Technische Universität Bergakademie Freiberg

Computational Material Science
Fakultät IV

Personal Programming Project

Programming of the 2D FE Implementation of the Stefan Problem and the Enthalpy Problem

Omkar Kolekar

Matriculation Number: 66808

07.10.2023

Supervised by
Dr. Ing. Stefan Prüger

Professor Incharge
Dr.-Ing. Arun Prakash
Dr.-Ing. Stefan Prüger

Abstract

The advancement in technology demand development of advanced tools. The manufacturing of these tools may involve laser manufacturing and phase change. This project is dedicated to the programming of laser-induced melting processes and the tracking of interface positions. There are two approaches programmed in this project. The first is the Enthalpy Problem approach, commonly known as fixed grid method and the second one is the Stefan Problem approach commonly known as the front tracking method. Both the methods present significant challenge. The Enthalpy approach is implemented for materials with sharp melting point, as well as materials that undergo melting over a range of temperatures. The Stefan problem is programmed only for materials that melt at a sharp melting point. We work with non-dimensionalized variables, and the techniques used for achieving this are explained in detail within this report. We utilize the Newton-Raphson-Scheme (NRS) to solve the set of non linear equations. Moreover, we employ the Euler implicit time integration scheme in the Enthalpy Problem and Euler forward or explicit time integration scheme for the Stefan Problem to track the position of the interface. In the Enthalpy Problem we aim to capture the sudden jumps in the Enthalpy that arises at the phase change. We have carried out necessary tests and validations to evaluate the results obtained.

Contents

List of Figures vii

List of Tables ix

1 Introduction 1
1.1 Motivation . . . 1
1.2 Objective . . . 1
1.3 Scope . . . 1

2 Mathematical Formulation 3
2.0.1 The Non Dimensionalization of Variables . . . 3
2.1 The Stefan Problem . . . 5
2.1.1 Weak Form . . . 6
2.1.2 Challenges associated with the Stefan Problem . . . 6
2.2 The Enthalpy Problem . . . 6
2.2.1 Weak Form . . . 7

3 Solution Techniques 11
3.1 Enthalpy Problem . . . 11
3.1.1 Time integration methods . . . 11
3.2 Stefan Problem . . . 12
3.2.1 Time integration methods . . . 13
3.3 Initial Conditions . . . 14
3.4 Flow Charts . . . 15

4 Testing 25
4.1 Unit Testing . . . 25
4.1.1 Test temperature enthalpy relation for amorphous material . . . 25
4.1.2 Test temperature enthalpy relation for crystalline material . . . 27
4.1.3 Test Shape Function Derivative . . . 28
4.1.4 Test Partition of Unity . . . 29
4.1.5 Test Jacobian . . . 30
4.2 Unit Tests 2D . . . 30
4.2.1 Test Partition of Unity . . . 30
4.2.2 Test Jacobian . . . 31
4.2.3 Test derivative of the shape function . . . 31

5 Verification **33**
5.1 Heat Transfer Verification in One Dimensional Settings for the Enthalpy Problem . . . 33

6 Results and Discussion **35**
6.1 The Enthalpy Problem . . . 36
6.1.1 Verification of Heat Transfer Results with Type I Boundary Conditions . . . 36
6.1.2 Verification of Heat Transfer Results with Type II Boundary Conditions . . . 38
6.1.3 Position of the Phase Change Boundary . . . 39
6.2 Stefan Problem . . . 41
6.2.1 Enthalpy Problem 2D results . . . 41

7 User Manual **51**
7.1 List of files and folders . . . 51
7.1.1 Enthalpy_Approach_1D . . . 51
7.1.2 Enthalpy_Approach_2D . . . 51
7.1.3 Heat_Transfer_Verification . . . 52
7.1.4 Testing . . . 52
7.1.5 Stefan_ Approach . . . 52
7.2 Libraries used . . . 53
7.3 Running the program . . . 53

8 Computation time study **55**

9 Conclusion **57**
9.1 Problems Encountered and Experiences . . . 58
9.2 Acknowledgement . . . 58
9.3 Milestones - status . . . 59

Nomenclature **61**
9.3.1 Constants . . . 61
9.3.2 Variables . . . 61
9.3.3 Subscripts . . . 61

Bibliography **63**
9.4 Git Log . . . 65

List of Figures

3.1 The mesh generation in Stefan problem 2D. (Figure taken from [4] 16
3.2 Structure of Class 16
3.3 Overview of the Enthalpy Problem Approach 17
3.4 Mesh 18
3.5 Main Class FEA 19
3.6 Class Initial Conditions 20
3.7 Class Elemental Subroutine 21
3.8 Overview of the Stefan Problem Approach 22
3.9 Stefan Problem Solver 23

5.1 Setup for the Heat Transfer Verification with type 1 boundary condition . 33
5.2 Setup for the Temperature Distribution Verification 34

6.1 Verification of the temperature distribution in Al for type I BC 36
6.2 Verification of the temperature distribution in SS304L for type I BC . . . 37
6.3 Convergence of the solution in Al for type I BC with variableΔt 38
6.4 Mesh convergence for Al using type I BC 39
6.5 RMS Error Convergence with Increasing Number of Terms in Series for analytical solution with type II BC. 40
6.6 Verification of the temperature distribution in AL for type II BC 41
6.7 Verification of the temperature distribution in SS304L for type II BC . . . 42
6.8 Phase Change Boundary Evolution (Al, No Latent Heat, 301 Nodes) . . . 42
6.9 Phase Change Boundary Evolution (Al, No Latent Heat, 201 Nodes) . . . 43
6.10 Phase Change Boundary Evolution (SS304L, No Latent Heat, 151 Nodes) 43
6.11 Phase Change Boundary Evolution (Al, 151 Nodes ,considering Latent Heat effects) 44
6.12 Enthlapy vs Temperature (Al, 151 Nodes ,no Latent Heat effects considered) 44
6.13 Enthlapy vs Temperature (Al, 301 Nodes ,considering Latent Heat effects) 45
6.14 Enthlapy vs Temperature (SS304L, 151 Nodes ,considering Latent Heat effects) 45
6.15 Temperature Distribution in Al (n = 301 nodes, without considering latent heat effects) 46
6.16 Temperature Distribution in SS304L (n = 301 nodes, without considering latent heat effects) 46
6.17 Position of the phase front from numerical scheme (considering the latent heat) and analytical scheme (neglecting the latent heat effects) 47

6.18 Temperature distribution in the liquid domain 47
6.19 Temperature distribution in the solid domain 48
6.20 Settings for the 2D Enthalpy problem 48
6.21 Temperature evolution from the node just below the heat source 49
6.22 Temperature distribution in the matrix at end time 49

List of Tables

2.1 Dimensionless Terms 8

6.1 Properties of Al 35
6.2 Properties of SS304 35

7.1 Libraries used and their versions 53

8.1 Computation time study 55

9.1 Milestone status of the project 59

1 Introduction

1.1 Motivation

From the automotive industry and space technology to medical applications, manufacturing stands as a cornerstone in equipment or tool production. Manufacturing requires precise knowledge of processes that may involve phase change in the material. Whether in the context of melting, ablation, or other transitions, phase change plays a critical role in determining the material properties. In the case of outer space applications, it becomes very important to explicitly track the phase change, as the equipment is subjected to an extreme environment. The phase change in crystalline materials occurs at a sharp temperature, whereas in amorphous materials, it occurs over a range of temperatures. Heat transfer plays an important role in these processes. Temperature represents the measure of thermal energy in the body and contributes to this phenomenon. The rate and direction of phase change may affect the properties and functionality of the manufactured tool, making this phenomenon interesting to simulate.

1.2 Objective

The objective of this project is to develop an FEA program using the Galerkin method[7] for the Enthalpy Problem[16] and Stefan Problem[14][16] in 2D settings that would explain the temperature distribution and track the phase change in the material over time for a constant external heat supply.

1.3 Scope

The Enthalpy problem approach is modelled for Aluminium and SS304, whereas the Stefan problem approach is modelled only for the aluminium metal, due to the requirement of strict sharp melting temperature condition.
In the Enthalpy Model[16], an analytical solution[5] to the temperature distribution for the 1D case is used to verify the results. The analytical solution for the evolution of interface boundary over time has been obtained using the Laplace transform of the one dimensional heat equation[16].

The following gives a brief overview of the report.

Chapter 2 explains the mathematical formulation for both the problems. The related weak forms and the necessity to non dimensionalize the variables is discussed in this

chapter.

Chapter 3 gives an overview of the solution techniques used to solve the problem.

Chapter 4 this chapter deals with the testing. Here, we discuss the unit testing carried out for each function, the inputs given, the expected output and the reason for the expected output.

Chapter 5 is the chapter related to verification. Here we discuss the setups used to verify our schemes, the various loading condition and the initial condition along with the required equations.

Chapter 6 the results obtained after running our program for the validation setup described in the previous chapter is discussed here.

Chapter 7 commands to run the provided code is given in this chapter.

Chapter 8 computational time study is carried out in this chapter.

Chapter 9 conclusion, problems faced and experiences are presented in this chapter.

2 Mathematical Formulation

Linear partial differential equations of the second order can be classified as one of the three types, hyperbolic, parabolic, and elliptic and can be reduced to an appropriate canonical or normal form[5]. Parabolic Partial Differential Equations describes the time evolution towards the steady state of an energy system. These equation involve the spatial and the temporal derivatives of the independent variables. This section discusses the parabolic PDEs for each approach separately and also the weak form of these equation used in the numerical scheme.
Temperature is a macroscopic properity which expresses the state of agitation or disordered motion of particles, and hence it is related to the kinetic energy of these particles. The energy of the system corresponding to the state of particle agitation is referred to as a form of internal energy of that system sometimes called, thermal energy[6]. Heat and temperature are interconnected through the fundamental principle of Fourier's Law. The temperature θ in the material is governed by the heat equation and in cylindrical coordinates is given by equation 2.1a along with the energy input given by the equation 2.1b.

$$\rho c \frac{\partial \theta}{\partial \tau} = \frac{k}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \theta}{\partial r}\right) + k\frac{\partial^2 \theta}{\partial \chi^2} \tag{2.1a}$$

$$k\frac{\partial \theta}{\partial \chi} = -I \tag{2.1b}$$

2.0.1 The Non Dimensionalization of Variables

The Stefan problem, as first formulated by Stefan in his research paper[14] in 1891, is a fundamental mathematical model employed to describe the phase transition of a material. It is a common convention used for simplicity and ease of comparison between different studies, to model the phase change at a temperature equal to zero. Non dimensionalization reformulates the problem and introduces the dimensionless temperature and other variables without altering the underlying physics. The dimensionless temperature is introduced in such a way, that the phase change occurs at '0' and any temperature below 0 represents phase 1 and any temperature above 0 represents phase 2.

We now introduce the dimensionless variables as follows

$$z = \frac{I_{\text{ref}}^2}{k(T_v - T_m)}\chi \tag{2.2a}$$

$$\bar{r} = \frac{1}{w_0} r \tag{2.2b}$$

$$t = \rho c \frac{I_{\text{ref}}^2}{k(T_v - T_m)^2}\tau \tag{2.2c}$$

$$T = \frac{\theta - \theta_m}{\theta_v - \theta_m} \tag{2.2d}$$

$$\epsilon = \frac{k^2(\theta_v - \theta_m)^2}{w_0^2 I_{\text{ref}}^2} \tag{2.2e}$$

Using the definition provided in (2.2a) to (2.2e) we can now reintroduce the heat equation (2.1a) in the dimensionless form together with influx of energy (2.1b) as follows

$$\frac{\partial T}{\partial t} = \epsilon \frac{1}{\bar{r}} \frac{\partial}{\partial \bar{r}} \left(\bar{r} \frac{\partial T}{\partial \bar{r}} \right) + \frac{\partial^2 T}{\partial z^2} \tag{2.3}$$

and

$$\frac{\partial T}{\partial z} = -\frac{I}{I_{\text{ref}}}, \quad z = 0 \tag{2.4}$$

where,
θ_m = melting Temperature in K
θ_v = vaporisation Temperature in K
z = depth
r = radial dimension
ρ = density in kg m^{-3}.
c = specific heat capacity
τ = time in s
θ = temperature in K
w_0 = waist of the heating source or laser beam
k = thermal conductivity in $Wm^{-1}K^{-1}$
I_{ref} = reference input laser intensity
I = Input Intensity

Typically, numerical methods for solving phase change problems can be characterized into three categories: front-tracking methods, front-fixing methods and fixed-domain methods[18]. The Stefan Problem and the Enthalpy Problem are classified as front-tracking method and fixed-domain method respectively. Both the methods are discussed in detail below.

2.1 The Stefan Problem

The Stefan Problem is a representation of the first-order phase transitions in matter through a boundary value problem for PDEs in which a discontinuity surface, internal to the domain, can move with time[12].
Consider the heat transfer in a solid liquid domain (Ω) $0 \leq$ z ≤ 1 . The portion (Ω_l) $0 \leq$ s(t) of this region is occupied by liquid and the portion (Ω_s) s(t)$\leq$1 is occupied by solid. The temperature in both the region is governed by the heat equation, in the dimensionless form it reads

$$\frac{\partial T_i}{\partial t} = \frac{\partial^2 T_i}{\partial z^2} \quad \text{in} \quad \Omega_i \quad \text{for} \quad i = s, l. \tag{2.5a}$$

At the boundary $z = 0$ the laser supplies an intensity $I = I(r, t)$,

$$\frac{\partial T_l}{\partial z} = -\frac{I}{I_{ref}}, \quad z = 0 \tag{2.6a}$$

In order for the phase change from solid to liquid in a material to transpire, we need to provide an additional amount of heat energy commonly referred to as Latent Heat of fusion. The latent in its dimensionless form is given below.

$$\lambda_f = \frac{L_f}{c(T_v - T_m)} \tag{2.7a}$$

At the solid liquid interface we have the well known *Stefan Condition* which expresses the absorption of heat needed for the phase change.

$$\left.\frac{\partial T_l}{\partial z}\right|_{z\uparrow s} - \left.\frac{\partial T_s}{\partial z}\right|_{z\downarrow s} = -\lambda_f \frac{ds}{dt}, \qquad z = s(t). \tag{2.8a}$$

The $z \uparrow s$ indicate the fact that we consider the derivative of the dimensionless temperature, under the limit when z approaches s (the liquid-solid interface) from the liquid phase (in terms of dimensionless temperature, we consider it from the positive side).

The $z \downarrow s$ indicate the fact that we consider the derivative of the dimensionless temperature, under the limit when z approaches s (the solid-liquid interface) from the solid phase (in terms of dimensionless temperature, we consider it from the negative side).

The dimensionless temperature T at the free boundary (moving phase front) is assumed to be continuous across the interface

$$T_s = T_l = 0 \qquad z = s(t). \tag{2.9a}$$

The boundary condition at infinity is given as follows

$$T_s \to T_a \qquad z \to \infty. \tag{2.10a}$$

where T_a is the dimensionless ambient temperature of the material.
The equation (2.5a) along with the boundary conditions (2.6a), (2.9a), (2.10a) and (2.8a) defines our problem statement. However, we need to find suitable initial conditions, which are discussed in the section 3.3.

2.1.1 Weak Form

The weak form of the Stefan problem reads as follows: -

$$\int_0^s \left\{ \frac{\partial T_l}{\partial t} v(z,t) + \frac{\partial T_l}{\partial z}\frac{\partial v}{\partial z} \right\} dz = \frac{I}{I_{ref}} v(0,t), \tag{2.11a}$$

and

$$\int_s^{z_b} \left\{ \frac{\partial T_s}{\partial t} w(z,t) + \frac{\partial T_s}{\partial z}\frac{\partial w}{\partial z} \right\} dz = 0 \tag{2.11b}$$

Where $v(z,t)$ and $w(z,t)$ are the test functions.

2.1.2 Challenges associated with the Stefan Problem

The Stefan Problem poses two challenges: -

1. With evolution of each time step the geometry of the phase changes.
   The geometry of this evolving boundary must be suitably described with a discrete model[13].

2. The temperature field must be discretised in such a way as the jump in the heat flux, i.e. in the temperature gradient, at the phase interface can be reproduced[13].

2.2 The Enthalpy Problem

To overcome the above mentioned challenges in the Stefan Problem, many researchers use the fixed domain method. The Enthalpy Problem is one of the fixed domain methods.

The advantage of this method is that it applies to the whole domain regardless of the phase at a particular material point, which in turn implies that the explicit tracking of the phase change interface is not required. Moreover, the enthalpy method is applicable to problems in which phase change occurs either at a sharp melting temperature or over a temperature range[18].
During sensible heating ($\rho c\theta$) the temperature of the material increases. It is imperative to provide sensible heat to the material to elevate it's temperature to the melting point. Subsequently an additional infusion of thermal energy in the form of Latent Heat L_f is also needed to transpire the phase transition event.
The region with temperature between the solidus and the liquidus temperature is referred to as *mushy region*[16]. The main objective of the enthalpy approach is to track the mushy region.
For nondimentionalisation purpose we introduce the terms described in the table 2.1.

We define the relation between the dimensionless enthalpy and dimensionless temperature as follows

$$\mathrm{H(T)} = \begin{cases} D_1 T, & \mathrm{T} < 0, \\ [0, D_1\lambda_f], & \mathrm{T} = 0, \\ D_1 T + D_1\lambda_f, & \mathrm{T} > 0 \end{cases} \qquad \text{for crystalline materials}$$

$$\mathrm{H(T)} = \begin{cases} D_2 T, & T \leq 0, \\ D_2 T + D_2\lambda_f \frac{T}{T_{liq}}, & 0 \leq T \leq T_{liq}, \\ D_2 T + D_2\lambda_f, & T > T_{liq} \end{cases} \qquad \text{for amorphous materials}$$

where,

$\lambda_f = \frac{L_f}{c(T_v - T_m)}$ constant for crystalline materials

$\lambda_f = \frac{L_f}{c(T_v - T_{sol})}$ constant for amorphous materials

2.2.1 Weak Form

The weak form of the Enthalpy problem is given as follows

$$\int_\Omega \frac{\partial H}{\partial t} v dz = D\frac{I}{I_{ref}} v(0) - D\int_\Omega \frac{\partial T}{\partial z}\frac{\partial v}{\partial z} dz \tag{2.19a}$$

Description	**Equation**
The enthalpy at vaporisation temperature	$\eta_v = \rho c \theta_v + \rho L_f$ (2.12)
The enthalpy at melting temperature for crystalline materials	$\eta_m = \rho c \theta_m$ (2.13)
The enthalpy at solidus temperature for amorphous material	$\eta_{sol} = \rho c \theta_{sol}$ (2.14)
The non-dimensional enthalpy for crystalline material	$H = \frac{(\eta - \eta_m)}{(\eta_v - \eta_m)}$ (2.15)
The non-dimensional enthalpy for amorphous materials	$H = \frac{(\eta - \eta_{sol})}{(\eta_v - \eta_{sol})}$ (2.16)
The constant D_1 defined for crystalline material	$D_1 = \frac{\rho c(\theta_v - \theta_m)}{\eta_v - \eta_m}$ (2.17)
The constant D_2 defined for amorphous material	$D_2 = \frac{\rho c(\theta_v - \theta_{sol})}{\eta_v - \eta_{sol}}$ (2.18)

Table 2.1: Dimensionless Terms

where $v(z)$ is the weight function.

Letting the enthalpy at the last node where $z = z_b$ (z_b is the total length under consideration) equal to the enthalpy at atmospheric temperature and maintaining a constant temperature at last node equal to atmospheric temperature for the entire time of the simulation $H_N(t) \equiv H(T_a)$ and $T_N(t) \equiv T_a$ we take care of the Dirichlet boundary condition.
In the next chapter, we discuss in detail the discretization and solution techniques.

3 Solution Techniques

The solution of the PDEs may not be trivial. Hence, numerical techniques are employed to find an approximate solution for the given PDE. Our choice for the numerical solution is FEM (Finite Element Method) because of its versatility in dealing with complex boundaries. The FEM technique focuses on dividing the problem into smaller subdomains while approximating the solution within each subdomain using the interpolation functions and assembling the element equation into a global system to ultimately solve the resulting system of equations while incorporating boundary conditions.

3.1 Enthalpy Problem

In FEM simulations, one of the most popular weighted residual methods is Galerkin's method[7]. Boris Grigoryevich Galerkin pioneered the method in the year 1915[7]. He found that the approximation is determined from the condition that the residual is orthogonal to a system of linearly independent test functions and the orthogonality is understood in the integral sense[11]. We employ his method in our FEM solution to find the approximate solution of the forms $\tilde{H}(z,t) = \sum_{i=0}^{N} H_i(t)\phi_i(z)$ and $\tilde{T}(z,t) = \sum_{i=0}^{N} T_i(t)\phi_i(z)$. Where, $\phi_i(z)$ is defined as the hat function.
While keeping the time = t, we divide our domain $0 \leq z \leq z_b$into N sub intervals [16].
The matrix representation of our variational form (2.19a) is as below[16].

$$\underline{\mathbf{M}}\frac{d\underline{\mathbf{H}}}{dt} = D\underline{\mathbf{b}} - D\underline{\mathbf{NT}} \tag{3.1a}$$

3.1.1 Time integration methods

The variables that are dependent on time (for example in our case the enthalpy is a time dependent variable $\dot{H}$), Euler backward (or Euler implicit) scheme [2] is employed for time integration scheme as it is relatively stable for a larger step size [1]. Hence, the equation (3.1a) can be rewritten as follows: -

$$\underline{\mathbf{H}}^{k+1,l} = \underline{\mathbf{H}}^{k+1,l-1} - (\partial\underline{\mathbf{G}}(\underline{\mathbf{H}}^{k+1,l-1})^{-1}\underline{\mathbf{G}}(\underline{\mathbf{H}}^{k+1,l-1})) \tag{3.2a}$$

Where, l = newton raphson step

$$M_{ij} = \int_0^{zb} \phi_i\phi_j dz, \qquad N_{ij} = \int_0^{zb} \frac{d\phi_i}{dz}\frac{d\phi_j}{dz}dz \tag{3.3a}$$

$$b_j = \frac{I}{I_{ref}}\phi_j(0) - T_a \int_0^{zb} \frac{d\phi_N}{dz}\frac{d\phi_j}{dz}dz \tag{3.4a}$$

$$\underline{\mathbf{G}}(\underline{\mathbf{H}}^{k+1}) = \underline{\mathbf{M}}(\underline{\mathbf{H}}^{k+1} - \underline{\mathbf{H}}^{k}) - \Delta t(\mathcal{F}(\underline{\mathbf{H}}^{k+1}, t^{k+1})) \tag{3.5a}$$

$$\mathcal{F}(H,t) = D(\underline{\mathbf{b}}(t) - \underline{\mathbf{NT}}(\underline{\mathbf{H}})). \tag{3.6a}$$

$$\partial\underline{\mathbf{G}}(\underline{\mathbf{H}}) = \underline{\mathbf{M}} + \Delta t D\underline{\mathbf{N}}\frac{\partial\underline{\mathbf{T}}}{\partial\underline{\mathbf{H}}}(\underline{\mathbf{H}}) \tag{3.7a}$$

The equations 3.3a and 3.4a are programmed in the material subroutine along with $\frac{\partial\underline{\mathbf{T}}}{\partial\underline{\mathbf{H}}}(\underline{\mathbf{H}})$ and equations 3.5a to 3.7a are programmed in element subroutine.

Our problem will be solved as per the algorithms given in section 3.4 along with the suitable initial conditions discussed in the section 3.3.

3.2 Stefan Problem

We again use the Galerkin's method[7] in the formulation of our FEM scheme. We divide the domains $0 \leq z \leq s(t)$ and $s(t) \leq z \leq z_b$ into N and M equal sub intervals respectively. We find the approximate solution of the forms

$$T_{h_l}(z,t) = \sum_{j=0}^{N} T_{l,j}(t)\phi_{l,j}(z,t), \qquad T_{h_s}(z,t) = \sum_{j=0}^{M} T_s, j(t)\phi_{s,j}(z,t). \tag{3.8a}$$

Incorporating the boundary conditions as $T_{l,N}(t) = 0$ and $T_{s,M}(t) = T_a$, we ensure the fulfillment of Dirichlet condition. Substituting the above approximations in the weak form (2.11a) and (2.11b) yields the two system of equations in the matrix form

$$\underline{\mathbf{M_l}}\frac{d\underline{\mathbf{T_l}}}{dt} + \underline{\mathbf{N_l}\mathbf{T_l}} = \underline{\mathbf{b_l}}, \tag{3.9a}$$

$$\underline{\mathbf{M_s}}\frac{d\underline{\mathbf{T_s}}}{dt} + \underline{\mathbf{N_s}\mathbf{T_s}} = \underline{\mathbf{b_s}} \tag{3.9b}$$

3.2.1 Time integration methods

We employ the Crank-Nicolson scheme for the time integration of the temperature distribution evaluation which is $O(\Delta t^2)$ and Euler Forward scheme for evaluating the position of the solid liquid interface. We now describe each term in the equations 3.9a and 3.9b which are coded in our element subroutine of the Stefan Problem Approach.

$$M_{l,0\ 0} = \frac{1}{2}h_l(t), \qquad M_{l,i\ i} = h_l(t) \quad i = 1, \ldots, N-1, \tag{3.10a}$$

$$N_{l,i-1\ i} = -\frac{1}{6}\frac{3i-2}{N}\frac{ds}{dt} - \frac{1}{h_l(t)}, \qquad N_{l,i+1\ i} = \frac{1}{6}\frac{3i+2}{N}\frac{ds}{dt} - \frac{1}{h_l(t)}, \tag{3.10b}$$

$$N_{l,0\ 0} = \frac{1}{6N}\frac{ds}{dt} + \frac{1}{h_l(t)}, \qquad N_{l,i\ i} = \frac{1}{3N}\frac{ds}{dt} + \frac{2}{h_l(t)}, \qquad i = 1, \ldots, N-1, \tag{3.10c}$$

$$b_{l,0} = \frac{I}{I_{ref}}, \tag{3.10d}$$

$$M_{s,i\ i} = h_s(t), \qquad b_{s,M-1} = T_a\left(\frac{1}{3}\frac{ds}{dt} + \frac{1}{h_s(t)}\right), \tag{3.10e}$$

$$N_{s,i-1\ i} = -\frac{1}{6}\frac{3M-3i+2}{M}\frac{ds}{dt} - \frac{1}{h_s(t)}, \quad N_{s,i+1\ i} = \frac{1}{6}\frac{3M-3i-2}{M}\frac{ds}{dt} - \frac{1}{h_s(t)}, \tag{3.10f}$$

$$N_{s,i\ i} = -\frac{1}{3M}\frac{ds}{dt} + \frac{2}{h_s(t)}. \tag{3.10g}$$

The position of the interface can be calculated from the Stefan Condition 2.8a, which is discretized as below

$$s^{k+1} = s^k + \frac{\Delta t}{\lambda_f}\left\{\frac{1}{h_l(t^k)}T^k_{l,N-1} + \frac{1}{h_s(t^k)}T^k_{s,1}\right\} \tag{3.11a}$$

Using the Crank-Nicolson scheme, we find the temperature distribution for the next time step in both the phases as follows

$$\begin{aligned}\left(\underline{\mathbf{I}} + \frac{1}{2}\Delta t(\underline{\mathbf{M}}_l^{k+1})^{-1}\underline{\mathbf{N}}_l^{k+1}\right)\underline{\mathbf{T}}_l^{k+1} = \left(\underline{\mathbf{I}} - \frac{1}{2}\Delta t(\underline{\mathbf{M}}_l^{k})^{-1}\underline{\mathbf{N}}_l^{k}\right)\underline{\mathbf{T}}_l^{k} + \\ \Delta t\left(\frac{1}{2}(\underline{\mathbf{M}}_l^{k+1})^{-1} + \frac{1}{2}(\underline{\mathbf{M}}_l^{k})^{-1}\right)\underline{\mathbf{b}}_l\end{aligned} \tag{3.12a}$$

and,

$$\left(\underline{\mathbf{I}} + \frac{1}{2}\Delta t(\underline{\mathbf{M}}_s^{k+1})^{-1}\underline{\mathbf{N}}_s^{k+1}\right)\underline{\mathbf{T}}_s^{k+1} = \left(\underline{\mathbf{I}} - \frac{1}{2}\Delta t(\underline{\mathbf{M}}_s^{k})^{-1}\underline{\mathbf{N}}_s^{k}\right)\underline{\mathbf{T}}_s^{k} + \Delta t\left(\frac{1}{2}(\underline{\mathbf{M}}_s^{k+1})^{-1}\underline{\mathbf{b}}_s^{k+1} + \frac{1}{2}(\underline{\mathbf{M}}_l^{k})^{-1}\underline{\mathbf{b}}_s^{k}\right) \tag{3.13a}$$

In the next section we discuss the derivation of the inital conditions.

3.3 Initial Conditions

For both the the schemes described above, we need a suitable initial condition. During the premelting stage the temperature in the material is governed by the heat equation (3.14a).

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial z^2} \tag{3.14a}$$

Assuming the following initial and boundary conditions.
The (dimensionless) energy supplied at the surface which is denoted by F.

$$\frac{\partial T}{\partial z} = -F, \quad z = 0 \tag{3.15a}$$

Subject to the boundary condition:

$$T \to T_a, \quad z \to \infty \tag{3.16a}$$

Initially we assume:

$$T = T_a, \quad t = 0 \tag{3.17a}$$

The solution to the equations (3.14a) to (3.17a) has been founded by using Laplace Transform by [16] in his paper

$$T(z,t) = F\left\{2\left(\frac{t}{\pi}\right)^{\frac{1}{2}}\exp\left(-\frac{z^2}{4t}\right) - z\,\mathrm{erfc}\left(\frac{z}{2t^{\frac{1}{2}}}\right)\right\} + T_a \tag{3.18a}$$

Using equation (3.18a) we find the initial temperature distribution in the material .

Since, for the Stefan Problem we assume some part of the material is already melted, we need an initial temperature distribution in the liquid as well as the solid phase of the material. The equation (3.19a) gives the initial temperature distribution in the liquid domain, satisfying the boundary condition at $z = s$ in the liquid domain. Where as, the equation 3.19b along with the equation 3.19c gives the initial temperature distribution in the solid domain that satisfy the boundary condition at $z = s$ and $z = z_b$ in the solid domain.

$$T_{l,j}(\Delta t) = \frac{I}{I_{ref}}(s - z_{l,j}), \qquad j = 0, \ldots, N-1 \tag{3.19a}$$

$$T_{s,j}(\Delta t) = |T_a| \exp\left(-\frac{(z_{s,j} - s)^2 F^2}{T_a^2 \pi}\right) - F(z_{s,j} - s)\,\mathrm{erfc}\left(\frac{(z_{s,j} - s)F}{|T_a|\sqrt{\pi}}\right) + T_a, \qquad j = 1, \ldots, M-1 \tag{3.19b}$$

Where,

$$F = \frac{I}{I_{ref}} - \lambda_f \frac{ds}{dt}. \tag{3.19c}$$

3.4 Flow Charts

We have used the Object Oriented Programming (OOP) to program both the schemes. The advantage of OOP is that it offers a combination of data (attributes) and functions (Methods) in a single unit known as class as shown in figure 3.2. A class is a blue print that facilitates instructions to create an object. An object in simple terms is a collection of attributes and methods. Methods control the behaviour of these attributes. A class is called child class when it inherits (acquires) the attributes and methods form another class. The class from which the attributes and methods are inherited is referred to as parent class. The advantage this method of programming is that, when we pass an object, a collection of attributes and methods is passed simultaneously making it easier to keep a track of variables, and the role and scope of each class is clearly defined from its attributes and methods.

An overview of the enthalpy problem approach program is given in the following flow charts. The flow chart for the material model for the enthalpy problem approach is not provided as the equations (3.3a) to (3.7a) are programmed in the material model for the enthalpy problem and element subroutine is not provided for the Stefan problem approach and equations 3.10a to 3.10g are programmed directly in the element subroutine, eliminating the need of the shape functions. Also, the processed inputs, variable updater and post processors are not shown here as they perform non dimensionalization

of the variables or plotting graphs. The variable updater is a program which converts the enthalpy to temperature and vice versa according to the material type. Also, the flow charts all the attributes are bold to show they are scalar, and the attributes in bold and with an underline represent vector or matrix. Methods are not represented here with an underline or in bold.

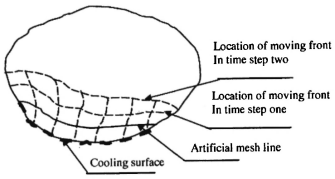

Figure 3.1: The mesh generation in Stefan problem 2D. (Figure taken from [4]

Child-Class
Parent class
PAttribute1 Parent class attribute 1 **PAttribute2** Parent class attribute 2
Parent Method 1 ()
Attribute 1 (with underline:- Vector/Matrix) **Attribute 2** (without undeline: - Scalar)
Methods()

Figure 3.2: Structure of Class

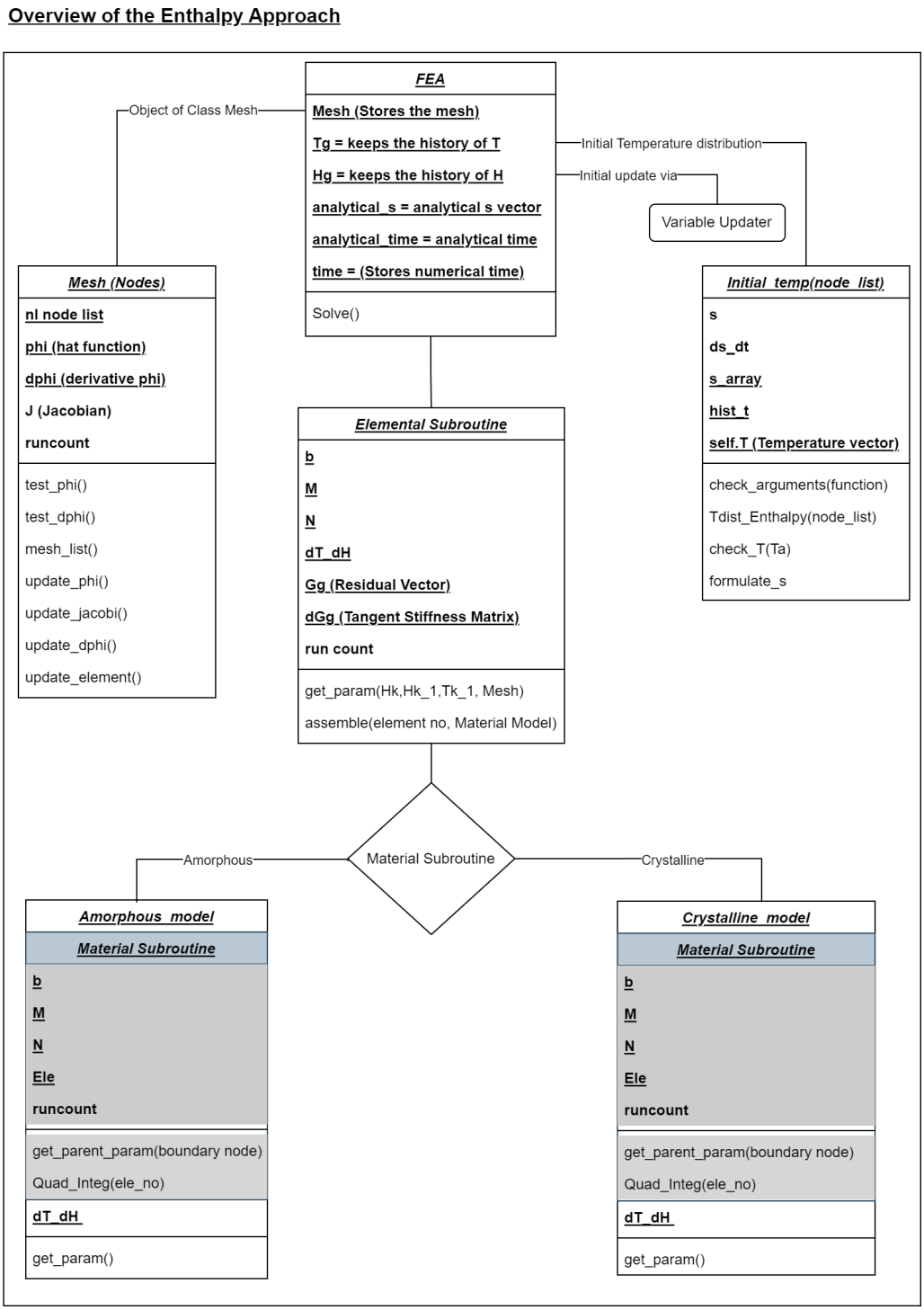

Figure 3.3: Overview of the Enthalpy Problem Approach

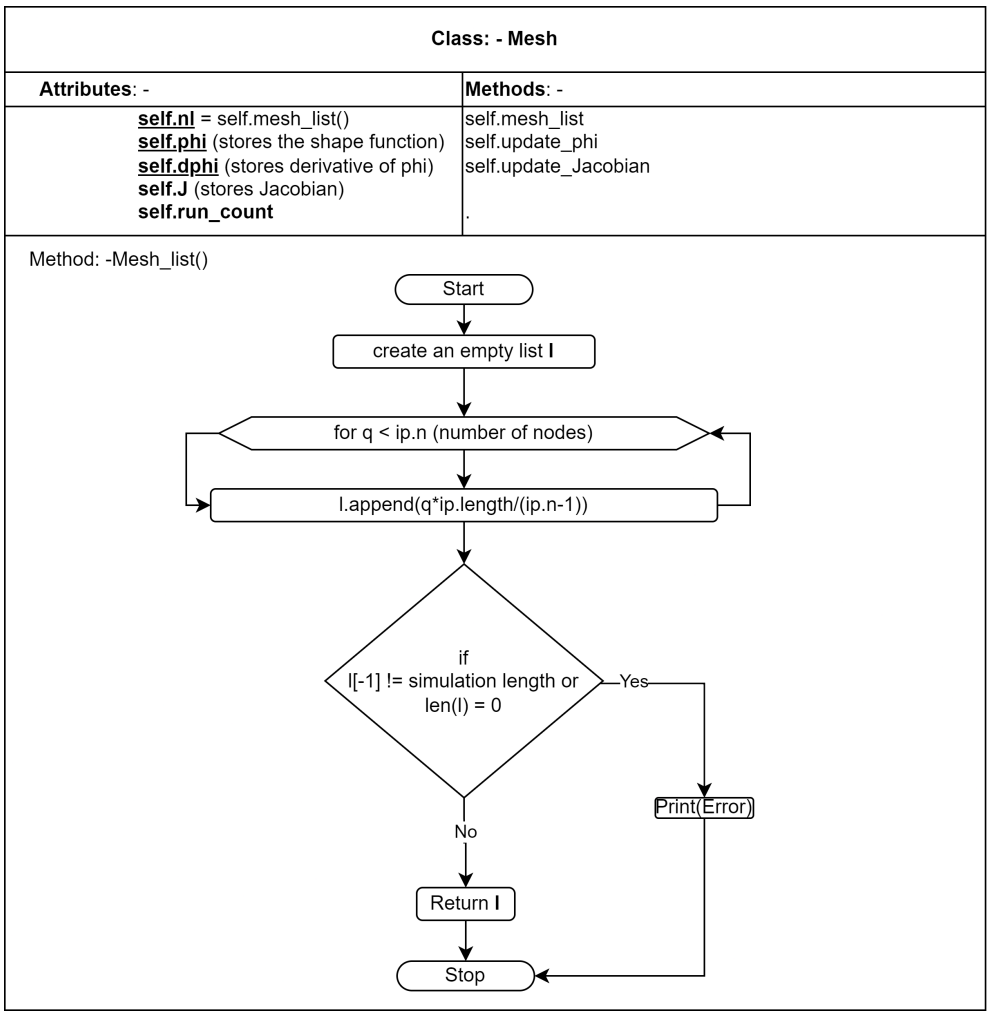

Figure 3.4: Mesh

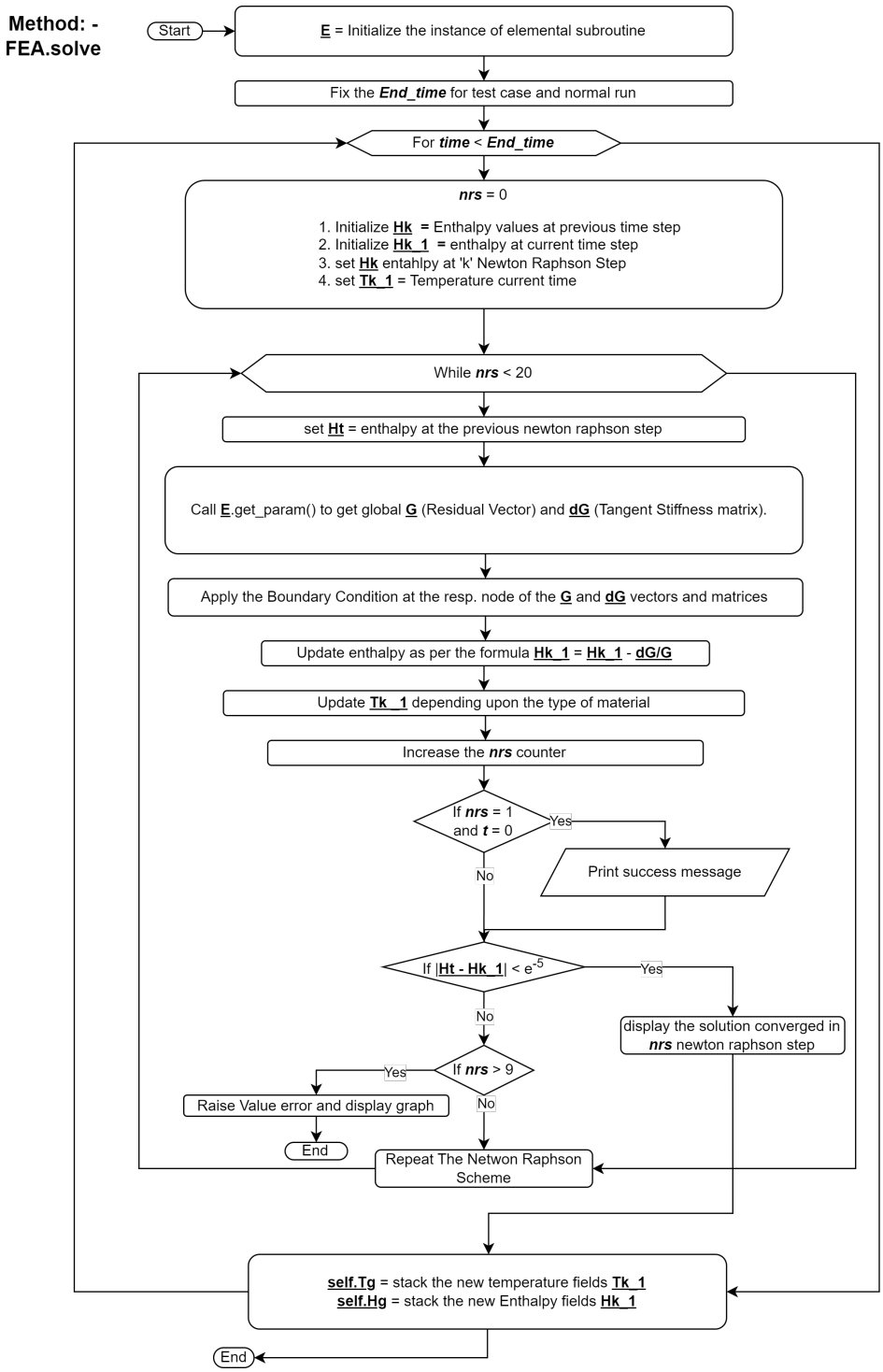

Figure 3.5: Main Class FEA

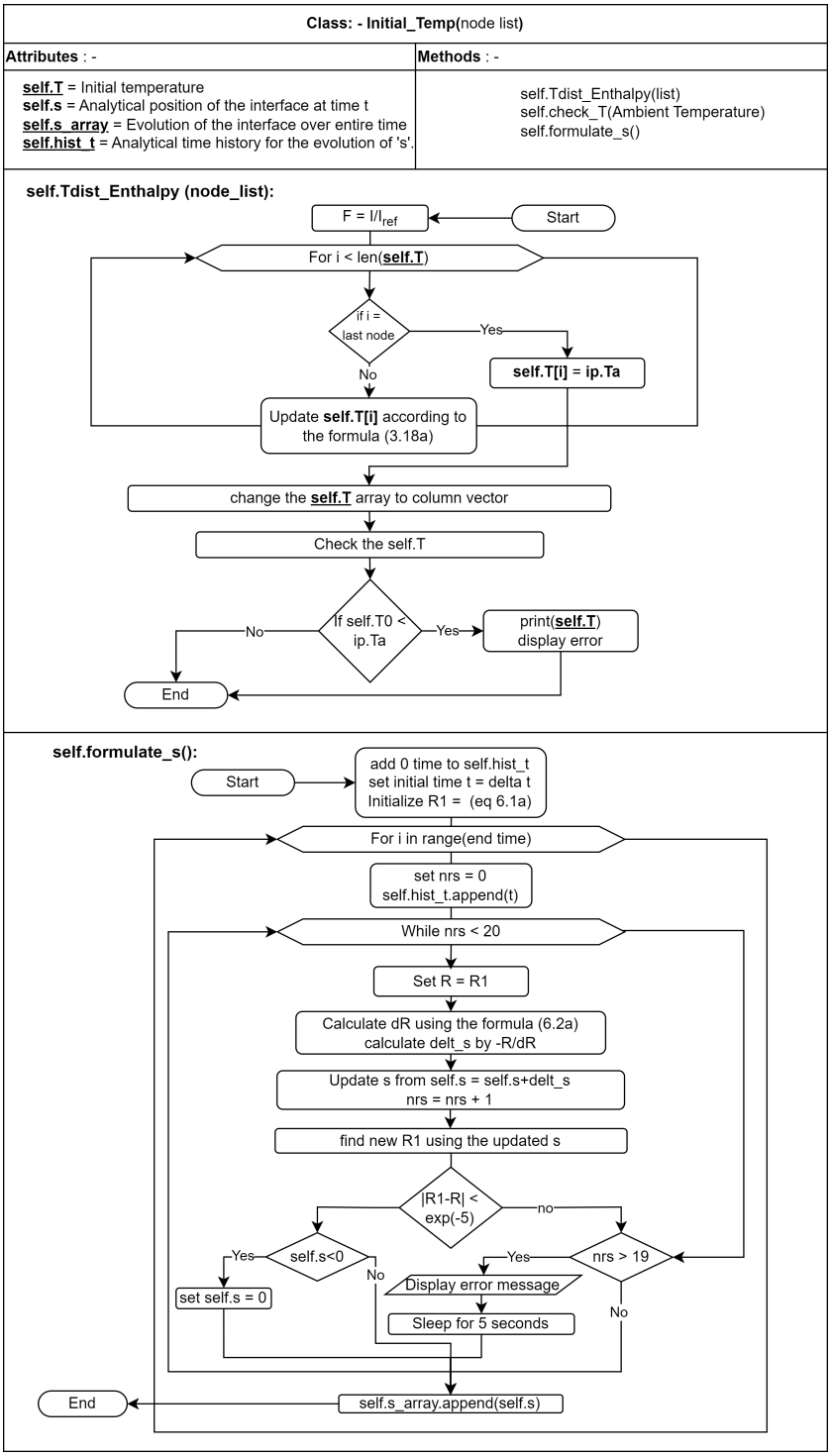

Figure 3.6: Class Initial Conditions

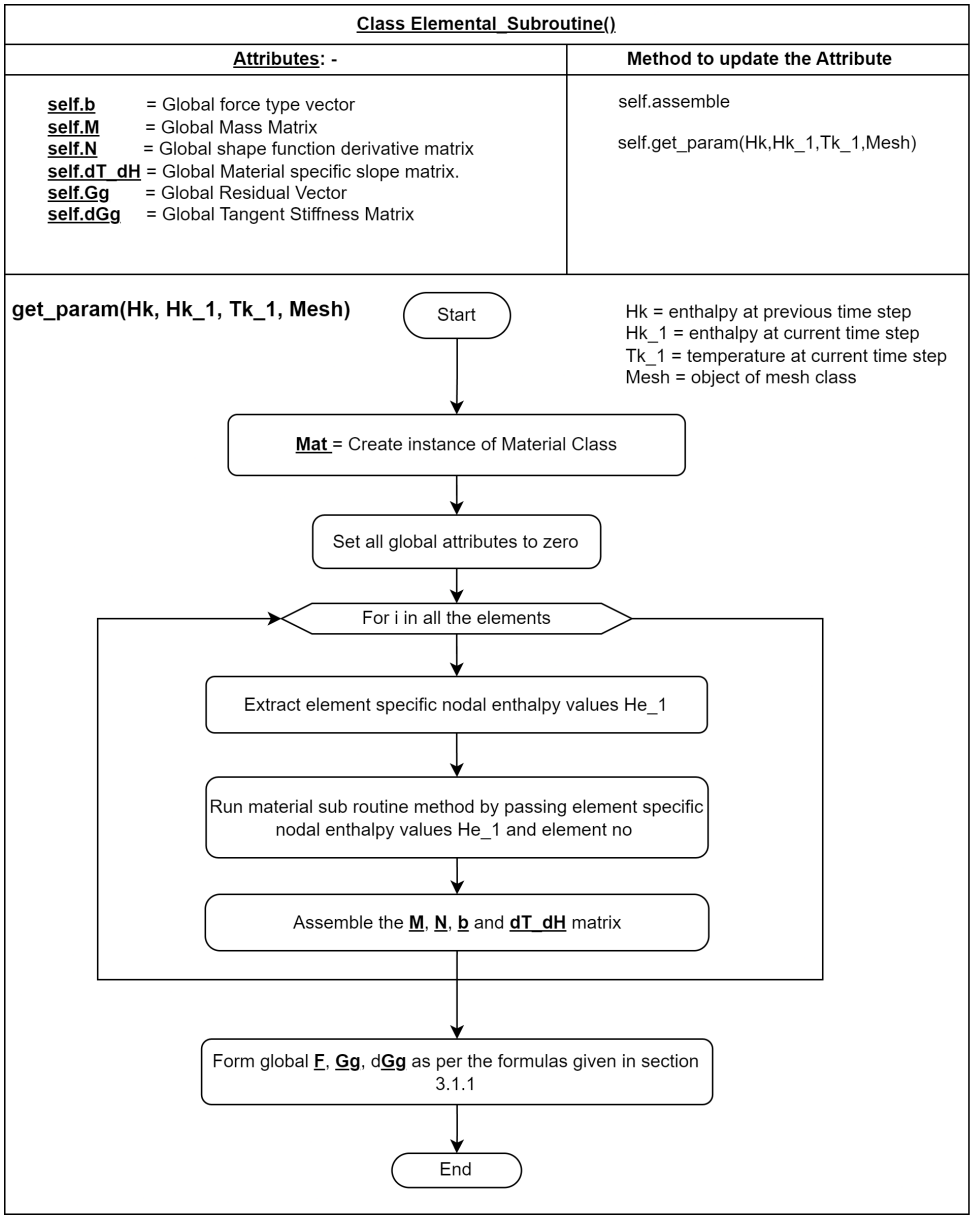

Figure 3.7: Class Elemental Subroutine

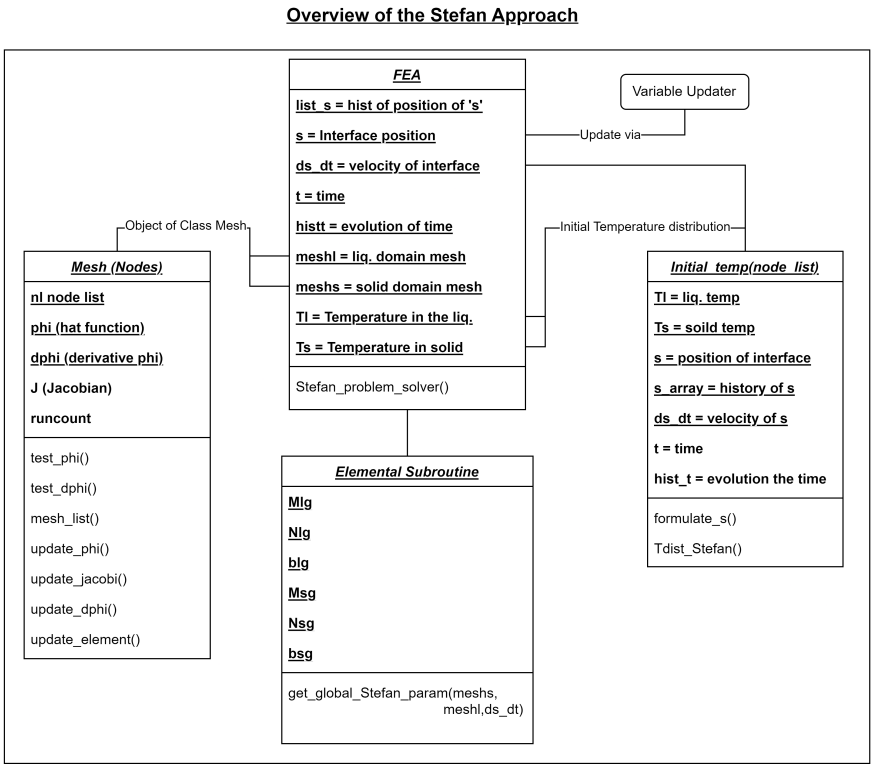

Figure 3.8: Overview of the Stefan Problem Approach

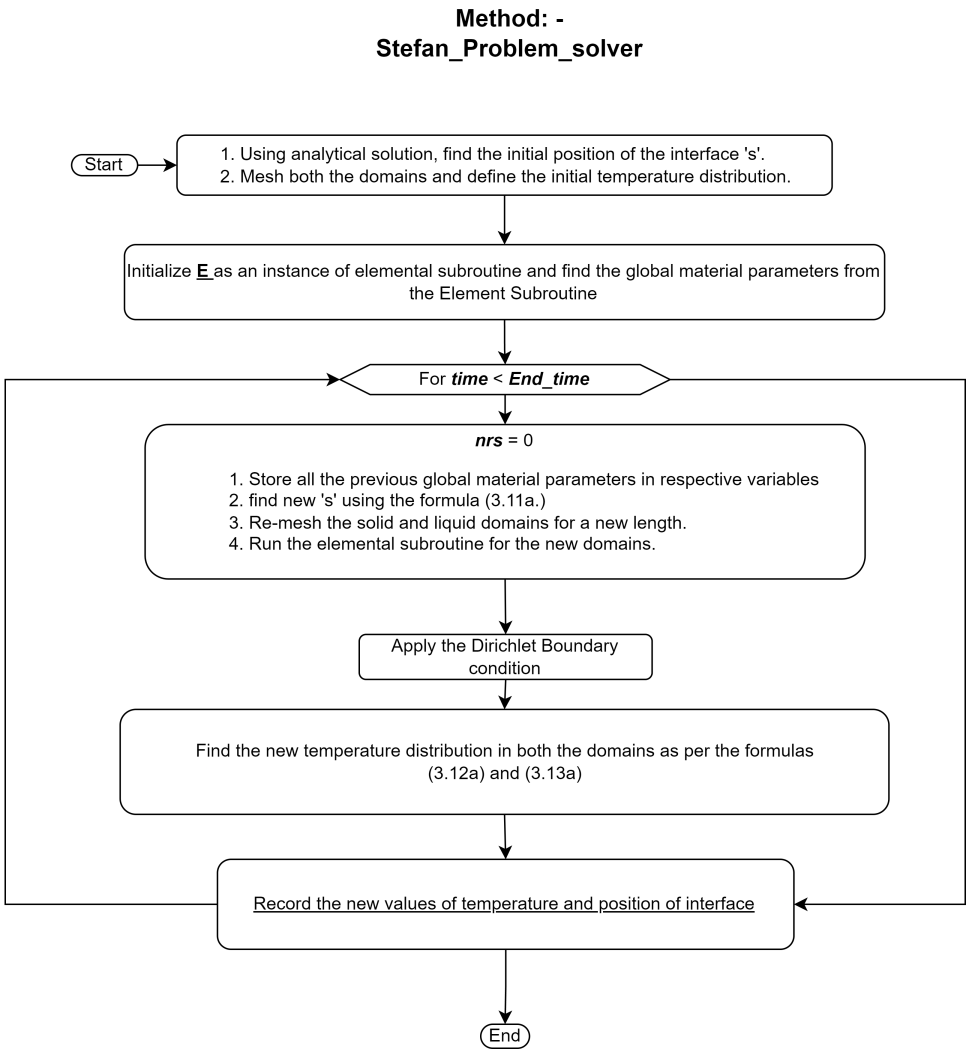

Figure 3.9: Stefan Problem Solver

4 Testing

4.1 Unit Testing

In this section we discuss the testing of the material specific libraries and the meshing library.
In the code, we have set the run count variable = 3, as for the very first run at time = 0 and run count = 0, the program outputs which type of material was detected as a check point to ensure proper functioning of the program. We refer to all the physical quantities as dimensionless quantities unless specified. For all testing purpose, we have assumed the length for the simulation to be 1 unit and the number of nodes to be 100.

4.1.1 Test temperature enthalpy relation for amorphous material

I will reiterate the temperature enthalpy relation for the amorphous material here for ease of reference within this section.

$$\mathrm{H(T)} = \begin{cases} D_2T, & T \leq 0, \\ D_2T + D_2\lambda_f \frac{T}{T_{liq}}, & 0 \leq T \leq T_{liq}, \\ D_2T + D_2\lambda_f, & T > T_{liq} \end{cases} \quad \text{for amorphous materials}$$

<u>Aim:</u> To check whether $\frac{\partial T}{\partial H}(H)$ in the user defined material model in the equation 3.7a functions correctly.

<u>Risks:</u> It is an important part of the material model as it defines the relation of temperature with the enthalpy. If this part of the code fails especially at the phase change, inaccurate slope values will be generated leading to an inaccurate position of the interface or may produce nonphysical results. The most important part is when the nodal non dimensional enthalpy is close to 0 or H_{liq} as these are the points of sharp transition and sudden changes in the slope. Hence, we mainly focus on these inputs.

<u>Values of interest:</u> The enthalpy value between 0 and H_{liq} are of concern as the slope changes here and attains the value $\frac{T_{\mathrm{liq}}}{DT_{\mathrm{liq}}+D\lambda_f}$. In the liquid and the solid phase the slope is simply defined by $\frac{1}{D}$.

<u>Inputs required:</u> For this function we need to provide the inputs in the following form dT_dH [material type, nodal Enthalpy values, run count], material type = 1 shows it is

a crystalline material and 2 indicates its an amorphous material. H_{liq} is the dimensionless enthalpy at the liquidus temperature calculated as per the equation 2.16, Below we discuss the various nodal values of the H_{liq} that were passed as an input.

Input: [-1, H_{liq}+0.5]

Output Expected:

$$\begin{bmatrix} \frac{1}{D} & 0 \\ 0 & \frac{1}{D} \end{bmatrix}$$

Reason for the Output Expected: Enthalpy = -1 indicates material is still in solid phase and the H_{liq}+0.5 indicate that material is in liquid state. For both the inputs the slope is defined as $\frac{1}{D}$

Status: Passed

Input: [0, H_{liq}]

Output Expected:

$$\begin{bmatrix} \frac{1}{D} & 0 \\ 0 & \frac{T_{\text{liq}}}{DT_{\text{liq}}+D\lambda_f} \end{bmatrix}$$

Reason for the Output Expected: Enthalpy equal to 0 indicates that the material is starting to undergo a phase change, and the temperature does not change. However, for amorphous materials, the slope changes after the enthalpy surpasses 0. H_{liq} signifies that the material is undergoing a phase change, and the slope is determined by $\frac{T_{\text{liq}}}{DT_{\text{liq}}+D\lambda_f}$.

Status: Passed

Input: [0.000001, H_{liq}+0.000001]

Output Expected:

$$\begin{bmatrix} \frac{T_{\text{liq}}}{DT_{\text{liq}}+D\lambda_f} & 0 \\ 0 & \frac{1}{D} \end{bmatrix}$$

Reason for the Output Expected: The first nodal enthalpy shows that the material is changing phase and the second nodal enthalpy indicates that the material is in liquid state.

Status: Passed

Input: [0, H_{liq}-0.000001]

Output Expected:

$$\begin{bmatrix} \frac{1}{D} & 0 \\ 0 & \frac{T_{\text{liq}}}{DT_{\text{liq}}+D\lambda_f} \end{bmatrix}$$

Reason for the Output Expected: The first nodal enthalpy shows that the material just started to change phase hence, considered to be in solid phase, and the second nodal enthalpy indicates that the material is in the phase change regime.

Status: Passed

Command for the test: pytest Test_Amorphous.py

4.1.2 Test temperature enthalpy relation for crystalline material

I will reiterate the temperature enthalpy relation for the crystalline material here for ease of reference within this section.

$$\text{H(T)} = \begin{cases} D_1 T, & \text{T} < 0, \\ [0, D_1\lambda_f], & \text{T} = 0, \\ D_1 T + D_1\lambda_f, & \text{T} > 0 \end{cases} \qquad \text{for crystalline materials}$$

Aim: The objective is similar as discussed in the above section. The only difference is that the value of the slope changes. Here, the slope is zero during the phase change.

Values of interest: The enthalpy value between 0 and $D\lambda_f$ are of concern as the slope changes to 0 here. In the liquid and the solid phase the slope is simply defined by $\frac{1}{D}$.

Input: [-1, $D\lambda_{\text{f}}$]

Output Expected:

$$\begin{bmatrix} \frac{1}{D} & 0 \\ 0 & 0 \end{bmatrix}$$

Reason for the Output Expected: Enthalpy = -1 indicates material is in solid state. $D\lambda_f$ shows that the material is in the phase change and the slope is 0.

Status: Passed

Input: [0, $D\lambda_{\text{f}} + 0.5$]

Output Expected:

$$\begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{D} \end{bmatrix}$$

Reason for the Output Expected: Enthalpy = 0 indicates material in phase change process and the slope is 0. $D\lambda_f$ +0.5 shows that the material is in the liquid phase.

Status: Passed

Input: $[\lambda_{\mathrm{f}}, -1]$

Output Expected:

$$\begin{bmatrix} \frac{1}{D} & 0 \\ 0 & \frac{1}{D} \end{bmatrix}$$

Reason for the Output Expected: Both the nodal enthalpies indicate material in liquid and solid state respectively.

Status: Passed

Input: [0, $D\lambda_f$]

Output Expected:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Reason for the Output Expected: Both the nodal enthalpies indicate material under going phase change.

Status: Passed

Command for the test: pytest Test_Crystalline.py

4.1.3 Test Shape Function Derivative

Aim: The derivative of the shape function describe the change of the open coefficients (in our case temperature) with respect to space. This is completely different than the slopes described above. In the above section, we tested the change of temperature with respect to enthalpy and not with respect to space. The objective is to check if the shape function were correctly derived.

Input: [2, 1]

Expected Output: -100.

Reason for the Output Expected: Since, the length of the material is 1 unit and the number of nodes are 100, we know that the elemental length l_e will be 0.01. By the definition of the derivative $\frac{T_2-T_1}{l_e}$ we have $\frac{1-2}{0.01}$ = -100.

Status: Passed

Input: [1, 3]

Expected Output: 200.

Reason for the Output Expected: Since, the length of the material is 1 unit and the number of nodes are 100, we know that 0.01 will be the elemental length l_e. By the definition of the derivative $\frac{T_2-T_1}{l_e}$ we have $\frac{3-1}{0.01}$ = 200.

Status: Passed

Input: [4, 4]

Expected Output: 0.

Reason for the Output Expected: Since, the length of the material is 1 unit and the number of nodes are 100, we know that the elemental length l_e will be 0.01.. By the definition of the derivative $\frac{T_2-T_1}{l_e}$ we have $\frac{4-4}{0.01}$ = 0.

Status: Passed
Command for the test: pytest Mesh_Test.py

4.1.4 Test Partition of Unity

Aim: The shape functions are used to interpolate the values of the open coefficients at the nodes within the element. The partition of unit test is used to check if the sum of the shape functions at any point within the element is equal to 1. Failing this check means that the physical quantity of interest is not conserved within the element leading to incorrect calculation or definition of the shape function.

Input:[0.5] (value of local co-ordinate).

Expected Output: 1

Reason for the Output Expected: Property of the shape function.

Status: Passed

Command for the test: pytest Mesh_Test.py

4.1.5 Test Jacobian

Aim: To check if the determinant of Jacobian is positive. Since we work with linear shape functions, our Jacobian will be exactly half of the elemental length. Failing this test means that the geometry is not being conserved correctly.

Input: [2] (element number).

Expected Output: $\frac{(x_3-x_2)}{2}$

Reason for the Expected Output: Here x_3 = global x co-ordinate of the third node and x_2 = global x co-ordinate of the second node

Status: Passed

Command for the test: pytest Mesh_Test.py

4.2 Unit Tests 2D

We have created an instance of 2D mesh class by creating 5 x 5 nodes and assuming the length of the square to be 1 unit. All the tests were performed for element 3.

4.2.1 Test Partition of Unity

Aim: This is similar to the 1D partition of unity test. However, here we pass the Gauss quadrature points and iterate over all the points, and for every iteration we check if the sum of the shape function is equal to 1. The weights for these points are unity.

Input: Gauss points = [-0.577350269189626, 0.577350269189626].

Expected Output: 1

Status: Passed

Command for the test: pytest Test_Mesh_2D.py

4.2.2 Test Jacobian

Aim: To check the determinant of the Jacobian is always positive and the sum of the determinant of the Jacobian over the Gauss quadrature is equal to the area of the element.

Input: Gauss points = [-0.577350269189626, 0.577350269189626].

Expected Output: $(x_2 - x_0) \times (y_2 - y_0)$

Reason for the Expected Output: The determinant of the Jacobian is equal to the area of the element. It is the scaling factor by which the area changes during the transformation of the global co-ordinates to isoparametric co-ordinates. Also, in our mesh for an element, $x_2 = x_1$, $x_0 = x_3$, $y_2 = y_1$ and $y_0 = y_3$.

Status: Passed

Command for the test: pytest Test_Mesh_2D.py

4.2.3 Test derivative of the shape function

Aim: It is the same test as the 'Test Shape Function Derivative' in 1D, here we check it for 2D.

Input: We check the derivative on the nodal points hence we pass $\xi = 1$ and $\eta = 1$ (local co-ordinates) for the third element. The nodal temperature was [1,2,3,4].

Expected Output: 1) $\frac{T_2 - T_3}{(x_2 - x_0)}$
2) $\frac{T_2 - T_1}{(y_2 - y_0)}$

Reason for the Expected Output: Due to the input local co-ordinates, the derivative of shape function wrt x has entries in 2 and 3 and for y we have entries in 1 and 2 all other entries in the dN vector are 0.

Status: Passed

Command for the test: pytest Test_Mesh_2D.py

5 Verification

This chapter discusses the verification carried out to ensure proper working of our code. The results for the same will be discussed in Results and Discussion 5.

5.1 Heat Transfer Verification in One Dimensional Settings for the Enthalpy Problem

To verify the temperature distribution obtained from the numerical scheme, we compare the results obtained from the numerical scheme to the analytical one. We assume a one dimensional bar of unit length. We have considered two different types of boundary conditions, in order to verify the temperature distribution.

A) Specified Temperature Boundary Condition (Type I BC): -
A boundary condition where both the ends of the bar are held at a constant fixed temperature is commonly referred to as specified temperature boundary condition[3]. We assume, both the ends of the rod are at fixed 0 dimensionless temperature and $T_{initial} = f(x)$ as the initial temperature distribution, this means that our material is in liquid phase and there is no phase change. Mathematically we can write: -

$$T(x,0) = T_{initial} = -\frac{1}{2}sin(3\pi x) + \frac{3}{2}sin(\pi x) \tag{5.1a}$$

$$T(0,t) \;=\; 0 \quad \text{and} \quad T(1,0) \;=\; 0 \tag{5.1b}$$

Figure 5.1 depicts our setup for the heat transfer verification.
The analytical solution for equation (2.3) subject to initial (5.1a) and boundary (5.1b) conditions is given by equation (5.2a)[9].

$$T_{(x,t)} = \frac{3}{2}sin(\pi x)e^{-\pi^2 t} - \frac{1}{2}sin(3\pi x)e^{-9\pi^2 t} \tag{5.2a}$$

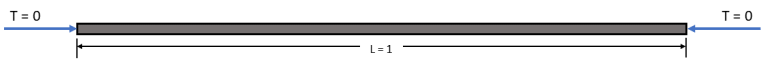

Figure 5.1: Setup for the Heat Transfer Verification with type 1 boundary condition

B) Specified Heat Flux Boundary Condition (Type II BC): -

A boundary condition where heat flux is specified at one end and the other end is assumed to be fixed at a dimensionless temperature is known as specified heat flux boundary condition[3]. In our case, we assume that the end x = 0 is insulated $\left(\frac{\partial T}{\partial x} = 0\right)$, where as the other end x = 1 is at a fixed 0 dimensionless temperature ($T_{1,t} = 0$) with $T_{initial} = T_{x,0}$ = 1 as the initial temperature distribution. The figure 5.2 shows the setup. The equation

Figure 5.2: Setup for the Temperature Distribution Verification

(5.3a) [9]is the analytical solution to the temperature distribution over time for the equation (2.3) subject to type 2 boundary condition.

$$T(x,t) = \frac{4T_{initial}}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n-1} cos\left(\frac{2n-1}{2}\pi x\right) exp\left(-\frac{(2n-1)^2\pi^2}{4}t\right) \tag{5.3a}$$

where: -
$T_{initial}$ = Constant Initial Temperature Distribution
t = Time
We have carried out the verification of the position of the interface for the one dimensional Stefan problem approach in the next chapter.

6 Results and Discussion

This chapter describes the results obtained from the above two schemes. The quantities used below in this section are dimensionless unless specified. We have considered pure Aluminum as a crystalline material with sharp melting point and SS304L as an amorphous material which melts over a range of temperature. The properties of Al are summarized in the table 6.1[16] and the properties of SS304 are summarized in the table 6.2[17].

Property	Value
Thermal conductivity k	2.3 x 10^2 $\frac{W}{mK}$
Density ρ	2.7 x 10^3 $\frac{kg}{m^3}$
Melting temperature T_m	9.3 x $10^2 K$
Vapour temperature T_v	2.5 x $10^3 K$
Specific heat of the solid c	9.0 x $10^2 \frac{J}{kgK}$
Latent heat of fusion L_f	3.6 x $10^5 \frac{J}{kg}$

Table 6.1: Properties of Al

Property	Value
Thermal conductivity k	22 $\frac{W}{mK}$
Density ρ	7.2 x 10^3 $\frac{kg}{m^3}$
Solidus temperature T_{sol}	16.7 x $10^2 K$
Liquidus temperature T_{sol}	17.27 x $10^2 K$
Vapour temperature T_v	3.375 x $10^3 K$
Specific heat of the solid c	7.0 x $10^2 \frac{J}{kgK}$
Latent heat of fusion L_f	2.47 x $10^5 \frac{J}{kg}$

Table 6.2: Properties of SS304

Further more we have assumed that the reference laser power I_{ref} is 1.5 x $10^{10} \frac{W}{m^2}$.

6.1 The Enthalpy Problem

The results of the enthalpy problem are discussed below.

6.1.1 Verification of Heat Transfer Results with Type I Boundary Conditions

The figure 6.1 shows the temperature distribution in Aluminium material obtained by the numerical scheme and the analytical scheme along with the initial temperature distribution. We considered 150 elements and Δt was assumed to be 0.0001 with input laser power I = 1.5 x 10^{10} $\frac{W}{m^2}$. The end time for this simulation was set to 0.2. The root mean square error for Al using these settings was found to be 0.092. Further, a reduction of error was observed as we decreased the time step Δt and is shown in figure 6.3.

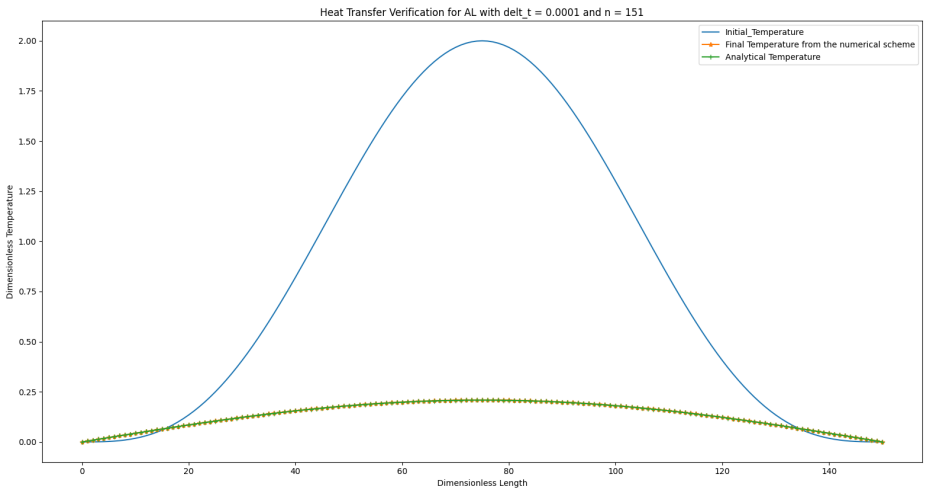

Figure 6.1: Verification of the temperature distribution in Al for type I BC

Similarly, the figure 6.2 shows the temperature distribution obtained from numerical and the analytical scheme for end time = 0.2 in SS304L for type 1 BC along with the initial temperature distribution. We considered 151 nodes and $\Delta t = 0.0001$ was assumed. The root mean square error obtained using the above mentioned settings was found to be 0.090.

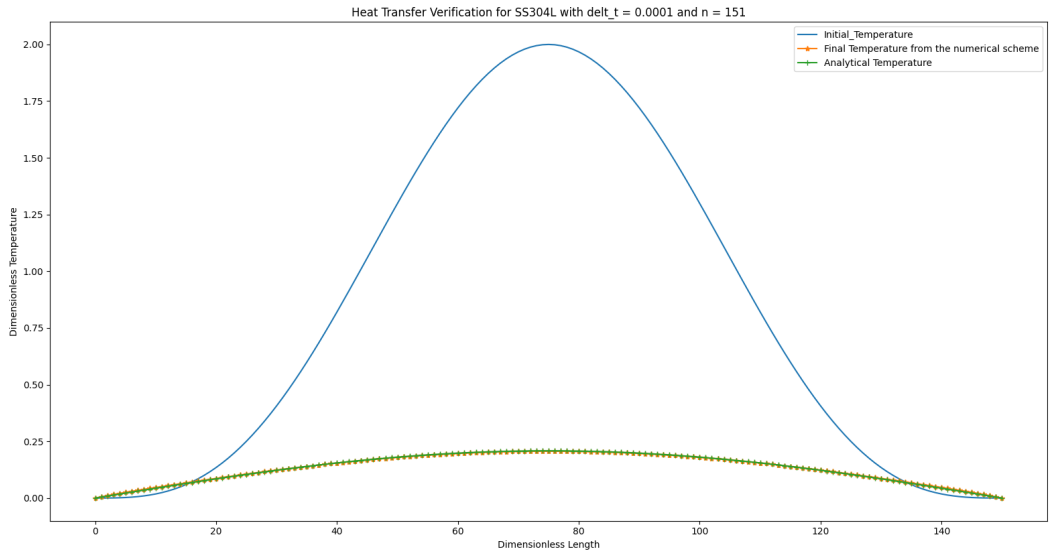

Figure 6.2: Verification of the temperature distribution in SS304L for type I BC

To investigate the impact of time step (Δt) on the convergence of our solution, we conducted four separate runs, each with a different Δt value. Throughout all runs, we maintained consistent settings, including a fixed input laser power of $1.5 \times 10^{10}\ \frac{W}{m^2}$, a total of 151 nodes, and a constant end time = 0.2. The only variable we altered was the time step.
In the initial run (Run 1), we set Δt to 0.1. Subsequently, in Run 2, we reduced Δt to 0.01. In Run 3, Δt was further decreased to 0.001. Finally, in Run 4, we employed a much smaller Δt of 0.0001.
The x-axis in our results in the figure 6.3 represents the run count, corresponding to the specific Δt used in each simulation run.

On a similar note,we conducted an examination of how the number of nodes influences the error. For this investigation, we maintained consistent settings, including a fixed input laser power of $1.5 \times 10^{10}\ \frac{W}{m^2}$, a constant Δt of 0.001, and an end time of 0.2. The only quantity varied was the number of nodes. The Figure 6.4 illustrates the error reduction as the number of nodes decreases. Notably, beyond 360 nodes, the error remains nearly constant.

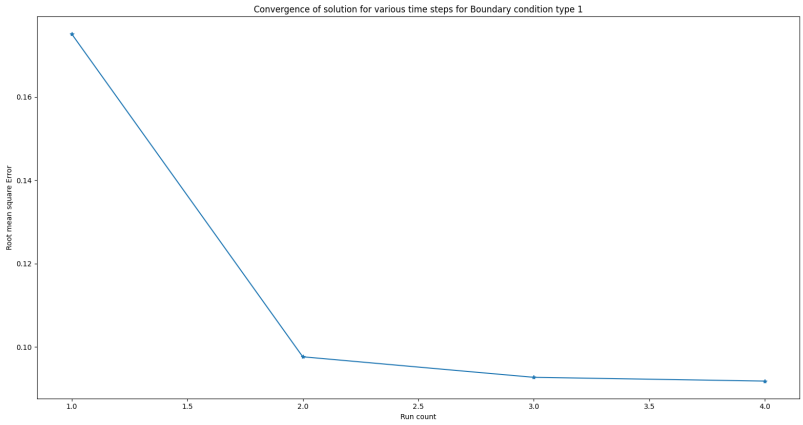

Figure 6.3: Convergence of the solution in Al for type I BC with variableΔt

6.1.2 Verification of Heat Transfer Results with Type II Boundary Conditions

We may approximate the solution (5.3a) by considering only its first term. This approximation is valid when the end time t is greater than $\frac{4}{\pi^2}$, as indicated by Hancock[9]. In our case, with an end time of $t = 0.1$, which is less than $\frac{4}{\pi^2}$, we utilized the first 4 terms of equation (5.3a). Our choice was influenced by the observation presented in Figure 6.5, where it is evident that the error approaches nearly zero after adding the first 2 terms. However, for enhanced accuracy, we included the first 4 terms in the series. This error was assessed by monitoring the changes in values as new terms were added to the solution. A similar setting is employed as we used in type I BC, we employ 150 elements and Δt was fixed to be 0.0001 with input laser power I = 1.5 x 10^{10} $\frac{W}{m^2}$. The end time for this simulation was set to 0.1 for both the materials. The figure 6.6 shows the temperature distribution in AL and the figure 6.7 shows the temperature distribution in SS304L respectively.

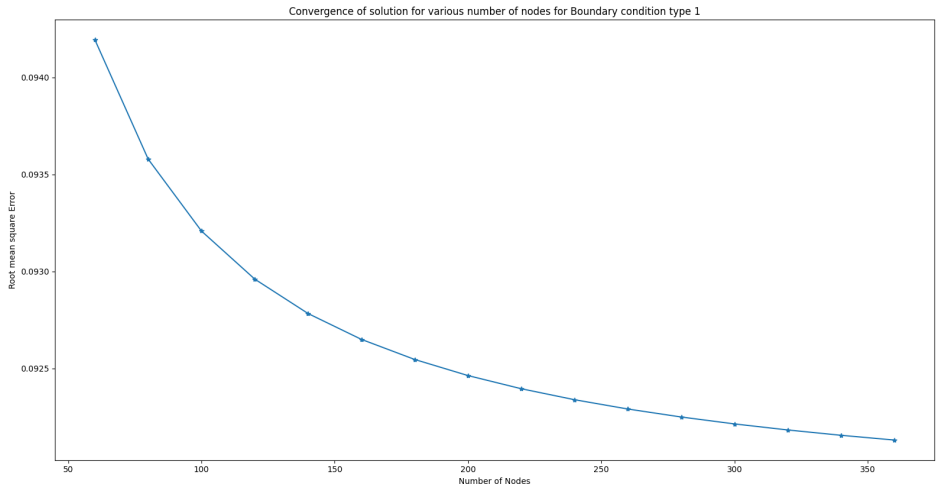

Figure 6.4: Mesh convergence for Al using type I BC

6.1.3 Position of the Phase Change Boundary

The analytical solution for the phase change boundary without considering the latent heat effects is given by the equation 6.1a[16][10].

$$R = \frac{I}{I_{\text{ref}}} \left\{ 2\sqrt{D} \left(\frac{t}{\pi}^{\frac{1}{2}} \right) \exp\left(-\frac{s^2}{4Dt} \right) - s \operatorname{erfc}\left(\frac{s}{2\sqrt{Dt}} \right) \right\} + T_a = 0 \tag{6.1a}$$

The position of the phase change boundary is found by Euler Forward Step and Newton Raphson Scheme. For the NRS we need to find the derivative of the equation 6.1a with respect to 's'. The equation 6.2a is the derivative of equation 6.1a with respect to 's' and was found manually by hand calculations.

$$\frac{\partial R}{\partial s} = -\frac{I}{I_{\text{ref}}} \left\{ \operatorname{erfc}\left(\frac{s}{2\sqrt{Dt}} \right) \right\} \tag{6.2a}$$

Using the equations 6.1a and 6.2a we have plotted the analytical solution in figures 6.8 to 6.11 for the verification of the position of the phase front in Al and SS304L for each time step.

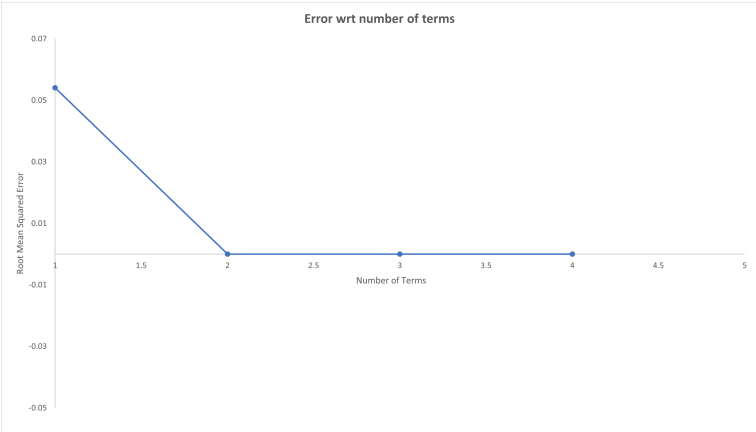

Figure 6.5: RMS Error Convergence with Increasing Number of Terms in Series for analytical solution with type II BC.

In the in figures 6.8 to 6.11 we observe the staircase-like behaviour which was also observed by Ben Hills[8] in his paper. It takes time for the interface to travel through an element where tracking of the interface is impossible due to absence of nodes within the elements. The author J.C.J. Verhoeven in his paper [16] has mentioned two techniques to overcome this issue. One method is to use finer mesh (which is already done in our work) and the other is a method suggested by Tacke[15] (which we did not use in this project).
We observe the impact of a finer mesh by comparing the results in Figure 6.8 (301 nodes) with Figure 6.9 (201 nodes). The influence of considering latent heat on the interface's evolution is evident in Figure 6.13, while Figures 6.12 and 6.13 illustrate the effect of latent heat on enthalpy evolution at node 1. In our Python simulation, node 1 corresponds to the node just below the one where heat flux is applied. The figure 6.13 and 6.14 also provides a visual verification for retention of material characteristic in our material model for crystalline and amorphous material respectively.

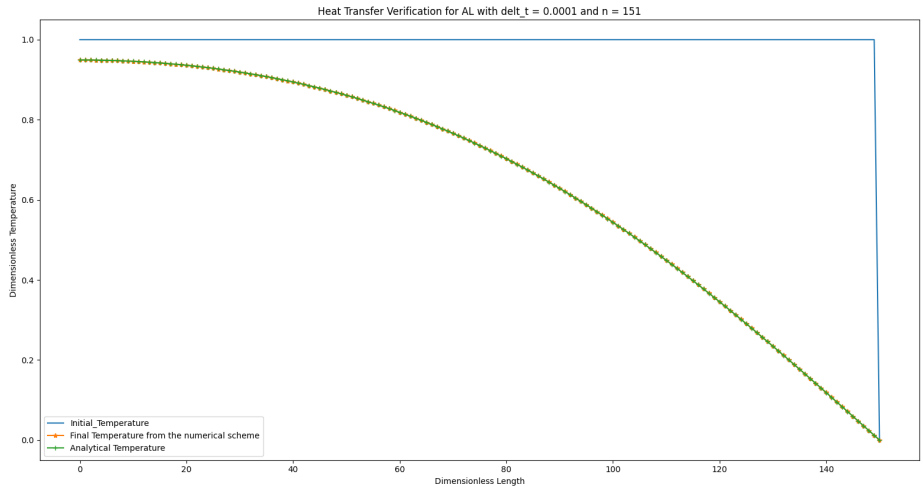

Figure 6.6: Verification of the temperature distribution in AL for type II BC

6.2 Stefan Problem

Similar to the enthalpy problem approach, we present the results form the Stefan problem approach. For the analytical solution of the position of interface over time we employ the Euler Forward time integration scheme on the equation 6.1 and equation 6.2a. This analytical solution does not account for the latent heat. Also, due to the boundary condition 2.8a, we cannot set our latent heat equal to zero. Doing so will not allow our material to melt. Also, it is important to note that this scheme is not applicable to the amorphous material as it does not have sharp melting point. We have assumed $I = 1.5 \times 10^{10} \frac{W}{m^2}$ laser input intensity and 2 nodes in the liquid domain and 51 nodes in the solid domain for the following simulation results. The decision to employ only 2 nodes in the liquid domain was based on the initial conditions, where the liquid domain is minimal, if not negligible. The Δt was assumed to be 0.00001.

6.2.1 Enthalpy Problem 2D results

Again we assumed a unit square. We supplied the top left corner node with the laser input. The bottom and the right edge were kept at constant ambient temperature. The results figure 6.20 shows the simulation setup.

The figure 6.21 shows the evolution of the temperature over the entire simulation time. The figure 6.22 shows the temperature distribution over the entire domain at the end of simulation time. We can see that first few nodes are in the liquid regime. It is also important to note that the scheme shows convergence issues, when the temperature at node 0 goes beyond unity as it physically means that the material is evaporated.

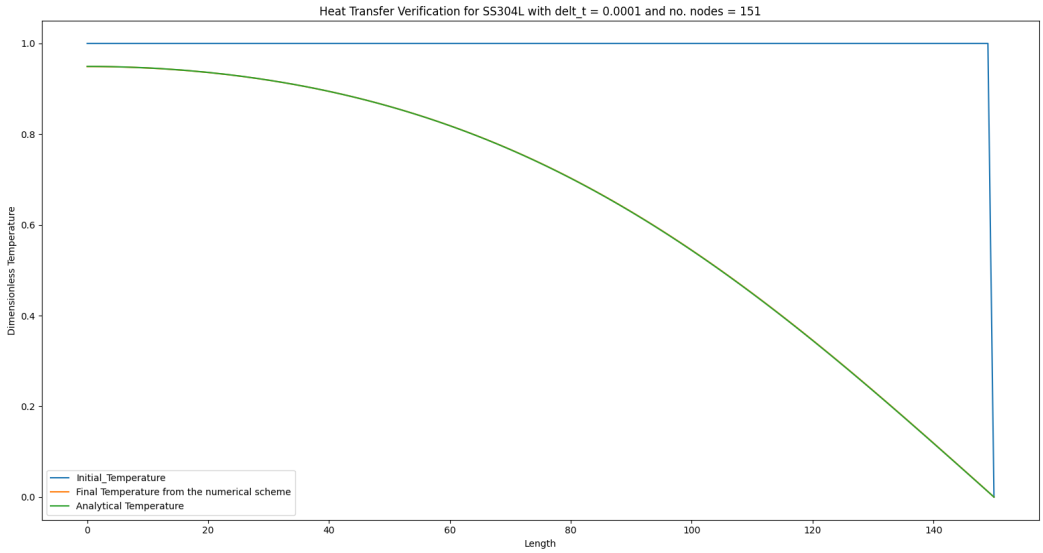

Figure 6.7: Verification of the temperature distribution in SS304L for type II BC

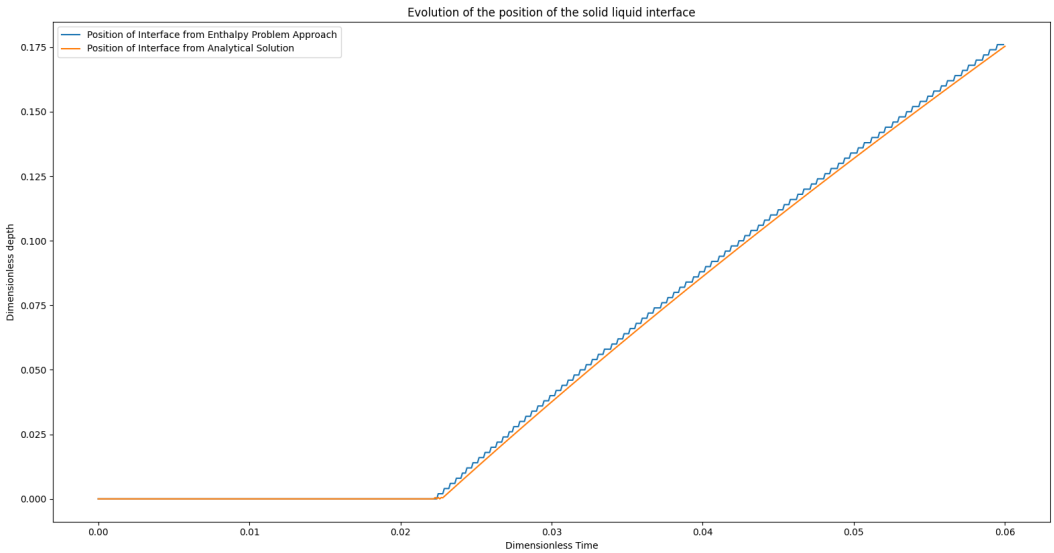

Figure 6.8: Phase Change Boundary Evolution (Al, No Latent Heat, 301 Nodes)

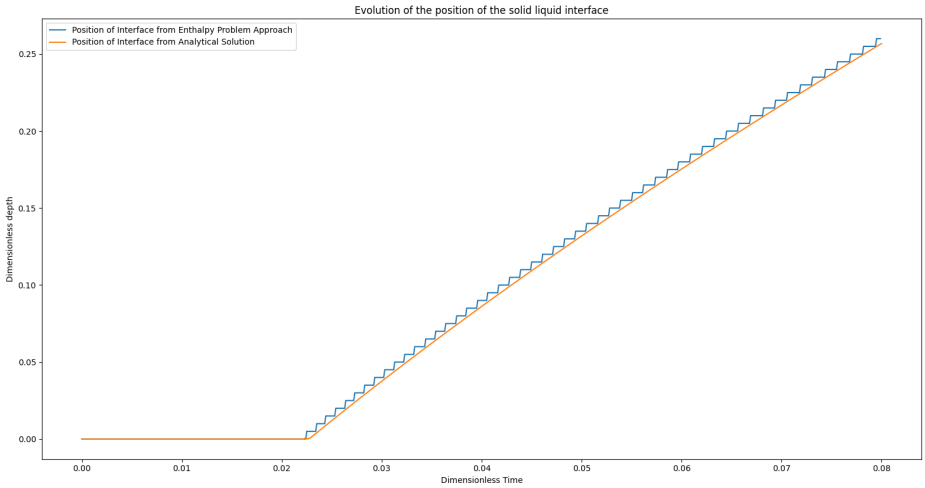

Figure 6.9: Phase Change Boundary Evolution (Al, No Latent Heat, 201 Nodes)

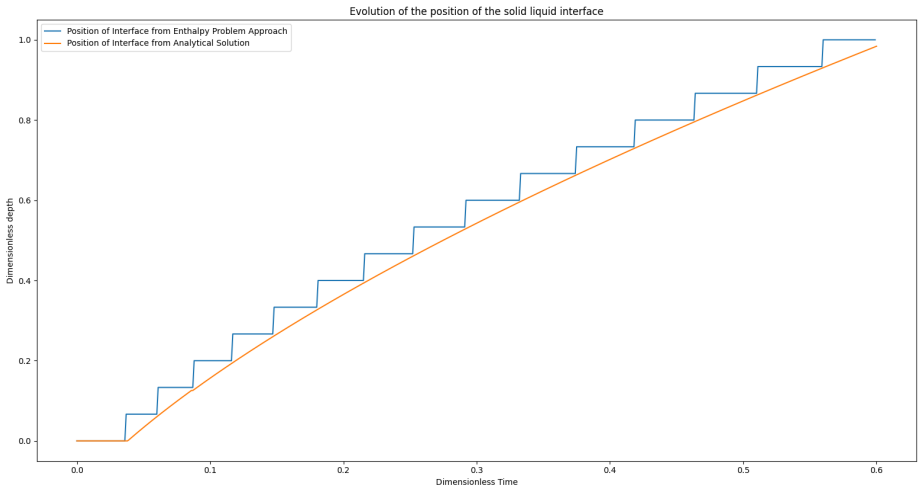

Figure 6.10: Phase Change Boundary Evolution (SS304L, No Latent Heat, 151 Nodes)

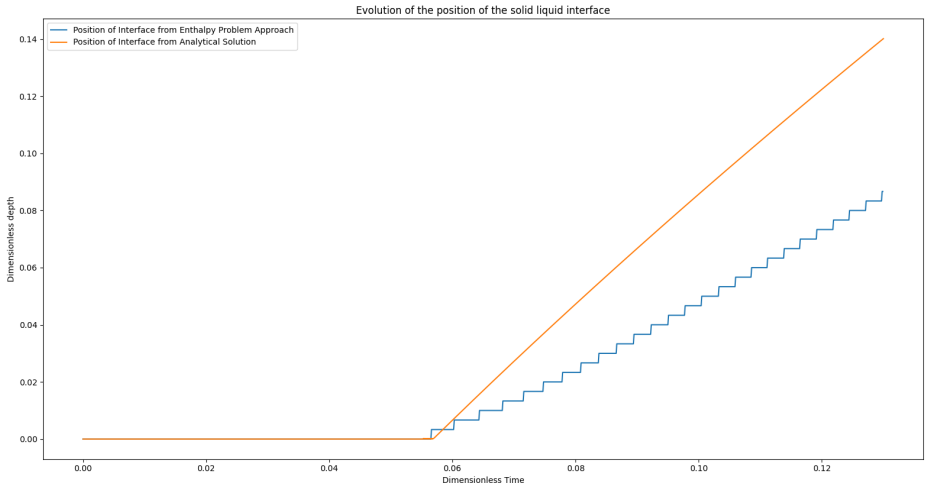

Figure 6.11: Phase Change Boundary Evolution (Al, 151 Nodes ,considering Latent Heat effects)

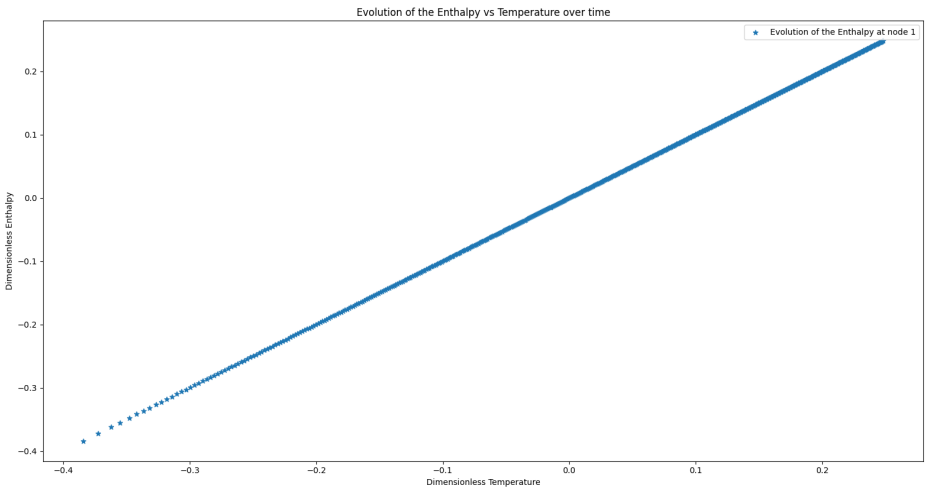

Figure 6.12: Enthlapy vs Temperature (Al, 151 Nodes ,no Latent Heat effects considered)

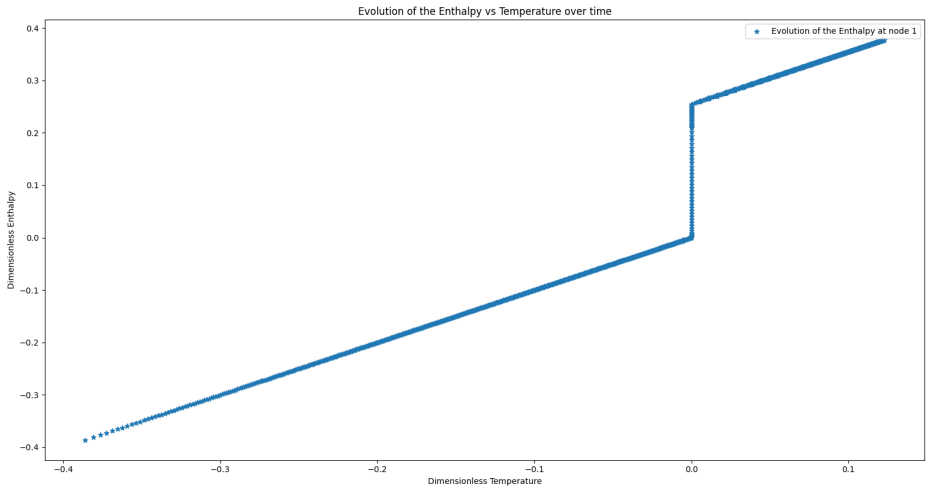

Figure 6.13: Enthlapy vs Temperature (Al, 301 Nodes ,considering Latent Heat effects)

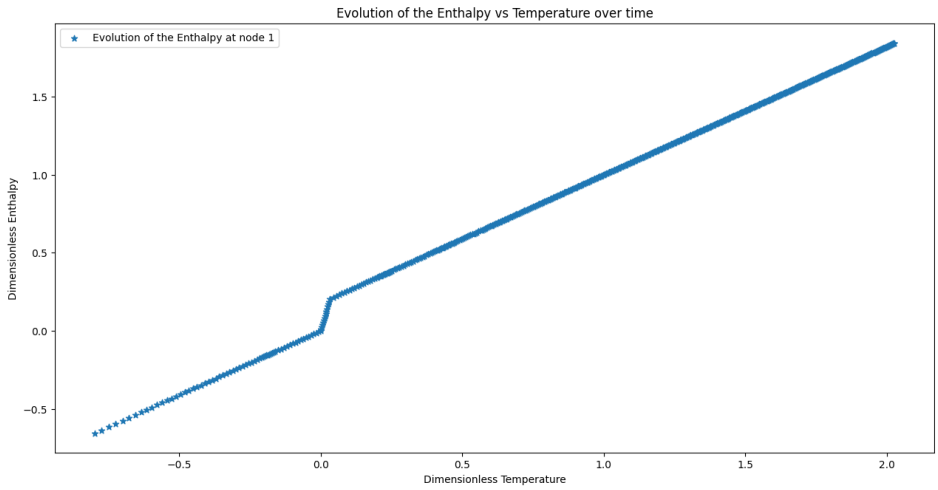

Figure 6.14: Enthlapy vs Temperature (SS304L, 151 Nodes ,considering Latent Heat effects)

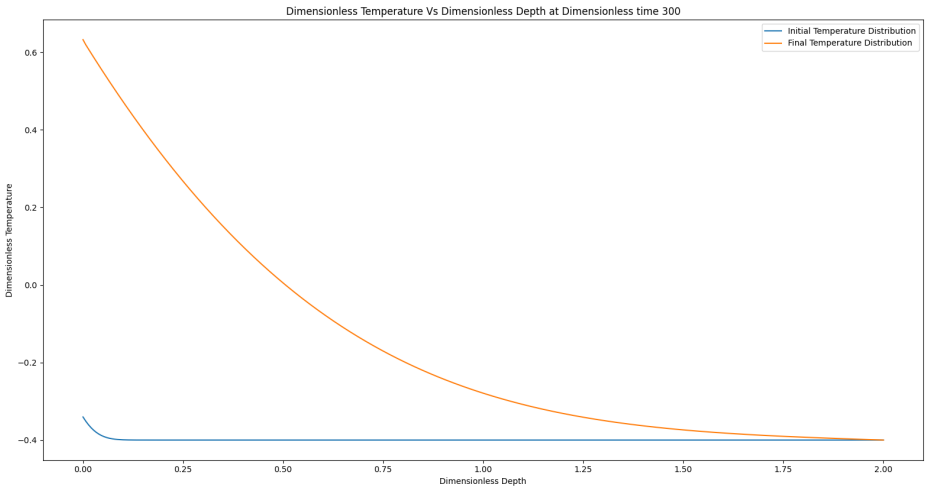

Figure 6.15: Temperature Distribution in Al (n = 301 nodes, without considering latent heat effects)

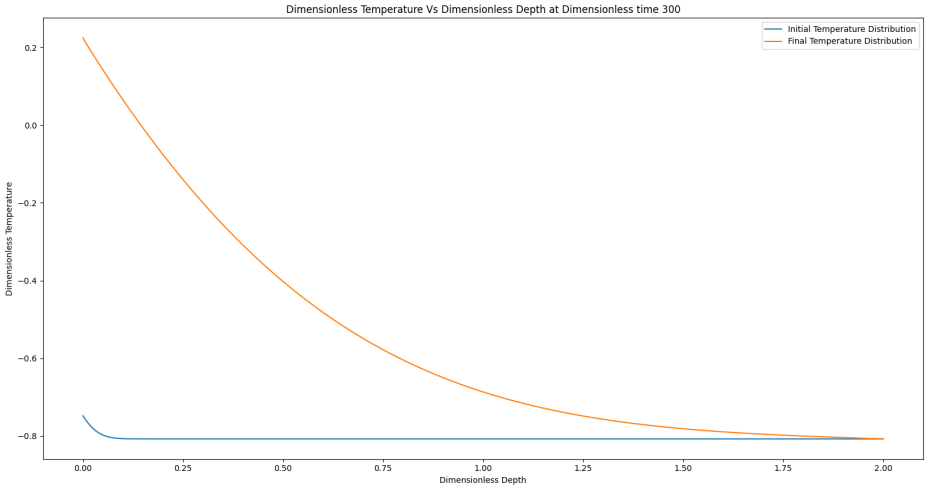

Figure 6.16: Temperature Distribution in SS304L (n = 301 nodes, without considering latent heat effects)

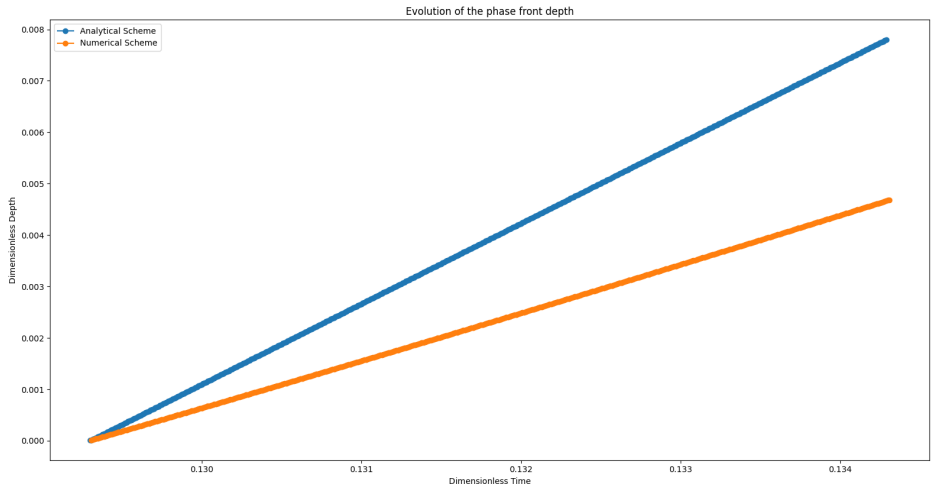

Figure 6.17: Position of the phase front from numerical scheme (considering the latent heat) and analytical scheme (neglecting the latent heat effects)

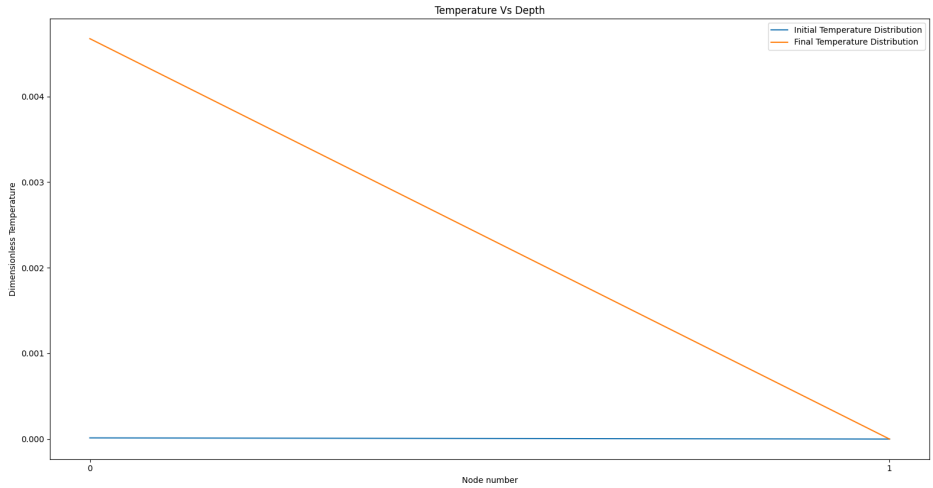

Figure 6.18: Temperature distribution in the liquid domain

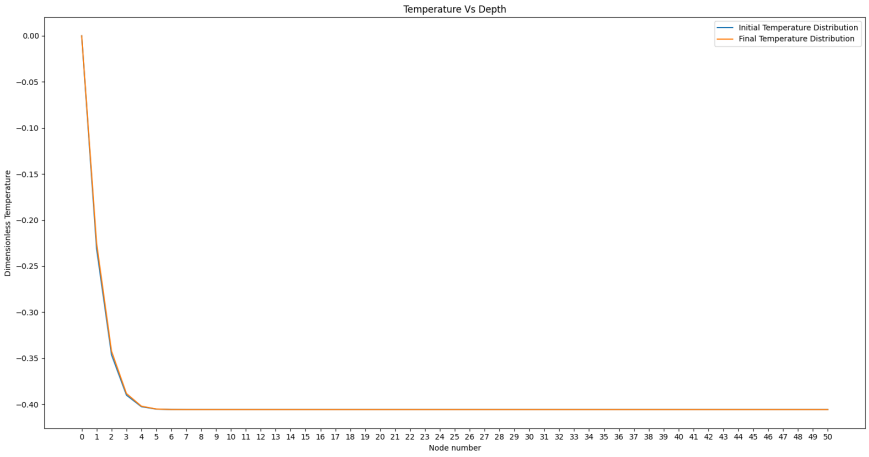

Figure 6.19: Temperature distribution in the solid domain

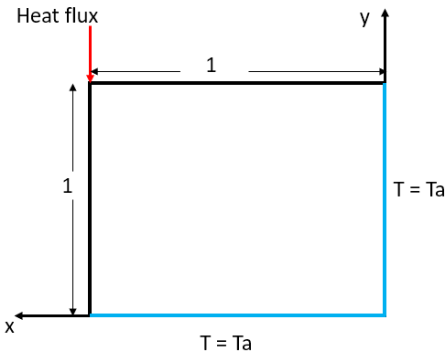

Figure 6.20: Settings for the 2D Enthalpy problem

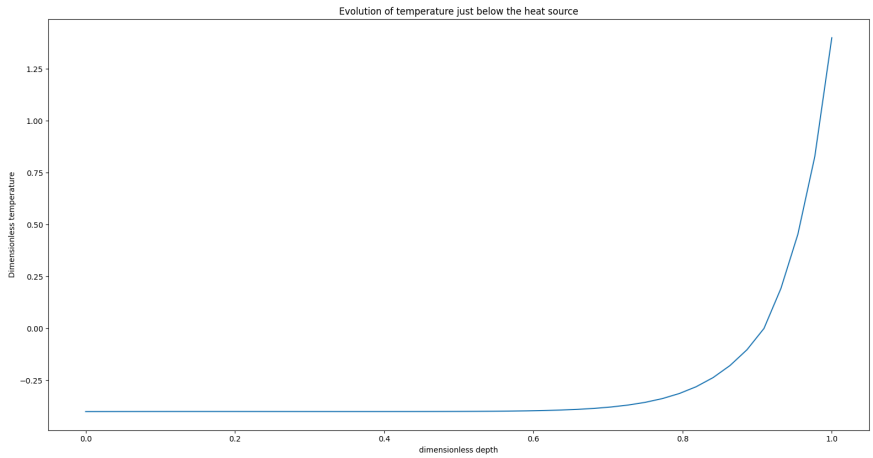

Figure 6.21: Temperature evolution from the node just below the heat source

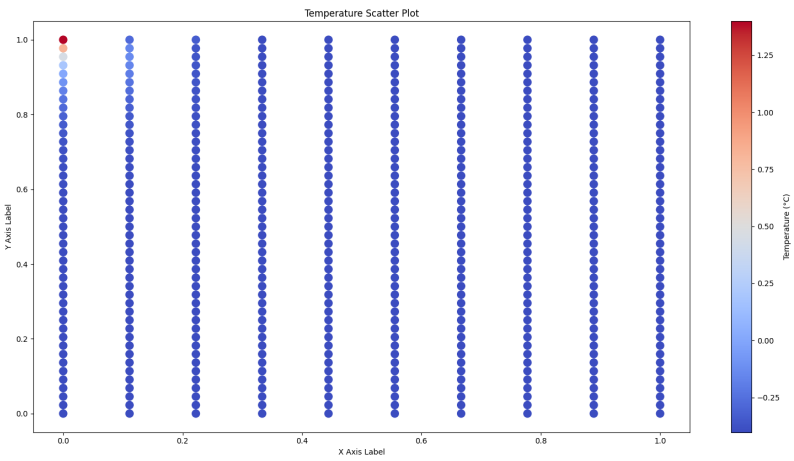

Figure 6.22: Temperature distribution in the matrix at end time

7 User Manual

7.1 List of files and folders

7.1.1 Enthalpy_Approach_1D

1. Amorphous_inputs.py
2. Crystalline_inputs.py
3. Elemental_Subroutine.py
4. FEA_solve.py
5. Initial_Temperature_Dist.py
6. Material_Subroutine.py
7. Mesh.py
8. postProcessor.py
9. Processed_inputs.py
10. variableUpdater.py

7.1.2 Enthalpy_Approach_2D

1. Amorphous_inputs.py
2. Crystalline_inputs.py
3. Elemental_Subroutine.py
4. FEA_solve.py
5. Gridify.py
6. Initial_Temperature_Dist.py
7. Material_Subroutine.py
8. Processed_inputs.py
9. variableUpdater.py

7.1.3 Heat_Transfer_Verification

Along with the above mentioned files the Heat_Transfer_Verification has one more additional file Heat_Transfer_Verification.py. The file Initial_Temperature_Dist.py is changed in this folder. Remaining all the files are same.

7.1.4 Testing

1. Amorphous_inputs.py
2. Crystalline_inputs.py
3. Gridify.py
4. Mesh.py
5. Mesh_Test.py
6. Processed_inputs.py
7. Processed_inputsA.py
8. slope_matrix.py
9. Slope_MatrixA.py
10. Test_Amorphous.py
11. Test_Crystalline.py
12. Test_Mesh_2D.py

The Gridify.py is the 2D mesh file, Slope_MatrixA.py and slope_matrix.py are part of Material_Subroutine.py. All the other files such as Processed_inputs.py and Processed_inputsA.py are same just the material selected is different.

7.1.5 Stefan_ Approach

1. Crystalline_inputs.py
2. Elemental_Subroutine.py
3. FEA_solve.py
4. Initial_Conditions.py
5. postProcessor.py
6. Processed_inputs.py
7. variableUpdater.py

7.2 Libraries used

The libraries we used in programming the simulation is shown in table 7.1.
The installation of the libraries mentioned in table 7.1 can be done as follows:-

Library / Software	**Version**
Python	3.11.2
Numpy	1.24.1
math	3.5
matplotlib	3.6.3
Time	3.11.2 (inbuilt)

Table 7.1: Libraries used and their versions

Installing the numpy.
$ pip install numpy
Installing the math.
$ pip install math
Installing the numpy.
$ pip install matplotlib

7.3 Running the program

1. To begin, please navigate to the folder that contains the files related to the simulation you desire to run.

2. The files 'Amorphous_inputs.py' and 'Crystalline_inputs.py' contains the inputs which can be edited for user defined inputs.

3. Next, select the type of material in the 'Processed_inputs.py' by setting the mat_type = 1 for the crystalline material and 2 for the amorphous material for the enthalpy problem approach. Note for Stefan problem approach this variable is always = 1.

4. Afterward the simulation can be run using the command line with the command python .\FEA_solve.py for standard Windows operating system.

5. Finally, the results will be displayed as graphs and on the command line as well.

8 Computation time study

The 'time' module is built into the Python interpreter, and we have utilized this module in our program to measure the time required for the execution of each module. The modules and their respective execution times are presented in Table 8.1.

It can be observed that the solver requires the most time to execute. In the previous

Library	**Time** (in s)
Mesh	0
Gridify	0.01
Initial_Temperature_Dist	0
variableUpdater	0
Elemental_Subroutine	0.491
FEA_solve	16.3031
postProcessor	9.6406

Table 8.1: Computation time study

versions of the program, where the assembly was done using the assignment matrix, the time required was quite large. Introduction of mesh list removed the need of assignment matrix. Hence, an improvement in the computational time was observed. Further, the computational time can be reduced by utilizing the 'scipy' package in python which includes a sparse matrix solver. This solver is considerably more efficient than computing the inverse of the stiffness matrix, which is computationally expensive. Additionally, the assembly is now done in a separate function, which allows parallelization of the execution.

9 Conclusion

In conclusion, we have successfully developed a program to track the location of the solid liquid phase front using two approaches: the Enthalpy problem approach and the Stefan problem approach. The former method is referred to as a fixed domain method, while the latter is referred to as a front tracking method.
In the fixed domain method, the temperature-enthalpy relation is used to track the phase front.On the other hand, the front tracking method employs the Arbitrary Euler-Lagrange Approach to track the position of the phase front. The fixed grid method is applicable to both amorphous and crystalline materials, while the variable space grid can only be employed for crystalline materials.
In the Enthalpy problem approach, for both amorphous and crystalline materials, we validated two cases for the heat transfer verification. The first case with both the ends at a fixed temperature, usually referred to as type I boundary condition, and the second case includes one end at a fixed temperature and other end insulated, commonly referred to as type II boundary condition. In both the cases, we observed a good agreement with the analytical solution. We also conducted a mesh convergence test and found an exponential decay in errors after 300 nodes.
For the type II boundary condition, we analyzed the effect of adding a term to the analytical solution. The location of the phase front was tracked successfully using this method and was compared with the analytical scheme. A staircase-like behaviour in the solution of the phase front location for lower number of nodes from the Enthalpy problem approach was observed. Additionally, we encountered a convergence issue during the phase transition in the 2D fixed grid method due to the the temperature at the first node crossing the vaporisation temperature, making the problem unphysical.
We also compared the location of interface obtained from the variable space grid method. The solution exhibited a good trend with the analytical one, but exact overlap is not achieved due to the latent heat effects. Further, in 2D settings, we saw for the fixed grid method we employed the new numerical approach as suggested by Zhao-chun Wu [4]. However, his method has issues with the velocity update.
The entire program was built using Object-Oriented Programming, as methods can be passed to a function or another method along with the attributes by just passing an instance of a class, allowing simplicity in programming.

9.1 Problems Encountered and Experiences

Earlier, for one dimensional Enthalpy problem approach, only type II boundary condition for the verification of the heat transfer was planned. The heat transfer results from the scheme showed an acceptable agreement with the analytical one with some stability issues.
Later, I decided to independently verify the results of heat transfer for a type I boundary condition to assess the stability condition. During this second verification, the results exhibited a bias toward the end where heat flux was applied. Upon performing manual calculations to investigate this bias, it was discovered that the input laser had not been correctly turned off. The scheme was so designed that laser power in the inputs script would be overridden by the material model. This was due to the Object-Oriented programming where the instance of the material model was not lost even after the function execution was completed. This led to inaccuracies throughout the scheme.
Lessons Learnt: While Object-Oriented Programming can be advantageous for passing attributes and methods, it's important to be aware that, unlike functions where the scope of a variable is limited to within the function, and the variable is lost once the function is completed, the attributes of a class are not lost and retain the data from the previous calculation unless explicitly reset or reinitialized. This also highlights the importance of the testing and verification of the code for various cases.

9.2 Acknowledgement

I extend my sincere gratitude to Dr. Ing. Stefan Prüger for his invaluable assistance at every stage of this project, particularly in the verification of the type I and II boundary conditions. His guidance and insights were of great help in addressing challenges, understanding the physics and enhancing the overall quality of the work.

I would also like to acknowledge the contributions of authors of the paper 'Modelling laser induced melting' Verhoeven J.C.J, Jansen, J. K. M., Mattheij, R. M. M., and Smith, W. R. whose mathematical formulation served as the foundation for implementing the Finite Element Method (FEM) scheme in our project.

Lastly, I am thankful to TU Berlin for providing the LaTeX template on Overleaf, enabling us to produce this work seamlessly and efficiently.

9.3 Milestones - status

Milestone	Status
Programming of the 1D Enthalpy problem	Yes
Heat Transfer Verification with type I boundary condition	Yes
Heat Transfer Verification with type II boundary condition	Yes
Verification of the position of the phase front	Yes
Programming of the 1D Stefan problem	Yes
Verification of the position of the phase front	Yes
Programming of the 2D Enthalpy problem	Yes
Programming of the 2D Stefan problem	Yes

Table 9.1: Milestone status of the project

Nomenclature

Tensors and vectors are represented in bold with an underline ($\underline{\mathbf{A}}$)

9.3.1 Constants

ρ	Density ($\frac{kg}{m^3}$)
c	Specific Heat Capacity ($\frac{J}{kgK}$)
k	Thermal Conductivity $\frac{W}{mK}$
I_ref	Reference laser energy $\frac{W}{m^2}$
θ_v	Vapourisation Temperature K
θ_m	Melting Temperature K
T_v	Non dimensionalised vapourisation temperature
T_m	Non dimensionalised melting temperature

9.3.2 Variables

τ	Time (s)
θ	Temperature (K)
χ	Depth in z direction (m)
r	Radius (m)
I	Laser input energy $\frac{W}{m^2}$
w_0	Waist m
z	Non dimensionalised depth
$\bar{r}$	Non dimensionalised radius
t	Non dimensionalised time
T	Non dimensionalised temperature
L_f	Latent heat of fusion
s	Position of the solid liquid interface

9.3.3 Subscripts

0	initial or starting value
s	solid domain
l	liquid domain
sol	Solidus
liq	Liquids

w_0	Waist m
z	Non dimensionalised depth
$\bar{r}$	Non dimensionalised radius
t	Non dimensionalised time
T	Non dimensionalised temperature

Bibliography

[1] Stuart Brorson. Mathematics libretexts. https://math.libretexts.org/Bookshelves/Differential_Equations/Numerically_Solving_Ordinary_Differential_Equations_(Brorson)/01%3A_Chapters/1.03%3A_Backward_Euler_method#:~:text=Stability%20of%20backward%20Euler%20for%20the%20exponential%20growth%20ODE,-Now%20we%20derive&text=Note%20that%20this%20stability%20region,only%20stable%20for%20small%20steps., July 2022.

[2] John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[3] Yunus A Cengel and Afshin J. Ghajar. *Chapter 02*, page 82–89. McGraw-Hill Education, fifth edition, 2020.

[4] Zhao chun Wu. A new numerical approach to two-dimensional stefan problem. *International Journal of Turbo and Jet Engines*, 22(4):281–288, 2005.

[5] L. Debnath. *Nonlinear Partial Differential Equations for Scientists and Engineers*. Birkhäuser Boston, 2010.

[6] Gaalen Erickson and Andrée Tiberghien. Heat and temperature. *Children's ideas in science*, pages 52–84, 1985.

[7] Boris Grigoryevich Galerkin. Series solution of some problems of elastic equilibrium of rods and plates. *Vestnik inzhenerov i tekhnikov*, 19(7):897–908, 1915.

[8] Ben Hills. An assessment of several numerical solutions to the stefan problem, 2016.

[9] Hancock Matthew. 18.303 linear partial differential equations, fall 2004, 01 2004.

[10] pahio (2872) pahio (2872). using laplace transform to solve heat equation, May 2015.

[11] Sergey Repin. One hundred years of the galerkin method. *Computational Methods in Applied Mathematics*, 17(3):351–357, 2017.

[12] Luca Salvatori and Niccolò Tosi. Stefan problem through extended finite elements: review and further investigations. *Algorithms*, 2(3):1177–1220, 2009.

[13] Luca Salvatori and Niccolò Tosi. Stefan problem through extended finite elements: Review and further investigations. *Algorithms*, 2(3):1177–1220, Sep 2009.

[14] Johan Stefan. Über die theorie der eisbildung, insbesondere über die eisbildung im polarmeere. *Annalen der Physik*, 278(2):269–286, 1891.

[15] Karl-Hermann Tacke. Discretization of the expliclt enthalpy method for planar phase change. *International Journal for Numerical Methods in Engineering*, 21:543–554, 1985.

[16] JCJ Verhoeven, JKM Jansen, RMM Mattheij, and WR Smith. Modelling laser induced melting. *Mathematical and Computer Modelling*, 37(3-4):419–437, 2003.

[17] J Zhou, H L Tsai, and T F Lehnhoff. Investigation of transport phenomena and defect formation in pulsed laser keyhole welding of zinc-coated steels. *Journal of Physics D: Applied Physics*, 39(24):5338, dec 2006.

[18] Y Zhou and A C Fernandez-Pello. An enthalpy-temperature hybrid method for solving phase-change problems and its application to polymer pyrolysis and ignition. *Combustion Theory and Modelling*, 4(4):477, dec 2000.

9.4 Git Log

```
commit 2b81d865ab751497ce9e1dbb9ef01f25441c9831
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Oct 8 16:37:35 2023 +0200

    Final Commit

commit d204a47235ab500882d47abcc909b06eebfb6514
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Oct 8 02:22:10 2023 +0200

    Update Class_intro.drawio

commit 9cbf846a13a01d76f92d603fb46f3941085c6b58
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Oct 8 02:17:08 2023 +0200

    Update Stefan_Overview.drawio

commit 6d8a808243403023375bce6a1d3065f1ddda4e5a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 20:45:54 2023 +0200

    Update Stefan_Overview.drawio

commit e2d0e92c1407142adbffaf0cd7162b6e049a3293
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 20:44:54 2023 +0200

    Update Stefan_Overview.drawio

commit 7c28f1059ecff0a5538253ba706351aa65188a64
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 20:43:05 2023 +0200

    Update Stefan_Overview.drawio

commit aea9fe257714c02240485a5b0cbeb244683c26e8
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 20:42:13 2023 +0200

    Update Stefan_Overview.drawio

commit 2a968fa62ecc0557c88f6af06cd8b51616b38bff
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 20:35:36 2023 +0200

    Update Stefan_Overview.drawio

commit 00129755e9ece456dee3d94cbe1f15491df548a6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 20:10:11 2023 +0200
```

```
    Update Stefan_Overview.drawio

commit a324f7fda426e9af0d0d5913e4cfc74142f02863
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 20:07:39 2023 +0200

    Update Stefan_Overview.drawio

commit 79975174cb40fdf332209557becad1ed82b076bd
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 20:03:08 2023 +0200

    Added Stefan_Overview.drawio

commit 961d2a022740789b257580ff01fd2d92e71dccf2
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 18:59:39 2023 +0200

    Update Stefan_Problem.drawio

commit 37799d1369a231924e80944cc93fd8496d2ebf0c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 18:59:08 2023 +0200

    Update Stefan_Problem.drawio

commit 455ee86919e142abd9e324775dad12fdafb556d7
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 18:58:10 2023 +0200

    Update Stefan_Problem.drawio

commit 8ba1ebac4ea31b7a8b5aad2d3fbc08c4a9c2cabe
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 18:55:25 2023 +0200

    Update Stefan_Problem.drawio

commit de34af419cbf35f0b0201ae13e3ee2451d380f25
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 18:47:41 2023 +0200

    Update Stefan_Problem.drawio

commit a6c8b66d0402120adb2d4483ad656fd62e863ce9
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 18:28:49 2023 +0200

    Added Stefan_Problem.drawio

commit 969b16503b0c0cf27ec563cb9d4186a88bed3a31
Author: Omkar Kolekar <okolekar123@gmail.com>
```

```
Date:   Sat Oct 7 03:06:25 2023 +0200

    Update Untitled Diagram.drawio

commit bcf66587595fca98963d133a31a25bd4f182ae12
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 03:04:49 2023 +0200

    Update Untitled Diagram.drawio

commit 805d5ac46a3a335a46343853b0e11bc1e4cf1f2d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 02:47:01 2023 +0200

    Update Mesh.drawio

commit 51ea95b5adb82a9ab35d15936da4dc4d7942212d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Oct 7 02:27:44 2023 +0200

    Update .drawio

commit 1baae5755df8e103c4cca72982fb129efab192a3
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:53:58 2023 +0200

    Update Untitled Diagram.drawio

commit 7ac3f3ece17bb7d94cb39f1edf0d1f28c8d99cd1
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:52:12 2023 +0200

    Update Untitled Diagram.drawio

commit 9bc5f00f26e3f4a9f6ff7f72f0b0f606fd7535ec
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:50:03 2023 +0200

    Update Untitled Diagram.drawio

commit 202a09c3b97839428c19c200c09284bcbda97ad9
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:15:55 2023 +0200

    Update .drawio

commit afddfdf69a2f114bc3af2ea86b5bc1e7f5f3f5b0
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:15:04 2023 +0200

    Update .drawio

commit 6ee287687b98d13b86813421b1369cfed9a0d054
```

```
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:13:36 2023 +0200

    Update .drawio

commit 503b185c006b46a8a199b88fa6f106a0e5a1b4cb
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:02:10 2023 +0200

    Update .drawio

commit 6d2b417d54f7877f57919046812ad44fa2a21443
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:01:59 2023 +0200

    Update .drawio

commit cf914963a8163150bb55fad71c990cdd9d5150dd
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Oct 6 01:00:55 2023 +0200

    Update .drawio

commit c7074d02095a5b8a02a7106efa3e717db15a3901
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 22:44:13 2023 +0200

    Update .drawio

commit 8d8b3dbb01d5e33dd04bab7a1f3347eacb1cff98
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 21:58:37 2023 +0200

    Update .drawio

commit 1c2d513a26906de9adf4e7c5a58bd610acf16f14
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 21:58:24 2023 +0200

    Update .drawio

commit 0894ad292bfb23baa34f3af4798b4010cdceb403
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 19:57:00 2023 +0200

    Update .drawio

commit 43efdee1649925921e4a866cbd8fc9a984ddff29
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 19:56:28 2023 +0200

    Update .drawio
```

```
commit d6d01648a791218679f40b8801b2d6ef1ba35158
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 19:46:13 2023 +0200

    Update .drawio

commit c03713ec86dea35c6d4584c3299b9aff02c69df8
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 19:40:07 2023 +0200

    Update Mesh.drawio

commit 1454fc8480d845c5dcf7987852b1c5452eb6af3f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 19:37:47 2023 +0200

    Update Mesh.drawio

commit a88bca653e7133c6ff5cc4c8898701a3f1965ff4
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 19:18:05 2023 +0200

    Update ElementalSubroution.drawio

commit e08e801e972f3eb8a222cfa495fb0c15885a8d80
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 19:04:44 2023 +0200

    Update ElementalSubroution.drawio

commit 6016bea50a988c6dd0105b2485efddabcd3938d2
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 19:04:33 2023 +0200

    Update ElementalSubroution.drawio

commit 6dd32488fcc6af6e956adb6b3c27cc601d227284
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Oct 5 18:07:46 2023 +0200

    Update ElementalSubroution.drawio

commit 113c8db202330b1245a37ce2394ff2577f6341c1
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Oct 4 21:27:33 2023 +0200

    Update Over_View.drawio

commit e4f85db2c44efe87a7fb1424d18147eb00248e9c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Oct 4 21:26:50 2023 +0200

    Update Over_View.drawio
```

```
commit 5e0423d8789b7826bc49a7eee720cd4642a75a96
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Oct 4 21:24:29 2023 +0200

    Update Over_View.drawio

commit b3a687908678bc0bf8838e718ae3d8e95484a8ae
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Oct 4 21:23:45 2023 +0200

    Update Over_View.drawio

commit 239c50d7cab5193dfc25e1d4f3f00c6c089c0f38
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Oct 4 21:11:37 2023 +0200

    Update Over_View.drawio

commit 82a37ab9d9f5484955b0da9b81001a1834b61a4e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Oct 4 21:00:12 2023 +0200

    Update Over_View.drawio

commit ba26b853f02ce762f9a014389fec2f88ea5074ab
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Oct 4 20:59:51 2023 +0200

    Update Over_View.drawio

commit 77a2df1e9423e2adfbbc411170e72218a4bcf1e5
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Oct 4 20:56:06 2023 +0200

    Update Over_View.drawio

commit 9ede74e337e5a55edcc9c08238873fb5df4db129
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Sep 30 15:14:21 2023 +0200

    Work in progress

commit d4e0bfd9cca306aa3839701fcf17bfe3a91a6be2
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Sep 30 15:13:23 2023 +0200

    Create Crystalline_inputs.py

commit 2c41d0baaf20cc35904448ed2071ee600c3ec3ab
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Sep 30 15:10:43 2023 +0200
```

```
    Add files via upload

commit f49e9000a865b53d0e33db4d5fd54e05632e4657
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Sep 30 15:09:41 2023 +0200

    Create Crystalline_inputs.py

    Verification for temperature done

commit 6d292ba940422d16f977570832f11ddbf2209ffd
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Sep 27 23:39:37 2023 +0200

    Mushy zone worked deleted the previous work

commit a540eb989cf7de9492ff45d445b9e0078fe61a6a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 02:49:45 2023 +0200

    Update Untitled Diagram.drawio

commit 050aac7645cf5a87856de94a30787ee6cdc29ade
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 02:47:49 2023 +0200

    Update Untitled Diagram.drawio

commit 2fa5df385eeda4cbe98c2dad2309160c45a16c2e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 02:40:06 2023 +0200

    Update .drawio

commit 79648e1101609664cae19a8fd9294aca7c8bfa25
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 02:00:00 2023 +0200

    Update .drawio

commit 24d2726c29bf77992c0403d9b31f56b92f20f618
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 01:52:21 2023 +0200

    Update Over_View.drawio

commit da994dba772a4ecfabfed1def9332fc4e7ad2e37
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 01:52:20 2023 +0200

    Update Over_View.drawio

commit 435947bbaf5119807a38ecd02f8e1326510d4194
```

```
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 01:41:31 2023 +0200

    Update Class_intro.drawio

commit 43ebe122d8f703cec192a6505dad0b1423ea999b
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 01:39:10 2023 +0200

    Update .drawio

commit dd537736505c02ebb1bc73c0f2f6fb28a0254679
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 01:38:59 2023 +0200

    Update .drawio

commit d23454480006288f2bf197dc9d64135613b91480
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Sep 22 01:25:52 2023 +0200

    Update .drawio

commit ad121319028aee24295da170941f37b3fc00fafc
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 21 21:21:34 2023 +0200

    Update .drawio

commit b19688b589828f74a074b8f187df458eb01670a7
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 21 13:20:13 2023 +0200

    Update .drawio

commit 8fed139e82553e92077d95db79f4615383ec054f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 21 13:15:08 2023 +0200

    Update Class_intro.drawio

commit 5d3cac1be94d40b1f742f1bbd031ea8763c92196
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Sep 12 15:59:08 2023 +0200

    Stefan_Analytical_Enthalpy

    I have added the Stefan analytical solution and the Enthalpy solution
         together to compare the results done for the crystalline inputs
        only.

commit 5be1d45a502b09ed6336a7cc5cbc5b9d72570095
Author: Omkar Kolekar <okolekar123@gmail.com>
```

```
Date:   Tue Sep 12 15:58:02 2023 +0200

    Create Crystalline_inputs.py

    I have added the Stefan analytical solution and the Enthalpy solution
         together to compare the results done for the crystalline inputs
        only.

commit 7699f7e6a31ada1acf5f6321642b5dfd460c6f49
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Sep 11 12:42:50 2023 +0200

    Melting_Tracked

commit 5cd04064899e07b3f61c2f7f56019aef0b4d879a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Sep 11 12:40:58 2023 +0200

    Melting Tracked

commit c9bd1c5a05418259e0162e91e44e78980860e10c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Sep 11 12:39:14 2023 +0200

    Melting Tracked

commit 47bac9bcb69627308c8d2a910a5be7009da61312
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:54:40 2023 +0200

    Update Over_View.drawio

commit 13ef4065ca83e6996ce6d8bf12b98b0bf21a1ad0
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:54:16 2023 +0200

    Update Over_View.drawio

commit 6de8366a0841ecf8a4c5b897848b6bc429152853
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:53:13 2023 +0200

    Update Over_View.drawio

commit b72e117882162a6a0efd2cf22070f797d2ba1536
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:51:33 2023 +0200

    Update Over_View.drawio

commit 4ff02703e412f3b7d2c3d90fce76ff68ea819069
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:50:50 2023 +0200
```

```
    Update Over_View.drawio

commit 15049be5f9d6a033e7ae98eaecdba9f5289244a7
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:46:18 2023 +0200

    Update Over_View.drawio

commit a4053e86b59708771c7e3f05e07a4131a98a2e82
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:45:41 2023 +0200

    Update Over_View.drawio

commit 8c6ed81030a52a0a7dd05b083c32818ca05c5d74
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:43:48 2023 +0200

    Update Over_View.drawio

commit 82a66833c2217ebb7ab263d11a24c674f523bb72
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:42:53 2023 +0200

    Update Over_View.drawio

commit 264251622b3d00b276215f93698c505f554b1f99
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:42:02 2023 +0200

    Update Over_View.drawio

commit 37683015596232d9e61d97d2858d19acf6f8e938
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:39:17 2023 +0200

    Update Over_View.drawio

commit 74b191a50ad71b36427cd04e28a51ee860482544
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:32:00 2023 +0200

    Update Over_View.drawio

commit f2824c93dad7a2a4190b47de24bb8099b1404e6d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:31:22 2023 +0200

    Update Over_View.drawio

commit 2a64b247dd71355c6a7aa44be935213799be9886
Author: Omkar Kolekar <okolekar123@gmail.com>
```

```
Date:   Thu Sep 7 17:27:43 2023 +0200

    Update Over_View.drawio

commit 34ba0ccfd6fa9078d3045b74e6ca8126e90c7ff0
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:26:52 2023 +0200

    Update Over_View.drawio

commit f8ba74fd03e180ccda1657dbe0248047e701e22e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:21:52 2023 +0200

    Update Over_View.drawio

commit 54792ae27169049241f30b34ee9e0579b8fec7a1
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Sep 7 17:21:16 2023 +0200

    Update Over_View.drawio

commit 76e842b7ce183086f4173e6d60acd00548eb823c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Sep 6 02:53:41 2023 +0200

    Update Gridify.py

    Updated the boundary nodes and the comments

commit 425ad9c3b3b292d99621795403df36d49c89d9e1
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Sep 6 00:34:54 2023 +0200

    Update Gridify.py

    Jacobian Updated and ready for action

commit 36dc74889becef80514d884be4e3a5c8a1a6b124
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Sep 6 00:11:08 2023 +0200

    Create Gridify.py

    The Jacobian is not updated

commit 681677d4fd3c8bc9c628eb65a3983b8b077dbf51
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Aug 31 18:45:31 2023 +0200

    Update Class_intro.drawio

commit 3341360709c49f0eb6047f6ea7be8e8a0a927f76
```

```
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Aug 31 18:22:31 2023 +0200

    Update Class_intro.drawio

commit 83fc89853e4dd994a6d0525920d628b939cbabaa
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Aug 31 18:15:27 2023 +0200

    Added Class_intro.drawio

commit 79c874aca8dc1045f593b91780e04b8414db3a51
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 17:52:44 2023 +0200

    Add files via upload

commit adc1a10bd6c9c035d22b509f2083f4960ef78a70
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 17:50:40 2023 +0200

    Update Crystalline_inputs.py

commit f718ff416f98062d87e26bc17ce74953d7b53c5f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 17:50:15 2023 +0200

    Update slope_matrix.py

    corrected the crzstalline slope

commit 248458f174635989146dcefd4f536dbe04a1e1dd
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 17:27:19 2023 +0200

    Add files via upload

commit 674f5d3006fb6164b6c316c47e5dfcf1e0f6ebce
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 17:16:49 2023 +0200

    Add files via upload

commit a384fb80eeb804a133244a9a01fbbbed4a29cf70
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 17:15:47 2023 +0200

    Create FEA_solve.py

    Temperature distribution achieved with the foll settings

commit 0ae6dcc4e3cb1cdb58c887f21afc2bc8ba3a83ca
Author: Omkar Kolekar <okolekar123@gmail.com>
```

```
Date:   Wed Aug 30 17:14:35 2023 +0200

    Add files via upload

commit 3c109637ec818c57e08452ce883bb6f6cfbe57a4
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:55:03 2023 +0200

    Add files via upload

commit 6f34499611c585468b0a3028f89b9f2ffacb3b3a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:54:20 2023 +0200

    Create FEA_solve.py

commit 944b60aada536137c30e68c124ff09d42f2ef655
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:39:09 2023 +0200

    Add files via upload

commit 9f0de3e22b70971a305d9eb96d56481b4592d0e9
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:38:44 2023 +0200

    Add files via upload

    Verification done

commit 81934106983c1668e200cfc6e14aa4f251db15cb
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:38:05 2023 +0200

    Create FEA_solve.py

    Done the verification

commit a0c979359002038ebcde2adf155d763210726d0e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:22:19 2023 +0200

    mushy amorphous

commit 844c9e918a5d510ab9882e9f0748e5943959aa8e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:21:51 2023 +0200

    mushy amorphous

commit 812cae5571feb16723ffdb9132e76735e7fcbc0a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:20:33 2023 +0200
```

```
    mushy_Amorphous

commit 904ade120181e3b11cb00eae659e94740dc411a4
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:19:58 2023 +0200

    mushy amorphous

    _coded_specifically_to_amorphous_mat_to_track mushy zone

commit 81b4aae3fb093db4c1e45e39b25bfad8e8275701
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 30 16:18:05 2023 +0200

    Create FEA_solve.py

    I have added the code to track the mushy zone

commit 354e86263c866f4d29e8b143bed374deaae47672
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:39:40 2023 +0200

    Update Mesh.drawio

commit ce259cf6af04c1534ece97fd176c891e9fa0b11c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:38:25 2023 +0200

    Update Mesh.drawio

commit 506bf107b4d50a5789139783697fefcbdb3cc3b9
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:38:16 2023 +0200

    Update Mesh.drawio

commit 3d6d88f252cf2376aa21daf81ed6772935fcf900
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:30:38 2023 +0200

    Update Mesh.drawio

commit a647bf468119b36158c31051d6a9f2e2ed934eb1
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:27:38 2023 +0200

    Update Over_View.drawio

commit 5d5998a9208acda654d0c6eb5ba665e567bc0a2e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:21:32 2023 +0200
```

```
    Update Over_View.drawio

commit 4d13dc8d722a72112168e0f410ac9e23e84d4146
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:19:14 2023 +0200

    Update Over_View.drawio

commit 1a027f50b5f79d61bd6c89154a0173d23cb1e1dc
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:18:33 2023 +0200

    Update Over_View.drawio

commit aa05074c60b2191d7c55a216c0122bdb970c1054
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 16:13:38 2023 +0200

    Update Over_View.drawio

commit de99d7f1fa89c4234ec7551f2b2a13980af7dca8
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 15:47:04 2023 +0200

    Update Over_View.drawio

commit 1bee1b9c99606d0083119fbc3c3a16f52573c93d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 15:43:55 2023 +0200

    Update Untitled Diagram.drawio

commit 8875cdde14c35ed3cf4c629277775d0c24a7f3b5
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 15:35:31 2023 +0200

    Update Over_View.drawio

commit dd753b29bdcfed218ea891e7d80c522b86d3d17c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 15:33:43 2023 +0200

    Update Over_View.drawio

commit 34236f9161f196ad283fab5e6a40c54cebde5884
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 15:20:54 2023 +0200

    Update .drawio

commit ca7c34643c90f31d57c66318dc7ac26a20fe804e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 15:20:34 2023 +0200
```

```
    Update .drawio

commit 101d31d14b37a6e51b9df50483c0a6446c1ce11f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 02:49:51 2023 +0200

    Update Over_View.drawio

commit 0504f959261d3cea49ddb98c7ef35cd992764522
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 02:48:54 2023 +0200

    Update Over_View.drawio

commit 4c6c239d91bae81e130fba3617a6645341d37b9b
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 02:48:27 2023 +0200

    Update Over_View.drawio

commit 8657bac89c1ee4b1c95cf194d8e5e0ea0cd0a41f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 02:46:10 2023 +0200

    Update Over_View.drawio

commit 1b9f9efab4065fabf5c74cc4ebc809f161011437
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 02:44:14 2023 +0200

    Update Over_View.drawio

commit 81e2b4a6ef197c2744d57bb384e1a01dbdc79080
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 02:24:04 2023 +0200

    Update Over_View.drawio

commit 1411a456ec1fbe98af90f077b90b19e34323f3d6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 01:32:06 2023 +0200

    Added Over_View.drawio

commit 54b9554db32b882cf1d3defb3bb0c94c7bb3e061
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 01:31:21 2023 +0200

    Update Mesh.drawio

commit 0d7afdc45ff987cb4ce1ebf2725ce860c5328261
Author: Omkar Kolekar <okolekar123@gmail.com>
```

```
Date:   Tue Aug 29 01:27:52 2023 +0200

    Update Mesh.drawio

commit 5cec5e84509dc9d28eb4e5746de808d7e8726478
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 01:26:46 2023 +0200

    Update Mesh.drawio

commit 2462498277af78510ec0528904db781c2efa1e3c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Aug 29 00:15:00 2023 +0200

    Update Mesh.drawio

commit 665ed12d58a42c8a0551758c05ee07fe0461c436
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 23:52:41 2023 +0200

    Added Mesh.drawio

commit 983f6851fcc13361600da63a3b726df2557002c5
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 23:47:57 2023 +0200

    Update .drawio

commit db0ef6268fd1212aa5b8c18eb30fc4a956263b8a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 23:46:02 2023 +0200

    Update .drawio

commit cc4c96db5f963fd9ade6a6c0dde24c946757c789
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 23:02:56 2023 +0200

    Update .drawio

commit d940a87def01f16f7f3fb3860dfc5767c6149aef
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 22:14:39 2023 +0200

    Initial_Temperature_dist.drawio

commit dd359470d500c2cf37fcd368d0b5092f66f5acc1
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 22:12:43 2023 +0200

    Update ElementalSubroution.drawio

commit 14e4ae890691994a7c74f4dd0ccbc8b4acdd6a7a
```

```
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 21:40:01 2023 +0200

    Update ElementalSubroution.drawio

commit b5aab17442babb04efc98bbc080eb852832ca686
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 21:39:32 2023 +0200

    Update ElementalSubroution.drawio

commit b45acae747412f00090b338f878419d6aa8e57c6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 21:34:44 2023 +0200

    Update ElementalSubroution.drawio

commit 5ad20dd0d7a54a5229941837413948c5e4ca3653
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 21:22:22 2023 +0200

    Update ElementalSubroution.drawio

commit 7b0f2dc8e97054007259b4f02039b89e4ac6e01f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 21:11:56 2023 +0200

    Update ElementalSubroution.drawio

commit acea1219763c9cd9aeb9f25bc6967e36e0bc8dc3
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 20:25:27 2023 +0200

    Update ElementalSubroution.drawio

commit 8504f01384fdaf22de5e587f3b3dd728244053c6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 19:59:52 2023 +0200

    Added ElementalSubroution.drawio

commit 79d33061ec3d7e994e68817f80352eb18fa8967d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 19:38:53 2023 +0200

    Update Untitled Diagram.drawio

commit 640d065eeeb625d771f99795e19ef27701f63920
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 19:36:18 2023 +0200

    Update Untitled Diagram.drawio
```

```
commit 8c93589cc2a566cd21b0b88dbfdedccf06ac9da7
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 19:32:44 2023 +0200

    Update Untitled Diagram.drawio

commit 3e7f8695923d8a0e6e743c798f0fd092d0b4eb8a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 18:37:07 2023 +0200

    Update Untitled Diagram.drawio

commit 6a333706413cd807d1426d876ced61ff47f24261
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 18:36:11 2023 +0200

    Update Untitled Diagram.drawio

commit fcdc59588786d8f26cc099bdfb0a16f1ab7cb16a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 16:42:15 2023 +0200

    Update Untitled Diagram.drawio

commit 787c50479b11d3d2392dfdf297188f84eba73221
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Aug 28 16:40:37 2023 +0200

    Added flow_chart_for_the_main.drawio

commit 8cfec4d9f807875b0cb4b590a2ffc5b357129ff2
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Aug 24 20:12:24 2023 +0200

    Update FEA_solve.py

    The changes made are to the boundary condition of Mass matrices and
        the Initial condition is now added in the Initial Conditions
        Script

commit 79a4eee8a1169394ec41d543b00c37d90a5c23db
Merge: 7a683e9 956499e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 23 18:42:24 2023 +0200

    Merge pull request #42 from okolekar/okolekar-patch-2

    element b corrected entry for m-1

commit 956499e9bad08b76e3c2bb1cf7643d304b8296f3
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 23 18:40:54 2023 +0200
```

```
    element b corrected entry for m-1

commit 7a683e9c67dcfd6babeb7ff00efede3c5bfb3594
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 02:22:00 2023 +0200

    Updated to run the Stefan and Enthalpy approach

commit 6e89636ccca07263b488b12e8534fe09c0705d8b
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 02:21:31 2023 +0200

    Updated to run the Stefan and Enthalpy approach

commit c9bfd031ec04b99bfbf300b2fdf4b11c13c386a4
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 02:20:49 2023 +0200

    Updated to run the Stefan and Enthalpy approach

commit 812867cd47926b2368ede10997262524e5c63785
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 02:20:23 2023 +0200

    Updated to run the Stefan and Enthalpy approach

commit fd5afdc9d738af5a67b8f66bb3b322f0778850a4
Merge: aaabad0 5e57bea
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 02:18:59 2023 +0200

    Merge pull request #40 from okolekar/okolekar-patch-1

    Create readme.txt

commit 5e57bea52665b110f3e0bdc50fc80ffc473a4cd6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 02:18:45 2023 +0200

    Create readme.txt

commit aaabad0b3c2e6125099e4a168e8e89812cb527a5
Merge: df93933 7e8b7c5
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 01:43:19 2023 +0200

    Merge pull request #39 from okolekar/Stefan_and_Enthalpy_approach

    Stefan and enthalpy approach

commit 7e8b7c560d0ea26c905bff978158567f0fa1866d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 01:39:00 2023 +0200
```

```
    Updating the module reference

commit 86da778fe5157f380f206b9825775d40d21df185
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 01:24:09 2023 +0200

    trial commit not necessary

commit 37b4185594185219c443c9f1083c27c05825cfc7
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Aug 20 00:47:43 2023 +0200

    Revert "Added the code to run the Stefan approach and Enthalpy
        approach as well."

    This reverts commit df93933214eb6c2511ed2600e3d3abd30a5600f0.

commit df93933214eb6c2511ed2600e3d3abd30a5600f0
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Aug 19 21:44:28 2023 +0200

    Added the code to run the Stefan approach and Enthalpy approach as
        well.

commit c7492fdfd09e9bf2b99b258706aa692c63161727
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Aug 19 21:32:19 2023 +0200

    Changed the folder for the two noded and the mesh list generator to
        the previous versions as its irrelevant in the push on 19 August
        2023.

commit 9c08957a39f8c4676bc353fb16bc7343874bad9a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Aug 19 02:05:12 2023 +0200

    Update FEA_solve.py

    Added the Stefan Approach and verified if it is giving any errors.
    Not giving error but there are issues.
    1) The temperature distribution is incorrect I think.
    2) The delt updater is not yet coded

commit 9e8355caaeeeffa4b08b4ad8e9164d9bc2d67ee3
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Aug 11 21:19:25 2023 +0200

    Add files via upload

commit a37df6f15b4ddecbc605882802bfffb6500e1203
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Aug 11 21:17:56 2023 +0200
```

```
    Create FEA_solve.py

commit bca3b6367958bfbeac45f6e15dbcf76101a4da45
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Aug 5 02:39:10 2023 +0200

    Create 3noded_Mesh.py

commit 74ac366b46fa860ac13e7b0fe2ce17c243fd4fbf
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Aug 4 23:45:39 2023 +0200

    Create Mesh_List_Generator.py

    Added the mesh list for the 2 noded element mesh

commit fa4de9b19d50afb7b5c6028cd186948bed096b96
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Aug 3 12:29:44 2023 +0200

    Rename Enthalpy_Approach_Amorphous to Enthalpy_Approach_Amorphous.py

    Rename the extension

commit 50ad78ee8bec5c6fe0e742d79efb045d4c631b84
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Aug 3 12:25:29 2023 +0200

    Reorganised the files and brought the new version to the front

commit d7aa3e7b65197f65fa6da92a554fefb17c8135da
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:45:12 2023 +0200

    Update Processed_Inputs.py

    Added a code that handles the default case

commit 2b84e67ded9d305e7e74e97347bef477f0014ce6
Merge: d6d5a91 17823b9
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:44:11 2023 +0200

    Merge pull request #28 from okolekar/
        Verification_Results_satisfactory-6

    Update Heat_Transfer_verification.py

commit d6d5a91ab478429bd859ff87096746eb2a5f3dd4
Merge: 0b89f8c 190cff6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:44:01 2023 +0200
```

```
    Merge pull request #29 from okolekar/
        Verification_Results_satisfactory-7

    Update Initial_Temperature_Dist.py

commit 0b89f8c325152e2b3c9a81fe775acdd57668ba53
Merge: 7029429 14fe562
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:43:47 2023 +0200

    Merge pull request #30 from okolekar/
        Verification_Results_satisfactory-8

    Update Mesh.py

commit 7029429b8622ce1b0a1cb99d7d8da4d3b683bdbf
Merge: 959da89 2322b04
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:43:33 2023 +0200

    Merge pull request #31 from okolekar/
        Verification_Results_satisfactory-9

    Update Processed_Inputs.py

commit 959da891b0eef599bee20a9914ab5bfcb8cf9740
Merge: d41a2c9 da3ae14
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:43:20 2023 +0200

    Merge pull request #32 from okolekar/
        Verification_Results_satisfactory-10

    Update Material_Subroutine.py

commit d41a2c9e260271f0b7f8d30a884fba4979eae13a
Merge: 6c4d40e 233f663
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:42:37 2023 +0200

    Merge pull request #33 from okolekar/
        Verification_Results_satisfactory-11

    Update slope_matrix.py

commit 6c4d40ebe4aadfefebed53c8a1a2eeef2bc520f5
Merge: 98c8bd4 32d6ba2
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:42:07 2023 +0200

    Merge pull request #34 from okolekar/
        Verification_Results_satisfactory-12
```

```
    Update variableUpdater.py

commit 98c8bd4863917821ea3f968d7e25f6ad0adb83ff
Merge: 67edb7c a35b39e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:41:54 2023 +0200

    Merge pull request #35 from okolekar/
        Verification_Results_satisfactory-13

    Update Crystalline_inputs.py

commit 67edb7c98961cfb0da856489010a6a4824d437c3
Merge: 125504d b01603d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Aug 2 19:41:40 2023 +0200

    Merge pull request #36 from okolekar/
        Verification_Results_satisfactory-14

    Update Amorphous_inputs.py

commit b01603dd342a8d095a4deec826c582791580bb2d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 21:19:53 2023 +0200

    Update Amorphous_inputs.py

    Added t_end

commit a35b39ea53fa6e8fd4c03febe95a35e26864fa85
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 21:19:10 2023 +0200

    Update Crystalline_inputs.py

    Added t_end

commit 32d6ba243855faa0b1f01fe087afbee13f75666d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 21:16:49 2023 +0200

    Update variableUpdater.py

    No change may be

commit 233f663c2fe339292507cbb4ddf101b43a54d31d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 21:13:19 2023 +0200

    Update slope_matrix.py
```

```
    Added Comments

commit da3ae14bd3eaa28b4da8c9806245654fadccc964
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 20:39:32 2023 +0200

    Update Material_Subroutine.py

    Added the determinant of the Jacobian to the integrals of b M and N.

commit 2322b04a490fb6e55c62611ad89a0dc7b9579038
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 20:34:12 2023 +0200

    Update Processed_Inputs.py

    Added the t_end to the program
    Added some comments as well

commit 14fe5626ec36e029a29ef7766a198fbeea3593bd
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 20:11:09 2023 +0200

    Update Mesh.py

    Made provision to run the tests once at the start of the program.
    Added the comments

commit 190cff6acadf328e52959fc5fbd150c3fec7f033
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 19:57:39 2023 +0200

    Update Initial_Temperature_Dist.py

    Added a comment
    in the except block, I changed the t_start to delt

commit 17823b90f78bc2737750337a44fc4e8267732dfa
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 19:46:42 2023 +0200

    Update Heat_Transfer_verification.py

    Removed the 2 to get a very accurate result.
    Added a comment

commit 125504dd03c5922a428deb059d7151e37a53ed32
Merge: f0d89a4 69e83d2
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 19:04:32 2023 +0200

    Merge pull request #27 from okolekar/
        Verification_Results_satisfactory-5
```

```
    Update FEA_solve.py

commit 69e83d2a2beddb44bb9b8288f48831595f119750
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 19:02:38 2023 +0200

    Update FEA_solve.py

    For the verification we pass the earlier inputs to the solver i.e. we
         pass Tg[1] vector
    Some comments changed

commit f0d89a4db466d3be9ea75bfc3f03c8e3744536d1
Merge: 818c57c 7ae61fd
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 18:11:02 2023 +0200

    Merge pull request #25 from okolekar/
        Verification_Results_satisfactory-3

    Update Crystalline_inputs.py

commit 7ae61fdfcba689740934f5479746158a21bef2b3
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 18:10:41 2023 +0200

    Update Crystalline_inputs.py

commit 818c57c86b6310e7cb29b8c4c8cc05a4b159d4c8
Merge: 6ea164e 2bf70e6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 18:08:13 2023 +0200

    Merge pull request #23 from okolekar/
        Verification_Results_satisfactory

    Update Amorphous_inputs.py

commit 2bf70e6ad500f7e89e3f9d25f491695cf1ed3e1c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 30 18:00:52 2023 +0200

    Update Amorphous_inputs.py

    Updated the Intensity of the laser as per the paper in the SS304L

commit 6ea164ebd9f47ab2e07860f1fef8ef266302755b
Merge: 17d94b5 07d4fc0
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Jul 29 02:11:26 2023 +0200

    Merge pull request #22 from okolekar/okolekar-patch-15
```

```
    Update Heat_Transfer_verification.py

commit 07d4fc09abfa2ba72f5eb2700dd1f2c9a7f8970e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jul 28 12:53:21 2023 +0200

    Update Heat_Transfer_verification.py

    Changed the name of the input file to Processed_Inputs.
    Thinking to change the formula to the Heat Transfer removing the 2 as
        well

commit 17d94b550bf64adeb19cf1840aa4722b03f9e0f0
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jul 28 01:51:44 2023 +0200

    Update and rename inputs.py to Amorphous_inputs.py

    Added some comments

commit f4c522d6d8fb67beee9f73334f46c4347a82c193
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jul 28 01:50:50 2023 +0200

    Update Crystalline_inputs.py

    Added comments

commit d9bd1549c671a9054de1c7e1c3ef9252b3709442
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jul 28 01:43:59 2023 +0200

    Update and rename inputs_Al_Material.py to Crystalline_inputs.py

    Name changed prosperties of the same material Al

commit cae3cb49f573dece0d191112207499493795bbcc
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jul 28 01:33:58 2023 +0200

    new variable added to Processed_Inputs.py

    I have added test_case to switch on and off the test case of the Heat
        Transfer.
    Also renamed the input labraries

commit 212f3137d7099a34a8f258136a3e85a109f0e913
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jul 28 01:30:57 2023 +0200

    Added test case functionality
```

```
    Added the option to run the test case and other case as well

commit adf0b17b05cf5511c540ec362fb3c69932224f65
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 23:57:43 2023 +0200

    Update FEA_solve.py

    Added a few comments

commit 56c7fb54a04407ff5b43a0afc9ccc7f2fc026640
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 23:24:07 2023 +0200

    Update Elemental_Subroutine.py

    Updated the comments

commit 48ae660b0117bc5b5ba715ac0a36c8aa87af16b8
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 23:23:19 2023 +0200

    Update variableUpdater.py

    Comments updated and a run as an attribute added for the test reason

commit 2aae0e9088c4fe84995b8c6d60eac368e18751c6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 23:21:00 2023 +0200

    Update slope_matrix.py

    Changed the input library and also added a feature to check if the
        test for mesh were performed
    Also changed the Crystalline and Amorphous Material number

commit 28ac85fad7fb892dc1fde34b72e67739702d8b5d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 23:18:33 2023 +0200

    Update Mesh.py

    Added a change to the inputs which is now the processed inputs.
    I have also made the changes to the script to make the tests run at
        least once and display the message that xyz test were performed
        for at least the first run

commit 9ceb4a4bcc73e2cb904ae0f9232582f1b15cb943
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 23:15:10 2023 +0200

    Update Initial_Temperature_Dist.py
```

```
    Changed some comments

commit 399f8e6b87312aaeafbf670ad3596603a2256906
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 23:04:14 2023 +0200

    Add files via upload

commit e2f5662855186a06d5f4ba2cfaa8f9f97903fc5d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 18:40:48 2023 +0200

    Update variableUpdater.py

    Added the comments

commit 16ad4ccfb084f3070a11fe2cb36ee4c8f9b356bc
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 18:38:36 2023 +0200

    Update slope_matrix.py

    Added comments.
    Changed the number for amorphous slope from 1 to 2 and for
        crystalline from 2 to 1

commit 0a7e3b5ee78e90576b3d04d3cbe57b26715dd204
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 18:36:14 2023 +0200

    Update Material_Subroutine.py

    Added the functionality to load the amorphous or crystalline material

commit 6e1647e1b6ba3134054d02ae8811705a1e34bacb
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 18:31:42 2023 +0200

    Update Elemental_Subroutine.py

    Added the functionality to select the amorphous or crystalline
        material models

commit 1e236534ff7d2c8d1ed0337907222e2697998e02
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 27 18:19:02 2023 +0200

    Update FEA_solve.py

    Added the functionality of choosing the material type.
    Removed the check input arguments decorater as processed inputs takes
         care of the challenge
```

```
commit a5e36e4bcae7dc584b95f5b4dcacfadad692808c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 12:42:44 2023 +0200

    Add files via upload

    Inputs for the Aluminum.

commit 6f6f93fd5b3fa3ca67447aebe7c4e60219071798
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:41:40 2023 +0200

    Update Check_Inputs.py

    Updated the comments

commit 5479c7356ad7504230a2b805e9678cae4aefcb3e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:41:05 2023 +0200

    Update Elemental_Subroutine.py

    Added commenst

commit ac6bbd00c3c5e0f2e8a81f9cf37e645d63531a52
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:40:34 2023 +0200

    Update Heat_Transfer_verification.py

    Added the comments

commit e079fc7d6313e7794da1e00764e94aeeb30a50aa
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:40:03 2023 +0200

    Update Initial_Temperature_Dist.py

    Updated comments

commit d985659e34d068e1d6f03d0b23c692e76b9299da
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:39:33 2023 +0200

    Update Material_Subroutine.py

    Updated the comments

commit a3a42672c6af9cee808f4374739761acd4455c76
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:38:54 2023 +0200

    Update Mesh.py
```

```
    Updated with comments

commit a5146866be0792715182d2d452423afc95130390
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:37:42 2023 +0200

    Update slope_matrix.py

    Incorrectly uploaded the last time

commit d019cd17486c4bc1d07c34f9ca96934225855e22
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:36:42 2023 +0200

    Update slope_matrix.py

    Added comments

commit a5701206f5ef96dc0cb236be24fabbc6717d4e30
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:36:14 2023 +0200

    Update variableUpdater.py

    Updated a wrong library last time

commit ebea2dc0fc65359eeb63e4a21f02f47387f4cf60
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:35:34 2023 +0200

    Update variableUpdater.py

    Added comments

commit 906dca76a7167c4302cafe3033290045cad68028
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:34:50 2023 +0200

    Update inputs.py

    Adapted the code to the new verification format

commit 66d580f2592aa5aefd69649e920f7a5e2342e80f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 26 01:34:04 2023 +0200

    Update FEA_solve.py

    Added comments

commit 9fd4ae6a5ac96d3cec5645b63b0cb3fa930eb675
Author: Omkar Kolekar <okolekar123@gmail.com>
```

```
Date:   Tue Jul 25 02:01:50 2023 +0200

    Update slope_matrix.py

    Added comments

commit 2e37beab60de51d4c14c175b329f6c6dae81cb8f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Jul 25 01:37:02 2023 +0200

    Update Material_Subroutine.py

    Added the comments

commit dcab022c95f39fd93680b1966cb6c9d649291233
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Jul 25 00:08:04 2023 +0200

    Update Elemental_Subroutine.py

    Added the comments

commit c0f3ce104071928d3a3e229fc5de015c510974d8
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 23:50:59 2023 +0200

    Update FEA_solve.py

commit de167d8319db87d4ce204e239f12d21226853f5f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 22:51:51 2023 +0200

    Update Initial_Temperature_Dist.py

    Added comments

commit 1b03727f10d2b734ccbff38e610043f7c821e6f5
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 21:39:34 2023 +0200

    Update FEA_solve.py

    Added the useful comments

commit 51b2409aa20a666e7af810d2abf48e0613a05314
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 01:06:19 2023 +0200

    Add files via upload

    The decorator that checks that input variables are at least realistic
        ..
```

```
commit f55a5b1f760c41c448c59524e3827ee501bf36aa
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 00:54:56 2023 +0200

    Update variableUpdater.py

    The program now takes the inputs from the main input file which is
        verified at the start of the code

commit 665a5aeb11b249694180342c96ac8ad8bfb02235
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 00:52:52 2023 +0200

    Update slope_matrix.py

commit 0837a4afeba3628748cc04b093095c0fa93a3703
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 00:52:08 2023 +0200

    Update postProcessor.py

    Many functionalities added to have multiple graphs

commit a4bb1b0fb6d1dca6f04f9461c9274aabb6081258
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 00:50:22 2023 +0200

    Update inputs.py

    Conductivity added to the code

commit 8f7ee403f2c32102bd59684f2a3844f19f55c3e1
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 00:41:36 2023 +0200

    Delete dT_dH.py

    This file is now ugraded to  slope_matrix.py

commit 521be1c18866ae4bf9728b2cfd186a27fb2ff3f0
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 00:28:55 2023 +0200

    Update Mesh.py

    Not many changes here but the testing part can be improved

commit ab59b7d8abaf640b393639b3ed6cf0e30ae45c6a
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 24 00:01:11 2023 +0200

    Update Material_Subroutine.py
```

```
    mesh is now received as an argument to the init

commit 431d6d2d9f42a27c530b68dfe4242aecd24e64bf
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 23 23:58:24 2023 +0200

    Update Initial_Temperature_Dist.py

    Instead of passing a bunch of input variables, we now have imported
        them directly from the input file which is checked at the start of
         the solver program

commit 5717b26b380a5f42f948ac405df19025e70a0ed2
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 23 23:49:30 2023 +0200

    Update FEA_solve.py

    Mesh is now an attribute of the solver class.
    A decorator is added to check the input file.
    Analytical solution to the problem for the heat transfer is added in
        this version
    plot has changed a lot

commit ac28a7193178f0e611d70987a8a86056b6fc4b89
Merge: d1976b8 763ccea
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 23 23:40:21 2023 +0200

    Merge pull request #21 from okolekar/okolekar-patch-14

    Update Elemental_Subroutine.py

commit 763ccea992972c5a5fb1a6ee9b6f9abf1b07e092
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 23 23:40:04 2023 +0200

    Update Elemental_Subroutine.py

    Not many changes only now the mesh instance from the solver is passed
         everywhere

commit d1976b812dc7b4c7e6fa7eccb1e7e3b95c32947b
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 23 23:38:20 2023 +0200

    Update and rename #Heat_Transfer_verification.py to
        Heat_Transfer_verification.py

commit c834ac3765de0e8f96a1b6d51fb67d0dd8fda325
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 23 23:38:03 2023 +0200
```

```
    Update Heat_Transfer_verification.py

    Removed the unnecessary commented lines and now the verification can
        be done directly using the main program

commit 3164f370f240c85904b282bc93b24af8beeaed1c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jul 21 07:54:34 2023 +0200

    Add files via upload

commit 90d880dd7a3394d15b9cd07ac0742e4bc278b402
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 16 18:58:39 2023 +0200

    Added the entire library for the enthalpy scheme

    In this library
    1) Entire program is object oriented
    2) Fixed the issues such as the improper gauss quadrature.
    3) Now the NRS scheme convergence is independent of the element
        number.
    4) Added the tests for the Jacobian, shape functions and the
        derivative of the shape function.
    5) Added the tests for the inputs as well.

commit 4b885cc0af9f2f44557953a4ee8b7fef5f2ce8d4
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jul 16 18:54:29 2023 +0200

    Create FEA_solve.py

    In this upload I have,
    1) Made the entire program object oriented.
    2) I have added the Newton Raphson Scheme and the Time Integration
        Scheme.
    -> The rest is is made as a seperate class

commit 76329ef5e1df714b1fecae1152f81a8b668c40f0
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jul 14 16:29:26 2023 +0200

    Mesh added

    Made the program object oriented
    added the test to Jacobi to check if it is positive and equal to the
        elemental length for a gauss loop
    added tests such as partition of unity to phi and dphi

commit 44a768594a4577679eb23911391403ee4121a0d8
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jul 13 21:34:39 2023 +0200
```

```
    Create slope_matrix.py

    This program is the updated version of the dT_dH.py.
    I kept the earlier version as well to have a reference for the output
        .
    I have the tests to the program which tests if the no. of inputs are
        correctly provided then the D should always be positive and then
        the Hliq is positive the lambda is positive.
    A check is added if the off diagonal entries are zero

commit 3349d866213b29619007aefde2259264c8e6b02d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 12 18:48:47 2023 +0200

    Update variableUpdater.py

    Added a checker wrapper for the no of input variables

commit 1860c6b1631ec56e0bccf956b78e785f1bb17ef6
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 12 17:50:46 2023 +0200

    Update and rename Initial_Condition.py
        Initial_Temperature_Distribution.py

    Changes made.
    1) Made the program object oriented.
    2) Added the following test:-
        checks whether correct number of input arguments are correct
        checks whether ratioI is above 0
        checks whether the last node has a temperature equal or lower
            than that of first node.
         checks whether node list is not scalar and does not have just a
             single node

commit 18fc8bd57b09b4c20419efd787b677136d564c3c
Merge: aeef756 c5e9f4b
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Wed Jul 12 15:53:22 2023 +0200

    Merge pull request #20 from okolekar/okolekar-patch-12

    Update Enthalpy_Approach_Amorphous

commit c5e9f4b180b14cc4f9054c35541eb774e63c1572
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 10 03:15:34 2023 +0200

    Update Enthalpy_Approach_Amorphous

    Made minor changes to the code removal of some H and T

commit aeef7566eec48bad1da6008a29e72df30d296b61
```

```
Merge: bf09f97 56a74fb
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 10 02:35:06 2023 +0200

    Merge pull request #19 from okolekar/okolekar-patch-11

    Update Enthalpy_Approach_Amorphous

commit 56a74fb2546be48210488461e60fe82d8142c80c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 10 02:34:29 2023 +0200

    Update Enthalpy_Approach_Amorphous

    Added the display option as well which displays the header as H1 H2
        and so on.
    Also re moved the return of Ht as it was  not needed

commit bf09f970f7faba425754db571edd57ded72f0b28
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 10 02:04:47 2023 +0200

    Update variableUpdater.py

    Added the amorphous updater as well as corrected the invalid
        implementation of the H and T

commit 1fce59939b855047e00b5ed66e77a7b0a93f9c25
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 10 01:59:55 2023 +0200

    Added the dT_dH rule file here

    This is only for the Amorphous zone

commit 1b60e7377b459e716f8af140e147c7dbcaadadbb
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 10 01:58:23 2023 +0200

    Create Enthalpy_Approach_Amorphous

    This code was an extended version of the original code of the
        Enthalpy Approach. I have made a few changes to the original
        enthalpy code

commit 8fd7081a8c9fcf8db038f452eb66c9f5683987f1
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Jul 8 13:40:53 2023 +0200

    Add files via upload

commit 1bbd1e781565a17ccd754f7c689558af55aa69ef
Author: Omkar Kolekar <okolekar123@gmail.com>
```

```
Date:   Sat Jul 8 13:40:17 2023 +0200

    Update EnthalpyApproach.py

    In this update I have added the update variable method that updates
        the variables H and T according to the conditions set in the paper

commit 504a931dc0a28cbd2941d9f732cf68fa93852290
Merge: bc7047c d263de9
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Jul 8 13:38:25 2023 +0200

    Merge pull request #17 from okolekar/okolekar-patch-10

    Update EnthalpyApproach12.py

commit d263de97efbc5cdf3f11a93f622339ad55abbd13
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Jul 4 14:14:15 2023 +0200

    Update EnthalpyApproach12.py

    In this new version, the history of Temperature and the Enthalpy is
        stored

commit bc7047cf388352b02f2dab100790c0ed257f226e
Merge: a59ca34 64a6359
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Jul 4 14:12:48 2023 +0200

    Merge pull request #16 from okolekar/okolekar-patch-9

    H as a time History

commit a59ca34b8661b60c7dc76fd7d6ab69bcac033981
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Jul 4 02:37:31 2023 +0200

    Added the postProcessing

commit 64a6359997a4898a93dd31fd9c3a97f1d246c79e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Jul 4 02:35:58 2023 +0200

    H as a time History.py

    1) H now stores the enthalpy of all the timesteps.
    2) dT/dH is now a matrix
    3) Added the plotting file postProcessing
    4) G is now based on the previous time step of the enthalpy and not
        the previous NRS scheme as this was the error earlier.

commit 22b67f859ae478ab2b14381a01b480836bce2e83
```

```
Merge: b790b0f 681a45c
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Tue Jul 4 02:27:19 2023 +0200

    Merge pull request #14 from okolekar/okolekar-patch-8

    Fixed dT_dH.py

commit 681a45c7f4d2eef8a2b2050c8b66f630ed872221
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 3 02:27:29 2023 +0200

    Fixed dT_dH.py

    1) Fixed the  dT_dH reseversal problem
    2) Fixed the Enthalpy as now in this new issue the Enthalpy is a
       dimensionless and is considred to be H = D*T
    3) Created the dT_dH  matrix

commit b790b0f4fc467625f3c97c9b9e9af5e90cec6ffb
Merge: 5990a77 6814270
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 3 02:20:45 2023 +0200

    Merge pull request #10 from okolekar/okolekar-patch-6

    Fixed_Boundary_Conditions.py

commit 5990a7772a2ed51a1ddb02f76a672f4d105b30ab
Merge: 2e69675 4208aff
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Mon Jul 3 02:20:18 2023 +0200

    Merge pull request #11 from okolekar/okolekar-patch-7

    Update Initial_Condition.py

commit 4208aff09d30babba810a436b1a85c61dfc17c85
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Jul 1 01:55:33 2023 +0200

    Update Initial_Condition.py

    Added the code that find the initial condition for the enthalpy
       approach.

commit 68142706747f582f4f49e68eccd9b7bac2f02699
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Jul 1 01:44:56 2023 +0200

    Fixed_Boundary_Conditions.py

    fixed the boundary conditions and hence the Non invertible dg Matrix
```

```
commit 2e69675c79a282553222f410a0045d011042a74d
Merge: 98c34d0 971f6e4
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jun 29 23:45:43 2023 +0200

    Merge pull request #8 from okolekar/okolekar-patch-5

    Added the element wise implementation of jacobi

commit 98c34d0e11474628805f664904f9ab1c9a39d1ed
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jun 29 02:49:09 2023 +0200

    Initial_condition_finder

    Python code to find the initial condition. Not yet complete

commit 9ea14ed63425ef84be40ae0e928446b402ac3cca
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jun 29 02:47:13 2023 +0200

    Reference calculation of phase interface position

    In this upload I have added the reference file to calculate the
        position of the solid liquid interface. Its an excel file.

commit 971f6e47413ee526bc6717286d0d71e40a85f4e5
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jun 29 00:02:38 2023 +0200

    Added the element wise implementation of jacobi

commit 5a11cbd0f0556f71229ed96ef9ee8385a7b7b03f
Merge: 37faeef 87bac55
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jun 25 18:39:49 2023 +0200

    Merge pull request #7 from okolekar/okolekar-patch-4

    Timestep Update EnthalpyApproach.py

commit 87bac55490603b523887a46ac28a35db597c4b7b
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jun 25 18:39:15 2023 +0200

    Timestep Update EnthalpyApproach.py

    Added the time step.
    Also, the dG matrix is now visible.  Made it to observe the dG matrix
         as it is not invertable.

commit 37faeef9074b6319ec814796f5b776157fefde2d
```

```
Merge: 920e4c5 7116897
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jun 25 00:54:42 2023 +0200

    Merge pull request #6 from okolekar/okolekar-patch-3

    Update README.md

commit 71168974459a24c66b4acfbc9c7d9161938bc221
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jun 25 00:54:27 2023 +0200

    Update README.md

commit 920e4c57d1d825d6158a842b21ec9804015320e6
Merge: 47bb2f9 1b2c65d
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jun 25 00:53:58 2023 +0200

    Merge pull request #5 from okolekar/okolekar-patch-2

    Create README.md

commit 1b2c65da1555fb953638708755726937623fa485
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jun 25 00:52:41 2023 +0200

    Create README.md

commit 47bb2f9654d1a7682d9614bb43b969490cdd5878
Merge: f383f15 7650b0f
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jun 25 00:43:31 2023 +0200

    Merge pull request #1 from okolekar/okolekar-patch-1

    Update Boundary condition  EnthalpyApproach.py

commit 7650b0f37f877316c68e10a340198e3682e3a8ae
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun Jun 25 00:42:41 2023 +0200

    Update Boundary condition  EnthalpyApproach.py

    1)Removed some comments as the program works for the changes made.
    2)Made the boundary conditions
    3)Initial Condition not yet added
    4)dG matrix is singular completely singular

commit f383f1588df8c4c1e34479bd6a313004a71db100
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sat Jun 24 20:40:43 2023 +0200
```

```
    Update EnthalpyApproach.py

    Activated the Boundary Conditions

commit 16cd590e53a175268f7857cf8f451ba23fc57b0e
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Fri Jun 23 18:19:30 2023 +0200

    Update EnthalpyApproach.py

    1) Corrected some syntactical  errors. Made the changes to the
        functions and merged them into a material model function.
    2) Created a list of the mesh. But is needed to be updated.
    3) Decision pending about the dG( ) as Te is a vector and decision is
         to be made whether to take into account each and every data
        points or the average of the vector.
    4) dG IS A SINGULAR MATRIX!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    5) Initial Conditions are not yet entered!!!!!!!!!!!!!!!

commit 65d89488442998b936888a06f8fa914d6ac4e3b2
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jun 22 19:44:25 2023 +0200

    Update EnthalpyApproach.py

    Still Euler scheme is not implemented. Also the problem with the
        condition of the NRS scheme.
    Also the problem dg cause I have implemented the Tmelting and its a
        problem cause it has the dimentions

commit 6b242c3737b31dfacf99bd9404f26e606a210e67
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Thu Jun 22 19:39:31 2023 +0200

    Update EnthalpyApproach.py

commit 15a1c09dc9b58a8f1ede289bc14e3879f7f1c416
Author: Omkar Kolekar <okolekar123@gmail.com>
Date:   Sun May 14 22:30:00 2023 +0200

    Add files via upload
```