Learning Phase-Field Modeling with MicroSim

Hasitha.S

B.E Materials Science and Engineering

College of Engineering Guindy

June 17, 2025

1 Introduction

As part of my internship at ICME, I explored **MicroSim**, a phase-field simulation software designed for modeling microstructural evolution in materials. MicroSim supports various thermodynamic models including the *Grand Potential*, *CALPHAD*, and *Allen-Cahn* formulations. Simulations are carried out using finite difference methods over a structured grid, with outputs generated in the `.vtk` format for visualization in ParaView.

In this exercise, I used the `Grand_potential_Finite_difference_2D_serial` solver available from the official MicroSim GitHub repository. The simulation was run using a sample `Input.in` file along with a simple `filling.in` file that initializes the computational domain.

The modeled system is a binary alloy (Al–Cu) with two distinct phases: `alpha` and `beta`. Over time, the simulation captures the diffusion-driven phase transformation behavior, with the chemical potential field (`Mu_Al`) showing characteristic changes. The results are saved as a series of VTK files corresponding to different time steps and were successfully visualized using ParaView.

2 Workflow

The following steps were followed to successfully run a microstructure simulation using the `MicroSim` framework:

1. **Cloning the repository:** The official MicroSim GitHub repository was cloned using the `git clone` command. This provided access to the source code, example input files, and necessary build files.

2. **Understanding the Directory Structure:** The cloned folder contained multiple solvers categorized by model type. The Grand Potential model using Finite Difference (2D Serial) was selected for simulation.

3. **Preparing Input files:** A sample `Input.in` and corresponding `Filling.in` file were used. The `Input.in` file defines simulation parameters (mesh size, time steps, phases, components, physical properties), while the `Filling.in` file specifies initial conditions for the simulation domain.

4. **Compiling the Solver:** The executable was run with three arguments: the input file, the filling file, and the desired output filename.

   Example command:

   ```
   ./microsim_gp Input.in Filling.in output
   ```

   Output VTK files were generated in a new `DATA` directory.

5. **Visualizing the Output:** The VTK files (e.g., `output_iso_1000.vtk`, `output_iso_2000.vtk`, etc.) were opened in ParaView as a time series group. By selecting different scalar fields such as `Mu_Al`, the evolution of the chemical potential over time was visualized.

3 Input

Input.in

```
##Geometrical dimensions of the simulation domain
DIMENSION = 2;
MESH_X = 50;
MESH_Y = 50;
MESH_Z = 1;
##Discretization, space and time
DELTA_X = 2.0;
DELTA_Y = 2.0;
DELTA_Z = 2.0;
DELTA_t = 0.08;
##Number of phases and composition
NUMPHASES = 2;
NUMCOMPONENTS = 2;
#Running and saving information
NTIMESTEPS = 1000;
NSMOOTH = 10;
SAVET = 1000;
STARTTIME = 0;
RESTART = 0;
## Component and Phase names
# COMPONENTS = {Al,Cu,B};
COMPONENTS = {Al, Cu};
PHASES = {alpha, beta};
##Material properties
##GAMMA={12, 13, 14, 23, 24...}
GAMMA = {1.0};
# Diffusivity = {Diagonal:0/1, phase, 11,22,33, 12, 13, 23...};
DIFFUSIVITY = {1, 0, 1};
DIFFUSIVITY = {1, 1, 1};
```

```
##Gas constant and molar volume
R = 1.0;
V = 1.0;
##Elasticity related parameters
EIGEN_STRAIN = {0, 0.01, 0.01, 0.0, 0.0, 0.0, 0.0};
EIGEN_STRAIN = {1, 0.01, 0.01, 0.0, 0.0, 0.0, 0.0};
VOIGT_ISOTROPIC = {0, 270, 187.5, 125.0};
VOIGT_ISOTROPIC = {1, 270, 187.5, 125.0};
#VOIGT_CUBIC = {phase, c11, c12, c44};
#VOIGT_TETRAGONAL = {phase, c11, c12, c13, c33, c44, c66};
##Boundary conditions
#0: Free, 1: Neumann, 2: Dirichlet, 3: Periodic, 4: Complex
#Boundary = {phase, X+, X-, Y+, Y-, Z+, Z-}
BOUNDARY = {phi, 1, 1, 1, 1, 0, 0};
BOUNDARY = {mu, 1, 1, 1, 1, 0, 0};
BOUNDARY = {c, 1, 1, 1, 1, 0, 0};
BOUNDARY = {T, 1, 1, 1, 1, 0, 0};
# Boundary = {phi, 1, 1, 0};
# Boundary = {"u", 3, 3, 2, 2};
#Boundary_value = {Value X+, Value X-, Value Y+, Value Y-,
Value Z+, Value Z-}
BOUNDARY_VALUE = {phi, 0, 0, 0, 0, 0, 0};
BOUNDARY_VALUE = {mu, 0, 0, 0, 0, 0, 0};
BOUNDARY_VALUE = {c, 0, 0, 0, 0, 0, 0};
BOUNDARY_VALUE = {T, 0, 0, 0, 0, 0, 0};
##Type of simulation
ISOTHERMAL = 1;
BINARY = 1;
#TERNARY
DILUTE = 0;
T = 0.96;
##FILEWRITING and OUTPUTTING TO SCREEN
## WRITEFORMAT ASCII/BINARY
##TRACK_PROGRESS: interval of writing out the progress of the simulation
to stdout.
WRITEFORMAT = BINARY;
TRACK_PROGRESS = 10;
##Model-specific parameters: Grand-potential model
##Phase-field parameters; epsilon:interface width;
it is not the gradient energy coefficient
epsilon = 8.0;
tau = 1.31;
Tau = {0.28};
##Anisotropy functions
##Anisotropy mode, FUNCTION_ANISOTROPY=0 is isotropic
Function_anisotropy = 1;
Anisotropy_type = 4;
dab = {0.04};
```

```
#Rotation_matrix = {0, 1, Euler_x(ang), Euler_y(ang), Euler_z(ang)};
Rotation_matrix = {0, 1, 0, 0, 0};
##Potential function
Function_W = 1;
Gamma_abc = {};
#Shifting of domain for infinite domain simulations
Shift = 1;
Shiftj = 30;
#Writing of composition fields along with the chemical potential fields
Writecomposition = 0;
#Noise
Noise_phasefield = 0;
Amp_Noise_Phase = 0.001;
##Temperature
Equilibrium_temperature = 1.0;
Filling_temperature = 1.0;
#TEMPGRADY={BASETEMP, DELTAT, DISTANCE, OFFSET, VELOCITY}
Tempgrady = {0.96, 0.06, 800.0, 0, 0.016};
##Function_F
Function_F = 1;
A = {0, 1};
A = {1, 1};
ceq = {0, 0, 0.78125};
ceq = {0, 1, 0.5};
ceq = {1, 1, 0.5};
ceq = {1, 0, 0.5};
cfill = {0, 0, 0.78125};
cfill = {0, 1, 0.5};
cfill = {1, 1, 0.5};
cfill = {1, 0, 0.5};
slopes = {0, 0, 0.45};
slopes = {0, 1, 0.45};
slopes = {1, 0, 0.45};
slopes = {1, 1, 0.45};
```

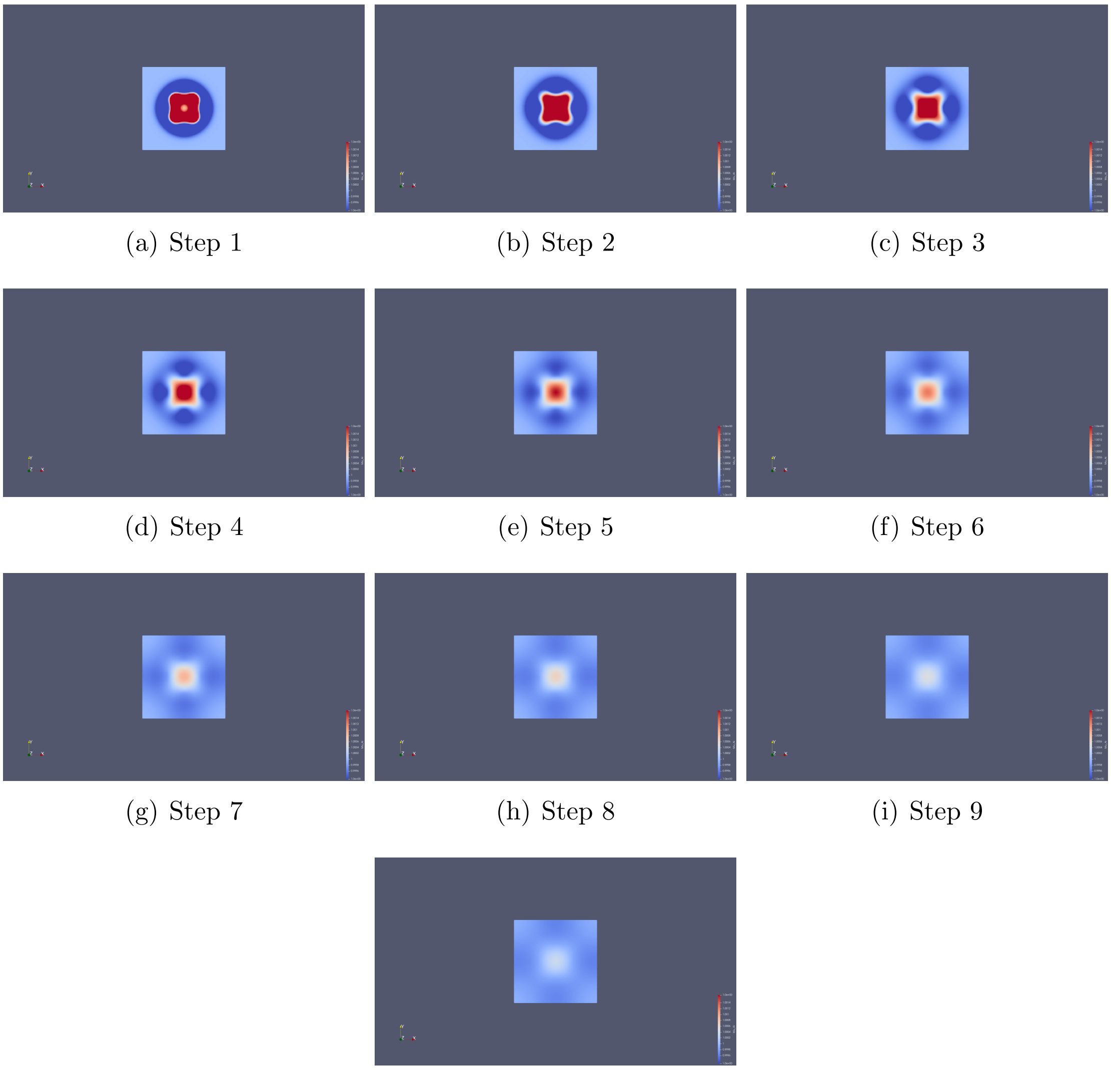
4 Fill

Fill.in

```
FILLCUBE = {0, 10, 10, 0,  20, 20, 0};
FILLCUBE = {0, 0, 0, 0,  100, 20, 0};
##FILLCYLINDER = {phase, x_centre, y_centre, z_start,
z_end, radius}
##FILLCYLINDER = {0, 10, 30, 0, 0, 15};
#FILLCYLINDER = {0, 50, 50, 0, 0, 15};
###FILLELLIPSE = {phase, x_center, y_center, major_axis,
eccentricity, rotation_angle_deg}
#FILLELLIPSE = {0, 50, 50, 0, 10, 0.1, 10};
#FILLSPHERE = {0, 50, 50, 0, 10};
```

5 Output images:

The simulation generates a series of output files in the VTK format, which contain the evolving phase and composition fields over time.
These files were visualized using ParaView, an open-source scientific visualization tool. The figures shown below represent snapshots of the simulation at various timesteps.



(a) Step 1

(b) Step 2

(c) Step 3

(d) Step 4

(e) Step 5

(f) Step 6

(g) Step 7

(h) Step 8

(i) Step 9

(j) Step 10

Figure 1: Images captured at different timesteps during the simulation (Step 1 to 10).

6 Conclusion:

Through this exercise, I successfully explored the capabilities of the MicroSim framework for simulating microstructural evolution using a phase-field approach. By setting up and running a sample simulation based on the Grand Potential model, I was able to understand the workflow involved—from preparing input files to visualizing results using ParaView.

This report documents the complete process, and the results obtained serve as a foundation for future experimentation or model modifications.