2)*T*T*(3-2*T)*cEigenStrain
    -(lambda1-lambda2)*(I*tr(T*T*(3-2*T)*cEigenStrain)))
    +(mu1_-mu2_)*strain;

fvScalarMatrix TEqn
(
    Tau*Epsilon*fvm::ddt(T)
    - 2.0*Gamma*Epsilon*fvm::laplacian(DTs, T)
    ==

```

```

-(18.0*Gamma/(sig*Epsilon))*(T)*(1-T)*(1-2.0*T)
-(1/sig)*6.0*T*(1-T)*(0.5*(deltaSigmaD&&(symm(fvc::grad(D))
-T*T*(3-2*T)*cEigenStrain))-(Sigma&&cEigenStrain));
);

TEqn.solve();

sumT += gSum(T());
sum2T+= gsum((T*T)());
sum3T += fvc::ddt(T)*deltaT.value();

scalar lambdas=sum3T/(6*(sumT-sum2T));

T+= - 6*lambdas*gSum((T*(1-T)());

```

3.6 Numerical test case

3.6.1 Model parameters

This section constitutes the material parameters and the non-dimensionalization scheme that is going to be used for simulation. This study is limited to cubic systems in 3Ds, so that we can use a simplified stiffness tensor. The following short-hand notations are used for the non-zero stiffness components, $C_{11} = C_{1111}$, $C_{22} = C_{2222}$, $C_{12} = C_{1122}$ and $C_{44} = C_{1212}$ where $C_{11} = C_{22}$ due to symmetry constraints. These components can be obtained by using material parameters, that are Poisson's ratio (ν) the Zener anisotropy (A_z), and shear modulus (μ) and can be written as

$$C_{44} = \mu, \quad C_{12} = 2\nu \left(\frac{C_{44}}{1 - 2\nu} \right), \quad C_{11} = C_{12} + \frac{2C_{44}}{A_z} \quad (3.17)$$

The eigenstrain matrix will be considered diagonal in the Cartesian coordinate system and reads

$$\epsilon^* = \begin{bmatrix} \epsilon_{xx}^* & 0 & 0 \\ 0 & \epsilon_{yy}^* & 0 \\ 0 & 0 & \epsilon_{zz}^* \end{bmatrix} \quad (3.18)$$

A non-dimensionalization scheme where the energy scale is set by the interfacial energy scale 1.0 J/m^2 divided by the scale of the shear modulus $1 \times 10^9 \text{ J/m}^3$ that yields a length scale $l^* = 1 \text{ nm}$. All the parameters hereafter, will be reported in the terms of non-dimensional units. Unless otherwise specified, all results are produced with $\mu_{mat} = 125$, $\nu_{ppt} = \nu_{mat} = 0.3$. When $A_z = 1.0$; elastic constants become isotropic. When A_z is greater than unity, elastically soft directions are $<100>$; whereas elastically hard directions are $<110>$: Similarly, in the case where A_z is less than unity, elastically soft (hard) directions are $<110>$ ($<100>$). For all the cases, the precipitate and matrix have the same magnitude of A_z .

The misfit strain or eigenstrain (ϵ^*) can be dilatational, i.e., the same along principle directions ($\epsilon_{xx}^* = \epsilon_{yy}^* = \epsilon_{zz}^*$) or tetragonal, i.e., different along principle directions ($\epsilon_{xx}^* \neq \epsilon_{yy}^* \neq \epsilon_{zz}^*$)

We have utilized a three-dimensional system of matrix with embedded precipitate. It is having a lattice mismatch at the interface that is coherent in nature. Periodic boundary conditions are used in simulations. To avoid any interaction of displacement field with the boundaries, ratio of the initialized precipitate size (equivalent radius) to the matrix size is maintained as 0.08. This is similar to the condition of an infinitely large matrix containing an isolated precipitate without any influence of external stress. The interfacial energy between the matrix and precipitate is assumed to be isotropic until specified having a magnitude of 0.15. In our model formulation, we have assumed that precipitate is softer than matrix, which is represented by parameter $\delta < 1$

3.6.2 Cubic anisotropy in elastic energy with dilatational misfit

Here, the Zener anisotropy parameter (A_z) is greater than one i.e. $A_z = 3.0$. Thus the faces of the precipitate prefer to be aligned along 100 directions. The misfit strain is dilatational i.e. $\epsilon^* = 0.01$, $\mu_{mat} = 125$, $\delta = 0.5$. Here, a precipitate with $R = 30$ (equivalent radius); acquires cubic shape with rounded corners as an effect of cubic anisotropy in the elastic energy. The precipitate faces are normal to $<100>$ directions, which are the elastically soft directions. In contrast to this, the precipitates with radius equal 50 possess rectangular morphologies with rounded corners and elongated faces along one of the $<100>$ directions. Depending upon the orientation of the initial configuration of the precipitate, i.e., the ellipsoidal shape for a given equivalent radius, it converges to a shape in which the precipitate symmetry breaks from four fold to two fold along one of the $<100>$ directions.

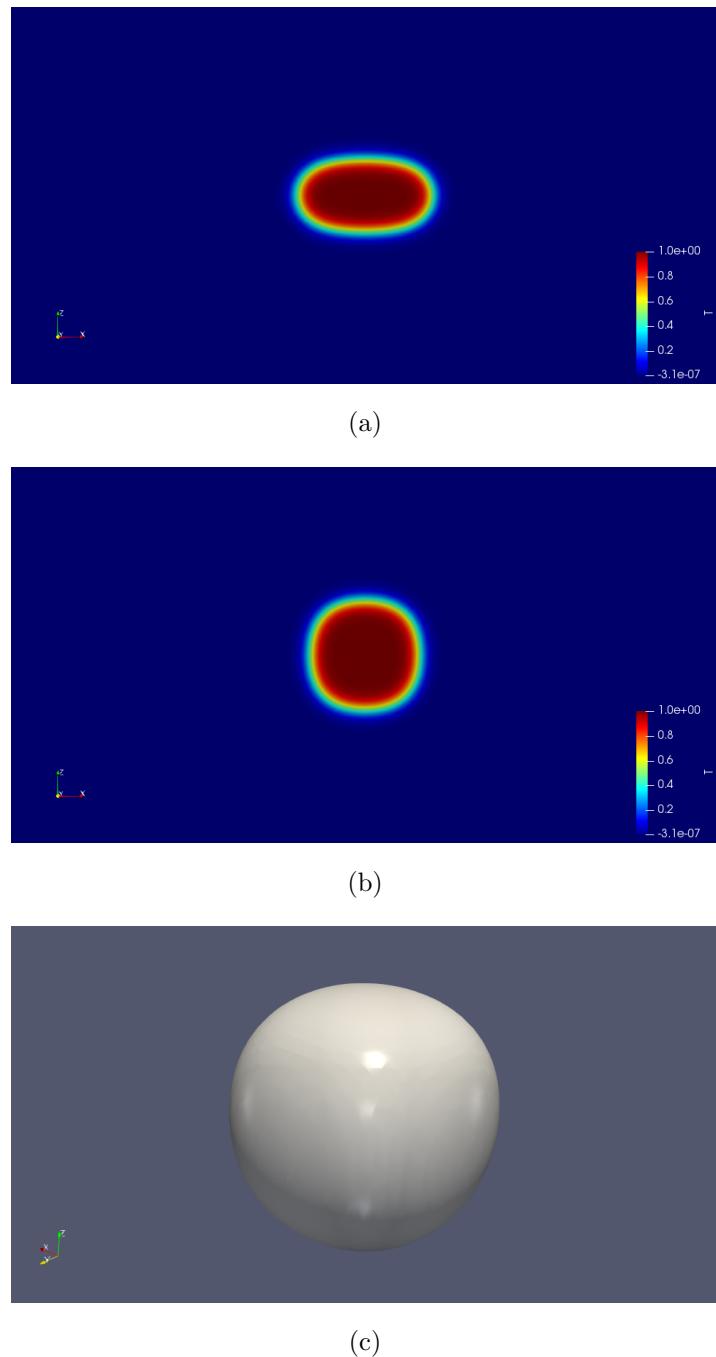


Figure 3.1: $Az=3.0$, $R=30$, 2D section normal to X axis (a) Initial configuration (b) Final configuration (c) 3D equilibrium shape.

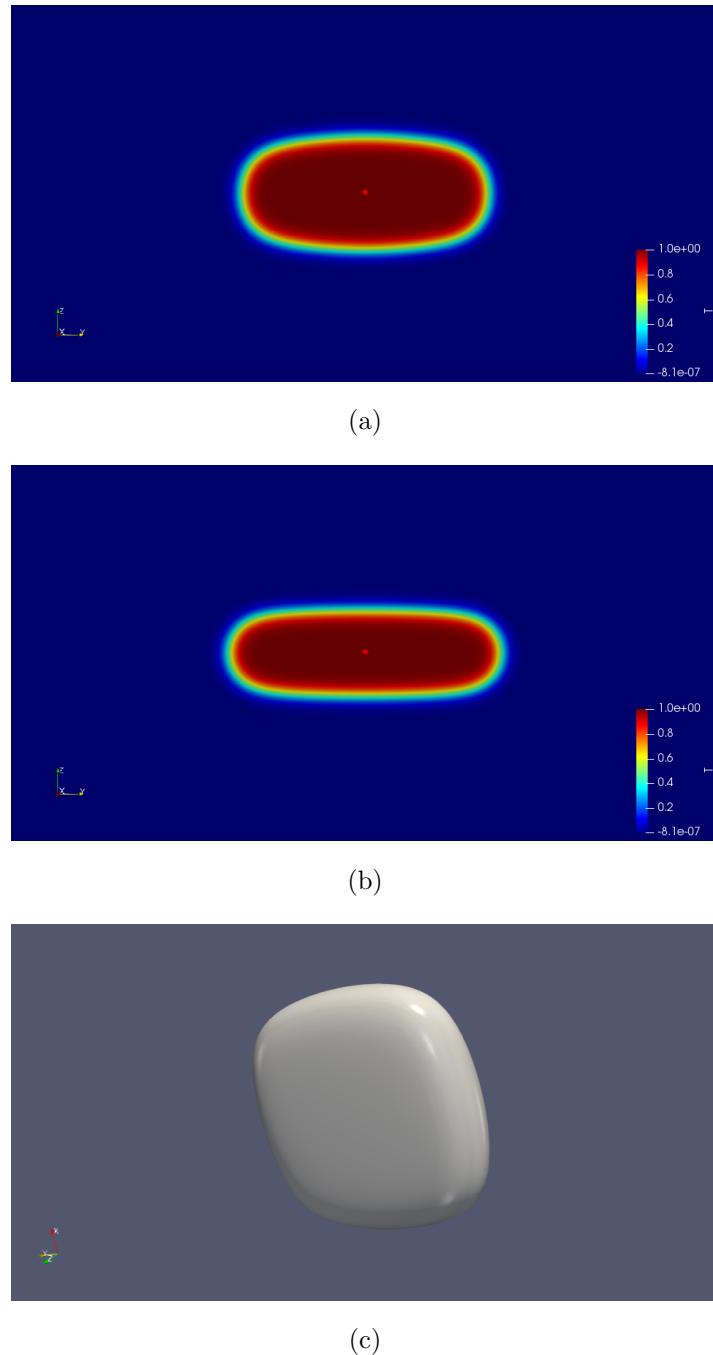


Figure 3.2: $A_z=3.0$, $R=50$, 2D section normal to X axis (a) Initial configuration (b) Final configuration (c) 3D equilibrium shape

3.6.3 Anisotropy in misfit strain

In this case the the elastic energy is isotropic i.e. $A_z=1.0$, $\delta = 1.25$, $\mu_{mat} = 100$ while the misfit strain is not dilatational anymore. The misfit along 010 and 001 (YZ plane) is $\epsilon_{yy}^* = \epsilon_{zz}^* = 0.005$

whereas misfit along 100 direction is twice compared to other two directions i.e. $\epsilon_{xx}^* = 0.01$. Thus, misfit along 100 direction is stiffer compared to other two directions. Thus the precipitate spreads in YZ plane and grows thinner along 100 direction as shown in the below figures:

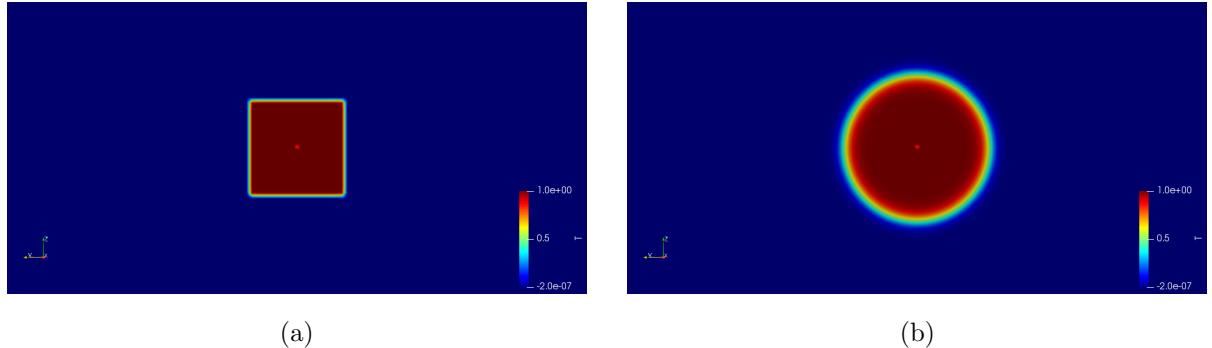


Figure 3.3: 2D section normal to X axis, R=55(a)Initial configuration(b) Final configuration.

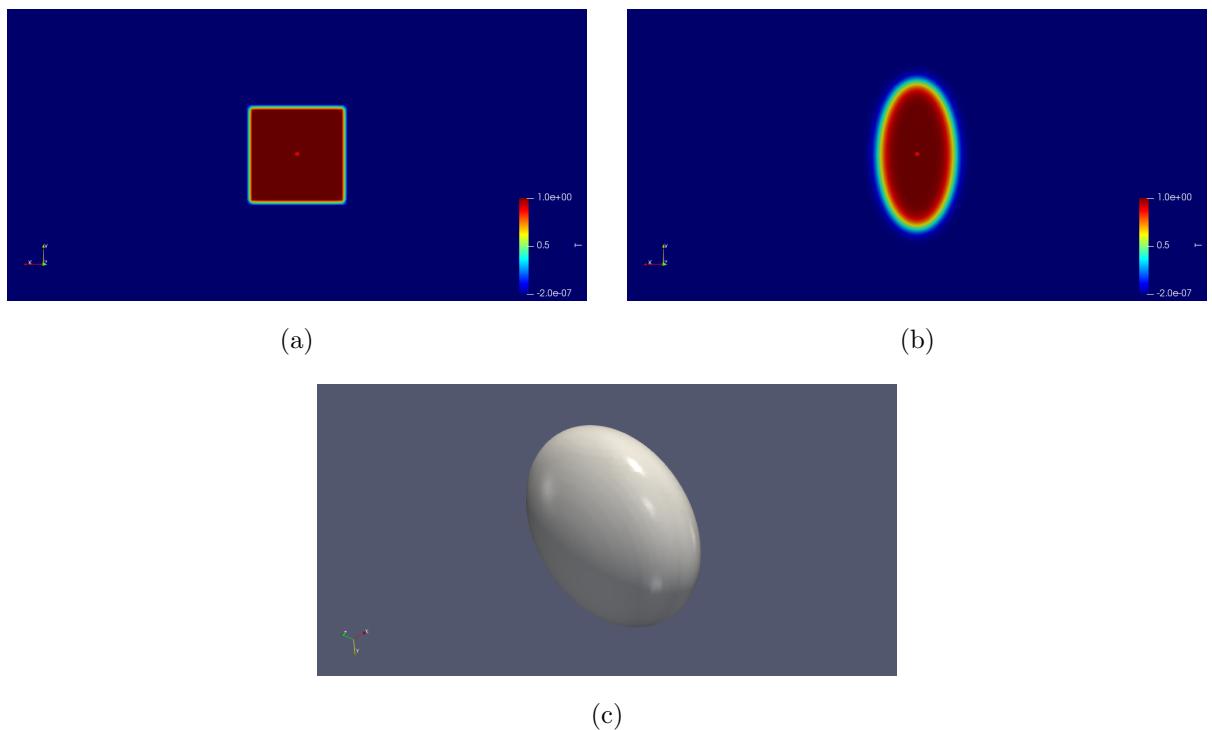


Figure 3.4: 2D section normal to Z axis,R=55(a)Initial configuration(b) Final configuration.(c) 3D equilibrium shape

Chapter 4

Dendrite tip selection:LGK theory and phase-field simulations

4.1 Background

Solidification occurs when liquid is cooled. After nucleation, solidification proceeds by movement of interface. This process generates heat due to release of latent heat. Similarly, solute partitioning happens due to difference of solute solubility in liquid and solid phase. This accumulation of heat and solute ahead of the interface leads to undercooling/supercooling of liquid ahead of solidification front. The interface thus becomes unstable due to thermal noise or impurities and in certain favourable circumstances solidification becomes dendritic. Since the interface is unstable at all points on growth front, it tends to branch. The origin of the term dendrite comes from this branching which gives it a tree-like character.

Dendritic growth involves two important length scale. The first one is R_{tip} which is radius of the dendrite tip and the second is primary arm spacing of array of dendrites during their coordinated growth. R_{tip} is a microstructural scale that is uniquely selected as a function of the growth conditions. It depends upon (a) amount of undercooling (b) imposed thermal gradient (c) velocity during directional solidification conditions (in case of alloys) (d) strength of anisotropy in the solid- liquid interfacial energy. The primary dendrite spacing is also a function of processing parameters that mostly varies depending upon thermal and compositional history. Both of these length scales are however functions of the solute-diffusivities that influence the diffusional length scales [20]. This chapter deals with the development and application of analytical and phase-field simulations to understand the relation between dendrite tip radius (R_{tip}) and undercooling.

The growing dendrite can be defined by the tip radius R_{tip} and the tip velocity v_{tip} . If we consider the tip to be a parabola (in 2-D) or a paraboloid of revolution (in 3-D) with parabolic tip radius R_p and the steady dendrite is isothermal with the solid at the melting temperature, Ivantsov [21] proposed the most widely quoted relationship for dendrite operating state for a purely thermally-controlled growing dendrite, comprising the relationship between external imposed undercooling $\Delta = (T_m - T_\infty) / (L/C_p)$ and the thermal Peclet number at the tip $P_{eT} = \frac{R_p v_{tip}}{2\alpha}$ as $\Delta = I_v(P_{eT})$, where T_m is the melting temperature, T_∞ is the temperature of the undercooled melt, L is the latent of fusion, C_p is the specific heat, α is the thermal diffusivity and $I_v(x) = \sqrt{\pi x} \exp(-x) \operatorname{erfc}(\sqrt{x})$ is the Ivantsov function (in 2-D). Ivantsov's theory predicted that for a given undercooling, there were infinite pairs of (R_{tip}, v_{tip}) for the solution of the linked expression since only their product (Peclet number) could be determined. While convenient, this implication is in conflict with experiment where R_{tip} and v_{tip} are invariant for a fixed undercooling.

Two subsequent approaches were then developed by the introduction of a selection constant defined as σ^* . The marginal stability theory was developed by Langer and Muller-Krumbhaar [22] [23] [24] and involved another relationship between the tip radius and velocity given by Oldfield as [25]:

$$\frac{R_p^2 v_{tip}}{d_0 \alpha} = \left(\frac{1}{\sigma^*}\right)^2 \quad (4.1)$$

where d_0 is the material thermal capillary length. Drawing on a stability analysis based on the allowable shape of a perturbed, non-flat solid-liquid interface, they proposed $\sigma^* = \frac{1}{(2\pi)}$. Langer and Muller-Krumbhaar's [22] [23] [24] analysis goes by the name of marginal stability, and Eqn. 4.1 with $\sigma^* = \frac{1}{(2\pi)}$, is referred to as the marginal stability criterion. Ben-Jacob et al [26] [27] and Kessler et al [28] [29] developed other approaches that allowed for anisotropic surface energy to give a single, paired solution for R_p and v_{tip} , deduced from the fastest growing mode of perturbed solid-liquid interface, which led to an expression similar to the one given by Langer and Muller-Krumbhaar [22] [23] [24] i.e. $R_p^2 v_{tip} = \text{constant}$. Kessler and Levine [28] extended this idea and found that the dendrite tip shape computed in this way generally displays a cusp (non-zero slope) at the tip and at a unique (R_p, v_{tip}) pair; the cusp reduces to a smooth shape with zero slope at the tip, which is called the microscopic solvability condition.

Lipton et al. [30] developed a model which simplifies the analytical procedure for computing a unique pair of (v^*, R_{tip}) for a given fixed undercooling, or a unique pair $(R_{tip}, \Delta T)$ with

an imposed velocity like in constraint growth. This model assumes solid dendrites to be a paraboloid of revolution. The overall undercooling is then given by:

$$\Delta T = \frac{L}{C_p} \Delta_T + \frac{k \Delta T_0 \Delta_c}{1 - (1 - k) \Delta_c} + \frac{2\Gamma}{R_p} \quad (4.2)$$

where the three terms on the right correspond to thermal, solutal and capillary/curvature undercooling, respectively. $\Delta T_0 = \frac{|m|c_\infty(1-k)}{k}$ is the equilibrium freezing range corresponding to c_∞ and Γ is the Gibbs-Thomson coefficient. $\Delta_T = \frac{(T_{tip} - T_\infty)}{(L/C_p)}$ is the dimensionless thermal undercooling. In our case, we have neglected the thermal undercooling term. Thus, the final equation comes out to be:

$$\Delta T = \frac{k \Delta T_0 \Delta_c}{1 - (1 - k) \Delta_c} + \frac{2\Gamma}{R_p} \quad (4.3)$$

At this point we have one equation and two unknowns solutal Peclet number and R_{tip} . Lipton et al.[30] found a way out by assuming that R_{tip} can be estimated from the expression for λ_{min} from linear stability analysis of planar front. The final expression for R_{tip} after performing some algebraic manipulation comes out to be:

$$R_{tip} = 8\pi^2 \Gamma_{sl} \left(-\frac{2Pe_c m_l C_0 (1 - k_0)}{(1 - (1 - k_0) Iv_{2D}(Pe_c))} \right)^{-1} \quad (4.4)$$

Eqs. 4.3 and 4.4 together uniquely determine the tip radius and Peclet number with a given undercooling.

4.2 Research goals

Our objective in this chapter will be to find the variation of dendrite tip radius with undercooling using phase-field simulations and comparing it with analytical results.

4.3 Outline

In the following sections, we firstly describe the model, followed by discussion on the phase-field results. The results obtained through phase-field simulation is then compared with the analytical solutions.

4.4 Physical model

The basic idea of the phase-field model consists in the description of the evolution of individual phases using the corresponding set of order parameters (the phase-field variables). For each

phase, the associated phase-field variable is equal to 1 inside the phase and vanishes to 0 outside the area occupied by the considered phase. The transition of phase fields from 1 to 0 at the phase boundaries is continuous and smooth, i.e., the phase boundaries are represented by diffuse interfaces with a finite thickness. Phase evolution is determined by the phenomenological minimization of the functional, which is formulated as the grand-potential functional [31]. The evolution equation for N phase-field variable can be written as

$$\omega\epsilon \frac{\partial\phi_\alpha}{\partial t} = \epsilon \left(\nabla \cdot \frac{\partial a(\phi, \nabla\phi)}{\partial \nabla\phi_\alpha} - \frac{\partial a(\phi, \nabla\phi)}{\partial \phi_\alpha} \right) - \frac{1}{\epsilon} \frac{\partial w(\phi)}{\partial \phi_\alpha} - \frac{\partial \Psi(T, \mu, \phi)}{\partial \phi_\alpha} \quad (4.5)$$

The double well potential $w(\phi)$ can be written as

$$w(\phi) = \gamma\phi^2(1-\phi)^2 \quad (4.6)$$

We have only two phase i.e; $(\phi_\alpha + \phi_\beta = 1)$. $a(\phi, \nabla\phi)$ is the gradient energy density. In our case a is only function of $(\nabla\phi)$ and has the form

$$a(\nabla\phi) = \gamma[a_c(q)]^2|q|^2 \quad (4.7)$$

where $q = |\nabla\phi|$ is the normal vector to the interface. γ controls the interfacial energy of the system and is known as surface energy density. $a_c(q)$ describes the anisotropy of evolving phase boundary. In case of 4 fold symmetry $a_c(\nabla\phi) = 1 + \delta\cos(4\theta)$ where θ is the orientation of the interface and is computed in 2D as [32]

$$\tan\theta = \left(\frac{\partial\phi}{\partial y} \right) / \left(\frac{\partial\phi}{\partial x} \right) \quad (4.8)$$

where x and y define the coordinate system. Using above expression for θ , anisotropy can be represented as:

$$a_c(\nabla\phi) = 1 - \delta \left[3 - \frac{4((\nabla_x\phi)^4 + (\nabla_y\phi)^4 + (\nabla_z\phi)^4)}{|\nabla\phi|^4} \right] \quad (4.9)$$

where δ is the strength of anisotropy. $\nabla_x\phi$ is the x-component of $\nabla\phi$ and so on. Since a is only function of $\nabla\phi$, thus from the evolution equation of phase field $\frac{\partial a(\phi, \nabla\phi)}{\partial \phi_\alpha} = 0$. The first term on the right hand side of evolution equation is then written as

$$\frac{\partial a}{\partial \nabla\phi} = 2\gamma a_c(\nabla\phi) \frac{\partial a_c}{\partial \nabla\phi} |\nabla\phi|^2 + 2\gamma \nabla\phi [a_c(\nabla\phi)]^2 \quad (4.10)$$

The last term in equation ?? is related to the thermodynamic driving force. We express the grand potentials $\Psi(\mu, T)$ as a linear expansion about the equilibrium chemical potential μ_{eq} :

$$\Psi_\alpha(T, \mu) = \Psi_\alpha(T, \mu_{eq}) + \frac{\partial \Psi_\alpha(T, \mu)}{\partial \mu} \Big|_{\mu_{eq}} (\mu - \mu_{eq}) \quad (4.11)$$

The driving force ΔF_α is then written as

$$\Delta F^\alpha = [\Psi_\alpha(T, \mu) - \Psi_\beta(T, \mu)] \frac{\partial h_\alpha(\phi)}{\partial \phi_\alpha} \quad (4.12)$$

$$= \left(\frac{\partial \Psi_\alpha(T, \mu)}{\partial \mu} \Big|_{\mu_{eq}} - \frac{\partial \Psi_\beta(T, \mu)}{\partial \mu} \Big|_{\mu_{eq}} \right) \\ \times (\mu - \mu_{eq}) \frac{\partial h_\alpha(\phi)}{\partial \phi_\alpha} \quad (4.13)$$

$$= - [c^\alpha(\mu_{eq}, T) - c^\beta(\mu_{eq}, T)] (\mu - \mu_{eq}) \frac{\partial h_\alpha(\phi)}{\partial \phi_\alpha} \quad (4.14)$$

where $h_\alpha(\phi)$ is an interpolation function given as

$$h(\phi) = \phi^2 (6\phi^3 - 15\phi + 10) \quad (4.15)$$

Calculations for growth into an undercooled melt starting from a small solid ($\phi = 0$) seed with appropriate levels of anisotropy typically exhibit needlelike growth with parabolic growth forms in the directions with the smallest values of σ . Calculations performed using a coarse mesh will usually exhibit side branching typical of real dendrites because discretization errors introduce noise into the calculation. (A coarse square mesh can also induce a synthetic four-fold symmetry.) As the computational mesh is refined, however, side branches necessarily disappear. Therefore, noise has been introduced at controlled levels to induce side branching in most simulations. This is typically implemented using random fluctuations of a source term added to the phase-field equation that is localized to regions where ϕ is between 0 and 1; i.e., in the interfacial region. This permits the inclusion of nucleation processes into simulations using the phase-field method[32]. Thus, phase field equation is modified by adding a noise term which is given as

$$\frac{\partial \phi}{\partial t} \rightarrow \frac{\partial \phi}{\partial t} - 6rM\phi^2(1-\phi)^2 \quad (4.16)$$

where r is a random number distributed uniformly between 0 and 1, and a new number is generated for every point of the grid, at each time-step. M is an amplitude of fluctuations

which in our case is taken as 0.003.

The phase-field equation for two phases after incorporating above equation can be given as:

$$\begin{aligned} \omega\epsilon \frac{\partial\phi_\alpha}{\partial t} = & \epsilon \nabla \cdot \frac{\partial a(\nabla\phi)}{\partial \nabla\phi_\alpha} - 18 \frac{\gamma_{\alpha\beta}}{\epsilon} \phi(1-\phi)(1-2\phi_\alpha) \\ & + \frac{1}{2}\epsilon [c^\alpha(\mu_{eq}, T) - c^\beta(\mu_{eq}, T)] \\ & \times (\mu - \mu_{eq}) \frac{\partial h_\alpha(\phi_\alpha)}{\partial \phi_\alpha} + 6rM\phi^2(1-\phi)^2 \end{aligned} \quad (4.17)$$

In this model instead of solving evolution equation for concentration fields, we propose to solve directly for the thermodynamic variable μ , which relate the phase concentrations c_i^α instead of solving for phase concentrations themselves. This is possible because the concentrations $c_i^\alpha(\mu, T)$ are written as explicit functions of the thermodynamic variable μ . The evolution equation for chemical potential can be written as:

$$\begin{aligned} & \left(\frac{\partial c^\alpha(\mu, T)}{\partial \mu} h_\alpha(\phi) + \frac{\partial c^\beta(\mu, T)}{\partial \mu} [1 - h_\alpha(\phi)] \right) \frac{\partial \mu}{\partial t} \\ & = \nabla \cdot \left[\left(D^\alpha g_\alpha(\phi) \frac{\partial c^\alpha(\mu, T)}{\partial \mu} \right. \right. \\ & \quad \left. \left. + D^\beta [1 - g_\alpha(\phi)] \frac{\partial c^\beta(\mu, T)}{\partial \mu} \right) \nabla \mu \right] \\ & \quad - [c^\alpha(\mu, T) - c^\beta(\mu, T)] \frac{\partial h_\alpha(\phi)}{\partial t} \end{aligned} \quad (4.18)$$

where $c^{\alpha,\beta}(\mu)$ are the phase concentrations as functions of the independent chemical potential μ . D^α, D^β are the independent interdiffusivities in the two respective phases. It is noteworthy that this equation looks very similar to the evolution equation of the temperature field in pure materials. The last term on the right-hand side $c^\alpha(\mu, T) - c^\beta(\mu, T)$ corresponds to a source term for rejection of mass at the interface during growth, which is analogous to the release of latent heat in pure material solidification. In this case we have assumed that diffusivity in solid phase is negligible. Thus, $D^\alpha = 0$. This then becomes the case of one sided diffusion.

For the case of one- sided diffusion, it has been shown in various previous works that there exists a thin-interface defect called solute trapping when simulations are performed with interface thicknesses, orders of magnitude larger than those of a real interface. The methodology proposed to correct this effect is the incorporation of an antitrapping current in the evolution equation of the chemical potential. We follow the formulations described in literature [33] [34] and incorporate the antitrapping term as an additional flux of solute from the solid to the liquid in the normal direction to the interface. The modified evolution equation for the chemical

potential along with the antitrapping term is

$$\begin{aligned} & \left(\frac{\partial c^\alpha(\mu, T)}{\partial \mu} h_\alpha(\phi_\alpha) + \frac{\partial c^\beta(\mu, T)}{\partial \mu} [1 - h_\alpha(\phi_\alpha)] \right) \frac{\partial \mu}{\partial t} \\ &= \nabla \cdot \left([1 - g_\alpha(\phi_\alpha)] \frac{\partial c^\beta(\mu, T)}{\partial \mu} \nabla \mu - j_{at} \right) \\ & - [c^\alpha(\mu, T) - c^\beta(\mu, T)] \frac{\partial h_\alpha(\phi_\alpha)}{\partial t} \end{aligned} \quad (4.19)$$

To make sure that the antitrapping current appears in the first-order correction to the chemical potential, we formulate the antitrapping current of the following form

$$j_{at} = s(\phi_\alpha) \epsilon [c^\beta(\mu, T) - c^\alpha(\mu, T)] \frac{\partial \phi_\alpha}{\partial t} \frac{q_{\alpha\beta}}{|q_{\alpha\beta}|} \quad (4.20)$$

where $s(\phi_\alpha)$ is a function such that the chemical potential jump vanishes at the interface. $q_{\alpha\beta}$ is the normal vector to the interface given by $(\phi_\alpha \nabla \phi_\beta - \phi_\beta \nabla \phi_\alpha)$. For the case of only two phases, the expression of the antitrapping current can be reduced to

$$j_{at} = -s(\phi_\alpha) \epsilon [c^\beta(\mu, T) - c^\alpha(\mu, T)] \frac{\partial \phi_\alpha}{\partial t} \frac{\nabla \phi_\alpha}{|\nabla \phi_\alpha|} \quad (4.21)$$

All terms in the above equation are used in the nondimensional form, so ϵ is the nondimensional parameter related to the interface width and t is the nondimensional time. $s(\phi)$ is a shape function which in our case is taken as a constant value of $\frac{1}{2\sqrt{2}}$.

4.5 Implementation in OpenFOAM

The phase-field equation and chemical potential equation is implemented in OpenFOAM as:

```
Random obj(1);
const scalar randNumber(obj.scalar01());
#include "dAdgradPhi_mod.H"
fvScalarMatrix alphaEqn
(
    omega*epsilon*dimt*fvm::ddt(alpha)           //alpha=1 liquid phase
    ==
    2.0*epsilon*gamma*dimx*dimx*fvm::laplacian(ac_01*ac_01, alpha)
    + 2*gamma*epsilon*dimx*fvc::div(dadgradPhi)
    - 18.0*(gamma/epsilon)*(alpha)*(1-alpha)*(1-2.0*alpha)
    + 6*noise_mag*alpha*(1.0-alpha)*alpha*(1-alpha)*randNumber
```

```

-(c_Sol-c_Liq)*(mu - (2*A*c_eq - 2*A*(T0-T)/m_1))
*30.0*alpha*alpha*(1.0-alpha)*(1.0-alpha)

);

alphaEqn.solve();

volVectorField n=dimx*fvc::grad(alpha)
/(1E-20+mag(dimx*fvc::grad(alpha)));
fvScalarMatrix muEqn
(
    (0.5)*dimt*fvm::ddt(mu)
    ==
    diff_Liq*0.5*dimx*dimx*fvm::laplacian(alpha,mu)
    + (c_Sol-c_Liq)* dimt*fvc::ddt(alpha)
    *30.0*alpha*alpha*(1.0-alpha)*(1.0-alpha)
    - anti_trap*epsilon*(c_Sol-c_Liq)*dimx
    *fvc::div((n*dimt*fvc::ddt(alpha)))
);

muEqn.solve();
Info<< "Min/max-alpha:" << min(alpha).value() << ''
<< max(alpha).value() << endl;

```

dAdgradPhi_mod.H file solves for gradient energy density and anisotropy of the of evolving phase boundary. It is written in OpenFOAM code as:

```

volVectorField q=dimx*fvc::grad(alpha);
Info<< "Min/max-magq:" << min(mag(q)).value() << ''
<< max(mag(q)).value() << endl;
ac_01 = 1.0 - delta_01*(3.0 - 4.0*(pow(q.component(0),4)
+ pow(q.component(1), 4)
+ pow(q.component(2), 4))/(1E-20 +pow(mag(q),4)));

```

```

dAdq01 = 16.0*delta_01*((pow(q.component(0),3)
/(1E-20 +pow(mag(q),4))
- q.component(0)*(pow(q.component(0),4)
+ pow(q.component(1), 4)+ pow(q.component(2), 4))
/(1E-20 +pow(mag(q),6)))*vector(1,0,0)

+(pow(q.component(1),3)/(1E-20 +pow(mag(q),4))
- q.component(1)*(pow(q.component(0),4)
+ pow(q.component(1), 4)+ pow(q.component(2), 4))
/(1E-20 +pow(mag(q),6)))*vector(0,1,0)

+(pow(q.component(2),3)/(1E-20 +pow(mag(q),4))
- q.component(2)*(pow(q.component(0),4)
+ pow(q.component(1), 4) + pow(q.component(2), 4))
/(1E-20 +pow(mag(q),6)))*vector(0,0,1));

dadgradPhi = (ac_01*dAdq01*mag(q)*mag(q));

Info<< "Min/max-ac_01:" << min(ac_01).value() << ''
<< max(ac_01).value() << endl;

Info<< "Min/max-dadgradPhi:" << min(dadgradPhi).value() << ''
<< max(dadgradPhi).value() << endl;

```

4.5.1 Analytical code in Matlab

```

m=0.45;
global c0;
c0=0.5;
k=0.78125/0.5
Tm=1;
k1=0.45/0.56;
global Delta_T
for Delta_T=0.065
term=@lgk;
pe0=0.003;

```

```

y=fsolve(term,pe0);
Iv=sqrt(pi*y)*exp(y)*(1-erf(sqrt(y)));
R=k1*((-m*c0*Iv*(k-1))/(1-Iv*(1-k))+Delta_T)^(1)
end
function y=lgk(pe)
m=0.45;
global c0;
k=0.78125/0.5;
global Delta_T;
y=-m*c0*(1-k)*(pe/(2*pi*pi)+sqrt(pi*pe)*exp(pe)*(1-erf(sqrt(pe))))/
(Delta_T*(1-(1-k)*sqrt(pi*pe)*exp(pe)*(1-erf(sqrt(pe)))))^1
end

```

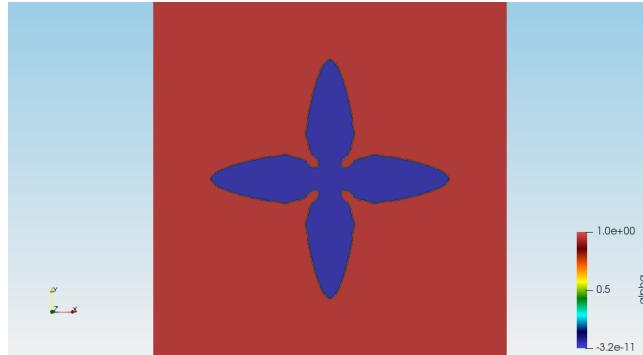
4.6 Numerical test case based on Phase-field model

Two test cases are made. First test case is used to study the effect of isothermal undercooling on dendrite tip radius. This result is then compared with analytically results obtained by solving Eqns.?? 4.2. Second test involves modelling of directionally solidified dendrite when subjected to a positive thermal gradient ahead of the solid-liquid interface. The domain size used for first case is 2000×2000 with $\Delta x = 12$. The following table shows the non dimensional values of different variables used in this simulation. Three different undercooling is studied for the first

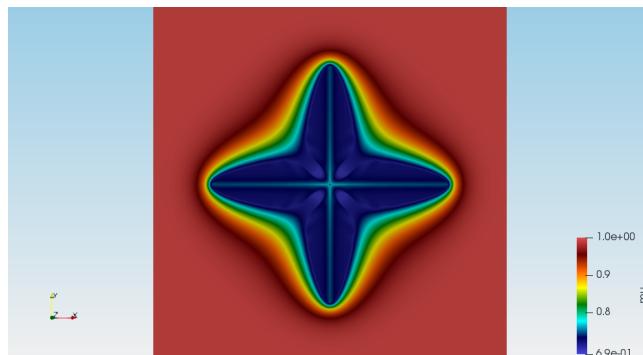
Properties	Values
Slope liquidus(m_1)	0.45
Slope solidus(m_0)	0.45
Relaxation coefficient(omega)	1.687
Surface energy(gamma)	1
Interface width(epsilon)	48
Composition of solid(c_Sol)	0.78
Composition of liquid(c_Liq)	0.45
Equilibrium composition(c_eq)	0.50
Anti trapping coefficient(anti_trap)	0.35355
Diffusivity in liquid(diff_Liq)	1
Thermal gradient(G)	1.7E-05
Velocity (v)	0.001
Strength of anisotropy (delta_01)	0.02
Melting Temp (T0)	1

Table 4.1: Parameters used in this simulation

case. After running the simulation in OpenFOAM, the data obtained is used for dendrite tip radius calculation. A python script (AppendixA) is used to fit the dendrite tip on parabola. This is carried out at regular time steps until the steady state dendrite tip radius is achieved.



(a)



(b)

Figure 4.1: Final profile for $\Delta T=0.065$ (a)Order parameter, α (b) Chemical potential, μ .

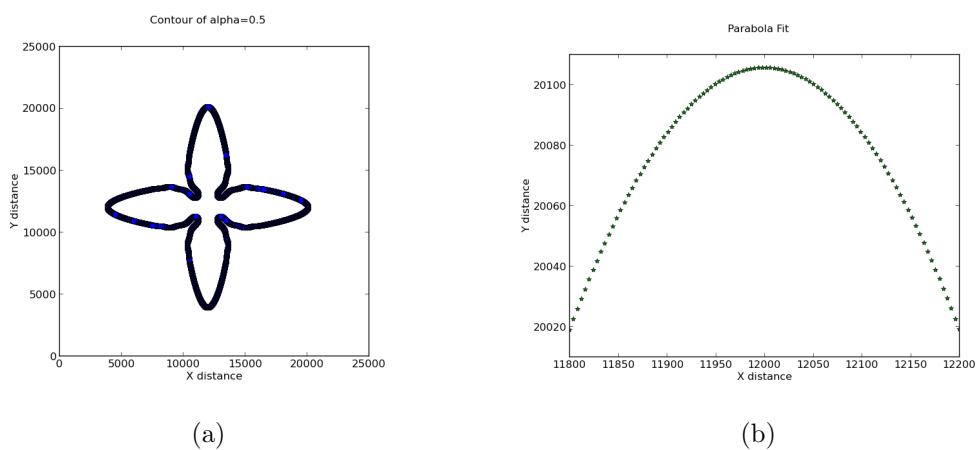


Figure 4.2: Final profile for $\Delta T=0.065$ (a) α contour at 0.5 (b) Parabolic fit at dendrite tip.

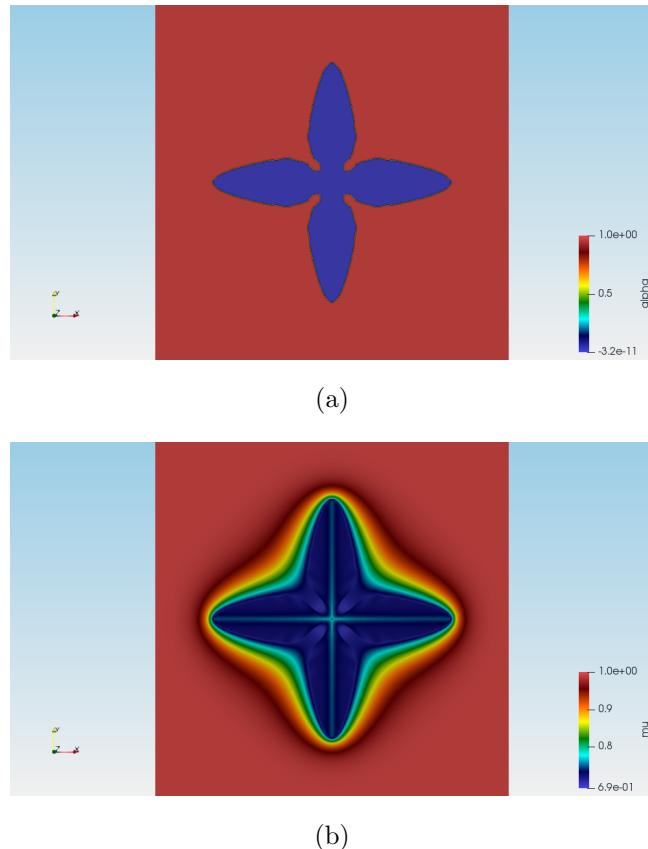


Figure 4.3: Final profile for $\Delta T = 0.065$ (a) Order parameter, α (b) Chemical potential, μ .

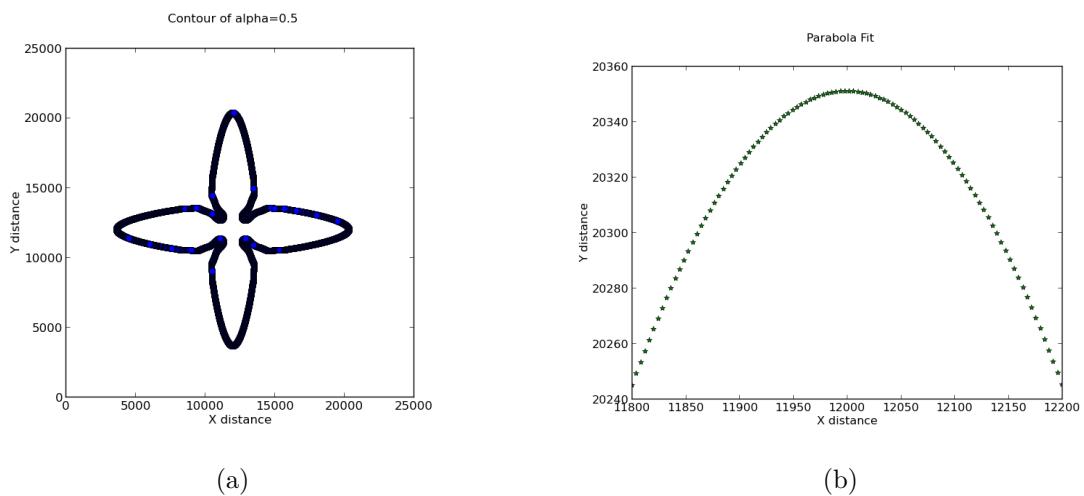


Figure 4.4: Final profile for $\Delta T = 0.07$ (a) α contour at 0.5 (b) Parabolic fit at dendrite tip.

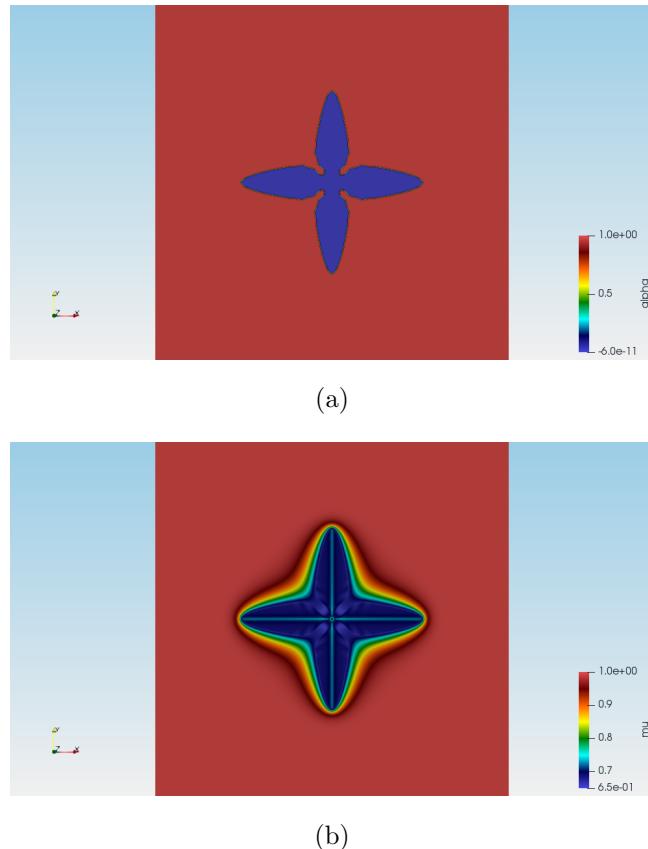


Figure 4.5: Final profile for $\Delta T=0.075$ (a) Order parameter, α (b) Chemical potential, μ .

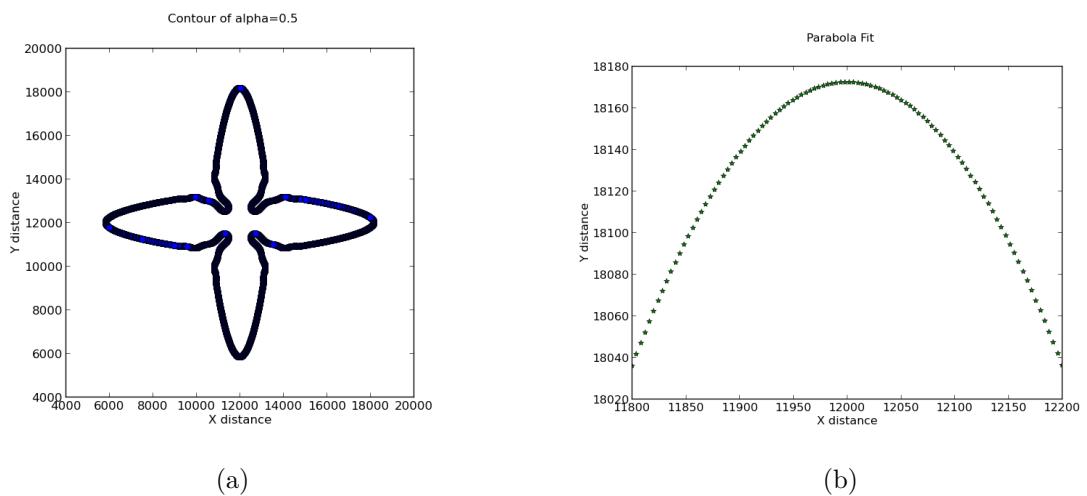


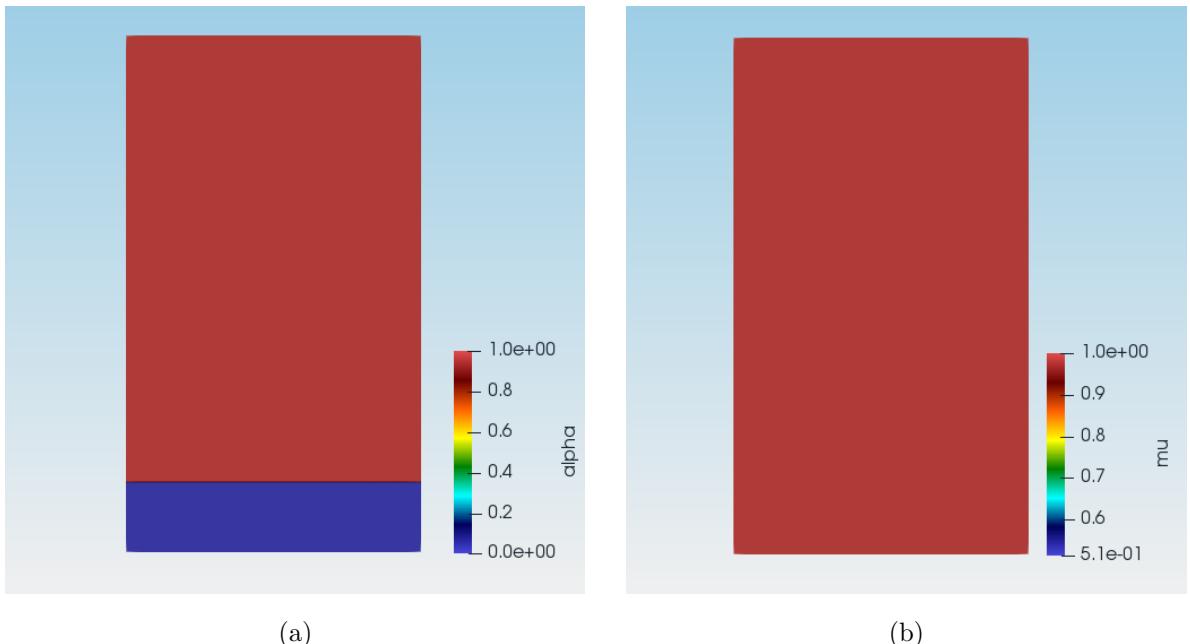
Figure 4.6: Final profile for $\Delta T=0.075$ (a) α contour at 0.5 (b) Parabolic fit at dendrite tip.

ΔT	R_{tip} Analytical (LGK model)	R_{tip} Phase-field simulation
0.065	295	230
0.07	204	188
0.075	141	146

Table 4.2: Comparison of dendrite tip radius using different methods

It can be easily observed from Table 4.2 that with an increase in undercooling, the equilibrium dendrite tip radius decreases. The simulation results are of the same order as the analytical solution.

In first case an uniform undercooling was given to the system which chooses its own velocity at equilibrium. This is known as free growth. In the second case a uniform velocity as well as thermal gradient is applied to the system and its response is observed. We have used a positive thermal gradient in this case. This is a simulation of directional solidification. The domain used in this is 3500×2000 with $\Delta x=12$. The parameters used are given in Table 4.1. The order parameter as well as the chemical potential at different time steps obtained from OpenFOAM simulation is as below:

Figure 4.7: Profile at $T=1$ (a)Order parameter, α (b) Chemical potential, μ .

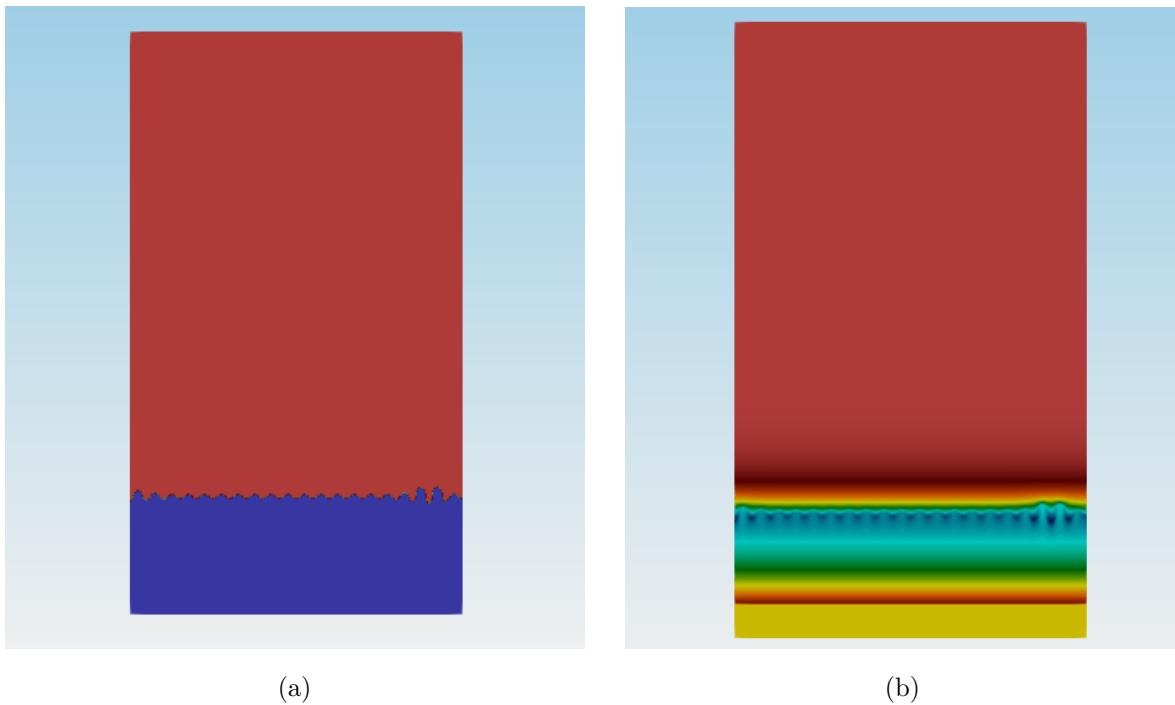


Figure 4.8: Profile at $T=3$ (a)Order parameter, α (b) Chemical potential, μ .

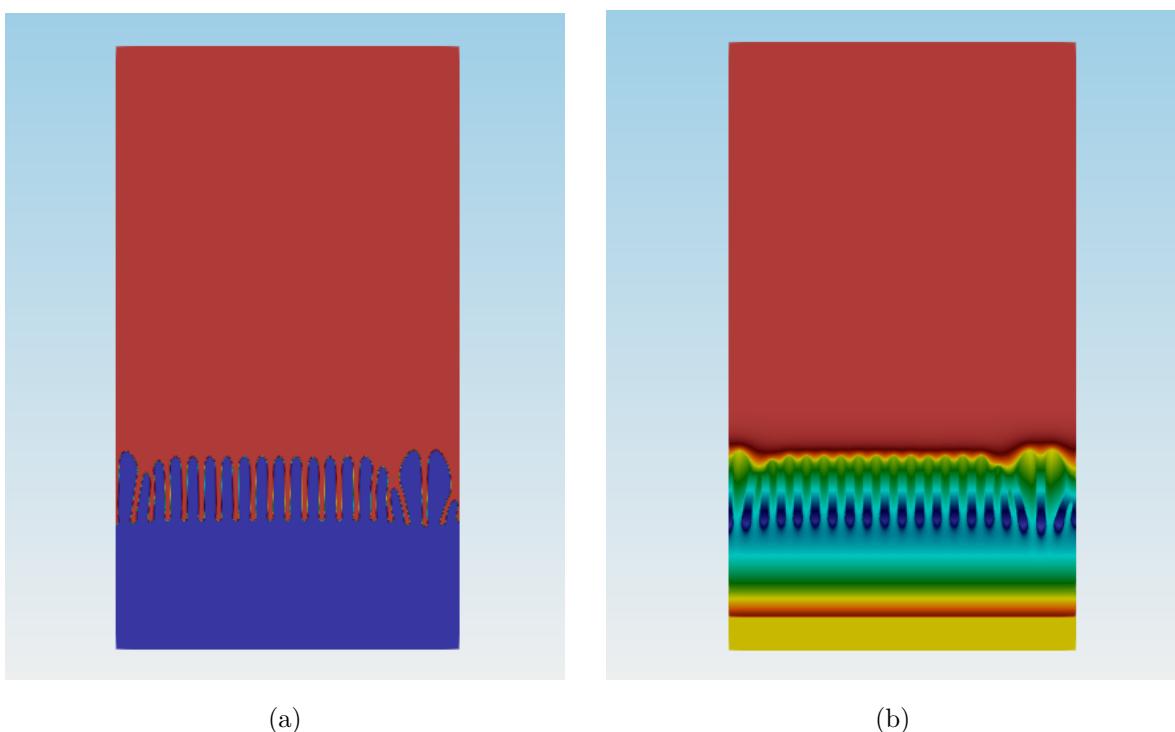


Figure 4.9: Profile at $T=5$ (a)Order parameter, α (b) Chemical potential, μ .

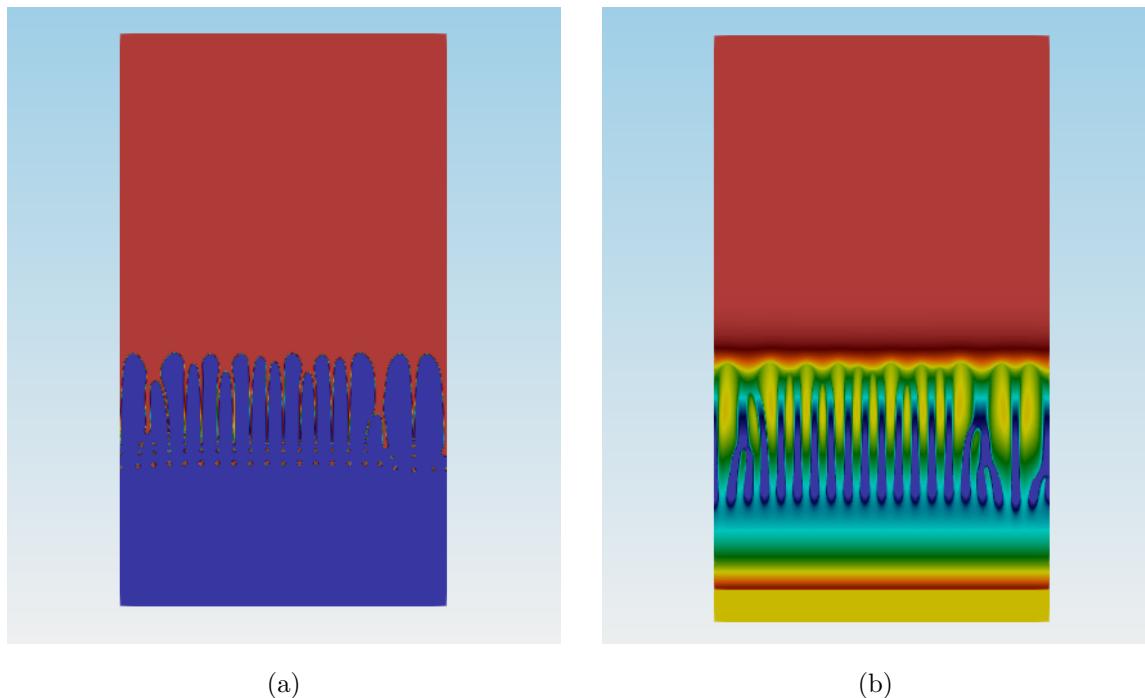


Figure 4.10: Profile at $T=10$ (a)Order parameter, α (b) Chemical potential, μ .

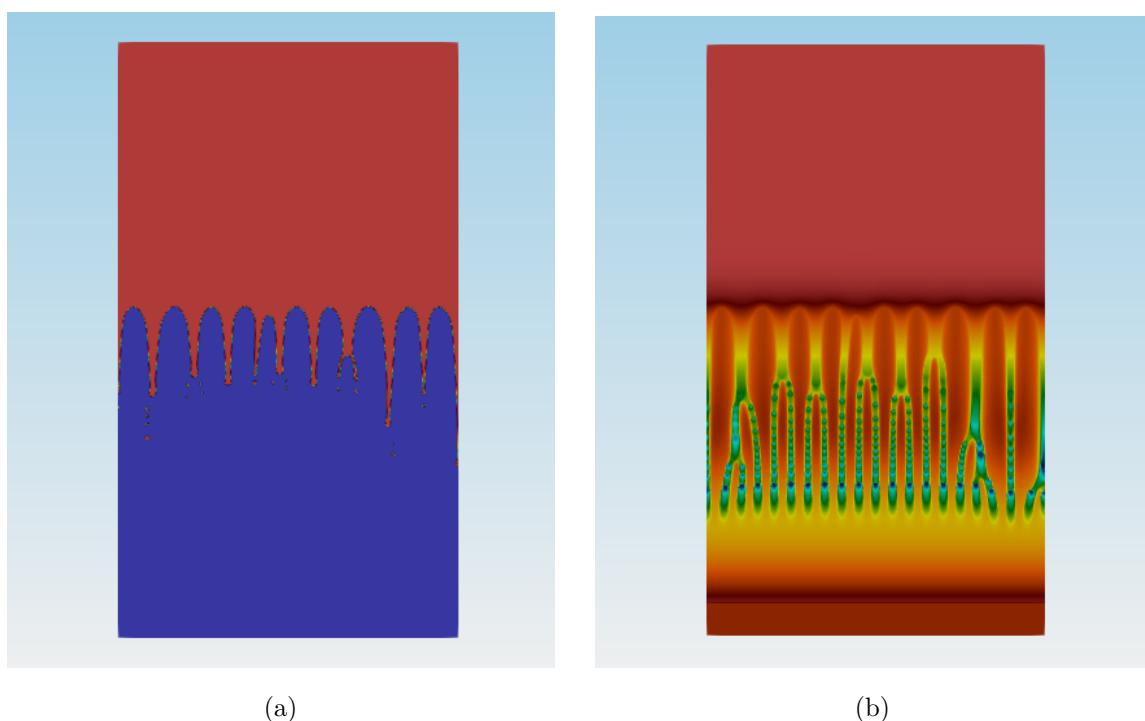


Figure 4.11: Profile at $T=15$ (a)Order parameter, α (b) Chemical potential, μ .

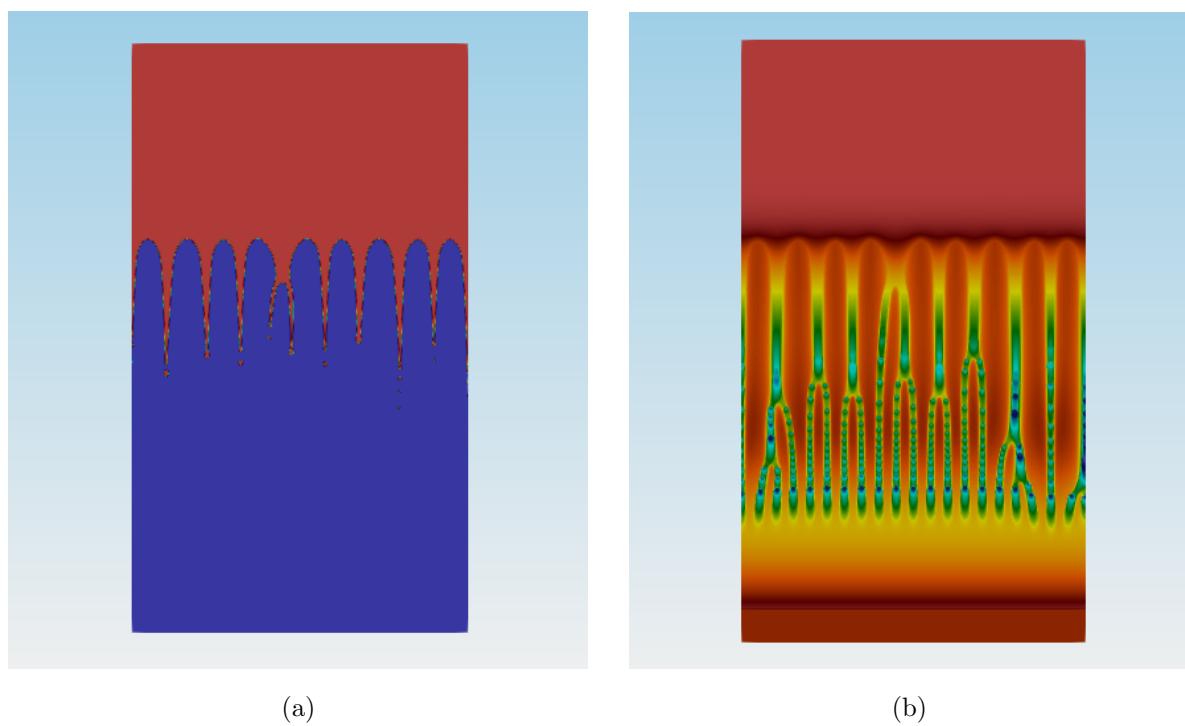


Figure 4.12: Profile at $T=20$ (a)Order parameter, α (b) Chemical potential, μ .

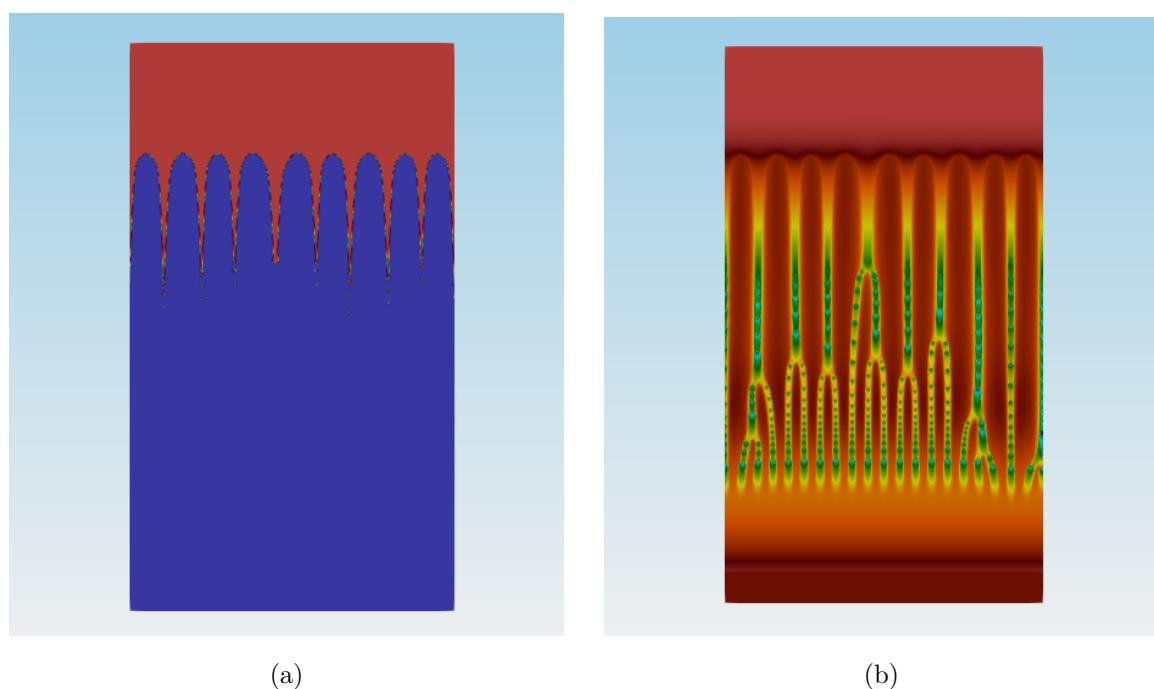


Figure 4.13: Profile at $T=25$ (a)Order parameter, α (b) Chemical potential, μ .

It can be clearly seen that the interface becomes unstable after few timesteps due to the perturbation given in the form of noise. A large number of initial dendrite arms starts growing together. After some more time, the system tries to take only maximally growing wavelength and other wavelengths dies down. As solidification proceeds, the solute rejected from the growing dendrites gets segregated in inter-dendritic region. This segregated region can be clearly observed in μ profile. Due to this segregation, the undercooling available in these spaces comes down. Thus, inter-dendritic spacing is the region which solidifies at last.

Chapter 5

Conclusions

We have studied both microscopic as well as macroscopic phenomena using OpenFOAM software.

1. In first chapter modelling of directional solidification setup was done to find out the thermal gradient at solid-liquid interface. This gradient is then further used for determining the critical velocity above which the interface becomes unstable. Experiments were carried out to validate the results obtained from the model. The experimental obervations were in line with the modelling outputs.
2. Chapter 2 deals with the modelling of state of the art technology in Additive Manufacturing. This chapter brings out different aspects involved in additive manufacturing. Alongwith shape and size of weld pool, thermal distortion and residual stress calculation were carried out assuming a elastic perfectly plastic material. The themal results obtained are useful to determine the layer thickness. The residual stress results are useful for further studying fatigue crack initiation/propagation and hot cracking susceptibility.
3. Precipitate strengthened alloys finds its application in high-temperature environment, where, the strengthening is achieved through interaction between precipitate and dislocation. Mostly precipitate size, morphologies, and their distribution drives the mechanical properties of these high temperature alloys. In third chapter phase-field model is used to determine equilibrium shape of precipitates. Different cases are studied with anisotropy in elastic energy and anisotropy in misfit strain. These different shapes of precipitates plays an important role in interactions with dislocations thereby influencing the mechanical properties.
4. This is again a microscopic modelling in which phase-filed simulations results are compared

with the analytical solutions to determine the steady state dendrite tip radius in an isothermally undercooled sample. The results obtained by modelling are of the same order as phase-field simulation outputs. There is an increase in dendrite tip radius with increase in undercooling. A case of directional solidification is also studied by giving a uniform thermal gradient and velocity. The profiles of chemical potential and order parameter were observed at different saved intervals.

Appendices

Appendix A

```
# trace generated using paraview version 5.6.0
#
# To ensure correct image size when batch processing, please search
# for and uncomment the line '# renderView*.ViewSize = [*,*] '
from __future__ import print_function
#### import the simple module from the paraview
from paraview.simple import *
#### disable automatic camera reset on 'Show'
import sys
#sys.path.insert
(0, '/opt/ParaView-5.2.0-Qt4-OpenGL2-MPI-Linux-64bit/lib/python2.7/site-
                                packages')
import numpy as np
import matplotlib.pyplot as plt
from vtk.numpy_interface import dataset_adapter as dsa
from vtk.numpy_interface import algorithms as algs
# for taking all files as input
import os
import glob
import re
paraview.simple._DisableFirstRenderCameraReset()

# create a new 'OpenFOAMReader'
foamfoam = OpenFOAMReader(FileName='foam.foam')
foamfoam.MeshRegions = ['internalMesh']

# Properties modified on foamfoam
foamfoam.CaseType = 'Decomposed Case'

# get active view
renderView1 = GetActiveViewOrCreate('RenderView')
```

```
# uncomment following to set a specific view size
# renderView1.ViewSize = [993, 281]

# show data in view
foamfoamDisplay = Show(foamfoam, renderView1)

# trace defaults for the display properties.
foamfoamDisplay.Representation = 'Surface'
foamfoamDisplay.ColorArrayName = [None, '']
foamfoamDisplay.OSPRayScaleArray = 'alpha'
foamfoamDisplay.OSPRayScaleFunction = 'PiecewiseFunction'
foamfoamDisplay.SelectOrientationVectors = 'None'
foamfoamDisplay.ScaleFactor = 2400.0
foamfoamDisplay.SelectScaleArray = 'None'
foamfoamDisplay.GlyphType = 'Arrow'
foamfoamDisplay.GlyphTableIndexArray = 'None'
foamfoamDisplay.GaussianRadius = 120.0
foamfoamDisplay.SetScaleArray = ['POINTS', 'alpha']
foamfoamDisplay.ScaleTransferFunction = 'PiecewiseFunction'
foamfoamDisplay.OpacityArray = ['POINTS', 'alpha']
foamfoamDisplay.OpacityTransferFunction = 'PiecewiseFunction'
foamfoamDisplay.DataAxesGrid = 'GridAxesRepresentation'
foamfoamDisplay.SelectionCellLabelFontFile =
foamfoamDisplay.SelectionPointLabelFontFile =
foamfoamDisplay.PolarAxes = 'PolarAxesRepresentation'
foamfoamDisplay.ScalarOpacityUnitDistance = 213.81569235902685

# init the 'GridAxesRepresentation' selected for 'DataAxesGrid'
foamfoamDisplay.DataAxesGrid.XTitleFontFile =
foamfoamDisplay.DataAxesGrid.YTitleFontFile =
foamfoamDisplay.DataAxesGrid.ZTitleFontFile =
foamfoamDisplay.DataAxesGrid.XLabelFontFile =
foamfoamDisplay.DataAxesGrid.YLabelFontFile =
foamfoamDisplay.DataAxesGrid.ZLabelFontFile =

# init the 'PolarAxesRepresentation' selected for 'PolarAxes'
foamfoamDisplay.PolarAxes.PolarAxisTitleFontFile =
foamfoamDisplay.PolarAxes.PolarAxisLabelFontFile =
foamfoamDisplay.PolarAxes.LastRadialAxisTextFontFile =
foamfoamDisplay.PolarAxes.SecondaryRadialAxesTextFontFile =

# reset view to fit data
```

```
renderView1.ResetCamera()

# get the material library
materialLibrary1 = GetMaterialLibrary()

# get animation scene
animationScene1 = GetAnimationScene()

# update animation scene based on data timesteps
animationScene1.UpdateAnimationUsingDataTimeSteps()
times = foamfoam.TimestepValues
n = 22
renderView1.ViewTime = times[n]
# create a new 'Contour'
contour1 = Contour(Input=foamfoam)
contour1.ContourBy = ['POINTS', 'alpha']
contour1.Isosurfaces = [0.49999998134471824]
contour1.PointMergeMethod = 'Uniform Binning'

# Properties modified on foamfoam
foamfoam.CellArrays = ['alpha', 'mu']

# Properties modified on contour1
contour1.Isosurfaces = [0.5]

# show data in view
contour1Display = Show(contour1, renderView1)

# trace defaults for the display properties.
contour1Display.Representation = 'Surface'
contour1Display.ColorArrayName = [None, '']
contour1Display.OSPRayScaleArray = 'Normals'
contour1Display.OSPRayScaleFunction = 'PiecewiseFunction'
contour1Display.SelectOrientationVectors = 'None'
contour1Display.ScaleFactor = 411.15410156250005
contour1Display.SelectScaleArray = 'None'
contour1Display.GlyphType = 'Arrow'
contour1Display.GlyphTableIndexArray = 'None'
contour1Display.GaussianRadius = 20.557705078125
contour1Display.SetScaleArray = ['POINTS', 'Normals']
contour1Display.ScaleTransferFunction = 'PiecewiseFunction'
contour1Display.OpacityArray = ['POINTS', 'Normals']
```

```

contour1Display.OpacityTransferFunction = 'PiecewiseFunction'
contour1Display.DataAxesGrid = 'GridAxesRepresentation'
contour1Display.SelectionCellLabelFontFile =
contour1Display.SelectionPointLabelFontFile =
contour1Display.PolarAxes = 'PolarAxesRepresentation'

# init the 'GridAxesRepresentation' selected for 'DataAxesGrid'
contour1Display.DataAxesGrid.XTitleFontFile =
contour1Display.DataAxesGrid.YTitleFontFile =
contour1Display.DataAxesGrid.ZTitleFontFile =
contour1Display.DataAxesGrid.XLabelFontFile =
contour1Display.DataAxesGrid.YLabelFontFile =
contour1Display.DataAxesGrid.ZLabelFontFile =

# init the 'PolarAxesRepresentation' selected for 'PolarAxes'
contour1Display.PolarAxes.PolarAxisTitleFontFile =
contour1Display.PolarAxes.PolarAxisLabelFontFile =
contour1Display.PolarAxes.LastRadialAxisTextFontFile =
contour1Display.PolarAxes.SecondaryRadialAxesTextFontFile =

# hide data in view
Hide(foamfoam, renderView1)

# update the view to ensure updated data information
renderView1.Update()

# change representation type
contour1Display.SetRepresentationType('Wireframe')

#### saving camera placements for all active views

# current camera placement for renderView1
renderView1.CameraPosition = [12000.0, 12000.0, 65569.2193832923]
renderView1.CameraFocalPoint = [12000.0, 12000.0, 0.0]
renderView1.CameraParallelScale = 16970.562748901404

rawData = servermanager.Fetch(contour1)
data = dsa.WrapDataObject(rawData)

size_arr = np.size( (data.PointData['mu']) )
x_interface, z_interface = np.zeros(size_arr), np.zeros(size_arr)
count = 0

```

```

for i in range(size_arr):
    x_interface[count] = data.Points.Arrays[0][i][0]
    z_interface[count] = data.Points.Arrays[0][i][1]
    count += 1

plt.plot(x_interface, z_interface, 'o')
plt.gca().set_aspect('equal', adjustable='box')
plt.suptitle('Contour of alpha=0.5')
plt.xlabel('X distance')
plt.ylabel('Y distance')
plt.savefig('contour_1_' + str(n) + '.png')
plt.close()

x_para = []
z_para = []

for i in range(size_arr):
    if x_interface[i] > 11800 and x_interface[i] < 12200:
        if z_interface[i] > 15000:
            x_para.append(x_interface[i])
            z_para.append(z_interface[i])

x_para_np = np.asarray(x_para)
z_para_np = np.asarray(z_para)

para_coeffs = np.polyfit(x_para_np, z_para_np, 2)
para_func = np.poly1d(para_coeffs)

x_axis = np.linspace(11800, 12200, 100)
plt.plot(x_axis, para_func(x_axis), '*g')
plt.ticklabel_format(useOffset=False)
plt.suptitle('Parabola Fit')
plt.xlabel('X distance')
plt.ylabel('Y distance')
plt.savefig('para_fit_1_' + str(n) + '.png')
para_a = para_coeffs[0]
para_b = para_coeffs[1]
para_c = para_coeffs[2]

den = (1 + (2*para_a*12000 + para_b)**2)**1.5
num = 2*para_a
curvature = num/den

```

```
radius_para = abs(1/curvature)
print(renderView1.ViewTime, radius_para)

##### uncomment the following to render all views
# RenderAllViews()
# alternatively, if you want to write images, you can use SaveScreenshot
# (...).

##### uncomment the following to render all views
# RenderAllViews()
# alternatively, if you want to write images, you can use SaveScreenshot
# (...).
```

Bibliography

- [1] P. W. Bridgman, “Certain physical properties of single crystals of tungsten, antimony, bismuth, tellurium, cadmium, zinc, and tin,” in *Proceedings of the American Academy of Arts and Sciences*, vol. 60, pp. 305–383, JSTOR, 1925. [1](#)
- [2] Q. Xu, H. Zhang, X. Qi, and B. Liu, “Multiscale modeling and simulation of directional solidification process of turbine blade casting with mca method,” *Metallurgical and Materials Transactions B*, vol. 45, no. 2, pp. 555–561, 2014. [1](#)
- [3] W. W. Mullins and R. Sekerka, “Stability of a planar interface during solidification of a dilute binary alloy,” in *Dynamics of Curved Fronts*, pp. 345–352, Elsevier, 1988. [8](#)
- [4] Z. S. Saldi, “Marangoni driven free surface flows in liquid weld pools,” 2012. [v](#), [11](#)
- [5] J. A. Dantzig, “Modelling liquid–solid phase changes with melt convection,” *International Journal for Numerical Methods in Engineering*, vol. 28, no. 8, pp. 1769–1785, 1989. [12](#)
- [6] X. He, P. Fuerschbach, and T. DebRoy, “Heat transfer and fluid flow during laser spot welding of 304 stainless steel,” *Journal of Physics D: Applied Physics*, vol. 36, no. 12, p. 1388, 2003. [12](#)
- [7] V. R. Voller and C. Prakash, “A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems,” *International Journal of Heat and Mass Transfer*, vol. 30, no. 8, pp. 1709–1719, 1987. [13](#)
- [8] A. Brent, V. Voller, and K. Reid, “Enthalpy-porosity technique for modeling convection-diffusion phase change: application to the melting of a pure metal,” *Numerical Heat Transfer, Part A Applications*, vol. 13, no. 3, pp. 297–318, 1988. [13](#)
- [9] V. Voller and C. Swaminathan, “Eral source-based method for solidification phase change,” *Numerical Heat Transfer, Part B Fundamentals*, vol. 19, no. 2, pp. 175–189, 1991. [13](#)

- [10] V. Manvatkar, A. De, and T. DebRoy, “Spatial variation of melt pool geometry, peak temperature and solidification parameters during laser assisted additive manufacturing process,” *Materials Science and Technology*, vol. 31, no. 8, pp. 924–930, 2015. [20](#)
- [11] T. DebRoy and S. David, “Physical processes in fusion welding,” *Reviews of modern physics*, vol. 67, no. 1, p. 85, 1995. [20](#)
- [12] T. Tang, “Implementation of solid body stress analysis in openfoam,” 2013. [21](#)
- [13] T. Mukherjee, W. Zhang, and T. DebRoy, “An improved prediction of residual stresses and distortion in additive manufacturing,” *Computational Materials Science*, vol. 126, pp. 360–372, 2017. [viii, 27, 28, 36](#)
- [14] B. Song, S. Dong, H. Liao, and C. Coddet, “Process parameter selection for selective laser melting of ti6al4v based on temperature distribution simulation and experimental sintering,” *The International Journal of Advanced Manufacturing Technology*, vol. 61, no. 9-12, pp. 967–974, 2012. [29](#)
- [15] N. Moelans, B. Blanpain, and P. Wollants, “An introduction to phase-field modeling of microstructure evolution,” *Calphad*, vol. 32, no. 2, pp. 268–294, 2008. [37](#)
- [16] P. Wu, X. Ma, J. Zhang, and L. Chen, “Phase-field model of multiferroic composites: Domain structures of ferroelectric particles embedded in a ferromagnetic matrix,” *Philosophical Magazine*, vol. 90, no. 1-4, pp. 125–140, 2010. [37](#)
- [17] W. Johnson and J. Cahn, “Elastically induced shape bifurcations of inclusions,” *Acta metallurgica*, vol. 32, no. 11, pp. 1925–1933, 1984. [38](#)
- [18] B. Bhadak, R. Sankarasubramanian, and A. Choudhury, “Phase-field modeling of equilibrium precipitate shapes under the influence of coherency stresses,” *Metallurgical and Materials Transactions A*, vol. 49, no. 11, pp. 5705–5726, 2018. [38](#)
- [19] H. Garcke, B. Nestler, B. Stinner, and F. Wendler, “Allen–cahn systems with volume constraints,” *Mathematical Models and Methods in Applied Sciences*, vol. 18, no. 08, pp. 1347–1381, 2008. [39](#)
- [20] A. Lahiri and A. Choudhury, “Dendrite tip selection during isothermal free growth in multi-component alloys: Marginal stability theories and insights from phase-field simulations,” *Computational Materials Science*, vol. 158, pp. 209–218, 2019. [52](#)

- [21] G. Ivantsov, “Temperature field around a spherical, cylindrical, and needle-shaped crystal, growing in a pre-cooled melt,” *Temperature field around a spherical, cylindrical, and needle-shaped crystal, growing in a pre-cooled melt Transl. into ENGLISH of” Temperaturnuye Pole Vokrug Sharobraznogo Tsilindricheskogo i Igloobrazno go Kristalla Rastushchego v Pereokhlazhdennom Rasplave” Doklady, Akademiya Nauk SSR, v. 58, 1947 p 567-569*, 1985. 53
- [22] J. Langer and H. Müller-Krumbhaar, “Theory of dendritic growth—i. elements of a stability analysis,” *Acta Metallurgica*, vol. 26, no. 11, pp. 1681–1687, 1978. 53
- [23] J. Langer and H. Müller-Krumbhaar, “Theory of dendritic growth—ii. instabilities in the limit of vanishing surface tension,” in *Dynamics of Curved Fronts*, pp. 289–295, Elsevier, 1988. 53
- [24] H. Müller-Krumbhaar and J. Langer, “Theory of dendritic growth—iii. effects of surface tension,” *Acta Metallurgica*, vol. 26, no. 11, pp. 1697–1708, 1978. 53
- [25] W. Oldfield, “Computer model studies of dendritic growth,” *Materials Science and Engineering*, vol. 11, no. 4, pp. 211–218, 1973. 53
- [26] E. Ben-Jacob, N. Goldenfeld, J. Langer, and G. Schön, “Dynamics of interfacial pattern formation,” *Physical Review Letters*, vol. 51, no. 21, p. 1930, 1983. 53
- [27] E. Ben-Jacob, N. Goldenfeld, J. Langer, and G. Schön, “Boundary-layer model of pattern formation in solidification,” *Physical Review A*, vol. 29, no. 1, p. 330, 1984. 53
- [28] D. A. Kessler and H. Levine, “Stability of dendritic crystals,” *Physical review letters*, vol. 57, no. 24, p. 3069, 1986. 53
- [29] D. A. Kessler, J. Koplik, and H. Levine, “Pattern selection in fingered growth phenomena,” *Advances in physics*, vol. 37, no. 3, pp. 255–339, 1988. 53
- [30] J. Lipton, M. Glicksman, and W. Kurz, “Equiaxed dendrite growth in alloys at small supercooling,” *Metallurgical and Materials Transactions A*, vol. 18, no. 2, pp. 341–345, 1987. 53, 54
- [31] A. Choudhury and B. Nestler, “Grand-potential formulation for multicomponent phase transformations combined with thin-interface asymptotics of the double-obstacle potential,” *Physical Review E*, vol. 85, no. 2, p. 021602, 2012. 55

- [32] W. J. Boettinger and J. A. Warren, “The phase-field method: simulation of alloy dendritic solidification during recalescence,” *Metallurgical and materials transactions A*, vol. 27, no. 3, pp. 657–669, 1996. [55](#), [56](#)
- [33] A. Karma, “Phase-field formulation for quantitative modeling of alloy solidification,” *Physical Review Letters*, vol. 87, no. 11, p. 115701, 2001. [57](#)
- [34] R. F. Almgren, “Second-order phase field asymptotics for unequal conductivities,” *SIAM Journal on Applied Mathematics*, vol. 59, no. 6, pp. 2086–2107, 1999. [57](#)