Linear Planar Model for Solidification

Hasitha S
B.E Materials Science and Engineering
College of Engineering Guindy

June 12, 2025

1 Introduction

The Linear Planar IRF model is used to understand how the temperature at the solid-liquid interface of an alloy changes with solidification velocity. It's particularly useful for analyzing rapid solidification processes in metallurgy and materials science.

As the solidification velocity increases, solute trapping and kinetic effects alter the behavior of the alloy, making it deviate from equilibrium. This model helps predict the interface temperature under such non-equilibrium conditions, especially for multicomponent alloys.

2 Objective

In this project, I implemented the Linear Planar Interface Response Function (IRF) model using Python. The aim was to simulate how the temperature at the solid-liquid interface changes with different solidification velocities in a multi-component alloy system.

The model uses a set of thermophysical parameters provided in a .json input file and calculates the non-equilibrium interface temperature across a range of velocities. These values are then plotted to visualize how the interface responds during solidification.

By doing this, I gained hands-on experience in:

- Extracting and using data from `JSON` files,
- Performing scientific computations using `NumPy`
- Visualizing physical behavior using `Matplotlib`
- Understanding the role of each material property in solidification

3 Inputs and Outputs

3.1 Inputs

The model requires the following parameters, usually provided via a JSON input file:

- **Solutes**: List of solute elements
- **Alloy Composition**: Mole fraction of each solute in the alloy
- **Equilibrium Partition Coefficient**: Ratio of solute concentration in solid to liquid under equilibrium
- **Equilibrium Liquidus Slope**: Slope of the liquidus line in a phase diagram
- **Solvent Melting Point**: Melting point of the base alloy
- **Kinetic Coefficien**t: Controls kinetic undercooling with velocity
- **Interface Diffusive Speed**: Affects how much the actual partitioning deviates from equilibrium
- **Minimum Velocity**
- **Maximum Velocity**
- **Velocity Step**

3.2 Outputs

For each velocity, the model computes:

- **Solidification velocity** (m/s):
- **Interface temperature** T (K)
- **Non-equilibrium partition coefficient** k_v (for each solute)
- **Non-equilibrium liquidus slope** m_v (for each solute)

4 Python process flow

1. **Import Required Libraries**

   We use `json` to read input, `numpy` for numerical operations, and `matplotlib` for plotting.

2. **Read Input JSON File**

   The file `Planar_H13_delta.json` is opened, and its contents are converted into a Python dictionary using `json.loads()`.

3. **Extract Input Values** The dictionary keys are used to extract solutes, composition values, equilibrium parameters, and velocity limits.

4. **Define get_mv function**
   This function calculates the non-equilibrium liquidus slope (mv) using the formula:

$$m_v = \frac{m_e}{1-k_e}\left(1 - k_v\left(1 - \ln\left(\frac{k_v}{k_e}\right)\right)\right)$$

5. **Loop Over Velocities**
   Starting from `vmin`, we loop up to `vmax` in steps of `vstep` (with smaller steps after 1 m/s). For each velocity:

   - Calculate k_v

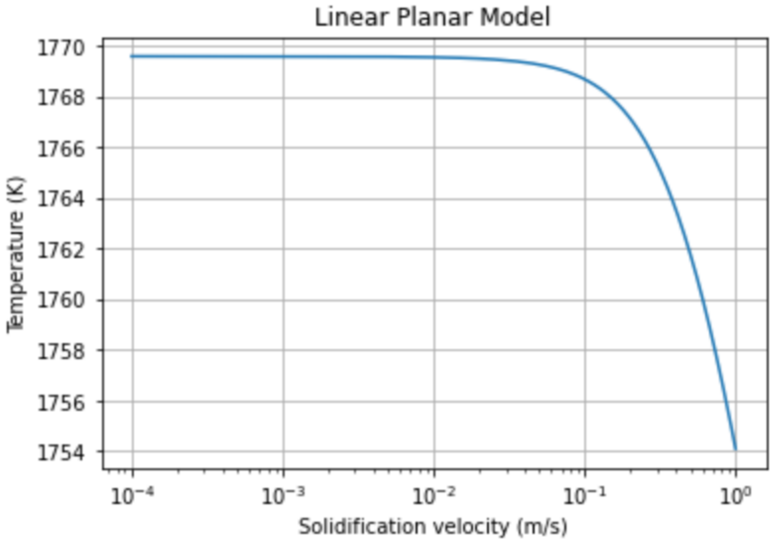
$$k_v = \frac{k_e + \frac{v}{v_{di}}}{1 + \frac{v}{v_{di}}}$$

   - Use k_v to get m_v
   - Compute interface temperature T

$$T = T_m + \sum m_v \cdot c_o - \frac{v}{\mu_k}$$

   - Print all values

6. **Plot Results**
   A plot of Temperature vs Velocity is generated, with a logarithmic scale on the velocity axis.



5 Conclusion

Through this task, I learned how to use `Python` to implement the Linear Planar IRF model, extract inputs from a `JSON` file, perform numerical calculations with `NumPy`, and create plots using `Matplotlib`.