



Norwegian University of  
Science and Technology

# Numerical Simulation of Solidifying Flow

Numerisk simulering av størking

**Olav Rømcke**

Master of Science in Mechanical Engineering

Submission date: June 2018

Supervisor: Reidar Kristoffersen, EPT

Norwegian University of Science and Technology  
Department of Energy and Process Engineering



EPT-M-2018-74

**MASTER THESIS**

for

Student Olav Rømcke

Spring 2018

Numerical Simulation of Solidifying Flow

*Numerisk simulering av storking***Background and objective**

The biggest challenges of low-pressure casting are good molding without turbulence, temperature control and air ventilation from the casting mold cavities. The aim is to get a good metallic bond between a solid body and liquid aluminum that flows past and solidify on the body. The flow field around the solid body has been shown to be of great importance to the degree of strength in the metallic bond.

The objective is through collaboration between *Department of Materials Science and Engineering* and *Department of Energy and Process Engineering* to obtain an in-house CFD-code based on the SOLA algorithm that is capable of simulating solidifying flows.

**The following tasks are to be considered:**

1. Literature study of Solid/Fluid interaction.
2. Implementing a Conjugate Gradient solver for the Poisson equation for the pressure.
3. Investigate the possibilities for simplifying the numerical modelling of the solidifying process.
4. Implementing a solidifying model in the SOLA algorithm.
5. Set up and perform a series of test case simulations.
6. Perform a simulation of a 3D solidifying case to validation/verification the program.

-- ” --

Within 14 days of receiving the written text on the master thesis, the candidate shall submit a research plan for his project to the department.

When the thesis is evaluated, emphasis is put on processing of the results, and that they are presented in tabular and/or graphic form in a clear manner, and that they are analyzed carefully.

The thesis should be formulated as a research report with summary both in English and Norwegian, conclusion, literature references, table of contents etc. During the preparation of the text, the candidate should make an effort to produce a well-structured and easily readable report. In order to ease the evaluation of the thesis, it is important that the cross-references are correct. In the making of the report, strong emphasis should be placed on both a thorough discussion of the results and an orderly presentation.

The candidate is requested to initiate and keep close contact with his/her academic supervisor(s) throughout the working period. The candidate must follow the rules and regulations of NTNU as well as passive directions given by the Department of Energy and Process Engineering.

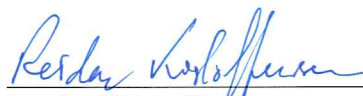
Risk assessment of the candidate's work shall be carried out according to the department's procedures. The risk assessment must be documented and included as part of the final report. Events related to the candidate's work adversely affecting the health, safety or security, must be documented and included as part of the final report. If the documentation on risk assessment represents a large number of pages, the full version is to be submitted electronically to the supervisor and an excerpt is included in the report.

Pursuant to "Regulations concerning the supplementary provisions to the technology study program/Master of Science" at NTNU §20, the Department reserves the permission to utilize all the results and data for teaching and research purposes as well as in future publications.

The final report is to be submitted digitally in DAIM. An executive summary of the thesis including title, student's name, supervisor's name, year, department name, and NTNU's logo and name, shall be submitted to the department as a separate pdf file. Based on an agreement with the supervisor, the final report and other material and documents may be given to the supervisor in digital format.

- Work to be done in lab (Water power lab, Fluids engineering lab, Thermal engineering lab)  
 Field work

Department of Energy and Process Engineering, 15. January 2018



Reidar Kristoffersen  
Academic Supervisor

Research Advisor:

# 1 Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science. The research has been conducted in the spring of 2018 at NTNU for the Department of Energy and Process Engineering.

The formulation of the thesis was inspired by a set of meetings with Chassis Norway AS (former Bentler Automotive Farsund), a Norwegian foundry who manufactures aluminum parts for the automotive industry. The physical process of melting and solidification peaked an interest and remains the main theme throughout this thesis. The problem formulation was developed by the author and supervisor Reidar Kristoffersen.

The work condensed into developing a new numerical approach for solving solid-liquid phase transition problems. Rather than delivering the thesis on a report format, an article based thesis was suggested. A short introduction and documentation on the work done this semester is supplied, but the main body of the thesis is two articles. A preprint of the first article has been submitted to Applied Mathematical Modelling, and the second article will be submitted for publication at a later date.

## 2 Abstract

A 3D model for solving convection dominated solid-liquid phase transitions has been developed. The model for solving the fluid fields utilizes the projection method, which is also the basis for the iterative strategy for solving the phase transition. Throughout this report a conjugate gradient method has been used for solving the various sparse matrix equations encountered.

Comparisons are made to a square cavity melting experiment, which is a common example from the literature demonstrating the influence convective currents have on the evolution of the solid-liquid interphase.

Development of the code has its basis in an educational SOLA solver by Reidar Kristoffersen and the previous master thesis by Arne Bøckmann utilizing the same fluid solver. The code has been manipulated into handling the changing fluid domain, as the material changes phase.

The main body of the thesis is two articles followed by a section presenting, in detail, the numerical discretization, boundary conditions, a preliminary verification of the fluid model and extension to phase transition of mixtures.

# Contents

<b>1</b>	<b>Preface</b>	<b>3</b>
<b>2</b>	<b>Abstract</b>	<b>4</b>
<b>3</b>	<b>Article 1: A Projection Method for Convection Dominated Phase Transitions</b>	<b>6</b>
<b>4</b>	<b>Article 2: A 3D Study of Convection Dominated Phase Transitions Using the Projection Method</b>	<b>32</b>
<b>5</b>	<b>Trails and Tribulations</b>	<b>51</b>
5.1	Notes on the Buckingham-II Theorem . . . . .	51
5.2	Discretization and Boundaries . . . . .	52
5.2.1	FTCS - Momentum Equations . . . . .	52
5.2.2	Pressure Correction . . . . .	53
5.2.3	BTCS - Energy Equation . . . . .	54
5.2.4	Solid Fraction Correction . . . . .	54
5.2.5	Boundary Conditions . . . . .	55
5.3	Conjugated Gradient and Storage Format . . . . .	57
5.4	Pressure Driven Channel . . . . .	58
5.5	Phase Transition of Mixtures . . . . .	60
<b>6</b>	<b>Concluding Remarks</b>	<b>65</b>
<b>A</b>	<b>Code Snippets</b>	<b>67</b>
A.1	Preliminary Velocities . . . . .	67
A.2	Velocity Boundary Conditions . . . . .	69
A.3	Solve the Fluid Equations . . . . .	69
A.4	Generate the Pressure Correction Matrix System . . . . .	70
A.5	Solve the Thermal Equations . . . . .	73
A.6	Generate the Temperature Matrix System . . . . .	73
A.7	Solid Fraction Correction . . . . .	75
A.8	Thermal Boundary Conditions . . . . .	76
A.9	Conjugated Gradient . . . . .	76
A.10	Solid Fraction Temperature Relations . . . . .	78

### **3 Article 1**

#### **A Projection Method for Convection Dominated Phase Transitions**



# A Projection Method for Convection Dominated Phase Transitions

Olav Rømcke<sup>\*a,b</sup>, Reidar Kristoffersen<sup>a,c</sup>

<sup>a</sup>*NTNU, Department of Energy- and Process Engineering, NO-7491 Trondheim, Norway*

<sup>b</sup>*olavromc@stud.ntnu.no*

<sup>c</sup>*reidar.kristoffersen@ntnu.no*

---

## Abstract

A simple model for solving convection/diffusion phase transition problems will be described. Pure substances are the focus of this paper, but extension to more complex temperature dependent phase transitioning behavior is also addressed. This method utilizes a non-deforming, staggered, cartesian grid. A Boussinesq approximation is the driving force for the natural convection, while the Poisson pressure-correction equation is solved with a conjugate gradient method. A simple marking method for liquid and solid cells are updated at every time step, such that the pressure-correction only needs to be solved in the domain where the substance is liquid. Solving the thermal fields (i.e. solid fraction and temperature) utilizes a projection method derived in this paper. A description of the Darcy source term for handling fluid flow in the solid region, as well as a Source-based method for solving the thermal fields are also presented, and compared to the method derived in this paper. Model validation is done by comparison with experimental results of a 2D cavity convection/diffusion case with gallium.

*Keywords:* Phase Transition, Projection, Convection, 2D, Gallium, Numerical Simulation

---

## 1. Introduction

Phase change occur in many natural and industrial processes. Melting and formation of ice, boiling and condensation of water, welding and casting of metals, drying and freezing food for conservation are but a few examples of a wide range of situations where a phase transition occur. The scope of this

## Nomenclature

$*$	Non-Dimensional	$H$	Length Scale
$\alpha$	Thermal Diffusivity	$h, k$	# Steps in Spacial Direction
$\beta$	Thermal Expansion	$i, j$	Spacial Direction
$\epsilon$	Convergence Criterion	$K$	Thermal Conductivity Ratio
$\hat{\phantom{x}}$	Preliminary	$k$	Thermal Conductivity
$\lambda$	Porosity	$L$	Laten Heat
$\mu$	Viscosity, Kinetic	$l$	Liquid, Subscript
$\nu$	Viscosity, Kinematic	$M$	Generic Matrix
$\omega$	Relaxation Factor	$m$	# Iterations
$\rho$	Density	$M_G$	Global Mass Conservation
$\theta$	Temperature, Non-Dim	$n$	# Discrete Time Steps
$(\vec{\phantom{x}})$	Vector	$nb$	Neighbour Node
$\xi()$	Correction Function	$P$	Pressure, Non-Dim
$A$	Porosity Function	$p$	Pressure
$a$	Discretization Parameter	$PC$	Press-Corr, Subscript
$A_r$	Aspect Ratio	$Pr$	Prantl Number
$B$	Generic Vector	$r$	Residual Vector
$b$	Darcy Porosity Constant	$Ra$	Rayleigh Number
$C$	Cold, Subscript	$ref$	Reference
$C$	Darcy Porosity Constant	$res$	Residual Norm
$c$	Center Node	$s$	Solid, Subscript
$C_P$	Volumetric Heat Capacity Ratio	$Ste$	Stefan Number
$c_p$	Heat Capacity	$T$	Energy, Subscript
$F()$	Function	$T$	Temperature
$F_B$	Force, Buoyancy	$t$	Time
$F_D$	Force, Darcy Porosity	$TT$	Thermal, Subscript
$F_G$	Global Solid Fraction	$U$	Velocity, Non-Dim
$f_S$	Solid Fraction	$u$	Velocity
$F_O$	Time, Non-Dim	$W$	Cavity Width
$g$	Gravitational acceleration	$X$	Position, Non-Dim
$H$	Hot, Subscript	$x$	Position
		$z$	Unlimited Solid Fraction

paper is the temperature driven phase change occurring when a substance crosses the solid-liquid fusion temperature. Experiments have shown that the shape and position of the solid-liquid interphase depend on the amount of convection the material is able to induce [1] [2], hence the diffusive energy transport alone is not enough to explain the behavior of the substance. Later experiments and simulations by Ben-David et.al. [3] also points out the importance of 3D flow structures.

The moving boundary, phase transitioning problems governed by the heat equation is known as Stephan problems, and relatively few analytical solutions are known [4]. When introducing a velocity field as well, a numerical approach is needed. A review article by Voller et.al. [5], and later by König-Haagen et.al. [6] recaps most work on fixed-grid methods for phase transition problems. In general, numerical fixed grid solution methods of these kind of problems can be categorized into Apparent Heat Capacity method, Enthalpy method and Source method.

The idea behind the Apparent Heat Capacity method is to incorporate the latent heat of fusion into the heat capacity. The released/absorbed heat during phase change is accounted for in the heat capacity change and no source term is needed. This method renders the source term obsolete and deals with a temperature dependent heat capacity. Implementation with existing code could be possible with some knowledge of the solid fraction-temperature relationship. However, if the temperature range where material solidifies is small (i.e. pure substances, where solid fraction and temperature has a shock-like relationship), this method risk not accounting for the latent heat added or removed during phase change.

Enthalpy methods substitutes temperature with enthalpy in the governing heat equations. For a good choice of substitution the issues with a discontinuity in the solid fraction-temperature relationship are bypassed [6] [7] [8] [9].

For Source based methods the focus is to handle the source term occurring in the energy equation due to the phase transition [10]. As a melting material will absorb energy, and a solidifying material will release energy there is some correct balance between change in solid fraction and change in temperature that can be found by an iterative scheme at every time step. In general the temperature field and the solid fraction is calculated and corrected iteratively. This procedure is repeated until a prescribed convergence criterion is satisfied.

As experiments have shown, the convective energy transport plays a signifi-

cant role, thus the velocity field needs to be resolved such that this transport is correctly accounted for. Key aspects of solving the velocity lies in knowing the driving force of the fluid movement, how to simplify the Navier-Stokes equations and how to handle velocities in solidified regions of the domain.

For a thermal cavity, as utilized in [3] [6] [8] [11] [12] [13], natural convection is driven by the small density changes induced by temperature differences. The Boussinesq approximation for coupled laminar flow and heat transfer have shown to yield satisfactory results.

There are several techniques for solving the incompressible Navier-Stokes equations, with FVM methods such as PISO and the family of SIMPLE-algorithms to mention a few. The solution strategy chosen here is a FDM Projection method similar to the one proposed by Chorin et.al. [14]. There are a couple of methods utilized in the literature to ensure zero, or close-to-zero velocity in the solidified regions of the calculation domain. One simple approach utilized by Gartling [15] is to increase the viscosity such that any calculated velocity in the solid region has a negligible influence. Morgan [13] simply reduces the velocity to zero in the solid regions. However, a widely adopted approach is the Darcy porosity source term resisting the flow in the solid region [8] [11] [12]. The combination of enthalpy and porosity approach is supported in both Fluent [16] and Comsol [3].

This paper presents an alternative phase transition solution strategy based on the idea behind the projection method from [14]. The PhD thesis by N.R. Morgan [17] presents a similar solution strategy for liquid-vapor film boiling with a level-set method to separate the liquid and vapor phases. A source based approach is also derived and compared to the experimental results. A live-dead cell approach is utilized such that the known zero velocity in the solid region is not calculated. Comparisons to the Darcy-porosity approach is also presented. The focus areas of this paper are prediction accuracy, computational efficiency and simplicity.

## 2. Model

For the mathematical formulation of convection and diffusion driven phase transitioning model, the following assumptions are made:

1. The physics can be described in 2D.
2. Heat transfer is governed by convection and conduction.
3. The fluid flow is laminar and incompressible.

4. Viscous dissipation is negligible.
5. Density changes only effect the free convection. Change in density are only considered for the Boussinesq approximation.
6. Physical properties ( $c_p$ ,  $\mu$ ,  $k$  ...) only differ between solid and liquid phases. They do not vary with temperature.
7. A phase field function is a sufficient representation of the average local phase fraction.
8. The phase transition is isothermal.
9. Solidified material is in full contact with the boundary walls

### 2.1. Mathematical formulation

Based on the assumptions above, the governing fluid equations state

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (1)$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_j u_i)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{1}{\rho} \frac{\partial}{\partial x_j} \left( \mu \frac{\partial u_i}{\partial x_j} \right) + \frac{1}{\rho} F_i \quad (2)$$

where (1) is the incompressible mass continuity constraint, while (2) represents the momentum equations.  $\rho$  and  $\mu$  is density and kinetic viscosity respectively.  $F_i$  is the sum of volumetric forces acting on the fluid. The Boussinesq approximation is one component of  $F_i$ :

$$F_B = -\rho g_i \beta (T - T_{ref}) \quad (3)$$

A common method to accounting for the flow in the solid region is to estimate the material as a porous medium governed by Darcy's law. According to [11] this is implemented by adding a volumetric force to  $F_i$ :

$$F_D = -C \frac{(1 - \lambda)^2}{\lambda^3} u_i = A u_i \quad (4)$$

where  $\lambda$  is the local porosity and  $C$  is a geometric constant. The focus of this paper is a live-dead fluid cell methodology, but for comparison purposes the Darcy term is included.

Accounting for conduction and convection, the energy equation states:

$$\frac{\partial T}{\partial t} + \frac{\partial (u_j T)}{\partial x_j} = \frac{1}{\rho c_p} \frac{\partial}{\partial x_j} \left( k \frac{\partial T}{\partial x_j} \right) + \frac{L}{\rho c_p} \frac{\partial f_S}{\partial t} \quad (5)$$

where  $c_p$  and  $k$  represents heat capacity and thermal conductivity, while  $L$  and  $f_S$  represents volumetric latent heat and local fraction of solid respectively. The latent heat ( $L$ ) enters the equation as a volumetric energy source proportional to the temporal change in solid fraction. Depending on the nature of the material, the relationship between solid fraction and temperature can have different relationships. Pure substances have a distinct temperature where the phase transition occurs. Wax, polymers and glass are examples of materials that typically solidifies continuously over a temperature range. The solidification of metal alloys have a phase change region where some components of the mixture solidifies at higher temperature than others. The solidifying region can consist of complicated crystalline micro structures. As mentioned, this paper use a phase field function ( $f_S$ ) to represent the local fraction of solid. This is further generalized to have some specific relation to temperature:

$$f_S = F(T) \quad (6)$$

Different forms of the function  $F(T)$  is indicated in figure 1. Here the functions are simplified, but other well known relationships are the Scheil equation and the Lever Rule equation [18]

By utilizing the scaling proposed below,

$$\begin{array}{lll} U_i = \frac{u_i H}{\alpha_l} & \theta = \frac{T - T_{ref}}{\Delta T} & P = \frac{p H^2}{\rho_l \alpha_l^2} \\ X_i = \frac{x_i}{H} & A^* = \frac{A H^2}{\rho_l \alpha_l} & \alpha_l = \frac{k_l}{c_{pl} \rho_l} \\ Fo = \frac{t \alpha_l}{H^2} & \nu = \frac{\mu}{\rho_l} & \Delta T = T_H - T_{ref} \\ Pr = \frac{\nu}{\alpha_l} & Ra = \frac{g \beta \Delta T H^3}{\nu \alpha_l} & Ste = \frac{\rho_l c_{pl} \Delta T}{L} \\ A_r = \frac{W}{H} & K = \frac{k}{k_l} & C_P = \frac{c_p \rho}{c_{pl} \rho_l} \end{array}$$

the non dimensional form of the governing equations (1), (2), (5) and (6) are:

$$\frac{\partial U_i}{\partial X_i} = 0 \quad (7)$$

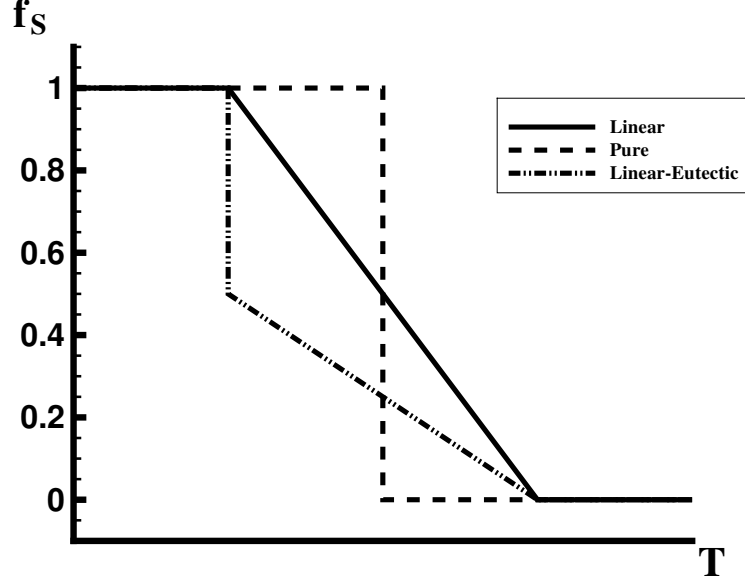


Figure 1: Different examples of solid fraction ( $f_s$ ) and temperature ( $T$ ) relation.

$$\frac{\partial U_i}{\partial Fo} + \frac{\partial (U_j U_i)}{\partial X_j} = -\frac{\partial P}{\partial X_i} + Pr \frac{\partial^2 U_i}{\partial X_j^2} + Ra Pr \theta \frac{\partial X_i}{\partial X_2} + A^* U_i \quad (8)$$

$$\frac{\partial \theta}{\partial Fo} + \frac{\partial (U_j \theta)}{\partial X_j} = \frac{\partial}{\partial X_j} \left( K \frac{\partial \theta}{\partial X_j} \right) + \frac{1}{Ste} \frac{\partial f_s}{\partial Fo} \quad (9)$$

$$f_s = F(\theta) \quad (10)$$

Note that for simplicity the dimensionless volumetric heat capacity ( $C_P$ ) is not a part of the non dimensional equations, as  $\frac{\rho_s c_{ps}}{\rho_l c_{pl}} \approx 1$  for gallium. However,  $\frac{k_s}{k_l} \approx 1.46$ , thus the dimensionless thermal conductivity ( $K$ ) will have a significant impact on the solution. The convention used from now on is the non-dimensional form of the governing equations (7) to (10). Three dimensionless numbers appears ( $Pr$ ,  $Ra$ ,  $Ste$ ), and a short description is provided.

The Prantl number ( $Pr$ ) represents the proportion between viscous and thermal diffusion rate. Liquid metals typically have a low  $Pr$ ; good thermal conductors and not particularly viscous. Oils typically have high  $Pr$ .

The Rayleigh number ( $Ra$ ) relates to buoyancy flows. This can be seen in the non-dimensional momentum equation (8) as it appears in the Boussinesq term.  $Ra$  indicates whether heat transfer in the fluid is convection, or conduction dominated. Above some critical  $Ra$ , depending on the problem, the fluid heat transfer is controlled by convection. Below that value thermal conduction generally moves faster than the convective transport. The effect of change in  $Ra$  for a cavity melting problem is indicated by Morgan [13]. Values of  $Ra$  are typically large for engineering purposes.

The Stefan number ( $Ste$ ) is the ratio of sensible heat to latent heat. This number relates to the change of phase. From (9) one can see that it controls the influence of the change in solid fraction. For large  $Ste$  the heat released (absorbed) during solidification (melting) is small relative to the thermal capacity of the fluid. Thus, the change of solid fraction have a small impact on the energy equation. For small  $Ste$ , the heat released/absorbed during a phase change is relatively large in respect to the capacity of the material, and the phase change have a large impact on the energy equation. In effect, a large  $Ste$  would yield a quick phase transition, while a small  $Ste$  yields slow phase transition.

## 2.2. Numerical Solution

As indicated by the mathematical formulation of a convection/conduction phase transitioning problem, the rate of change in temperature effects the rate of change in solid fraction, and visa versa. There are several techniques

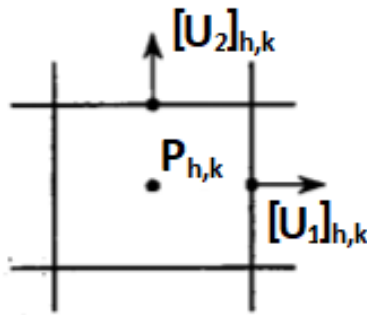


Figure 2: Cell position and numbering for a staggered grid.



developed for handling this coupled system of the thermal fields ( $f_S$  and  $\theta$ ). For comparison purposes the Source Method, together with the model developed for this paper, are presented here. The momentum fields ( $P$  and  $U$ ) are handled numerically by the projection method [14]. The mesh is a fixed cartesian staggered grid, where the velocity nodes are placed on the cell sides, while intensive properties (i.e. temperature, pressure, solid fraction, viscosity, heat capacity ect.) are placed in the cell centers, see figure 2.

### 2.2.1. Source Method

This method views the energy equation as a balance between the heat used to change the local temperature and heat used to change the local solid fraction. The energy equation stays as given in (9), but the interaction between energy transport and the latent heat source at every time step will be handled by an iterative scheme. The idea behind the solution algorithm presented here can be found in the article by Voller et.al. [5]. Since the variables and parameters defined in this paper differ from those defined in the literature, the method is derived here as well. The discretized energy equation (9) is a good starting point for Source based methods:

$$\begin{aligned} a_c[\theta_c]_m^n + \sum a_{nb}\theta_{nb} = \\ a_1[\theta_c]^{n-1} + a_0([f_S]_m - [f_S]^{n-1}) \end{aligned} \quad (11)$$

Superscript  $n$  represents time step, while subscript  $m$  represents iteration. Which time- and iteration step values used for the neighbouring nodes depends on the discrete scheme utilized. Knowing that upon convergence (11) should satisfy:

$$\begin{aligned} a_c F^{-1}([f_S]^n) + \sum a_{nb}\theta_{nb} = \\ a_1[\theta_c]^{n-1} + a_0([f_S]^n + \Delta f_S - [f_S]^{n-1}) \end{aligned} \quad (12)$$

where  $\Delta f_S$  is the correction in solid fraction needed to reach convergence. By subtracting (11) from (12) the solid fraction correction can be estimated as:

$$\Delta f_S \approx \frac{a_c}{a_0} (F^{-1}([f_S]^n) - [\theta_c]_m) \quad (13)$$

The solid fraction is simply updated by

$$[f_S]_m = [f_S]_{m-1} + \omega \Delta f_S \quad (14)$$

where  $\omega$  typically represents an underrelaxation constant. Voller et.al. [5] recommends  $\omega$  between 0.5 and 0.7. A physical limitation for  $f_S$  that the solid fraction is limited between 0 and 1.

$$\begin{aligned} f_S &= 0, \text{ if } f_S < 0 \\ f_S &= 1, \text{ if } f_S > 1 \end{aligned} \quad (15)$$

or

$$f_S = \min(1, \max(0, f_S)) \quad (16)$$

For the source method, this limiting of the solid fraction is applied at every iteration step. The iterative solution procedure for a single time step is as follows:

1. Initiate the first solid fraction field  $[f_S]_1$  by setting it equal to the solid fraction from the previous time step  $[f_S]^{n-1}$ .
2. Calculate the updated temperature field according to (11).
3. Calculate a correction to the solid fraction according to (13).
4. Update the solid fraction according to (14).
5. Limit the solid fraction according to (16).
6. Check for convergence. If not, repeat step 2 - 6.
7. If convergence is reached, replace the old thermal fields ( $[\theta]^{n-1}, [f_S]^{n-1}$ ) with the calculated ones ( $[\theta]_m, [f_S]_m$ ) and proceed to the next time step.

Here, (11) is discretized according to the BTCS scheme. This yields a system on the form  $M\vec{\theta} = \vec{B}$ , where  $M$  is a diagonal matrix and  $\vec{B}$  represents the forcing term from the discretized system.  $\vec{\theta}$  represents the temperature nodes for the whole domain ordered in a vector. This linear system is solved with a conjugate gradient method.

### 2.2.2. A Projection Method for Phase Transitions

As for the Source method, the Projection method views the energy equation as a balance between the heat used to change the local temperature and heat used to change the local solid fraction. The algorithm is derived accordingly: A temperature field  $\hat{\theta}$  is estimated by discretizing (9), ignoring the phase transition term:

$$\frac{\hat{\theta}_c - [\theta_c]^{n-1}}{\Delta Fo} = RHS \quad (17)$$

where

$$RHS = \frac{\partial}{\partial X_j} \left( K \frac{\partial \theta}{\partial X_j} \right) - \frac{\partial (U_j \theta)}{\partial X_j} \quad (18)$$

This initial projection of the temperature field is obviously wrong. However, with the right adjustment of the solid fraction, the final energy balance should satisfy:

$$\frac{[\theta_c]^n - [\theta_c]^{n-1}}{\Delta Fo} - \frac{1}{Ste} \frac{[f_S]^n - [f_S]^{n-1}}{\Delta Fo} = RHS \quad (19)$$

Subtracting (17) from (19) yields:

$$[\theta_c]^n - \hat{\theta}_c = \frac{1}{Ste} ([f_S]^n - [f_S]^{n-1}) \quad (20)$$

Substituting in (10) and rearranging (20) gives:

$$0 = [f_S]^n - [f_S]^{n-1} - Ste \left( F^{-1}([f_S]^n) - \hat{\theta}_c \right) \quad (21)$$

$$\xi(z) = z - [f_S]^{n-1} - Ste \left( F^{-1}(z) - \hat{\theta}_c \right) \quad (22)$$

Thus, a simple, pointwise, 1D iterative scheme can be formulated as finding the roots of the function  $\xi(z)$ , where  $\xi$  is defined as the function in (22) where  $z = [f_S]^n$ . Note that as for the Source method, the solution to  $\xi(z) = 0$  could yield unphysical values for  $f_S$ . The limiter defined in (16) is applied after the solution to (22) has converged. The solid fraction is then given directly. The temperature field is corrected according to (20). To summarize:

1. Estimate a temperature field ( $\hat{\theta}$ ) from (17).
2. Correct the solid fraction ( $f_S$ ) by finding the root of  $\xi$  from (22).
3. Limit the solid fraction to physical values according to (15)
4. Correct the temperature field according to (20)

Here, a BTCS-scheme is used for (17) and (18) and solved by a conjugate gradient method. Finding the root of  $\xi$  in (22) is done by the Secant Method. The Source Method iteratively corrects both temperature and solid fraction as a coupled system. The Projection method proposed here, however, decouples the system and corrects the temperature field after the solid fraction for the new time step has been found. A more graphical explanation of the Projection method derived here can be found in Appendix A.

### 2.2.3. Fluid Model

Consider the Navier-Stokes equations (7) and (8). The first describes the conservation of mass and the latter describes the momentum equation. However, the coupling between pressure and velocity is not clear. A projection method on a staggered cartesian grid is utilized in this paper. A more thorough derivation of a projection algorithm has been done in section 2.2.2. In the literature, Chorin [14] presented the projection algorithm for Navier-Stokes, while Harlow and Welch [19] presented the general MAC approach. A summary of the method for Navier-Stokes is still done here. The idea is to estimate the velocity and then correct both pressure and velocity, such that mass is conserved. For every time step the method can be summarized as follows:

1. Calculate a preliminary velocity field ( $\hat{U}$ ) from (23) based on the previous pressure field. This velocity field does not necessarily fulfill continuity (7).
2. A version of (8) based on the current pressure field should however fulfill continuity. The difference between these two equations will yield the pressure change necessary to uphold the continuity constraint.
3. By further taking the divergence of the difference between the two discrete velocity equations, (7) can be substituted in. This yields a Poisson equation for the pressure correction (24).
4. The discretized Poisson equation for the pressure correction gives a linear system of equations.
5. Velocity and pressure fields are corrected based on the calculated pressure correction (26) (25).

Equations for the N-S projection method are as follows:

$$\frac{\hat{U}_i - [U_i]^{n-1}}{\Delta Fo} = -\frac{\partial(U_j U_i)}{\partial X_j} - \left[ \frac{\partial P}{\partial X_i} \right]^{n-1} + Pr \frac{\partial^2 U_i}{\partial X_j^2} + RaPr\theta \frac{\partial X_i}{\partial X_2} + A^* U_i \quad (23)$$

$$\frac{\partial^2(\Delta P)}{\partial X_i^2} = \frac{1}{\Delta Fo} \frac{\partial \hat{U}_i}{\partial X_i} \quad (24)$$

$$[P]^n = [P]^{n-1} + \Delta P \quad (25)$$

$$[U_i]^n = \hat{U}_i - \Delta F o \frac{\partial(\Delta P)}{\partial X_i} \quad (26)$$

Here, (23) discretized by a FTCS-scheme. Several techniques for solving the diagonal matrix system from the Poisson pressure correction equation can be found in the literature, but a conjugate gradient method is utilized here [20].

#### 2.2.4. Handling Flow in the Solid Region

Ideally, the velocity in the solid region should be zero. As mentioned, three distinct methods are proposed by the literature. A simple but brute force way is to increase the viscosity in the solidified material, thus being a function of the solid fraction [15]. Initial tests with this approach showed that the difference in viscosity needed to be quite large, such that the velocity field could be neglected in the solid region. Additionally, the viscos increase also yielded a unfavorable stability criterion for the time step,  $\Delta F o$ . When viscosity increase, the travel speed of shear force information also increases. From a simulation perspective, the information cannot travel further than the width of one cell in one time step, thus the time step must decrease such that this is accounted for. This effect have a significant impact on the total simulation time and the method was abandoned.

Viewing solidifying cells as a porous medium is another method for handling flow in the solid region. As mentioned in the mathematical formulation, accounting for the resistance in the porosity, a force term  $A^*U_i$  is added to the momentum equation as indicated in (8).  $A^*$  is the dimensionless Darcy porosity function. The definition of porosity is equal to the liquid fraction, and a simple substitution makes the porosity function a function of the solid fraction:

$$A^* = -C^* \frac{(f_S)^2}{(1 - f_S)^3} \quad f_S = 1 - \lambda \quad (27)$$

Note that (27) will not be consistent as the solid fractions approaches unity. For simulation purposes a small constant,  $b$ , is added to the denominator.

$$A^* = -C^* \frac{(f_S)^2}{(1 - f_S)^3 + b} \quad (28)$$

The constant  $b$  needs to be significantly small such that velocity approaches zero in the solid region, but large enough to avoid resolution issues in the

computer. Voller and Prakash [11] defined  $b = 0.001$  and  $C = 1.6e3$ , which is also used here.

Both the viscosity method and the porosity method solves the Navier-Stokes equations in a region where the velocity is known to be zero. For a region with a lot of solidified cells, this approach is unnecessarily inefficient. A live/dead-cell method was implemented to account for this. The method can be summarized accordingly:

1. If a cell has half of its volume solidified the cell is marked as a dead fluid cell.
2. If a velocity node has both of its neighbouring cells marked as a dead fluid cell, then the velocities are known to be zero. This is in effect a no-slip boundary condition. The Neighbouring cells are defined as the east and west cell for the first spacial direction, and north and south cell for the second spacial direction.
3. If a pressure node has a neighbouring node marked as a dead fluid cell, then the spacial derivative in that direction is known to be zero. This is in effect a zero gradient boundary condition.

From an implementation aspect a new pressure-correction matrix needs to be established at every time step, accounting for new live or dead fluid cells. A high resolution is recommended, as the velocity and pressure boundary is interpreted as a "staircase" shaped interphase.

### 2.2.5. Model Verification Parameters

Model verification will be discussed in detail in section 3. However, a distinction is made between the different verification parameters in this section. During the solution procedure, several Matrix-Vector equations on the form  $M\vec{x} = \vec{B}$  are solved. The iterative CG-method has been applied throughout this paper. The  $L^1$  norm of the residual vector is used as a measurement for convergence:

$$res = ||\vec{r}|| = ||\vec{B} - M\vec{x}|| \quad (29)$$

Both the pressure-correction (24) and the energy equation from (17) and (11) yields such linear systems solved by the CG-method. In addition, the full thermal loop at every time step finds the correct balance of temperature change and solid fraction change by iteration. The resulting residuals from the thermal loop is defined as (30) for the Source method and as (31) for the Projection method. The  $L^1$  norm is also utilized here.

$$res = ||[\theta]_m - \theta_{m-1}|| \quad (30)$$

$$res = ||[z]_m - [z]_{m-1}|| \quad (31)$$

The different residuals are separated by a distinct subscript listed in table 1

Label	Procedure	Eq.	Method
$res_{PC}$	Pressure-Correction	(29)	CG
$res_T$	Energy Equation (BTCS)	(29)	CG
$res_{TT}$	Thermal	(30)/(31)	Source/Proj.

Table 1: Residual labels.

Global measurements are also utilized such as global mass conservation ( $M_G$ ) and global solid fraction ( $F_G$ ). Independent of how well the residuals perform under iteration, these global parameters should also yield satisfactory results.

$$M_G = \sum_1^n \sum_{h=1}^{h_{max}} \sum_{k=1}^{k_{max}} \frac{\Delta F_o}{HW} [\Delta U_1 \Delta X_2 + \Delta U_2 \Delta X_1]_{h,k} \quad (32)$$

$$F_G = \sum_{h=1}^{h_{max}} \sum_{k=1}^{k_{max}} \frac{1}{HW} [f_S \Delta X_1 \Delta X_2]_{h,k} \quad (33)$$

Mass conservation (32) should ideally equal zero, while global solid fraction (33) should remain unchanged under grid- and time step refinement.

### 3. Results and Discussion

A 2D cavity melting case with gallium has been used as a validation test. The domain is initially in solid state at a temperature ( $T_C$ ) below fusion temperature ( $T_{ref}$ ). The top and bottom walls are thermally insulated. The right wall has a constant temperature equal to the initial temperature. At time  $t = 0$  the temperature at the left wall is raised to some temperature ( $T_H$ ) above fusion temperature. Experimental results are taken from Gau and Viskanta [1]. Experimental uncertainty was not available. Here, an initial subcooling between 1 and 2 °C is indicated. Initial tests did not show significant difference between 1 and 2 °C, so  $T_C$  simply has a value of 1.5 °C below fusion temperature. Momentum boundary values are no-slip for the

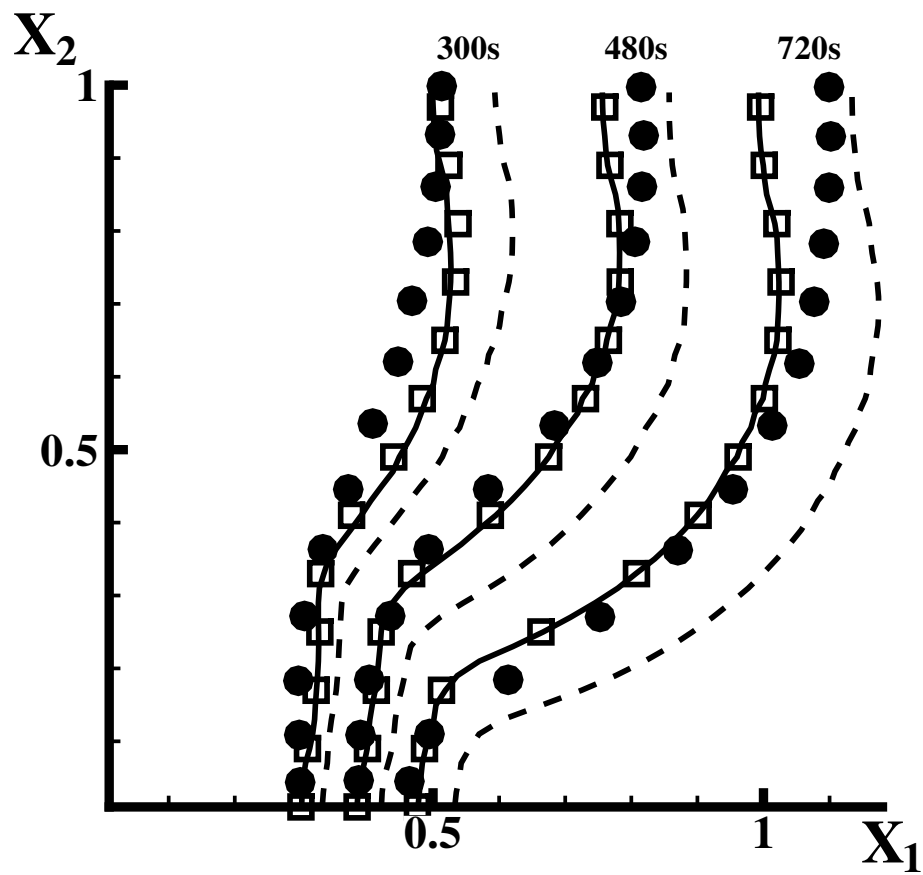


Figure 3: Solid-liquid phase front at different times. 100x50 grid, Solid line = Case A, Dashed line = Case B, Square = Case C, Circles = Experiment [1]. Note that Case A and Case C overlap. Time intervals are indicated on dimensional form



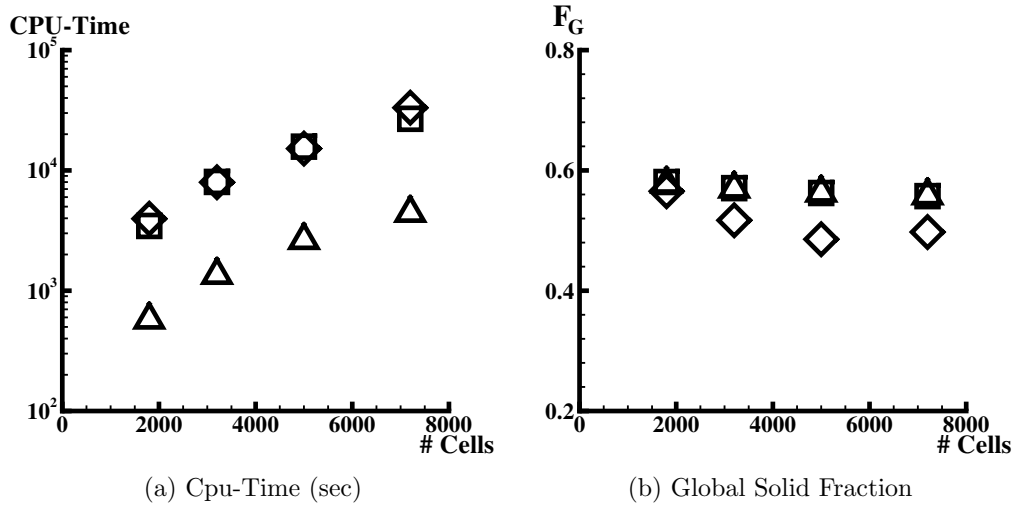


Figure 4: Grid refinement data. Delta = Case A, Diamond = Case B, Square = Case C. Note that case A and Case C overlap in fig. 4b.

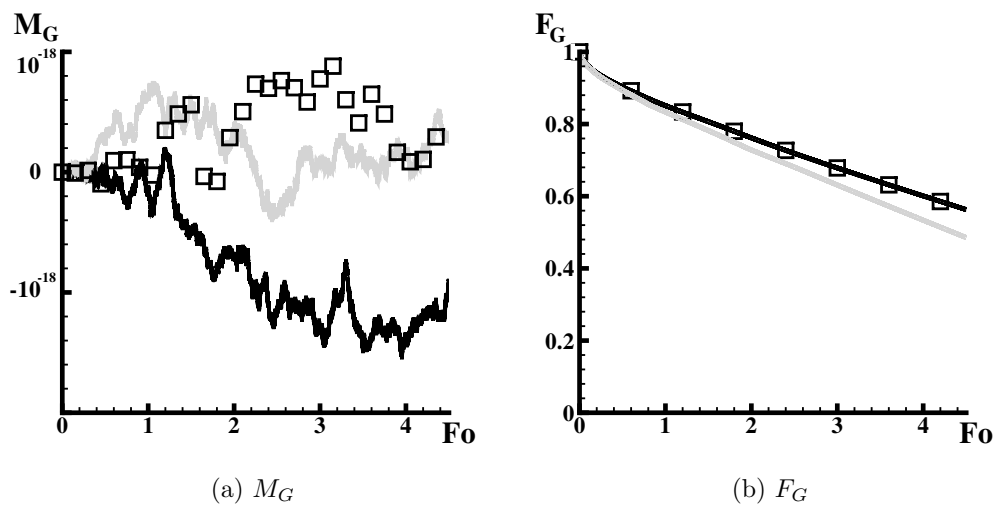


Figure 5: Monitor parameters for 100x50 grid as a function of dimensionless time ( $Fo$ ). Black = Case A, Light Gray = Case B, Square = Case C. Note that Case A and Case C overlap in figure 5b.

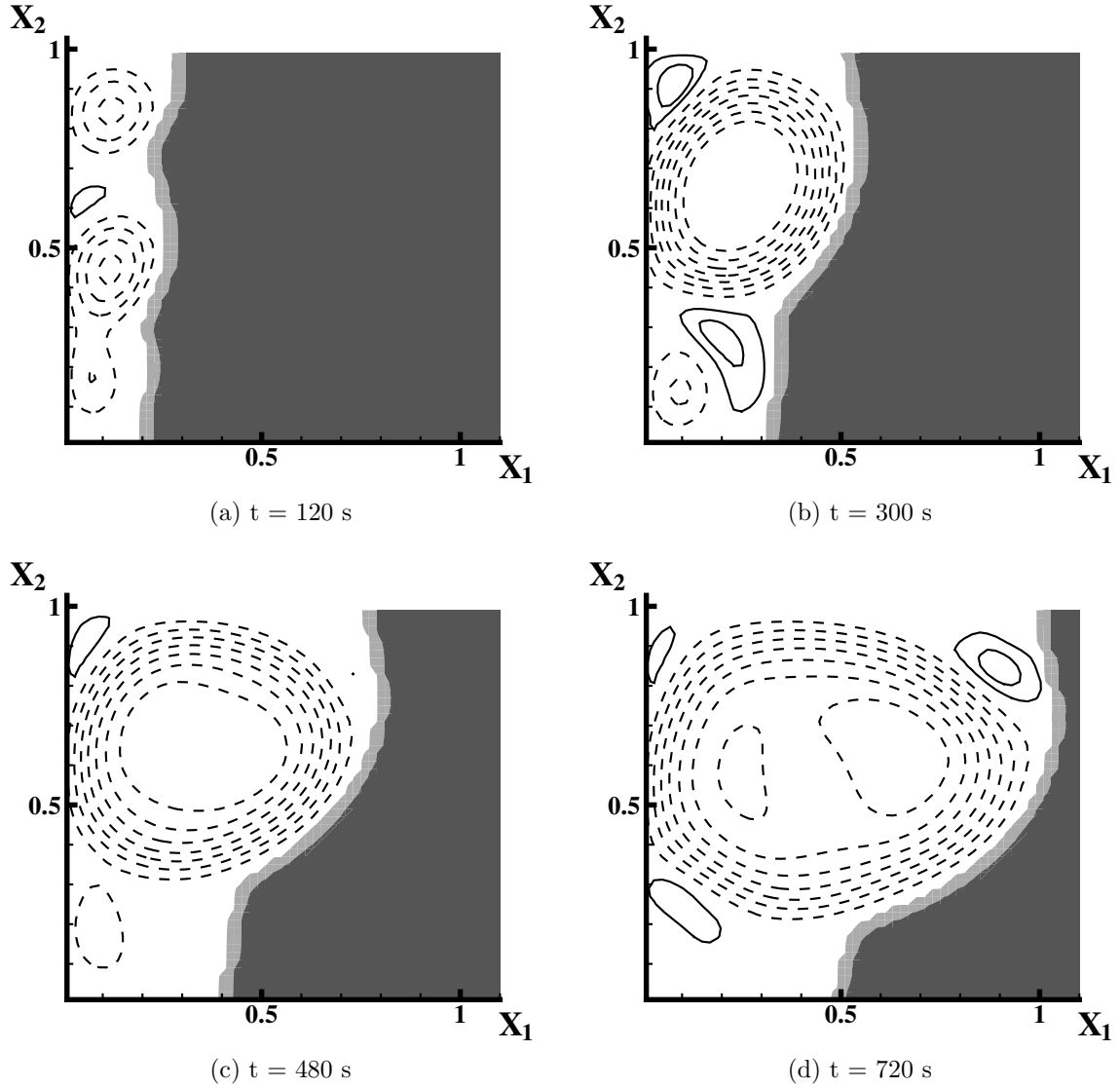


Figure 6: Case A: 100x50 grid, stream function iso-lines and solid fraction at different times. White:  $f_S < 0.05$ , light gray:  $0.05 < f_S < 0.95$ , gray:  $f_S > 0.95$ . Stream function iso-values:  $-7, -6, -5, -4, -3, -2, -1, 0.1, 0.5$ . Stream function iso-lines are dashed for negative values. Note that  $X_1$  only shows the range 0.0 to 1.1, and not 0.0 to 2.0 as indicated by the aspect ratio ( $A_r$ )

velocities and zero gradient for the pressure. Initially velocities and pressure are equal zero. The pressure has a fixed numerical value equal zero at the first computational cell in the bottom left corner such that numerical drifting is prevented. With these initial and boundary conditions three cases were tested with different combination of fluid and thermal modeling. This is listed and labeled in table 2. Simulation parameters are listed in table 3. As indicated by table 3 a constant time step ( $\Delta Fo$ ) has been utilized. The time step was refined until dampening from the BTCS discretization of the energy equation at the finest grid had minimal impact on the solution and the Navier-Stokes solver was stable.

Case	Thermal Model	Solid Velocity Model
<i>A</i>	Projection	Live Dead Cells
<i>B</i>	Source	Live Dead Cells
<i>C</i>	Projection	Darcy Porosity

Table 2: Combination of thermal and fluid models.

The predicted phase front for case A, B and C together with experimental results at different time intervals are plotted in figure 3. A study of grid independence can be seen in figure 4b. Here, the global solid fraction ( $F_G$ ) from (33) is plotted as a function of cell count. The solution was considered grid independent for a 100x50 mesh. The duration of the simulation is also an interesting parameter. This is plotted in figure 4a as a function of cell count. It is worth pointing out that Case A is in the range ten times faster than case B and C. Clearly, reducing the number of cells where the pressure correction is solved, as done by the live/dead cell method, has a huge impact on the simulation efficiency. Solving the discrete BTCS system for the energy equation only once, as done by the Projection method is clearly faster than the Source method, where the same discrete energy equation is solved multiple times. It is worth pointing out that the Source method derived and used here is a basic scheme. Faster converging implicit schemes and hybrid methods for the solid fraction correction have shown faster convergence [10]. The strength of the Source based method is the solid fraction correction scheme and no specific attention has been paid to optimize it in this paper.

Figures 5a and 5b shows the global mass conservation ( $M_G$ ) and solid fraction ( $F_G$ ) respectively.  $M_G$  shows result in the range of  $10^{-18}$  throughout the simulation, which is considered satisfactory.  $F_G$ , however, shows a difference

Property	Value	Dimensions
$T_H$	38.300	$^{\circ}C$
$T_{ref}$	29.765	$^{\circ}C$
$T_C$	28.265	$^{\circ}C$
$\rho_l$	6095.0	$\frac{kg}{m^3}$
$\rho_s$	5910.0	$\frac{kg}{m^3}$
$c_{pl}$	373.66	$\frac{J}{kg^{\circ}C}$
$c_{ps}$	396.40	$\frac{J}{kg^{\circ}C}$
$k_l$	27.82	$\frac{W}{m^{\circ}C}$
$k_s$	40.60	$\frac{W}{m^{\circ}C}$
$H$	0.0445	$m$
$\Delta Fo$	$3.0 * 10^{-5}$	—
$Ra$	$2.2 * 10^5$	—
$Pr$	0.021	—
$Ste$	0.042	—
$A_r$	2.0	—
$\epsilon_{PC}$	$10^{-7}$	—
$\epsilon_T$	$10^{-7}$	—
$\epsilon_{TT}$	$10^{-6}$	—

Table 3: Simulation parameters for the validation case.

between the Source method (Case B) and the Projection method (Case A and C). This difference is also indicated by the phase fronts in figure 3. Numerically this difference is approximately 6% at  $Fo = 4.5$ . The phase fronts predicted by the Projection method seems to yield a more accurate result than the Source method for the 2D case. However, it is worth pointing out that Ben-David et.al. [3] also predicted a faster melting rate for the 2D simulations, like the Source method in Case B shows here. Ben-David et.al. [3] did accomplished a more accurate result when the full 3D case was simulated. Solid fraction field and stream function plot of case A at different time intervals can be found in figure 6.

From an implementation aspect, the Darcy source term added for handling

flow in the solid region is a simple method easy to implement into existing codes. The live/dead cell method is somewhat more complicated. This method requires counting and marking the live fluid cells and a reconstruction of the pressure-correction matrix system at every time step. The Projection method and the Source method for phase transitions had no significant difference from an implementation point of view. Both methods requires a solution of the energy equation and an iterative procedure for finding the solid fraction. This is done as a coupled system for the Source method and as a decoupled system for the Projection method.

Interesting extensions of the Projection method presented here would be a full 3D simulation, as pointed out by Ben-David et.al. [3]. A faster converging iterative method for solving the different linear systems within the code would yield a more time efficient simulation. Several preconditioning algorithms for the CG-method is readily available in the literature [20]. A simple Secant method was used for the pointwise iteration of the solid fraction correction. This method is notoriously slow, but for a linear function  $\xi$  this method converged in two iterations. Faster Newton-Raphson methods could be necessary for more complicated versions of (10). Voller and Swaminathan [10] also presents several implicit- and hybrid methods for the solid fraction correction scheme in the family of Source based methods. Similarly, there could also be more efficient schemes for the solid fraction correction for the Projection method worth exploring.

#### 4. Conclusion

The Projection method for phase transition and the live/dead fluid cell method shown here have a clear advantage with respect to simulation efficiency. For the 2D case presented here, the Projection method also shows more accurate results than the Source method. However, according to Ben-David et.al. [3] a full 3D case develops a significant different flow pattern which again changes the evolution of the solid-liquid inter phase. The live/dead fluid cell method yielded the same results as the Darcy porosity approach. This approach was somewhat more complicated to implement than Darcy porosity, but had a significant advantage with respect to simulation efficiency.

## 5. References

- [1] C. Gau, R. Viskanta, Melting and solidification of a pure metal on a vertical wall, *Journal of Heat Transfer* 108 (1986) 174–181.
- [2] E. Sparrow, J. Ramsey, R. Kemink, Freezing controlled by natural convection, *Journal of Heat Transfer* 101 (1979) 578–584.
- [3] O. Ben-David, A. Levy, B. Mikailovich, A. Azulay, 3d numerical and experimental study of gallium melting in a rectangular container, *International Journal of Numerical Heat and Mass Transfer* 67 (2013) 260–271.
- [4] J. Ockendon, W. Hodgkins, *Moving boundary problems in heat flow and diffusion*, Oxford Univ. Press, Oxford, 1975.
- [5] V. Voller, C. Swaminathan, B. Thomas, Fixed grid techniques for phase change problems: a review, *International Journal for Numerical Methods in Engineering* 30 (1990) 875–898.
- [6] A. König-Haagen, E. Franquet, E. Pernot, D. Brüggermann, A comprehensive benchmark of fixed-grid methods for the modelling of melting, *International Journal of Thermal Sciences* 118 (2017) 69–103.
- [7] V. Voller, M. Cross, Accurate solutions of moving boundary problems using the enthalpy method, *International Journal Heat and Mass transfer* 24 (1980) 545–556.
- [8] V. Voller, M. Cross, N. Markatos, An enthalpy method for convection/diffusion phase change, *International Journal for Numerical Methods in Engineering* 24 (1987) 271–284.
- [9] V. Voller, Fast implicit finite-difference method for the analysis of phase change problems, *Numerical Heat transfer* 17 (1990) 155–169.
- [10] V. Voller, C. Swaminathan, General source-based method for solidification phase change, *Numerical Heat Transfer* 19 (1991) 175–189.
- [11] V. Voller, C. Prakash, A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems, *International Journal of Heat and Mass Transfer* 30 (1987) 1709–1719.

- [12] A. Mujumdar, Z. Gong, A finite element model for convection-dominated melting and solidification problems, *International Journal of Numerical Methods for Heat & Fluid Flow* 8 (1998) 393–408.
- [13] K. Morgan, A numerical analysis of freezing and melting with convection, *Computer Methods in Applied Mechanics and Engineering* 28 (1980) 275–284.
- [14] A. Chorin, Numerical solution of the navier-stokes equations, *Mathematics of Computations* 28 (1968) 745–762.
- [15] D. Gartling, Finite element analysis of convective heat transfer problems with change of phase, *Computer Methods in Fluids* 28 (1980) 257–284.
- [16] B. Niezgoda-Żelasko, The enthalpy-porosity method applied to the modelling of the ice slurry process during tube flow, *Procedia Engineering* 157 (2016) 114–121.
- [17] N. Morgan, A New Liquid-Vapor Phase Transition Technique For The Level-Set Method, Ph.D. thesis, Georgia Institute of Technology, 2005.
- [18] D. Porter, K. Easterling, M. Sherif, *Phase Transformation in Metals and Alloys*, CRC press, third edition, 2009.
- [19] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *The Physics of Fluids* 8 (1965) 2182–2189.
- [20] J. Shewchuk, An introduction to the conjugate gradient method without the agonizing pain (1994).

## Appendix A. A Graphical Interpretation

This section is a graphical interpretation of the Projection algorithm for solving solid-liquid phase transitions presented in this paper. One time step for three different temperature nodes  $[\theta_1, \theta_2, \theta_3]$  are presented here. In this example, node 1 is initially a solid node, node 2 is undergoing a phase transition, while node 3 is fully liquid. For simplicity a pure substance melting at temperature  $\theta_m$  is used as an example, but the principle is the same for any definition of the temperature-solid fraction relation.

Initially at a time step  $n$ , the temperature field from the previous time step is given as  $\theta^{n-1}$ . First, a guess of the temperature field is done by solving (17). Solving this system yields the preliminary temperature field  $\hat{\theta}$ , as seen in figure Appendix A.1.

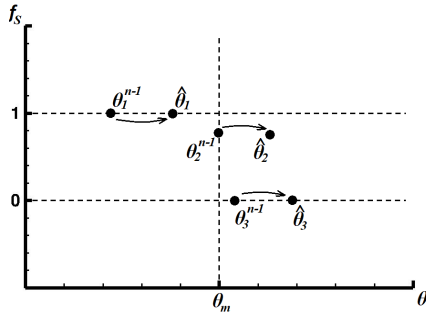


Figure Appendix A.1: Initial estimation of the temperature field  $\hat{\theta}$ .

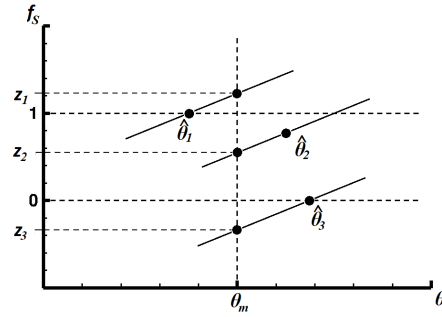


Figure Appendix A.2: Finding the corrected solid fraction by iteration.

The solution of the energy equation, (17), does not account for the change in solid fraction. The iterative scheme developed in this paper can be viewed as a search for the cross section between the solid fraction-temperature relation function defined as  $F(\theta)$  and the characteristic lines represented as diagonal lines in figure Appendix A.2. Physically, these diagonal characteristic lines is an energy balance, and represents what a change in temperature corresponds to as a change in solid fraction. Mathematically, these characteristics is represented by (20), but can be simplified into the relation  $\Delta\theta = \frac{\Delta f_s}{Ste}$ . The iterative procedure for finding the cross sectional points is the solution of function  $\xi(z) = 0$  from (22), and will yield a set of corrected solid fractions,  $[z_1, z_2, z_3]$ .



Notice from figure Appendix A.2 that  $z_1 > 1.0$  and  $z_3 < 0.0$ . This result is clearly not physical as a cell cannot be more than fully solid or fully liquid. This is where the limiter from (16) is implemented.  $z_1$  is simply limited to 1.0, while  $z_3$  is limited to 0.0. The resulting  $z$  now correctly represents the solid fraction for the current time step,  $[f_S]^n$ , and the new temperature field  $\theta^n$  can be calculated according to (20). The end result of one time step is represented in figure Appendix A.3.

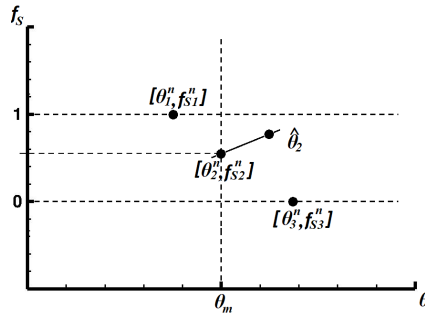


Figure Appendix A.3: Resulting solid fraction and temperature after one time step

It is worth noting that the resulting temperatures,  $\theta^n$ , in node 1 and 3 stayed the same as the initially estimated temperatures  $\hat{\theta}$ . This is due to the fact that the local solid fraction at these nodes did not change ( $\Delta f_S = 0$ ). An interesting advantage of this method is that there should not be any difficulty for the local solid fraction to change from fully solid to fully liquid in one time step. Another curiosity with this method is that the effect of the Stefan number ( $\mathbf{Ste}$ ) becomes clear. As seen from Appendix A.2 and the relation  $\Delta\theta = \frac{\Delta f_S}{\mathbf{Ste}}$ ,  $\mathbf{Ste}$  clearly corresponds to slope of the characteristic lines, and evidently the speed of the phase transition.

#### **4 Article 2**

**A 3D Study of Convection Dominated Phase Transitions Using the Projection Method**

# A 3D Study of Convection Dominated Phase Transitions Using the Projection Method

Olav Rømcke <sup>\*1</sup> and Reidar Kristoffersen <sup>†1</sup>

<sup>1</sup>*NTNU, Department of Energy- and Process Engineering, NO-7491 Trondheim, Norway*

June 8, 2018

## Abstract

A projection method for solving solid-liquid phase transitions is utilized for a 3D square cavity. This square cavity is compared to an experimental study of pure gallium. The difference between 3D and 2D results are also highlighted in this paper. In addition, a 2D study of the effects of the governing non-dimensional parameters and a demonstration of the importance of the convective currents are also done.

Keywords: Phase Transition, Dimensional parameters, Convection, 3D, Gallium, Numerical Simulation

## 1 Introduction

The melting square cavity case is a widely researched topic. Early experimental, analytical and numerical studies date back to the late 70's early 80's [1], [2], [3], [4], [5], [6]. A later review paper by König-Haagen et.al. [7] recaps the common numerical fixed grid methods for solving convection dominated solid-liquid phase transition up to 2017. Work by Ben-David et.al. [8] and Niezgodna-Zelasko [9] also demonstrates the most common method implemented in commercially available software like Comsol and Fluent respectively.

Today, the most common methods are categorized into three main solution strategies: Apparent Heat Capacity Method, Enthalpy Method and the Source Method. Interested readers are referred to the article by König-Haagen et.al. [7] and earlier review articles by Voller et.al. [10] [11].

This paper, however, investigates a new solution strategy developed by Rømcke et.al. [12]. The overall strategy shows some resemblance to the Source Method, but the idea behind the iterative scheme is based on the fractional step method commonly used for solving the Navier-Stokes equations. This new method will be summarized here, but for a thorough derivation and comparison to the Source Method, the reader is referred to the original paper [12]. It is worth noting that a similar solution strategy has been implemented in N.R Morgan's PhD thesis [13], a numerical study of film boiling utilizing the Level Set Method.

---

<sup>\*</sup>Corresponding Author: olavromc@stud.ntnu.no

<sup>†</sup>reidar.kristoffersen@ntnu.no

## Nomenclature

$\alpha$	Thermal Diffusivity	$i, j$	Spacial Direction
$\beta$	Thermal Expansion	$K$	Thermal Conductivity Ratio
$\epsilon$	Convergence Criterion	$k$	Thermal Conductivity
$\hat{\phantom{x}}$	Preliminary	$l$	Liquid, Subscript
$\mu$	Viscosity, Kinetic	$M$	Melting, Subscript
$\nu$	Viscosity, Kinematic	$M_G$	Global Mass Conservation
$\Omega$	Calculation domain	$n$	# Discrete Time Steps
$\rho$	Density	$P$	Pressure, Non-Dim
$\theta$	Temperature, Non-Dim	$p$	Pressure
$\xi()$	Correction Function	$PC$	Press-Corr, Subscript
$A_r$	Aspect Ratio	$Pr$	Prantl Number
$C$	Cold, Subscript	$r$	Residual Vector
$c$	Center Node, Subscript	$Ra$	Rayleigh Number
$C_P$	Volumetric Heat Capacity	$res$	Residual Norm
	Ratio	$s$	Solid, Subscript
$c_p$	Heat Capacity	$Ste$	Stefan Number
$D$	Cavity Width	$T$	Energy, Subscript
$F()$	Function	$T$	Temperature
$F_B$	Force, Buoyancy	$t$	Time
$F_L$	Global Solid Fraction	$TT$	Thermal, Subscript
$f_S$	Local Solid Fraction	$U$	Velocity, Non-Dim
$Fo$	Time, Non-Dim	$u$	Velocity
$g$	Gravitational acceleration	$x$	Position
$H$	Cavity Height	$X, Y, Z$	Position, Non-Dim
$H$	Hot, Subscript	$z$	Unlimited Solid Fraction

As indicated by experimental studies [1] [5] [8], the diffusive transport of energy is not enough to explain the development of the solid-liquid phase front. Depending on the parameters of the experiment (spacial scale, temperature range, material, ect.), the convective currents controls much of the development of the solid-liquid inter phase. A good description of the fluid flow is thus necessary for describing the physics involved. As indicated by Ben-David [8], 2D simulations do not account for the 3D flow structures needed to explain the square cavity melting case.

The main focus of this paper is to numerically describe the full 3D behavior of the square cavity melting case and compare it to experiments by Ben-David et.al. [8]. A numerical study of the role of the dimensionless numbers has also been done together with 2D simulations of some interesting mould-melt geometries.

## 2 Model

Similar to Rømcke et. al. [12] the physical assumptions which the mathematical model will be based on are:

1. Heat transfer is governed by convection and conduction.
2. The fluid flow is laminar and incompressible.
3. Viscous dissipation is negligible.
4. Density changes only effect the free convection. Change in density are only considered for the Boussinesq approximation.
5. Physical properties ( $c_p, \mu, k \dots$ ) only differ between solid and liquid phases. They do not vary with temperature.
6. A phase field function is a sufficient representation of the average local phase fraction.
7. The phase transition is isothermal.
8. Solidified material is in full contact with the boundary walls

## 2.1 Mathematical formulation

The convention used throughout this paper is the non-dimensional form of the governing equations. The scaling utilized here will be similar to that indicated by Rømcke et.al. [12], with the exception that the reference values ( $\rho_l, k_l, c_{pl} \dots$ ) are values at  $T_H$ . Note also that the temperature range is defined as  $T_H - T_C$ .  $T_H$  and  $T_C$  represents hot and cold temperatures, respectively

$$\begin{aligned}
 U_i &= \frac{u_i H}{\alpha_l} & \theta &= \frac{T - T_C}{\Delta T} & P &= \frac{p H^2}{\rho_l \alpha_l^2} \\
 X_i &= \frac{x_i}{H} & C_P &= \frac{c_p \rho}{c_{pl} \rho_l} & \alpha_l &= \frac{k_l}{c_{pl} \rho_l} \\
 Fo &= \frac{t \alpha_l}{H^2} & \nu &= \frac{\mu}{\rho_l} & \Delta T &= T_H - T_C \\
 Pr &= \frac{\nu}{\alpha_l} & Ra &= \frac{g \beta \Delta T H^3}{\nu \alpha_l} & Ste &= \frac{\rho_l c_{pl} \Delta T}{L} \\
 A_r &= \frac{D}{H} & K &= \frac{k}{k_l}
 \end{aligned}$$

the non dimensional form of the governing equations are:

$$\frac{\partial U_i}{\partial X_i} = 0 \quad (1)$$

$$\frac{\partial U_i}{\partial Fo} + \frac{\partial (U_j U_i)}{\partial X_j} = -\frac{\partial P}{\partial X_i} + Pr \frac{\partial^2 U_i}{\partial X_j^2} + Ra Pr \theta \frac{\partial X_i}{\partial X_2} \quad (2)$$

$$\frac{\partial \theta}{\partial Fo} + \frac{\partial (U_j \theta)}{\partial X_j} = \frac{\partial}{\partial X_j} \left( K \frac{\partial \theta}{\partial X_j} \right) + \frac{1}{Ste} \frac{\partial f_S}{\partial Fo} \quad (3)$$

$$f_S = F(\theta) \quad (4)$$

It is worth pointing out that  $X_1 = X$ ,  $X_2 = Y$  and  $X_3 = Z$ . The Navier-Stokes equations, (1) and (2), represents conservation of mass and the momentum equation respectively. (3) is the energy equation, while (4) is the solid fraction - temperature relation. Note that the reference temperature in this case is the cold

temperature, and not the melting temperature. As indicated in Rømcke et.al. [12] the dimensionless volumetric heat capacity ( $C_P$ ) is not a part of the non dimensional equations, as it varies insignificantly over the relevant temperature domain. However, the thermal conductivity ratio ( $K$ ) has a significant difference between solid and liquid state ( $\frac{k_s}{k_l} = 1.46$ ) and is thus accounted for in the mathematics. The convention used in this paper is the non-dimensional form of the governing equations (1) to (4). The dimensionless numbers  $Pr$ ,  $Ra$ ,  $Ste$  and  $Fo$  are worth noting as they represent the key parameters of the simulation. In addition, two derived parameters  $F_L$  and  $M_G$  represent global liquid fraction and global conservation of mass respectively. They are defined accordingly:

$$M_G = \int_0^{Fo} \frac{1}{Vol} \left[ \int_{\Omega} \nabla(\vec{U}) d\Omega \right] dFo \quad (5)$$

$$F_L = 1 - \frac{1}{Vol} \int_{\Omega} f_S d\Omega \quad (6)$$

$$Vol = \int_{\Omega} d\Omega \quad (7)$$

where  $\Omega$  represent the calculation domain, while  $Vol$  is the total domain volume. Ideally,  $F_L$  should remain constant under grid- and time step refinement, while  $M_G$  should remain close to zero during the whole simulation.

## 2.2 Numerical Solution

The reader is referred to the article by Rømcke et.al. for a thorough explanation of both the fluid and the thermal models. The same approach is utilized here with the only exception that the code has been expanded to 3D.

### 2.2.1 Thermal Model

A projection method for solid-liquid phase transitions has been utilized for the thermal modeling. For every discrete time step, this approach can be summarized accordingly:

1. Estimate a temperature field ( $\hat{\theta}$ ) from (8). The *RHS* is discretized according to the BTCS-scheme. Note that for this initial estimation, it is assumed that no material changes phase.
2. Correct the solid fraction ( $f_S$ ) by finding the root of  $\xi$  from (10). This is done by iteration and the Secant method has been utilized.
3. Limit the solid fraction to physical values according to (11). The roots of (10) does not necessarily yield physical values for the solid fraction, thus the corrected solid fraction values needs to be limited between 0.0 and 1.0.
4. Correct the temperature field according to (12). The temperature  $[\theta]^n$  now represents the temperature for the current time step.

The equations used are listed below. Here the discretized energy equation states:

$$\frac{\hat{\theta}_c - [\theta_c]^{n-1}}{\Delta Fo} = RHS \quad (8)$$

where

$$RHS = \frac{\partial}{\partial X_j} \left( K \frac{\partial \theta}{\partial X_j} \right) - \frac{\partial (U_j \theta)}{\partial X_j}. \quad (9)$$

The correct solid fraction is the solution to  $\xi(z) = 0$  where  $z$  represents the corrected solid fractions for the next time step.

$$\xi(z) = z - [f_S]^{n-1} - Ste \left( F^{-1}(z) - \hat{\theta}_c \right) \quad (10)$$

The roots found from (10) need to be limited between 0 and 1, as values outside this range do not represent physical values of the solid fraction.

$$\begin{aligned} f_S &= 0, \text{ if } f_S < 0 \\ f_S &= 1, \text{ if } f_S > 1 \end{aligned} \quad (11)$$

When the solid fraction for the next time step has been found, the initially estimated temperature field can be corrected according to:

$$[\theta_c]^n - \hat{\theta}_c = \frac{1}{Ste} ([f_S]^n - [f_S]^{n-1}) \quad (12)$$

The superscript  $n$  represents the time step. From an implementation aspect, it is worth noting that expansion to 3D was trivial.

### 2.2.2 Fluid Model

A FD Projection method has been utilized to solve the Navier-Stokes equation. For one time step, this method can be summarized accordingly:

1. Estimate a Velocity field ( $\hat{U}$ ) from (13). This equation is discretized according to the FTCS-scheme. Note that the pressure field from the previous time step is used for the initial velocity estimation.
2. Find the pressure correction from (14). By discretizing the Poisson pressure-correction equation a linear system of equation can be formulated. This matrix-vector system is then solved iteratively by a conjugate gradient method.
3. Based on the pressure-correction,  $\Delta P$ , the initially estimated velocities are corrected according to (16). The pressure is simply updated by (15).

The equations used for the fluid modelling are listed below.

$$\frac{\hat{U}_i - [U_i]^{n-1}}{\Delta Fo} = -\frac{\partial(U_j U_i)}{\partial X_j} - \left[ \frac{\partial P}{\partial X_i} \right]^{n-1} + Pr \frac{\partial^2 U_i}{\partial X_j^2} + RaPr\theta \frac{\partial X_i}{\partial X_2} \quad (13)$$

$$\frac{\partial^2(\Delta P)}{\partial X_i^2} = \frac{1}{\Delta Fo} \frac{\partial \hat{U}_i}{\partial X_i} \quad (14)$$

$$[P]^n = [P]^{n-1} + \Delta P \quad (15)$$

$$[U_i]^n = \hat{U}_i - \Delta Fo \frac{\partial(\Delta P)}{\partial X_i} \quad (16)$$

Flow in the solid region is simply handled by a live/dead fluid cell method [12]. For pure substances, a cell is marked as a dead fluid cell if the local solid fraction is above 0.5. When a cell is marked as a dead fluid cell the velocities should be zero and solving the Navier-Stokes equations for these cells is unnecessary. As the MAC approach [14] for location of pressure and velocity nodes is used here, the marking of live and dead fluid cells can be summarized accordingly:

1. If a velocity node has both of its neighbouring cells marked as a dead fluid cells, then the velocity is known to be zero (no-slip, no-penetration boundary).
2. If a pressure node has a neighbouring cell marked as a dead fluid cell, then the spacial derivative in that direction is known to be zero (zero gradient boundary).

By utilizing the information above, the Poisson pressure correction matrix system needs to be formulated at every time step. This has been shown more computationally efficient than other common methods for handling fluid flow in the solidified region [12].

### 3 Results and Discussion

Three main cases are presented in this section. As shown by previous work, accounting for the full 3D physics can have a significant impact on the evolution of the solid liquid interphase. A study of the governing dimensionless numbers (Ra, Pr, Ste) is also presented. Additionally, two mould-melt systems have been simulated with and without convective transport. The goal for this section is to show the importance of the 3D effects, how the dimensionless numbers qualitatively effect the physical system and the importance of the convective currents.

Property	Value	Dimensions	Property	Value	Dimensions
$T_H$	40.0	$^{\circ}C$	$H$	0.06	$m$
$T_M$	29.85	$^{\circ}C$	$\Delta Fo$	$2.0 * 10^{-5}$	—
$T_C$	25.0	$^{\circ}C$	$Ra$	$3.712 * 10^5$	—
$\rho_l$	6089.0	$\frac{kg}{m^3}$	$Pr$	0.0247	—
$\rho_s$	5910.0	$\frac{kg}{m^3}$	$Ste$	0.07444	—
$c_{pl}$	365.66	$\frac{J}{kg^{\circ}C}$	$A_r$	1.5	—
$c_{ps}$	396.40	$\frac{J}{kg^{\circ}C}$	$\epsilon_{PC}$	$10^{-6}$	—
$k_l$	28.32	$\frac{W}{m^{\circ}C}$	$\epsilon_T$	$10^{-6}$	—
$k_s$	40.60	$\frac{W}{m^{\circ}C}$	$\epsilon_{TT}$	$10^{-6}$	—

Table 1: Simulation parameters for the 3D validation case. Here, liquid physical properties are taken at  $T_H$  and solid properties at  $T_M$ . Values are found in [8].

#### 3.1 3D Simulation of the Cavity Melting Problem

As indicated by the literature [8] there is a difference in the flow field developed in a full 3D simulation compared to the 2D case. The convective transport has a significant impact on the shape and evolution of the solid liquid interphase, thus a thorough description of the flow field is important. A 2D and a 3D case have been simulated and compared to experimental results.

The system consists of a  $0.09 \times 0.06 \times 0.06$  m cavity with gallium at a temperature  $T_C$  below fusion temperature ( $T_M$ ). The front, back, top and bottom walls are



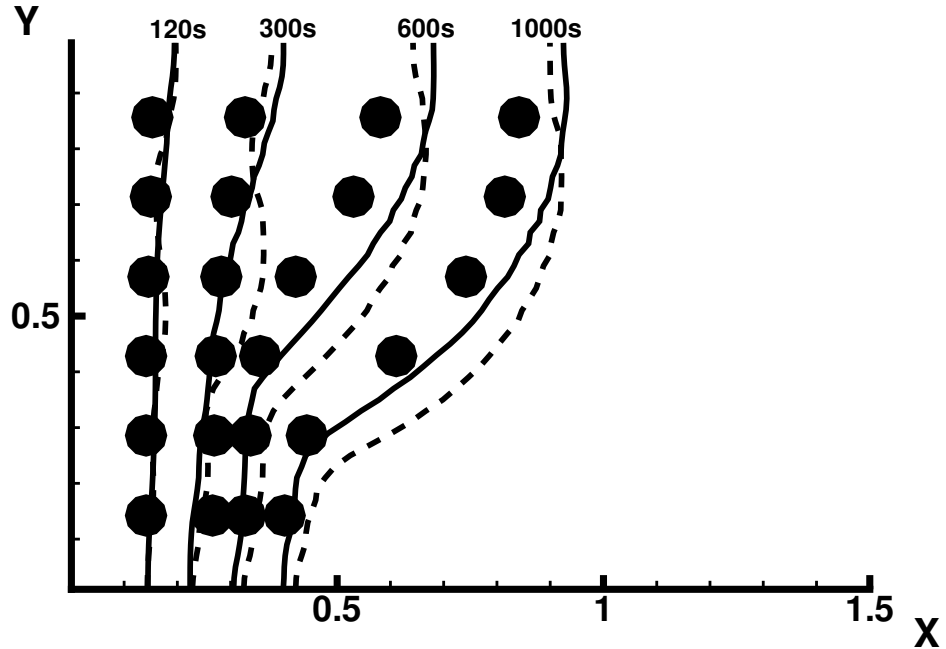


Figure 1: Solid-liquid phase front at different times. Circles: Experiment. Solid line: 3D. Dashed line: 2D. Experimental results are taken from [8]. Time intervals are indicated on dimensional form.

all thermally insulated, while the right wall has a constant temperature equal  $T_C$ . At time equal zero, the left wall has a sudden temperature rise to  $T_H$  above fusion temperature.

Simulation parameters for this case can be found in table 1. The experimental results are taken from Ben-David et.al. [8]. Figure 1 presents the 3D and 2D cases together with the experimental results. Figure 2 shows the vertical midplane crosssection, while figure 3 presents the horizontal midplane crosssection at different times. The simulation has been done on a  $100 \times 50 \times 50$  grid. The resulting residuals was ensured to never increase above the criterions indicated in table 1 and the global conservation of mass never exceeded  $M_G = 10^{-17}$ . A study of grid- and time step refinement was also done in order to verify a convergent result. 40% more cells, and a 25% finer time-step had less than 1% impact on the global solid fraction  $F_L$ .

The difference between the 2D and the full 3D case becomes clear in figure 1. The obvious result is that the 3D case shows a significantly slower moving interphase. Another key difference between the 2D and the 3D case is the development of the 2D vortexes early in the simulation [8] [12]. These are not present in the full 3D case. Both the 2D and the 3D case yielded significantly faster moving interphases than the experimental values indicate. The only variation in physical properties considered for this simulation is the thermal conductivity difference between solid and liquid state. How the properties vary with temperature has not been accounted for. From the physical properties used in Ben-David et.al. [8] viscosity and thermal conductivity shows some temperature

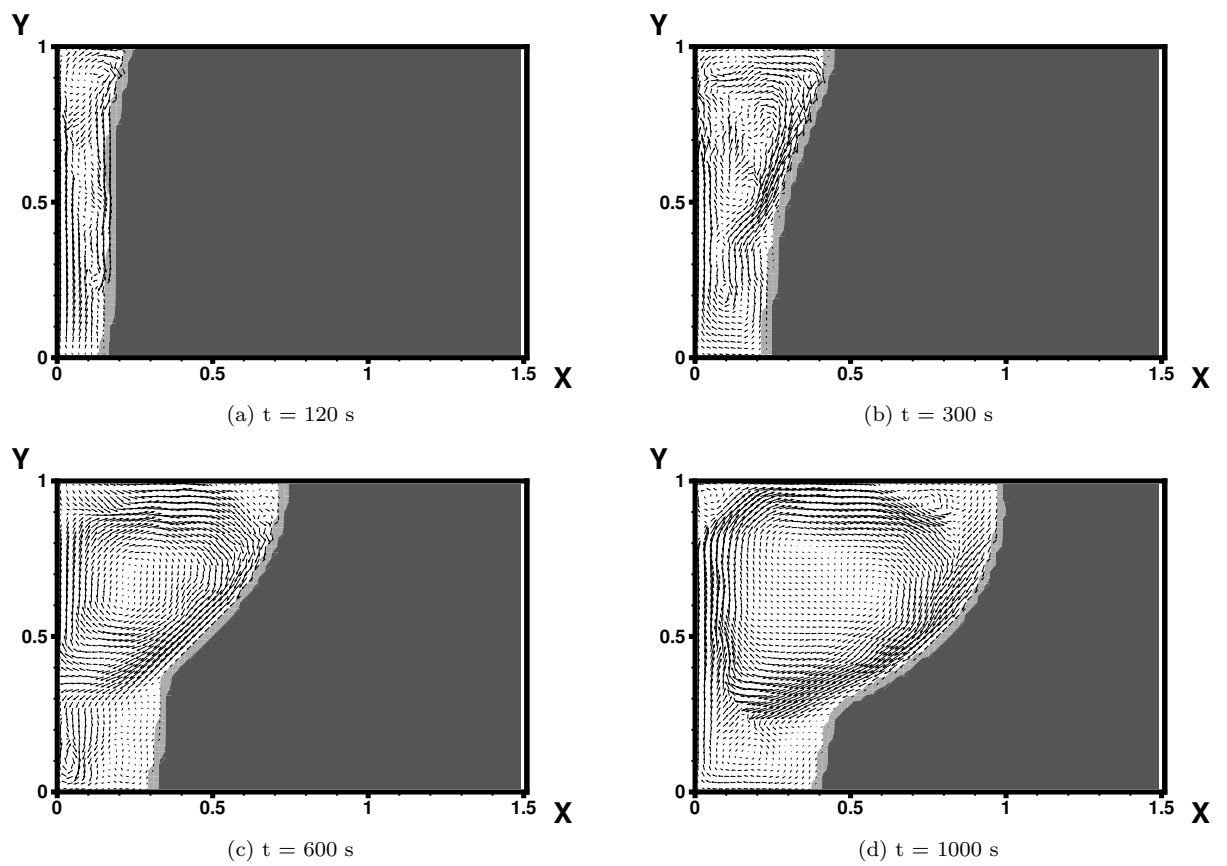


Figure 2: Velocity vector field and the solid fraction at the XY - midplane ( $Z = 0.5$ ). (dark gray:  $f_S > 0.95$ , light gray:  $0.05 < f_S < 0.95$ , white:  $f_S < 0.05$ ) at different times.

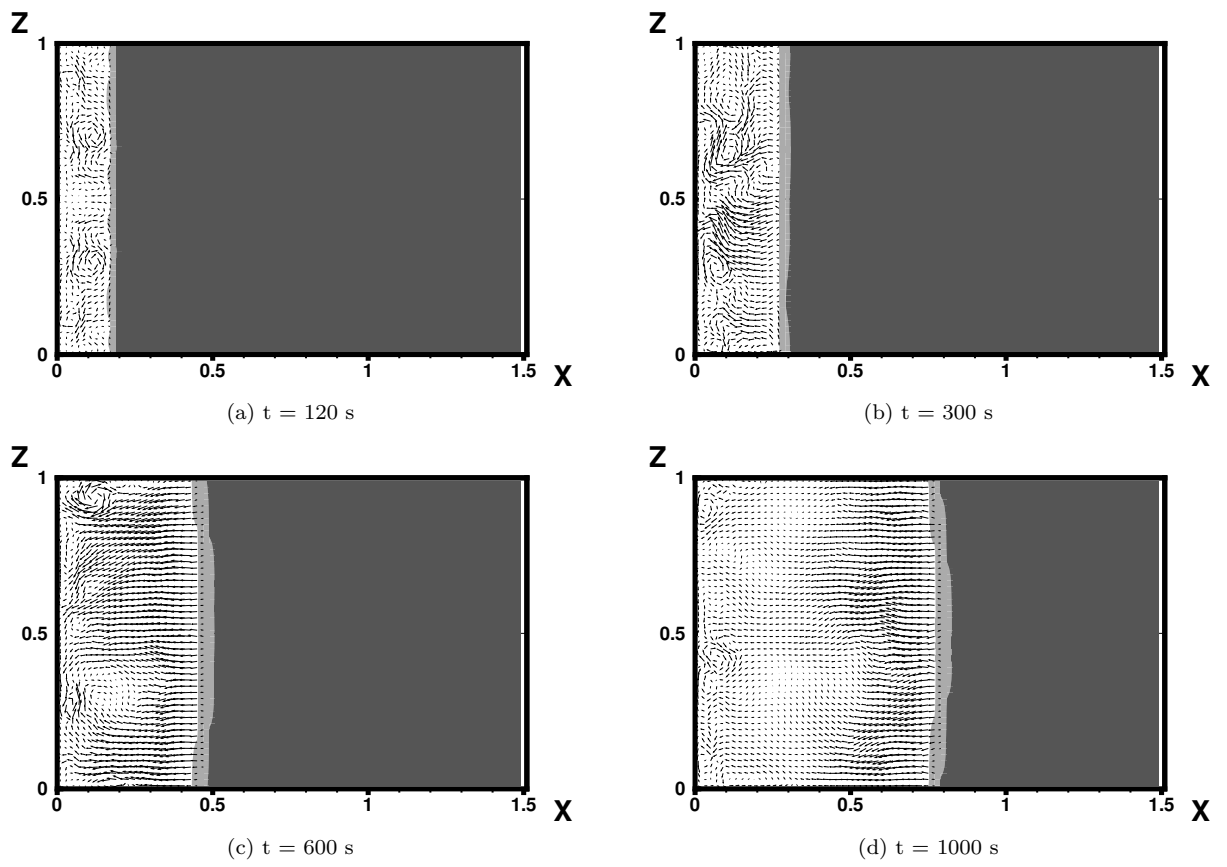


Figure 3: Velocity vector field and the solid fraction at the XZ - midplane ( $Y = 0.5$ ). (dark gray:  $f_S > 0.95$ , light gray:  $0.05 < f_S < 0.95$ , white:  $f_S < 0.05$ ) at different times.

Property	Value	Property	Value
$\theta_H$	1.0	$K$	1.0
$\theta_M$	0.5	$\epsilon_{PC}$	$10^{-6}$
$\theta_C$	0.0	$\epsilon_T$	$10^{-6}$
$\Delta Fo$	$2.0 * 10^{-5}$	$\epsilon_{TT}$	$10^{-6}$
$A_r$	1.0		

Table 2: Simulation parameters for the dimensional study.

Property	A	B	C
$Ra$	$10^4$	$10^5$	$10^6$
$Pr$	0.1	0.01	0.001
$Ste$	0.1	0.05	0.01
$Fo$	2.5	1.2	0.5

Table 3: Different values for the dimensional constants.

dependence that could yield a significant difference from the results presented here.

### 3.2 Effects of the Governing Dimensionless Parameters

Three dimensionless numbers ( $Ra$ ,  $Pr$ ,  $Ste$ ) appear in the governing equations (1) to (4). A small graphical study is presented in this section, followed by a sensitivity analysis in section 3.3. Values utilized in the graphical study can be found in table 2, where the dimensional constants used for the graphical representation can be found in table 3. Here, the simulations are only done in 2D.

Figure 4 indicates that an increase in any of the dimensional constants also increase the overall melting speed. However, both the Prantl number and the Rayleigh number seem to have some part in controlling the shape of the solid liquid interphase.

### 3.3 Sensitivity Analysis

A more extensive sensitivity analysis of the specific impact the dimensional parameters have on the global liquid fraction ( $F_L$ ) has also been done. From experimental studies [5] [15] correlation to some dimensional group ( $\Pi$ ) have been attempted. For the 2D case presented here, an attempt was made at finding a dimensional group for the global liquid fraction. From equation (17), the goal is to estimate the parameters  $c_1$  to  $c_5$ .

$$F_L = \Pi = c_1 Ra^{c_2} Pr^{c_3} Ste^{c_4} Fo^{c_5} \quad (17)$$

The author recognizes that the physics is not correctly described in 2D, as already shown in section 3.1. From a quantitative perspective, the values acquired for equation (17) will naturally not reflect the reality. However, the analysis will

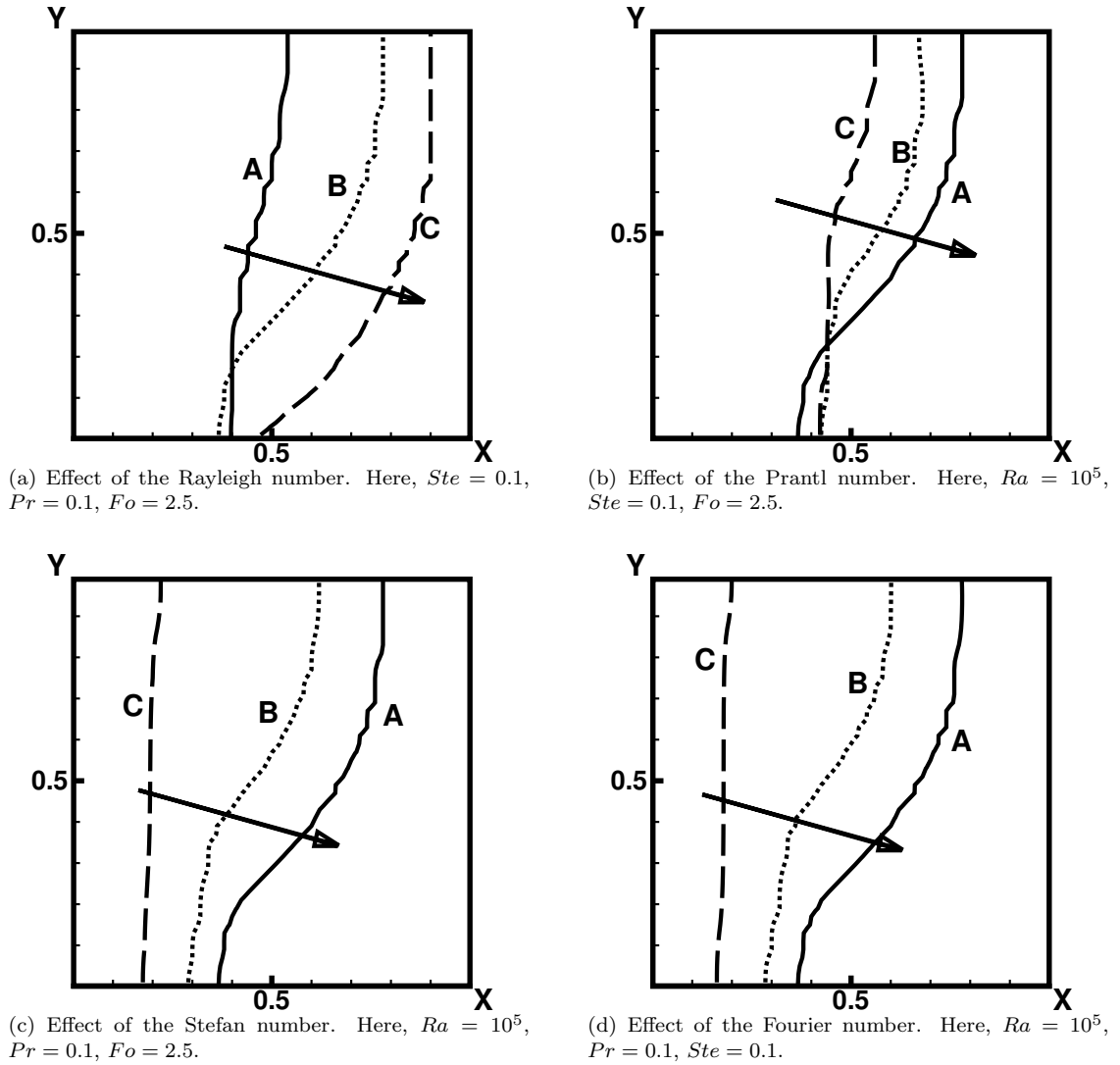


Figure 4: Isolated variation in the dimensional parameters. Simulation parameters can be found in table 2. For the three different cases A, B and C details are listed in table 3. The arrow indicates direction of increasing values of Rayleigh, Prantl, Stefan and Fourier, respectively. Note that the same relative increase in  $Fo$  and  $Ste$  results in almost equivalent evolution of the solid - liquid interphase.

Parameter	Min	Max
$Ra$	$10^{3.5}$	$10^{6.5}$
$Pr$	$10^{-2.3}$	$10^{-0.8}$
$Ste$	$10^{-3.3}$	$10^{-0.8}$
$Fo$	0.0	2.5

Table 4: Range of the dimensional parameters used in the sensitivity analysis.

Property	Value	Property	Value
$\theta_H$	1.0	$K$	1.0
$\theta_M$	0.01	$\epsilon_{PC}$	$10^{-6}$
$\theta_C$	0.0	$\epsilon_T$	$10^{-6}$
$A_r$	1.0	$\epsilon_{TT}$	$10^{-6}$

Table 5: Simulation parameters for the sensitivity analysis.

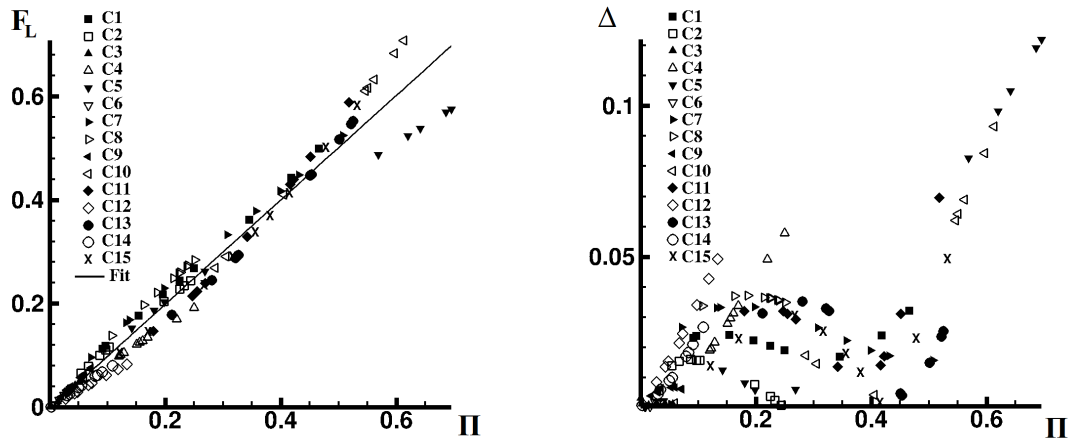
reflect some of the qualitative aspects of the system. In total 15 simulations were done on a 2D  $50 \times 50$  grid. Nine values of time ( $Fo$ ) were chosen for each simulation. In total 135 data points were acquired. All dimensional parameters were chosen at random between the ranges indicated in table 4. In order to neglect the dependence on the initial subcooling below fusion temperature, the melting temperature was very close to the initial temperature. Further simulation parameters can be found in table 5. The time step chosen was small enough for each simulation such that there were no issues with numerical instability. If the solidifying front reached the right wall, the simulation ended. From the resulting data acquired from the simulations it was sought to minimize the Cost function defined by equation (18).

$$Cost = |c_1 Ra^{c_2} Pr^{c_3} Ste^{c_4} Fo^{c_5} - F_L| \quad (18)$$

Figure 5a shows the simulated values together with a data fitted curve, while 5b presents the absolute deviation ( $\Delta$ ) from the fitted curve.  $\Pi$  was found to be:

$$\Pi = 0.577 Ra^{0.106} Pr^{0.068} Ste^{0.529} Fo^{0.619} \quad (19)$$

Some similarities with the previously found correlations by Gau et.al. [5] are worth noting (20). Gau et.al. tried to correlate the measured liquid fraction with the dimensional group  $\tau = SteFo$ ,  $Ra$  and  $A_r$ . In this paper, the Stefan and Fourier numbers are treated as separate variables.  $Fo$  and  $Ste$  seem to have the largest impact on the liquid fraction in (19). The difference between Gau et.al. and this analysis is the quantitative result. As Gau et.al. found  $\tau^{0.843}$  to correlate well with experimental result, the analysis undergone here yielded  $Ste^{0.529}$  and  $Fo^{0.619}$  respectively. However, a somewhat larger dependence on the Rayleigh number was found here ( $Ra^{0.106}$ ) compared to Gau et.al. ( $Ra^{0.0504}$ ). The Liquid fraction seems to have a weak dependence on the Prantl number ( $Pr^{0.068}$ ).



(a) Global liquid fraction,  $F_L$ , as a function of the dimensional group  $\Pi$ . The fitted line follows the curve  $F_L = 1.009\Pi - 0.003$ .  $R^2 = 0.97$

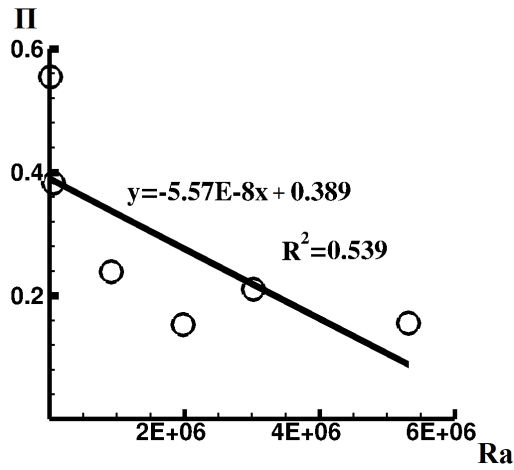
(b) Point wise deviation from the fitted line. Note the bump around  $\Pi = 0.4$ .

Figure 5: Results from the sensitivity analysis. Here, it was found that  $\Pi = 0.577Ra^{0.106}Pr^{0.068}Ste^{0.529}Fo^{0.62}$ .

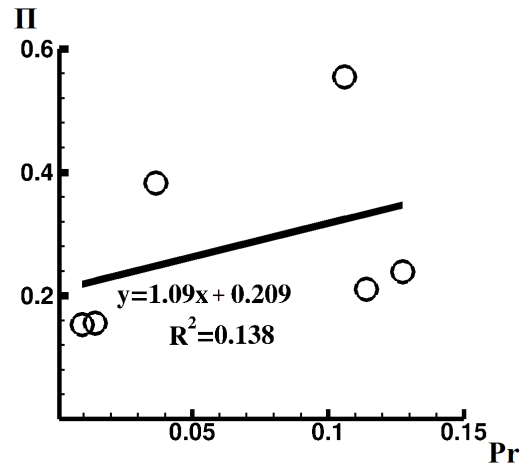
$$[F_L]_{Gau} = 2.708(SteFo)^{0.843}Ra^{0.0504}Ar^{-0.14} \quad (20)$$

It was observed that some simulations kept the solid liquid interphase a nearly vertical line, while others had a more curved interphase. Examples of this can be seen in figure 4. It is hypothesized, that the interphase shape says something about how convection dominated the system is. The interphase length  $S$  was used as a measure of this. It is worth pointing out that a perfectly conduction dominated system would have a constant interphase length of 1. Figure 6d shows that some of the simulations experience a sudden increase in interphase length  $S$ . This transition from conduction to convection is here characterized as the sudden increase in  $S$ . As seen in figure 6a, the point at which the transition occurs seems to happen at lower values of  $\Pi$  for higher values of  $Ra$ . Thus, large Rayleigh numbers would indicate a more convection dominated system, as expected. With regards to the Prantl number, the connection between figure 4b and 6b is not obvious. Figure 6b plots  $\Pi$  when the transition to convection dominated behaviour occurs as a function the Prantl number. Figure 4b on the other hand plots the position of the interphase for changes in  $Pr$  only. The two results (figure 4b and 6b) may seem contradictory, but no conclusive remark can be made due to the low coefficient of determination ( $R^2$ ) in figure 6b. The Stefan number seem to be independent on whether or not the system transitions to a convection dominated behaviour, but again the low  $R^2$  value makes any conclusive remarks difficult.

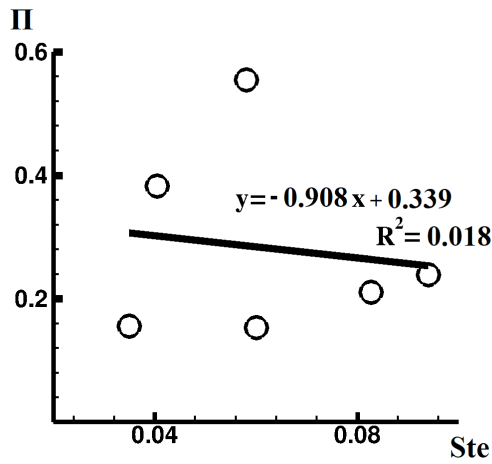
In summary, figure 4 in section 3.2 and figure 6 in section 3.3 states that Rayleigh and Prantl numbers seem to control the shape of the solid-liquid interphase, while the Stefan and Fourier numbers have a larger impact on the global liquid fraction.



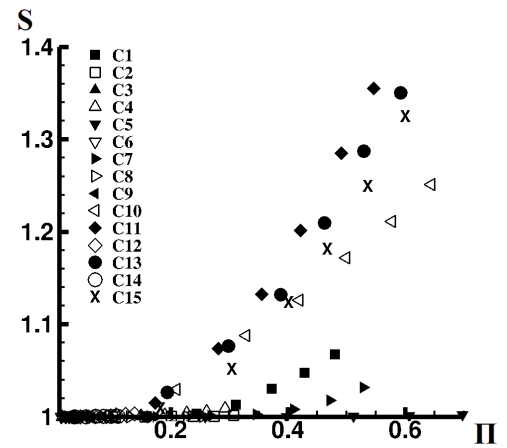
(a) Transition to convective dominated heat transfer as a function of the Rayleigh number



(b) Transition to convective dominated heat transfer as a function of the Prantl number



(c) Transition to convective dominated heat transfer as a function of the Stefan number



(d) Solid liquid interphase length  $S$  as a function of  $\Pi$

Figure 6: Transition to convective dominated behaviour. Figure 6d shows the sudden increase of the length ( $S$ ) of the solid liquid interphase.  $S$  is a variable attempting to say something about the shape of the solid liquid interphase. Figures 6a, 6b and 6c plots the value  $\Pi$  at which the simulation experienced a sudden increase in  $S$  as a function of Rayleigh, Prantl and Stefan numbers respectively. Coefficient of determination is indicated by  $R^2$ .



Property	Value	Property	Value
$\theta_H$	1.0	$Ra$	$2.0 * 10^7$
$\theta_M$	0.5	$Pr$	50
$\theta_C$	0.0	$Ste$	50
<i>Mould Height</i>	1.0	$\epsilon_{PC}$	$10^{-6}$
<i>Mould Width</i>	1.0	$\epsilon_T$	$10^{-6}$
$\Delta Fo$	$7.0 * 10^{-7}$	$\epsilon_{TT}$	$10^{-6}$

Table 6: Simulation parameters for the Mould - Melt systems.

### 3.4 Mould - Melt System

In casting technology it is important to control the evolution of the solid-liquid phase front. Though not accounted for in this paper, materials typically experience shrinkage during solidification. If a portion of melt is encapsulated within a solidified area, the solidified part will most lightly develop pores. Therefore, designing mould shapes that keep the last solidifying material in non-crucial areas of the mould is essential. This section utilizes the previously described model to emphasize the importance of including the convective currents occurring during solidification. The author recognizes that no attention has been paid to model undercooling, crystal precipitation or other effects typically influencing a solidifying material [16]. Two cross sectional shapes are represented here: a Pipe and a H-Beam. The simulations are in 2D, which could arguably be valid for systems that stretch far out in the third dimension, so end effects can be neglected. 3D flow structures could still develop, depending on the system. Simulations are carried out on a  $75 \times 75$  grid. Simulations was also carried out on a  $85 \times 85$  grid. Here, the time it took for the domain to fully solidify was within 2% of the  $75 \times 75$  grid, and the simulation was considered grid independent.

The shape is encapsulated in a mould which has pragmatically been given the same physical properties as the melt with the exception that it is fully solid during the whole simulation, and never undergoes a phase transition. Initially, the fluid is at rest with temperature  $\theta_H$  and the mould has a temperature  $\theta_C$ . The outer mould wall is kept at a constant cold temperature  $\theta_C$ . The material is considered pure, with a melting temperature of  $\theta_M$ . Two key parameters have been monitored. First, the placement of the nodes solidifying last and secondly the time taken for the mold cavity to be fully solidified. These two parameters have been compared to a cases where natural convection have been neglected. Figure 7 shows the evolution of the solid-liquid phasefront with and without

Shape	Convection	Solidification time (Fo)	Final solid point (X, Y)
<i>H - Beam</i>	<i>Yes</i>	0.032	(0.50, 0.59)
<i>H - Beam</i>	<i>No</i>	0.040	(0.50, 0.50)
<i>Pipe</i>	<i>Yes</i>	0.0284	(0.50, 0, 65)
<i>Pipe</i>	<i>No</i>	0.0285	<i>Concentric inner circle</i>

Table 7: Results for the Mould - Melt systems.

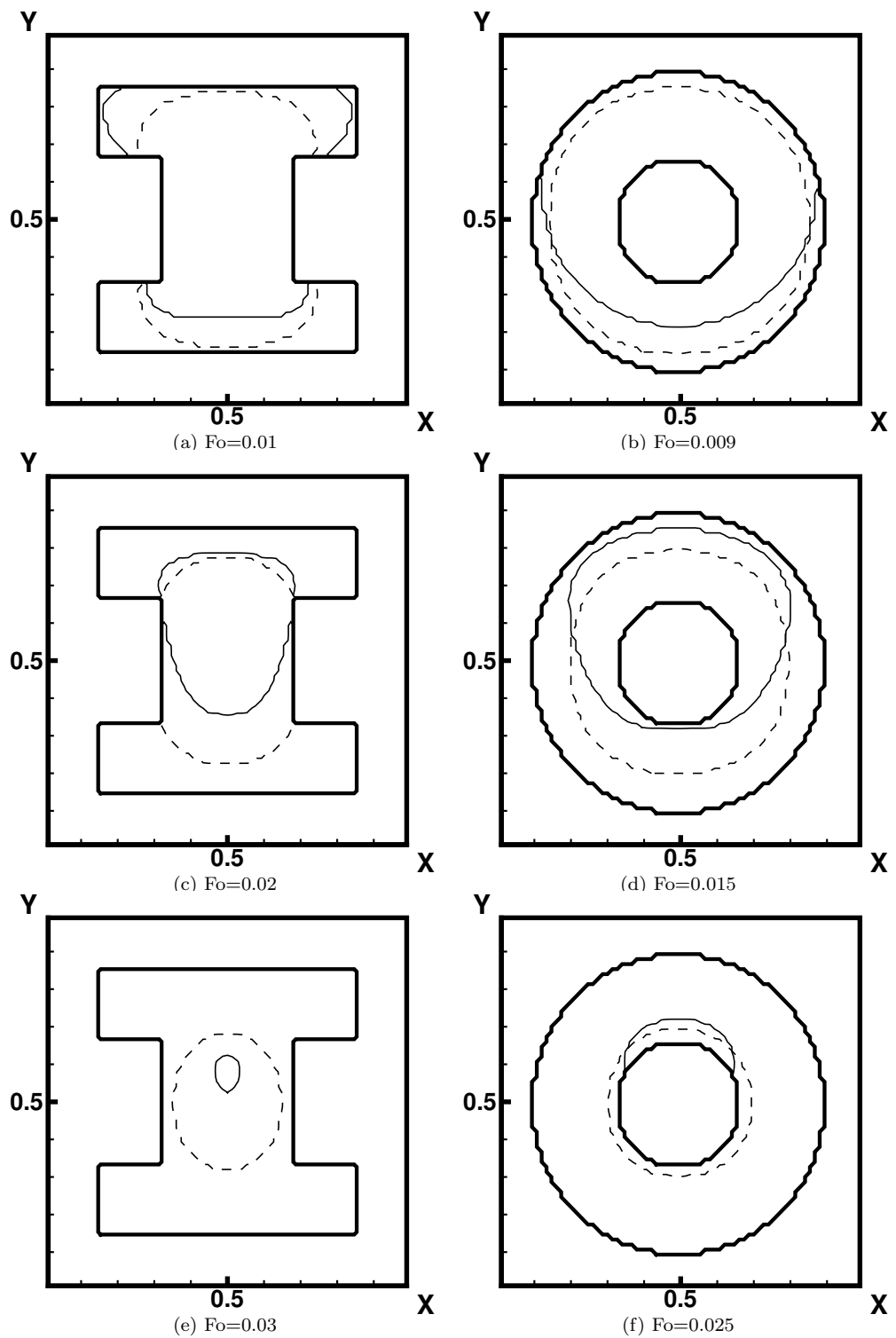


Figure 7: Solid liquid phase front ( $f_S = 0.5$ ) at different times. Dashed: Conduction only. Solid: Convection and conduction. The thick line indicates the mould shape.

natural convection. A clear difference is observed for all cases. Firstly, by including convection, the final solidifying point tends to move upwards. This is due to hot fluid being transported by convective currents in the positive Y-direction. Secondly, another effect induced by convection is that the fluid tends to solidify faster. For the tube, this effect is only marginally noticeable (less than 2%), but quite significant for the H-beam. The numerical values are listed in table 7.

## 4 Conclusion

Simulations shows that a full 3D description of the physics is necessary for achieving accurate results as demonstrated here and by Ben-David et.al. [8]. From an implementation aspect, extending the original 2D algorithm presented by Rømcke et.al. [12] to a full 3D model was straight forward. The 2D sensitivity analysis conducted here also revealed some similarities to previously found correlations, but a notable numerical difference. Interestingly, for the domain tested here, the Prantl and Rayleigh numbers had a relatively small influence on the evolution of the global liquid fraction. The dominating parameters with regards to the global liquid fraction were the Fourier and Stefan numbers. However, it is worth pointing out that Prantl and Rayleigh had an impact on the shape of the solid liquid interphase. Notably the low  $R^2$  values when trying to correlate the conduction-convection transition makes any conclusive remarks difficult. However, the approach presented here might be a useful framework for future studies. The importance of the convective currents has also been demonstrated for some interesting mould - melt systems.

## References

- [1] E.M. Sparrow, J.W. Ramsey, and R.G. Kemink. Freezing controlled by natural convection. *Journal of Heat Transfer*, 101:578–584, November 1979.
- [2] K. Morgan. A numerical analysis of freezing and melting with convection. *Computer Methods in Applied Mechanics and Engineering*, 28:275–284, 1980.
- [3] V.R. Voller and M. Cross. Accurate solutions of moving boundary problems using the enthalpy method. *International Journal Heat and Mass transfer*, 24:545–556, 1980.
- [4] J.R. Ockedon and W.R. Hodgkins. *Moving boundary problems in heat flow and diffusion*. Oxford Univ. Press, Oxford, 1975.
- [5] C. Gau and R. Viskanta. Melting and solidification of a pure metal on a vertical wall. *Journal of Heat Transfer*, 108:174–181, February 1986.
- [6] V.R. Voller and C. Prakash. A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems. *International Journal of Heat and Mass Transfer*, 30:1709–1719, 1987.
- [7] A. König-Haagen, E. Franquet, E. Pernot, and D. Brüggermann. A comprehensive benchmark of fixed-grid methods for the modelling of melting. *International Journal of Thermal Sciences*, 118:69–103, 2017.

- [8] O. Ben-David, A. Levy, B. Mikailovich, and A. Azulay. 3d numerical and experimental study of gallium melting in a rectangular container. *International Journal of Numerical Heat and Mass Transfer*, 67:260–271, 2013.
- [9] B. Niezgoda-Żelasko. The enthalpy-porosity method applied to the modelling of the ice slurry process during tube flow. *Procedia Engineering*, 157:114–121, 2016.
- [10] V.R. Voller, C.R. Swaminathan, and B.G. Thomas. Fixed grid techniques for phase change problems: a review. *International Journal for Numerical Methods in Engineering*, 30:875–898, 1990.
- [11] V.R. Voller and C.R. Swaminathan. General source-based method for solidification phase change. *Numerical Heat Transfer*, 19:175–189, 1991.
- [12] O. Rø mcke. A projection method for convection dominated phase transitions (unpublished). *Preprint submitted to Journal of Applied Mathematical Modelling*, 2018.
- [13] N.R. Morgan. *A New Liquid-Vapor Phase Transition Technique For The Level-Set Method*. PhD thesis, Georgia Institute of Technology, 2005.
- [14] F.H. Harlow and J.E Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8:2182–2189, 1965.
- [15] C.J. Ho and R. Viskanta. Heat transfer during melting from an isothermal vertical wall. *Journal of Heat Transfer*, 106:12–19, 1984.
- [16] D.A. Porter, K.E. Easterling, and M.Y. Sherif. *Phase Transformation in Metals and Alloys*. CRC press, third edition, 2009.

## 5 Trails and Tribulations

This section aims at presenting the work not necessarily presented in the articles, but was key stepping stones towards the development of the final model. The sparse matrix solver, choice of discretization and boundary conditions are fundamental with regards to this. In addition, the fluid model was compared to an analytic solution to the pressure driven channel problem and attempts at simulating more complex phase transition behaviour other than pure materials was also done. For a thorough derivation of the full model, the reader is referred to article 1.

### 5.1 Notes on the Buckingham- $\Pi$ Theorem

The Buckingham- $\Pi$  theorem loosely states that a system of  $v$  variables involving  $d$  physical dimensions can be described by a set of  $p = v - d$  non-dimensional parameters. On its simplest form with constant physical properties, the system studied in this thesis have  $v = 16$  variables ( $t, u_i, p, T, f_S, x_i, c_{pl}, k_l, \rho_l, \mu, g, \beta, H, D, \Delta T, L$ ) spanning  $d = 4$  fundamental dimensions ( $[Distance]$ ,  $[Mass]$ ,  $[Time]$ ,  $[Temperature]$ ). By the Buckingham- $\Pi$  theorem, the system should be described by  $p = 12$  non-dimensional groups. However, ignoring  $C_P$  and  $K$  which account for changing physical properties, only 10  $\Pi$ -groups are identified from the scaling of the governing equations in article 1 and 2:

$$\begin{aligned} \Pi_1 = \frac{u_i H}{\alpha_l} = U_i, \quad \Pi_2 = \frac{T - T_C}{\Delta T} = \theta, \quad \Pi_3 = \frac{p H^2}{\rho_l \alpha_l^2} = P, \quad \Pi_4 = \frac{x_i}{H} = X_i, \quad \Pi_5 = \frac{t \alpha_l}{H^2} = Fo \\ \Pi_6 = \frac{\nu}{\alpha_l} = Pr, \quad \Pi_7 = \frac{g \beta \Delta T H^3}{\nu \alpha_l} = Ra, \quad \Pi_8 = \frac{\rho_l c_{pl} \Delta T}{L} = Ste, \quad \Pi_9 = \frac{D}{H} = Ar, \quad \Pi_{10} = f_S \end{aligned}$$

Here,  $\alpha_l = \frac{k_l}{c_{pl} \rho_l}$  and  $\nu = \frac{\mu}{\rho_l}$ . All these non-dimensional groups ( $\Pi_1 - \Pi_{10}$ ) have an intuitive physical meaning. Some questions arise:

- What are the two remaining groups;  $\Pi_{11}$  and  $\Pi_{12}$ ?
- How can they (e.i.  $\Pi_{11}$  and  $\Pi_{12}$ ) be defined in such a way that they have some physical intuitive meaning?
- What effect do they have on the system?
- In article 2, the global liquid fraction  $F_L$  and the non-dimensional group  $\Pi$  were defined. Are they good candidates for  $\Pi_{11}$  and  $\Pi_{12}$ ?  $F_L$  clearly represents the global liquid fraction, but are there some intuitive physical interpretation of  $\Pi$  from article 2?
- Could some manipulation of the governing equation (derivation, integration, substitution, ect.) reveal a  $\Pi$ -group that the current form does not show?

This thesis does not investigate these questions any further, but it is recognized that answering them could potentially yield a deeper understanding of the physics involved.

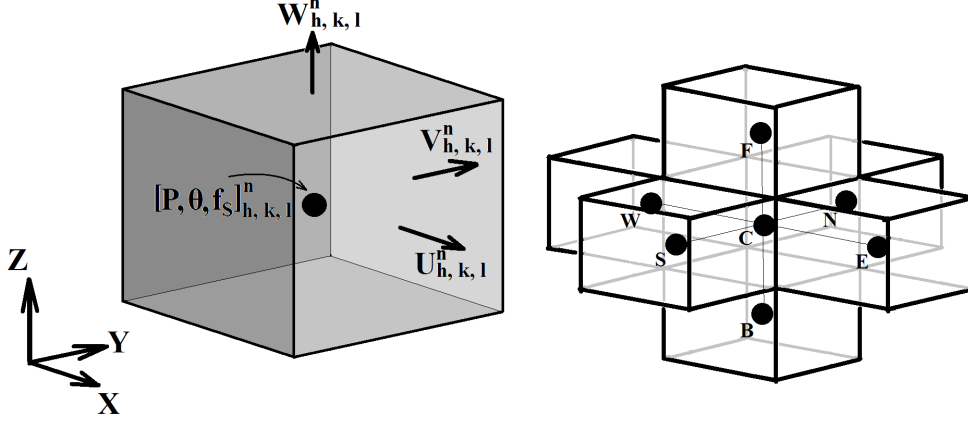


Figure 1: (Left) A 3D cell in a staggered grid. The velocity vectors ( $U$ ,  $V$ ,  $W$ ) are indicated on the cell phases, while the intensive properties ( $P$ ,  $\theta$ ,  $f_S$ ) are placed in the cell center. Superscript ( $n$ ) indicates discrete time step. Subscript ( $h$ ,  $k$ ,  $l$ ) indicates discrete position in ( $X$ ,  $Y$ ,  $Z$ ) direction respectively. (Right) Neighbouring cell notation.

## 5.2 Discretization and Boundaries

The discretization of the energy equation and the momentum equation is only mentioned in article 1 and 2. To summarize, the domain is meshed as a staggered grid. The preliminary velocity field is discretized by a FTCS scheme, while the preliminary temperature field is discretized by a BTCS scheme. Boundary conditions handled throughout this report is constant temperature, zero gradient pressure, no-slip for the velocity and zero gradient for pressure. Additionally, zero gradient for velocity and constant pressure is addressed in the pressure driven channel case. This section only deals with the 3D discretization. A single 3D grid cell is presented in figure 1. Note that the velocities are defined as  $U$ ,  $V$ ,  $W$  instead of  $U_1$ ,  $U_2$ ,  $U_3$  which has been the convention used in article 1 and 2. In this section the equations will be written out in its full form, thus it is easier to define the variables in terms of  $U$ ,  $V$ ,  $W$  to avoid a lot of subscripts that would make the syntax difficult to read. Code snippets are included in the appendix.

### 5.2.1 FTCS - Momentum Equations

For the calculation of the preliminary velocity field in the projection method, a Forward-Time-Central-Space (FTCS) scheme is used. The velocities are calculated accordingly:

X-Direction:

$$\hat{U}_{h,k,l} = U_{h,k,l}^{n-1} + \Delta Fo (-\nabla P_X - FU_X - FU_Y - FU_Z + Pr * Visc_U) \quad (1)$$

with

$$\begin{aligned} \nabla P_X &= \frac{P_{h+1,k,l}^{n-1} - P_{h,k,l}^{n-1}}{\Delta X} \\ FU_X &= \frac{(U_{h,k,l}^{n-1} + U_{h+1,k,l}^{n-1})^2 - (U_{h-1,k,l}^{n-1} + U_{h,k,l}^{n-1})^2}{4\Delta X} \\ FU_Y &= \frac{(V_{h,k,l}^{n-1} + V_{h+1,k,l}^{n-1})(U_{h,k,l}^{n-1} + U_{h,k+1,l}^{n-1}) - (V_{h,k-1,l}^{n-1} + V_{h+1,k-1,l}^{n-1})(U_{h,k-1,l}^{n-1} + U_{h,k,l}^{n-1})}{4\Delta Y} \\ FU_Z &= \frac{(W_{h,k,l}^{n-1} + W_{h+1,k,l}^{n-1})(U_{h,k,l}^{n-1} + U_{h,k,l+1}^{n-1}) - (W_{h,k,l-1}^{n-1} + W_{h+1,k,l-1}^{n-1})(U_{h,k,l-1}^{n-1} + U_{h,k,l}^{n-1})}{4\Delta Z} \\ Visc_U &= \frac{U_{h+1,k,l}^{n-1} - 2U_{h,k,l}^{n-1} + U_{h-1,k,l}^{n-1}}{\Delta X} + \frac{U_{h,k+1,l}^{n-1} - 2U_{h,k,l}^{n-1} + U_{h,k-1,l}^{n-1}}{\Delta Y} + \frac{U_{h,k,l+1}^{n-1} - 2U_{h,k,l}^{n-1} + U_{h,k,l-1}^{n-1}}{\Delta Z} \end{aligned}$$

Y-Direction:

$$\hat{V}_{h,k,l} = V_{h,k,l}^{n-1} + \Delta Fo (-\nabla P_Y - FV_X - FV_Y - FV_Z + Pr * Visc_V + F_B) \quad (2)$$

with

$$\begin{aligned}
F_B &= \frac{1}{2} Pr Ra (\theta_{h,k,l}^{n-1} + \theta_{h,k+1,l}^{n-1}) \\
\nabla P_Y &= \frac{P_{h,k+1,l}^{n-1} - P_{h,k,l}^{n-1}}{\Delta Y} \\
FV_X &= \frac{(U_{h,k,l}^{n-1} + U_{h,k+1,l}^{n-1})(V_{h,k,l}^{n-1} + V_{h,k+1,l}^{n-1}) - (U_{h-1,k,l}^{n-1} + U_{h-1,k+1,l}^{n-1})(V_{h-1,k,l}^{n-1} + V_{h,k,l}^{n-1})}{4\Delta X} \\
FV_Y &= \frac{(V_{h,k,l}^{n-1} + V_{h,k+1,l}^{n-1})^2 - (V_{h,k-1,l}^{n-1} + V_{h,k,l}^{n-1})^2}{4\Delta Y} \\
FV_Z &= \frac{(W_{h,k,l}^{n-1} + W_{h,k+1,l}^{n-1})(V_{h,k,l}^{n-1} + V_{h,k,l+1}^{n-1}) - (W_{h,k,l-1}^{n-1} + W_{h,k+1,l-1}^{n-1})(V_{h,k,l-1}^{n-1} + V_{h,k,l}^{n-1})}{4\Delta Z} \\
Visc_W &= \frac{V_{h+1,k,l}^{n-1} - 2V_{h,k,l}^{n-1} + V_{h-1,k,l}^{n-1}}{\Delta X} + \frac{V_{h,k+1,l}^{n-1} - 2V_{h,k,l}^{n-1} + V_{h,k-1,l}^{n-1}}{\Delta Y} + \frac{V_{h,k,l+1}^{n-1} - 2V_{h,k,l}^{n-1} + V_{h,k,l-1}^{n-1}}{\Delta Z}
\end{aligned}$$

Z-Direction:

$$\hat{W}_{h,k,l} = W_{h,k,l}^{n-1} + \Delta Fo (-\nabla P_Z - FW_X - FW_Y - FW_Z + Pr * Visc_W) \quad (3)$$

with

$$\begin{aligned}
\nabla P_Z &= \frac{P_{h,k,l+1}^{n-1} - P_{h,k,l}^{n-1}}{\Delta Z} \\
FW_X &= \frac{(U_{h,k,l}^{n-1} + U_{h,k,l+1}^{n-1})(W_{h,k,l}^{n-1} + W_{h+1,k,l}^{n-1}) - (U_{h-1,k,l}^{n-1} + U_{h-1,k,l+1}^{n-1})(W_{h-1,k,l}^{n-1} + W_{h,k,l}^{n-1})}{4\Delta X} \\
FW_Y &= \frac{(V_{h,k,l}^{n-1} + V_{h,k,l+1}^{n-1})(W_{h,k,l}^{n-1} + W_{h,k+1,l}^{n-1}) - (V_{h,k-1,l}^{n-1} + V_{h,k-1,l+1}^{n-1})(W_{h,k-1,l}^{n-1} + W_{h,k,l}^{n-1})}{4\Delta Y} \\
FW_Z &= \frac{(W_{h,k,l}^{n-1} + W_{h,k,l+1}^{n-1})^2 - (W_{h,k,l-1}^{n-1} + W_{h,k,l}^{n-1})^2}{4\Delta Z} \\
Visc_W &= \frac{W_{h+1,k,l}^{n-1} - 2W_{h,k,l}^{n-1} + W_{h-1,k,l}^{n-1}}{\Delta X} + \frac{W_{h,k+1,l}^{n-1} - 2W_{h,k,l}^{n-1} + W_{h,k-1,l}^{n-1}}{\Delta Y} + \frac{W_{h,k,l+1}^{n-1} - 2W_{h,k,l}^{n-1} + W_{h,k,l-1}^{n-1}}{\Delta Z}
\end{aligned}$$

Note that the pressure field from the previous time step is used here.  $Visc$  denotes the viscous term, while  $FU$ ,  $FV$  and  $FW$  denotes the advective terms of  $U$ ,  $V$  and  $W$  respectively in their indicated directions.

### 5.2.2 Pressure Correction

After the preliminary velocities are calculated, the correct pressure field needs to be established. The pressure correction also works as a basis for correcting the velocities. Discretizing the Poisson equation for the pressure correction results in a linear system of equations. One line of this matrix system can be presented accordingly:

$$\begin{bmatrix} a_F & a_N & a_E & a_C & a_W & a_S & a_B \end{bmatrix} \begin{bmatrix} \Delta P_{h,k,l+1} \\ \Delta P_{h,k+1,l} \\ \Delta P_{h+1,k,l} \\ \Delta P_{h,k,l} \\ \Delta P_{h-1,k,l} \\ \Delta P_{h,k-1,l} \\ \Delta P_{h,k,l-1} \end{bmatrix} = \frac{\hat{U}_{h,k,l} - \hat{U}_{h-1,k,l}}{\Delta X} + \frac{\hat{V}_{h,k,l} - \hat{V}_{h,k-1,l}}{\Delta Y} + \frac{\hat{W}_{h,k,l} - \hat{W}_{h,k,l-1}}{\Delta Z} \quad (4)$$

with

$$\begin{aligned}
a_E &= \frac{1}{\Delta X^2} & a_W &= \frac{1}{\Delta X^2} \\
a_N &= \frac{1}{\Delta Y^2} & a_S &= \frac{1}{\Delta Y^2} \\
a_F &= \frac{1}{\Delta Z^2} & a_B &= \frac{1}{\Delta Z^2} \\
a_P &= -(a_E + a_W + a_N + a_S + a_F + a_B)
\end{aligned}$$

The result is a sparse matrix system which is solved with a conjugated gradient method. When a satisfactory convergence criterion is reached pressure and velocities are updated according to (5) (6) (7) (8).

$$P^n = P_{h,k,l}^{n-1} + \Delta P \quad (5)$$

$$U_{h,k,l}^n = \hat{U}_{h,k,l} - \Delta F_o \frac{\Delta P_{h+1,k,l} - \Delta P_{h,k,l}}{\Delta X} \quad (6)$$

$$V_{h,k,l}^n = \hat{V}_{h,k,l} - \Delta F_o \frac{\Delta P_{h,k+1,l} - \Delta P_{h,k,l}}{\Delta Y} \quad (7)$$

$$W_{h,k,l}^n = \hat{W}_{h,k,l} - \Delta F_o \frac{\Delta P_{h,k,l+1} - \Delta P_{h,k,l}}{\Delta Z} \quad (8)$$

### 5.2.3 BTCS - Energy Equation

When calculating the preliminary temperature field ( $\hat{\theta}$ ) in the projection method for phase transition problems derived in article 1, a Backward-Time-Central-Space (BTCS) scheme is used. The result is a linear system of equations. One line of the general form of this matrix system can be presented accordingly:

$$\begin{bmatrix} a_F & a_N & a_E & a_C & a_W & a_S & a_B \end{bmatrix} \begin{bmatrix} \hat{\theta}_{h,k,l+1} \\ \hat{\theta}_{h,k+1,l} \\ \hat{\theta}_{h+1,k,l} \\ \hat{\theta}_{h,k,l} \\ \hat{\theta}_{h-1,k,l} \\ \hat{\theta}_{h,k-1,l} \\ \hat{\theta}_{h,k,l-1} \end{bmatrix} = a_0 \theta_{h,k,l}^{n-1} \quad (9)$$

with

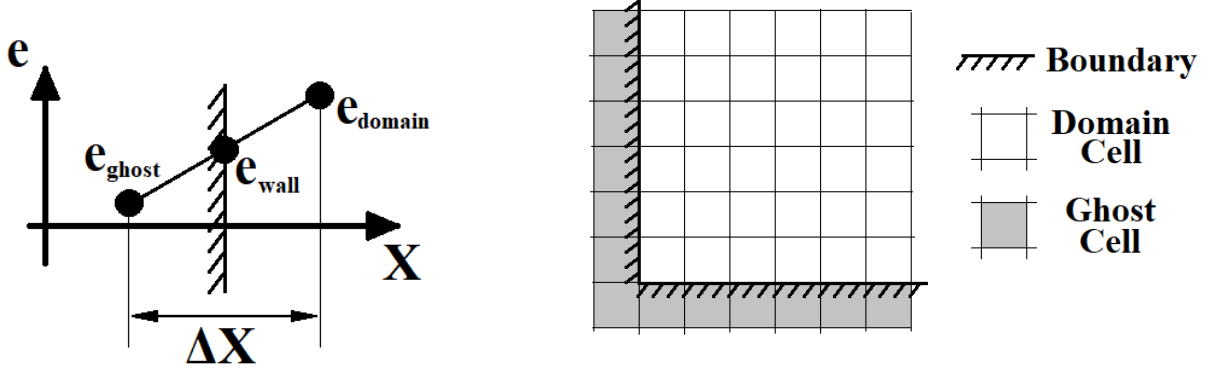
$$\begin{aligned} a_E &= \frac{U_{h,k,l}^n}{2\Delta X} - \frac{K_{h,k,l}^{n-1}}{\Delta X^2} - \frac{K_{h+1,k,l}^{n-1} - K_{h-1,k,l}^{n-1}}{4\Delta X^2} \\ a_W &= -\frac{U_{h-1,k,l}^n}{2\Delta X} - \frac{K_{h,k,l}^{n-1}}{\Delta X^2} + \frac{K_{h+1,k,l}^{n-1} - K_{h-1,k,l}^{n-1}}{4\Delta X^2} \\ a_N &= \frac{V_{h,k,l}^n}{2\Delta Y} - \frac{K_{h,k,l}^{n-1}}{\Delta Y^2} - \frac{K_{h,k+1,l}^{n-1} - K_{h,k-1,l}^{n-1}}{4\Delta Y^2} \\ a_S &= -\frac{V_{h,k-1,l}^n}{2\Delta Y} - \frac{K_{h,k,l}^{n-1}}{\Delta Y^2} + \frac{K_{h,k+1,l}^{n-1} - K_{h,k-1,l}^{n-1}}{4\Delta Y^2} \\ a_F &= \frac{W_{h,k,l}^n}{2\Delta Z} - \frac{K_{h,k,l}^{n-1}}{\Delta Z^2} - \frac{K_{h,k,l+1}^{n-1} - K_{h,k,l-1}^{n-1}}{4\Delta Z^2} \\ a_B &= -\frac{W_{h,k,l-1}^n}{2\Delta Z} - \frac{K_{h,k,l}^{n-1}}{\Delta Z^2} + \frac{K_{h,k,l+1}^{n-1} - K_{h,k,l-1}^{n-1}}{4\Delta Z^2} \\ a_P &= 4K_{h,k,l}^{n-1} \left( \frac{1}{\Delta X^2} + \frac{1}{\Delta Y^2} + \frac{1}{\Delta Z^2} \right) \\ a_0 &= \frac{1}{\Delta F_o} \\ a_C &= a_E + a_W + a_N + a_S + a_F + a_B + a_P + a_0 \end{aligned}$$

Here,  $K$  represents the thermal heat conduction ratio defined in article 1. The result is a sparse matrix system which is solved with a conjugated gradient method.

### 5.2.4 Solid Fraction Correction

The correction of the solid fraction is thoroughly explained in article 1. The Secant method will not be explained further here, but how this method is implemented in the program can be found in appendix A.7. A graphical interpretation of the overall thermal solution algorithm is presented in the appendix of article 1, and the reader is referred there.





(a) Node placement near boundary.  $e$  denotes a generic property (i.e.  $P$ ,  $f_S$ ,  $\theta$ , ect.).

(b) ghost cell placement

Figure 2: Property value near boundary (a) and position of ghost cells (b).

### 5.2.5 Boundary Conditions

This model utilizes one layer of ghost cells outside the domain in order to enforce the different boundary conditions. See figure 2b. This yields boundary conditions of the first order in space. Boundary conditions encountered in this paper can be divided into two types: Neumann (known gradient) and Dirichlet (known value). The Neumann boundaries are zero gradient for pressure at walls, zero gradient for temperature at thermally insulated walls and zero gradient for velocities at outlet and inlet boundary conditions where velocity is not necessarily known. The Dirichlet boundaries are no-slip and no penetration velocities at walls, known temperature and known pressure. Using the convention from figure 2a, zero gradient boundary conditions can be described by equation (10).

$$\frac{e_{domain} - e_{ghost}}{\Delta X} = 0 \quad (10)$$

$$e_{ghost} = e_{domain}$$

As an example the west node denoted  $W$  is a ghost node outside the domain. For a general linear system with zero gradient boundary, this will result in:

$$\begin{bmatrix} a_F & a_N & a_E & (a_C + a_W) & a_S & a_B \end{bmatrix} \begin{bmatrix} e_{h,k,l+1} \\ e_{h,k+1,l} \\ e_{h+1,k,l} \\ e_{h,k,l} \\ e_{h,k-1,l} \\ e_{h,k,l-1} \end{bmatrix} = Forcing\ term \quad (11)$$

Say both the west ( $W$ ) and south ( $S$ ) nodes are ghost nodes with zero gradient boundary, the linear system reduces to:

$$\begin{bmatrix} a_F & a_N & a_E & (a_C + a_W + a_S) & a_B \end{bmatrix} \begin{bmatrix} e_{h,k,l+1} \\ e_{h,k+1,l} \\ e_{h+1,k,l} \\ e_{h,k,l} \\ e_{h,k,l-1} \end{bmatrix} = Forcing\ term \quad (12)$$

This trend extends further if more neighbouring cells are ghost nodes with zero gradient boundary condition. Following the convention in figure 2a, the Dirichlet boundary states:

$$\frac{1}{2}(e_{domain} + e_{ghost}) = e_{wall} \quad (13)$$

$$e_{ghost} = 2e_{wall} - e_{domain}$$

As an example the west node denoted  $W$  is a ghost node outside the domain. For a general linear system this will result in:

$$\begin{bmatrix} a_F & a_N & a_E & (a_C - a_W) & a_S & a_B \end{bmatrix} \begin{bmatrix} e_{h,k,l+1} \\ e_{h,k+1,l} \\ e_{h+1,k,l} \\ e_{h,k,l} \\ e_{h,k-1,l} \\ e_{h,k,l-1} \end{bmatrix} = \text{Forcing term} - 2a_W e_{wall} \quad (14)$$

The extension is trivial if more neighbouring cells are a Dirichlet boundary. However, wall boundaries are simplified for pressure (zero gradient). The ghost cells are initially simply defined as dead fluid cells, and the live-dead fluid cell method described in article 1 is used. This upholds the zero gradient condition for pressure, but the the wall will in effect be half a computational cell outside the actual wall for the calculated velocities. Thus velocity boundary values at the wall for the FTCS scheme needs to be calculated. As a staggered grid has been used, all tangential velocities will follow the convention in figure 2a for the no-slip boundary. These are calculated according to (18) with the known tangential velocity at the wall equal zero. Due to the staggered grid, velocities normal to the wall are placed directly on the wall phase. To uphold the no penetration boundary these normal velocities on the wall are simply equal zero.

Constant pressure boundary is upheld by setting the pressure at the first computational cell inside the domain equal the known pressure. Knowing that the pressure correction should equal zero in this node, as pressure does not change, it is defined in the pressure correction sparse matrix by

$$\begin{bmatrix} 0 & 0 & 0 & a_C & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta P_{h,k,l+1} \\ \Delta P_{h,k+1,l} \\ \Delta P_{h+1,k,l} \\ \Delta P_{h,k,l} \\ \Delta P_{h-1,k,l} \\ \Delta P_{h,k-1,l} \\ \Delta P_{h,k,l-1} \end{bmatrix} = 0, \quad a_C = -10^{-8} \quad (15)$$

Notably, the method presented here needs no boundary conditions for the solid fraction. For cosmetic purposes a zero gradient boundary condition has been applied to the solid fraction ( $f_S$ ) at the walls. Code snippets are provided in appendix A.2 for the velocities and in appendix A.8 for the thermal boundary conditions. Keep in mind that boundary conditions for pressure and temperature are hard coded into the matrix-vector equations. The boundary conditions for temperature calculated in A.8 is purely cosmetic.

### 5.3 Conjugated Gradient and Storage Format

The conjugated gradient algorithm used here are in large part taken from the master thesis of A. Bøckmann [1]. The main changes done to the CG-subroutine is to be able to monitor iterations and residuals. Interested readers are also referred to [2]. By acknowledging that both the Poisson pressure correction matrix system (4) and the BTCS formulation of the energy equation (9) yields symmetric and positive definite matrices the conjugate gradient method is a viable solution strategy. The idea behind the Krylov subspace methods, which includes the CG-method, is to solve the matrix equation

$$Ax = b \quad (16)$$

by minimizing the function

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c \quad (17)$$

The CG algorithm only consists of a few line of code:

$$\begin{aligned} d_0 &= r_0 = b - Ax \\ \alpha_m &= \frac{r_m^T r_m}{d_m^T A d_m} \\ x_{m+1} &= x_m + \alpha_m d_m \\ r_{m+1} &= r_m - A d_m \\ \beta_{m+1} &= \frac{r_{m+1}^T r_{m+1}}{r_m^T r_m} \\ d_{m+1} &= r_{m+1} + \beta_{m+1} d_m \end{aligned} \quad (18)$$

Here, the iterations are ended when a maximum number of iterations have been reached or the  $L^1$  norm of the residual full fill some convergence criterion. An effect of this method is that if the eigenvalues of  $A$  is clustered, the algorithm converges faster. One common method to exploit this effect is to alter the system  $Ax = b$  by multiplying both sides by a matrix  $M^{-1}$ , which is sought to approximate  $A^{-1}$ . This is called preconditioning and several algorithms for finding the conditioner matrix  $M$  has been developed. Jacobi, Gauss-Seidel, Incomplete Cholesky factorization and Incomplete LU factorization are just a few. Though not implemented in this algorithm, much of the efficiency of the Krylov subspace methods lies in finding a good preconditioner matrix,  $M$ .

The largest sparse matrix systems generated during the simulations have been in the magnitude of 250000 by 250000. However, for such large systems, about 1 in 5000 entries of the matrix are occupied by non zero entries. Storing all entries in the sparse matrix is very inefficient. A compressed row storage format has thus been utilized. This format consists of one list of non-zero entries of the matrix  $A$ , one list indicating the row indices ( $A_i$ ) and another list for the column indices ( $A_j$ ). An example is provided in (19). This format might lead to some inefficiencies if one wants to extract a particular entry in  $A$  as a search through the row and column indices lists is necessary. However, The CG- algorithm only needs vector-vector and matrix-vector multiplications which is easily handled.

$$A = \begin{bmatrix} 0.0 & 2.8 & 5.2 & 0.0 \\ 0.0 & 3.6 & 0.0 & 2.5 \\ 7.1 & 8.5 & 1.9 & 0.0 \\ 0.0 & 0.0 & 0.0 & 2.6 \end{bmatrix} \quad A_i = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \\ 4 \end{bmatrix} \quad A_j = \begin{bmatrix} 2 \\ 3 \\ 2 \\ 4 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad (19)$$

How the sparse matrix-vector system for the pressure correction is generated can be found in appendix A.4 and how the matrix-vector system for the energy equation is generated can be found in appendix A.6. The CG-algorithm, matrix-vector and vector-vector multiplications are found in appendix A.9.

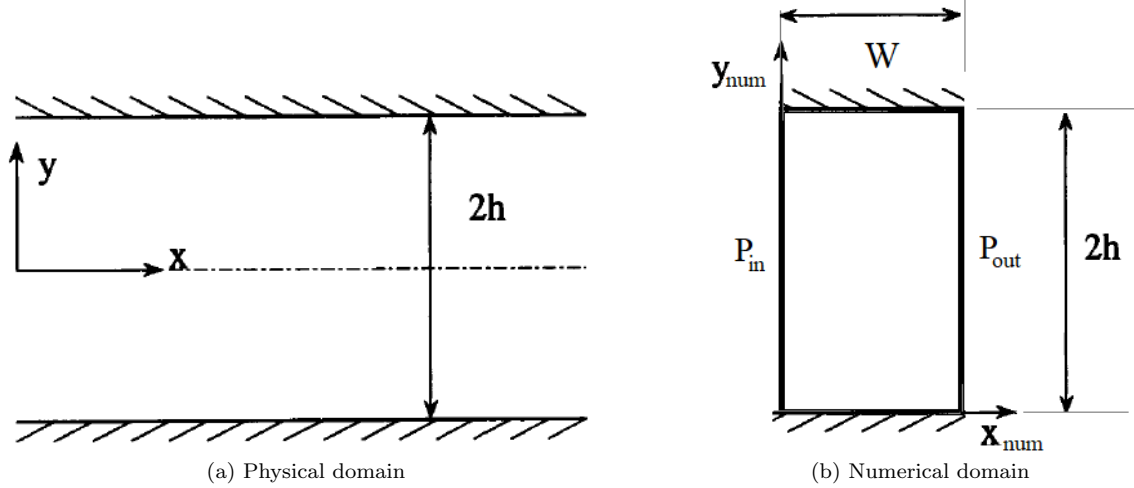


Figure 3: Physical (a) and numerical (b) domain.  $W$  represents channel length. Note the change in origo.

#### 5.4 Pressure Driven Channel

The pressure driven channel flow was used as an initial verification of the fluid model. Two parallel plates serves as the channel boundaries, while a constant pressure gradient in the  $x$  - direction is the driving force for the flow. The upper and lower walls have a no slip boundary for the velocity. This channel flow can be reduced to a 1D problem governed by (20) on dimensional form and (21) on non-dimensional form. An analytic solution to the 1D problem (21) is derived by R. Kristoffersen [3], and represented here by equation (22). This analytic solution is treated as a finite sum of 200 summations and compared to the simulated results.

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \frac{\partial^2 u}{\partial y^2} \quad (20)$$

With the scaling,

$$U = \frac{u}{u_{ref}}, \quad X = \frac{x}{h}, \quad Y = \frac{y}{h}, \quad P = \frac{p}{\rho u_{ref}^2}, \quad \tau = \frac{t\nu}{h^2},$$

the non-dimensional form of equation (20) can be written as

$$\frac{\partial U}{\partial \tau} = a + \frac{\partial^2 U}{\partial Y^2}, \quad a = -\frac{\partial P}{\partial X} Re \quad (21)$$

with corresponding analytic solution

$$U(Y, \tau) = 1 - Y^2 + \sum_{n=1}^{\infty} \frac{2a}{\lambda_n^3} (-1)^n \cos(\lambda_n Y) e^{-\lambda_n^2 \tau}, \quad \lambda = (2n - 1) \frac{\pi}{2} \quad a = 2 \quad (22)$$

The physical and numerical domains are shown graphically in figure 3. Numerically, this is solved as a 2D problem. Inlet and outlet have a constant pressures corresponding to the desired pressure gradient and the velocities are handled as zero gradient. At the walls however, the velocities are treated as a no-slip and no-penetration boundaries and the pressure as a zero gradient.

The simulation was conducted on a coarse 2 by 8 grid with a channel length of  $W = 0.5$  and a total channel height of  $2h = 2.0$ . Simulation time was set to  $\tau = 2.5$  with a time step of  $\Delta\tau = 10^{-4}$ . The inlet and outlet pressure was set to uphold  $a = Re(P_{in} - P_{out})/W = 2$ . Grid- and time step independence was insured. Resulting velocities are plotted in figure 4. Global conservation of mass ( $M_G$ , see article 1 and 2) stayed equal zero (smaller than the resolution of the floating point numbers) and residuals from the pressure-correction equation was kept below the convergence criterion of  $\epsilon_{PC} = 10^{-8}$ . The result was considered satisfactory.

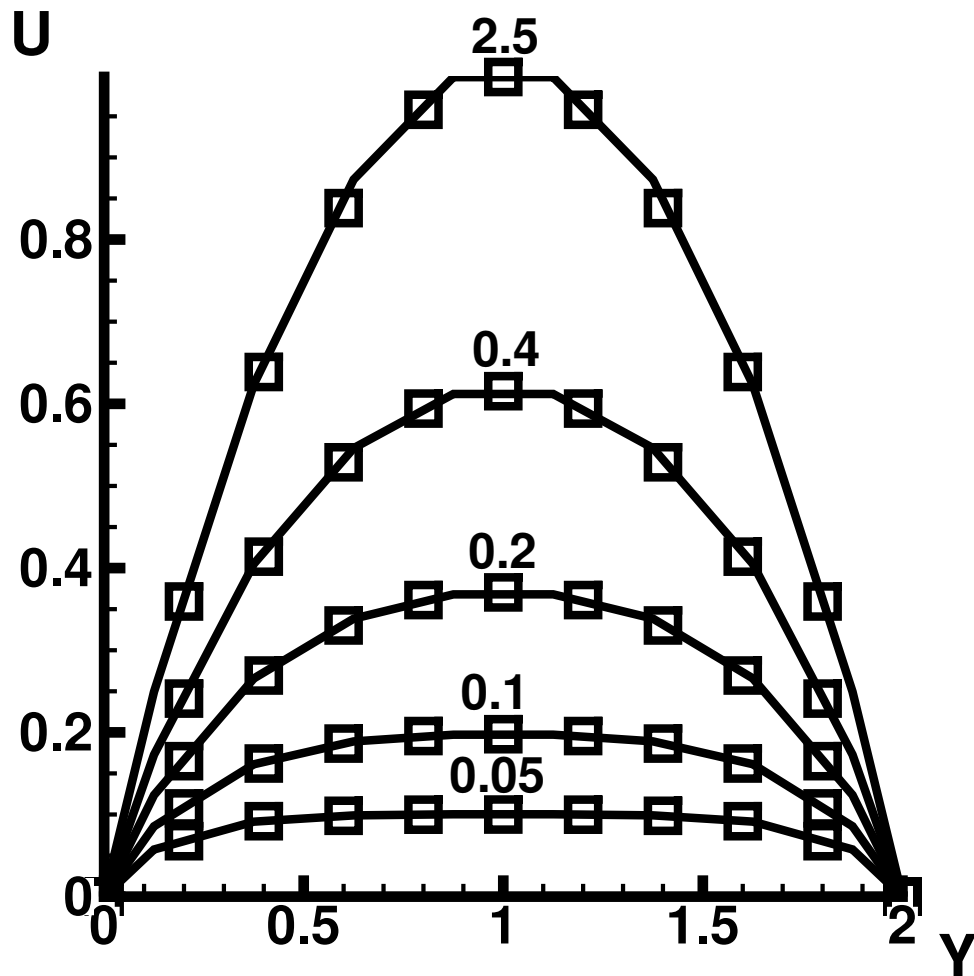


Figure 4: Resulting velocity profile at different time values indicated on the figure. Squares: Analytic, Line: Simulation. The  $Y$  values for the analytic solution is simply translated after calculation to correspond to the numerical domain

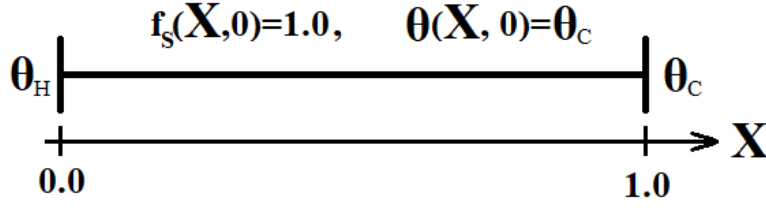


Figure 5: schematic representation of the calculation domain of the 1D unit bar.

## 5.5 Phase Transition of Mixtures

Both article 1 and article 2 deals with pure materials. For relatively pure materials the the inverse of the solid fraction - temperature relation ( $F(\theta)$ ) is simply equal the constant melting temperature. In both nature and industrial processes, pure materials are rarely encountered. For the phase transition algorithm derived in this thesis to be of any practical use, extension more complex phase transition behaviour needs to be addressed. Three different behaviours are studied in this section.

1. Linear ( $F_1(\theta)$ ): The substance gradually transitions with a linear relation from fully solid at  $\theta_S$  to fully liquid at  $\theta_L$ . From a numerical perspective, this should not have any significant difference from the pure case, as  $\xi()$  remains a first order polynomial.
2. Linear-Eutectic ( $F_2(\theta)$ ): At some constant temperature  $\theta_S$  the material melts as if it behaved like a pure material, but below some critical eutectic solid fraction  $f_e$ , the material melts with a linear relation with temperature until fully melted at  $\theta_L$ . Numerically, the interesting aspect is the transition between the two domains; eutectic transition (as if the material was a pure substance) to mushy zone transition (estimated as a linear relation).
3. Exponential ( $F_3(\theta)$ ): The substance gradually transitions from fully solid at  $\theta_S$  to fully liquid at  $\theta_L$  with an exponential relation. The mathematical description does not necessarily reflect nature, but from a numerical perspective the non-linear relation is an interesting test case for the algorithm.

The domain used to study the different solid fraction - temperature relations is a simple 1D unit length bar with constant temperature boundaries. Right boundary has a temperature  $\theta_C$  below, while the left wall has a temperature  $\theta_H$  above the fusion temperature range ( $\theta_S$  to  $\theta_L$ ). Initially the bar is fully solid and temperature is equal  $\theta_C$  in the whole domain. Any fluid movement is ignored, so the governing equations on non-dimensional form with constant physical properties is represented by the energy equation (23) and material dependent relation (24). The dimensional scaling is similar to that used in article 2.

$$\frac{\partial \theta}{\partial Fo} = \frac{\partial^2 \theta}{\partial X^2} + \frac{1}{Ste} \frac{\partial f_S}{\partial Fo} \quad (23)$$

$$f_S = F(\theta) \quad (24)$$

A schematic representation of the domain is presented by figure 5 and simulation parameters are listed in table 1.

As indicated in article 1 the inverse function of (24) is used during the simulation. Here, no specific care is taken towards modeling an actual physical substance other than being fully solid at  $\theta_S$  and fully liquid at  $\theta_L$ . The three inverse variations of (24) are defined as:

$$F_1^{-1}(f_S) = \theta_L + (\theta_S - \theta_L)f_S \quad (25)$$

$$F_2^{-1}(f_S) = \max(\theta_L + (\theta_S - \theta_L)\frac{f_S}{f_e}, \theta_S) \quad (26)$$

$$F_3^{-1}(f_S) = \theta_L + (\theta_S - \theta_L)\frac{e^{\beta f_S} - 1}{e^\beta - 1} \quad (27)$$

Parameter	Value
$Cells$	100
$Ste$	1.0
$\Delta Fo$	0.0001
$\theta_C$	0.0
$\theta_H$	1.0
$\theta_S$	0.2
$\theta_L$	0.8
$f_e$	0.5

Table 1: Simulation parameters for the 1D bar.

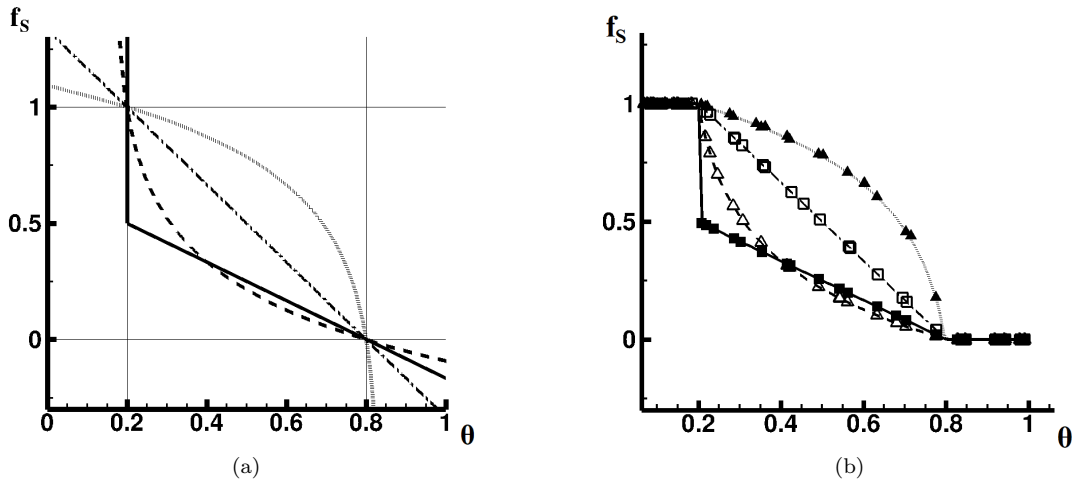


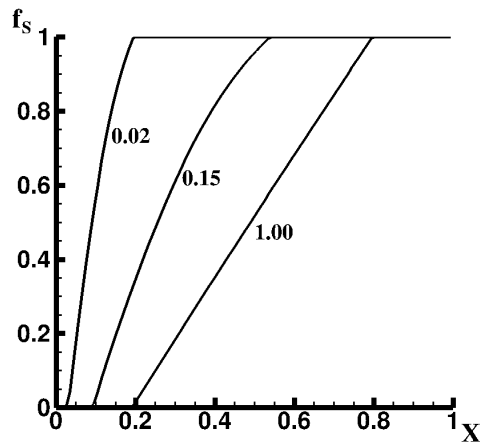
Figure 6: Solid fraction - Temperature relation. Dash-dot line/open square: linear, solid line/filled square: linear-eutectic, dashed line/open triangle: exponential ( $\beta = -3.0$ ), dotted line/filled triangle: exponential ( $\beta = 3.0$ ). 6a: Target  $f_S$ - $\theta$  plot. Phase transition domain indicated by horizontal and vertical lines. 6b: Simulated  $f_S$ - $\theta$  plot. Note how the solid fraction limiter ensures  $f_S$  in the range 0.0 - 1.0. Values are here taken at  $Fo = 0.02, 0.15$  and  $1.0$ . Not all nodes at all times are represented on the figure, as it would make the figure difficult to read.

Here,  $f_e$  indicates the eutectic solid fraction (26), while  $\beta$  controls the shape of the exponential relation (27). By l'Hôpital it can be shown that  $\lim_{\beta \rightarrow 0} F_3^{-1}(f_S) = F_1^{-1}(f_S)$ . The different relations are plotted in figure 6a. Note here that the only criterion is that the functions (25) to (27) passes through the points  $[f_S = 1.0, \theta = \theta_S]$  and  $[f_S = 0.0, \theta = \theta_L]$ , respectively indicating fully solid and fully liquid. How  $F^{-1}()$  is defined in the program has been listed in appendix A.10.

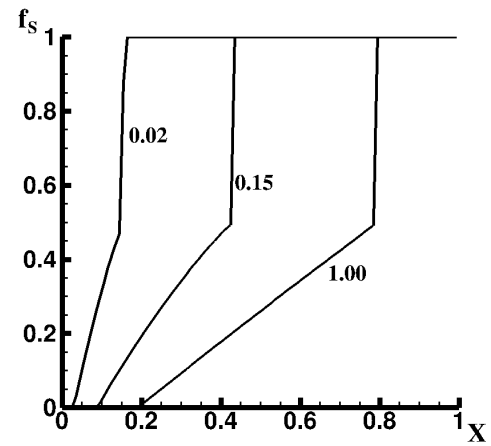
For the preliminary calculation of the temperature field,  $\hat{\theta}$ , (23) is discretized by a BTCS scheme. Following the algorithm outlined in article 1 the change in solid fraction is ignored during this step. The solid fraction correction is found using the pointwise Secant method as described in article 1.

As suspected, figure 8 indicates that there is some added resistance to the evolution of temperature when a material undergoes a change of phase. From (23) it is obvious that some of the energy is used to change the temperature, while some is used to change to solid fraction. However, the amount of resistance seem to greatly depend on the shape of function (24). Figure 6 shows the target and simulated values for the  $f_S$ - $\theta$  relation. The nodes seem to follow the solid fraction-temperature relation quite well.

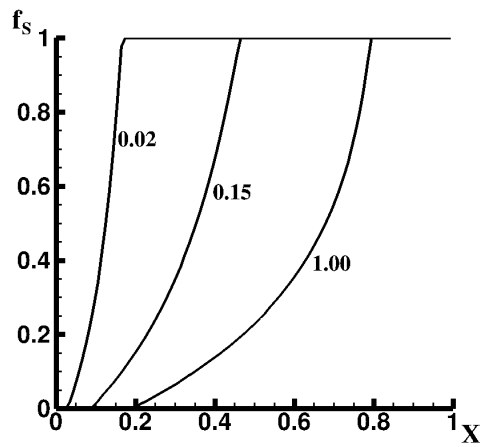
Both the linearly transitioning case (25) and the exponential transitioning (27) had no difficulty converging with the Secant method. However, the linear-eutectic case (26) showed some issues with converging. Both the purely transitioning and linearly transitioning parts of (26) by them selves only needs two iterations in order to find a convergent result. When transitioning from one domain to the other however, the algorithm needs some additional iterations in order for these nodes to reach convergence. As convergence is defined when a global convergence criterion is satisfied, this means that all nodes that



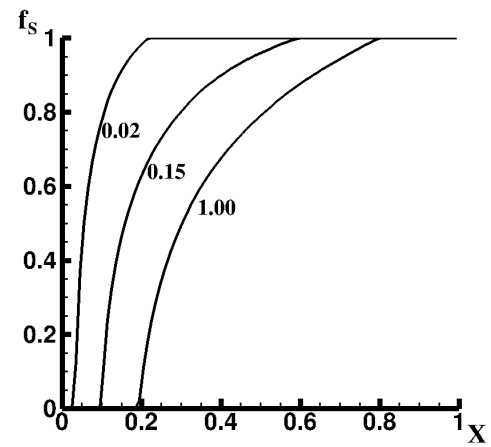
(a) Linear



(b) Linear-Eutectic



(c) Exponential ( $\beta = -3.0$ )



(d) Exponential ( $\beta = 3.0$ )

Figure 7: Evolution of the local solid fraction in a 1D bar. Dimensionless time ( $Fo$ ) is indicated on the figures



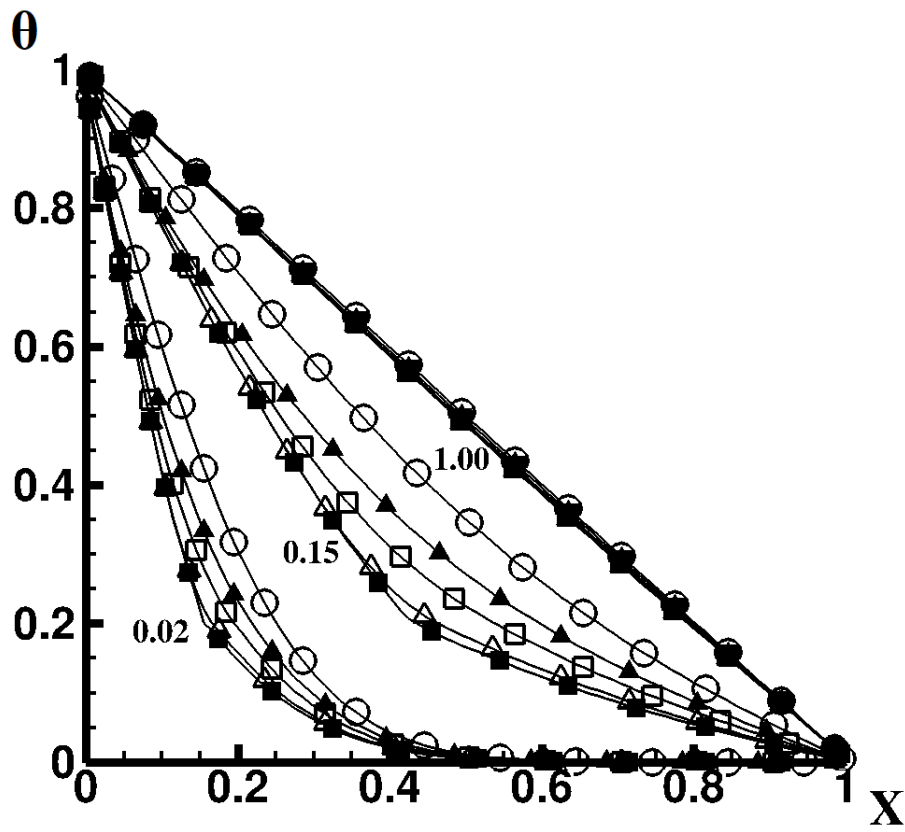


Figure 8: Temperature evolution in the bar. Dimensionless time ( $Fo$ ) is indicated on the figure. Open square: linear, filled square: linear-eutectic, open triangle: exponential ( $\beta = -3.0$ ), filled triangle: exponential ( $\beta = 3.0$ ), open circle: No phase change.

had converged still iterated to find better solutions. By it self this is not necessarily a problem, but the FORTRAN floating point numbers had some issues with dividing by close-to-zero values, which yielded strange results. This issue was simply solved by defining a local convergence criterion for each node and only iterating on nodes that had not satisfied this. The local convergence criterion was set to the global criterion divided by the number of cells. As a result, the algorithm need up to six iterations, and not two as in the linear case. However, this had no significant impact on simulation efficiency. Additionally, the algorithm was tested with a linear eutectic case with a Stefan number of  $Ste = 100$  and time step of  $Fo = 0.2$ . This test showed that the algorithm, from a numerical point of view, had no problem with transitioning several nodes from fully solid to fully liquid in one time step. The down side was that at some stages about 150 iterations was needed for the solid fraction correction and the simulation time was doubled.

## 6 Concluding Remarks

A fairly simple algorithm for solving convection dominated phase transitions has been derived and tested. The idea behind the phase transition algorithm is based on the Projection method used for solving Navier-Stokes, and should be familiar to those who have worked with CFD. The algorithm has been proven relatively accurate, efficient and robust. It has been expanded to 3D and yielded OK fit with experimental results. Further, the nature of the phase transition algorithm and not solving Navier-Stokes in the solid region had a significant impact on the computational efficiency. In addition, the algorithm have been proven able to handle discontinuities in the solid fraction - temperature relation, and able to fully transition several nodes in one time step.

Although the algorithm had its benefits, it still took a couple of days for a full 3D simulation on a  $50 \times 50 \times 75$  grid to finish. A large part of the computational time was spent on solving the pressure correction equation. One idea was to extend the model such that a multiphase flow could be handled. This could open the model to further use of industrial interest, such as mould filling during casting, shrinkage during phase transition, ect. The master thesis by A. Bøckmann [1] is an extensive report on a similar fluid model implemented with a level-set solver. Similarly, the conjugate gradient solver was identified as one of the aspects that needed improvement for a more efficient simulation. If the model is to be extended or larger calculation domain is required, it is recommended that the current matrix equation solver is revised. This is one of the reasons why the extensive sensitivity analysis in article 2 was conducted in 2D.

A short literature study was conducted on solid - liquid interaction. Due to the success of the zeroth order live-dead fluid cell method shown in article 1, no attempts were made at higher order interpolation scheme for a better resolution of the solid-liquid interphase. However, it would be an interesting extension. There are a lot of literature on the immersed boundary methods for solid liquid interaction on arbitrary shapes or with non-conformal meshes [4], [5], [6], [7], [8], [9].

As briefly mentioned in article 2, the process of solidification can be a quite complex process. As demonstrated in the literature, it was deemed difficult to get a consistent result for the solidifying cavity case [10]. Crystal precipitation, growth of dendrites and undercooling are some of the effects that can have a significant impact on the solution. This makes the over all transformation kinetics quite complex and was considered beyond the scope of this thesis. However, the Avrami-equation [11] have been identified as one possible way of expanding the solid fraction - temperature relation to include some of this behaviour.

A final note on further work would be to point out the two "missing" non-dimensional parameters, which was identified in section 5.1. Answering the questions outlined there could yield a deeper understanding of the physical process.

## References

- [1] A. Bøckmann. Efficiency improvement of numerical flow simulation. Master thesis, The Norwegian University of Science and Technology, Department of Energy and Process Engineering, 2007.
- [2] J.R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. 1994.
- [3] R. Kristoffersen. A navier-stokes solver using the multigrid method. Dissertation, The Norwegian Institute of Technology, Department of Applied Mechanics, Thermo- and Fluid Dynamics, 1994.
- [4] E. Khalili. Fluid-structure interaction and immersed boundary method for the compressible navier-stokes equations using high order methods. Phd thesis, The Norwegian University of Science and Technology, Department of Energy and Process Engineering, 2018.
- [5] A. A. Skøien. Cartesian grid methods for the compressible navier-stokes equations. Master thesis, The Norwegian University of Science and Technology, Department of Energy and Process Engineering, 2012.
- [6] Y. Tseng and J. H. Ferziger. A ghost cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192:593–623, 2003.
- [7] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.
- [8] D. Z. Noor, M. Chern, and T. Horng. An immersed boundary method to solve fluid-solid interaction problems. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.
- [9] D. V. Le, B. C. Khoo, and K. M. Lim. An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains. *Computer methods in applied mechanics and engineering*, 197:2119–2130, 2008.
- [10] C. Gau and R. Viskanta. Melting and solidification of a pure metal on a vertical wall. *Journal of Heat Transfer*, 108:174–181, February 1986.
- [11] D.A. Porter, K.E. Easterling, and M.Y. Sherif. *Phase Transformation in Metals and Alloys*. CRC press, third edition, 2009.

# A Code Snippets

Selected subroutines from the Fortran program are listed in this appendix

## A.1 Preliminary Velocities

```
SUBROUTINE VEL
! Purpose: Calculate tentative velocity field
! at current time level (FTCS)
USE FIELD
IMPLICIT NONE
DOUBLE PRECISION :: XFAC, YFAC, ZFAC, FUX, FUY, FUZ, FVX, FVY, FVZ, FWX, FWY, FWZ, &
VISU, VISV, VISW, DIFFE, DIFFW, DIFFN, DIFFS, DIFFF, DIFFB, DPX, DPY, DPZ

DOUBLE PRECISION, DIMENSION(IMAX, JMAX, KMAX) :: DELTAU, DELTAV, DELTAW

XFAC = 1.0/(DELX*DELX)
YFAC = 1.0/(DELY*DELY)
ZFAC = 1.0/(DELZ*DELZ)

DELTAW(:,:,:) = 0.0
DELTAV(:,:,:) = 0.0
DELTAW(:,:,:) = 0.0

DO I=2, IM1
DO J=2, JM1
DO K=2, KM1

! -----U-----
! Convective terms
FUX = ((U(I,J,K)+U(I+1,J,K))**2-(U(I-1,J,K)+U(I,J,K))**2)*0.25*RDX
FUY = ((V(I,J,K)+V(I+1,J,K))*(U(I,J,K)+U(I,J+1,K))-(V(I,J-1,K)&
+ V(I+1,J-1,K))*(U(I,J-1,K)+U(I,J,K)))*0.25*RDY
FUZ = ((W(I,J,K)+W(I+1,J,K))*(U(I,J,K)+U(I,J,K+1))-(W(I,J,K-1)&
+ W(I+1,J,K-1))*(U(I,J,K-1)+U(I,J,K)))*0.25*RDZ

! Viscous terms
DIFFN=U(I,J+1,K)-U(I,J,K)
DIFFS=U(I,J,K)-U(I,J-1,K)
DIFFE=U(I+1,J,K)-U(I,J,K)
DIFFW=U(I,J,K)-U(I-1,J,K)
DIFFF=U(I,J,K+1)-U(I,J,K)
DIFFB=U(I,J,K)-U(I,J,K-1)

! hardcode s-l BC
! North border
IF ((FNP(I,J+1,K).EQ.0).OR.(FNP(I+1,J+1,K).EQ.0)) THEN
DIFFN=-2*U(I,J,K)
ENDIF

! South border
IF ((FNP(I,J-1,K).EQ.0).OR.(FNP(I+1,J-1,K).EQ.0)) THEN
DIFFS=2*U(I,J,K)
ENDIF

! Front border
IF ((FNP(I,J,K+1).EQ.0).OR.(FNP(I+1,J,K+1).EQ.0)) THEN
DIFFF=-2*U(I,J,K)
ENDIF

! Back border
IF ((FNP(I,J,K-1).EQ.0).OR.(FNP(I+1,J,K-1).EQ.0)) THEN
DIFFB=2*U(I,J,K)
ENDIF
VISU=(DIFFE-DIFFW)*XFAC + (DIFFN-DIFFS)*YFAC + (DIFFF-DIFFB)*ZFAC

! Pressure forces
DPX=P(I+1,J,K)-P(I,J,K)

! -----V-----
! Convective terms
FVX = ((U(I,J,K)+U(I,J+1,K))*(V(I,J,K)+V(I+1,J,K))-(U(I-1,J,K)&
+ U(I-1,J+1,K))*(V(I-1,J,K)+V(I,J,K)))*0.25*RDX
FVY = ((V(I,J,K)+V(I,J+1,K))**2-(V(I,J-1,K)+V(I,J,K))**2)*0.25*RDY
FVZ = ((W(I,J,K)+W(I,J+1,K))*(V(I,J,K)+V(I,J,K+1))-(W(I,J,K-1)&
+ W(I,J+1,K-1))*(V(I,J,K-1)+V(I,J,K)))*0.25*RDZ

! Viscous terms
DIFFN=V(I,J+1,K)-V(I,J,K)
DIFFS=V(I,J,K)-V(I,J-1,K)
DIFFE=V(I+1,J,K)-V(I,J,K)
```

```

DIFFW=V(I,J,K)-V(I-1,J,K)
DIFFF=V(I,J,K+1)-V(I,J,K)
DIFFB=V(I,J,K)-V(I,J,K-1)

! hardcoded s-1 BC
IF ((FNP(I+1,J,K).EQ.0).OR.(FNP(I+1,J+1,K).EQ.0)) THEN !East border
    DIFFE=-2*V(I,J,K)
ENDIF
IF ((FNP(I-1,J,K).EQ.0).OR.(FNP(I-1,J+1,K).EQ.0)) THEN !West border
    DIFFW=2*V(I,J,K)
ENDIF

IF ((FNP(I,J,K+1).EQ.0).OR.(FNP(I,J+1,K+1).EQ.0)) THEN !Front border
    DIFFF=-2*V(I,J,K)
ENDIF
IF ((FNP(I,J,K-1).EQ.0).OR.(FNP(I,J+1,K-1).EQ.0)) THEN !Back border
    DIFFB=2*V(I,J,K)
ENDIF
VISV=(DIFFE-DIFFW)*XFAC + (DIFFN-DIFFS)*YFAC + (DIFFF-DIFFB)*ZFAC

! Pressure forces
DPY=P(I,J+1,K)-P(I,J,K)

! -----W-----
!Convective terms
FWX = ((U(I,J,K)+U(I,J,K+1))*(W(I,J,K)+W(I+1,J,K))-(U(I-1,J,K)&
+ U(I-1,J,K+1))*(W(I-1,J,K)+W(I,J,K)))*0.25*RDY
FWY = ((V(I,J,K)+V(I,J,K+1))*(W(I,J,K)+W(I,J+1,K))-(V(I,J-1,K)&
+ V(I,J-1,K+1))*(W(I,J-1,K)+W(I,J,K)))*0.25*RDY
FWZ = ((W(I,J,K)+W(I,J,K+1))*2-(W(I,J,K-1)+W(I,J,K))*2)*0.25*RDZ

!Viscous terms
DIFFN=W(I,J+1,K)-W(I,J,K)
DIFFS=W(I,J,K)-W(I,J-1,K)
DIFFE=W(I+1,J,K)-W(I,J,K)
DIFFW=W(I,J,K)-W(I-1,J,K)
DIFFF=W(I,J,K+1)-W(I,J,K)
DIFFB=W(I,J,K)-W(I,J,K-1)

! hardcoded s-1 BC
IF ((FNP(I+1,J,K).EQ.0).OR.(FNP(I+1,J,K+1).EQ.0)) THEN !East border
    DIFFE=-2*W(I,J,K)
ENDIF
IF ((FNP(I-1,J,K).EQ.0).OR.(FNP(I-1,J,K+1).EQ.0)) THEN !West border
    DIFFW=2*W(I,J,K)
ENDIF
IF ((FNP(I,J+1,K).EQ.0).OR.(FNP(I,J+1,K+1).EQ.0)) THEN !North border
    DIFFN=-2*W(I,J,K)
ENDIF
IF ((FNP(I,J-1,K).EQ.0).OR.(FNP(I,J-1,K+1).EQ.0)) THEN !South border
    DIFFS=2*W(I,J,K)
ENDIF
VISW=(DIFFE-DIFFW)*XFAC + (DIFFN-DIFFS)*YFAC + (DIFFF-DIFFB)*ZFAC

! Pressure forces
DPZ=P(I,J,K+1)-P(I,J,K)

! Bouynacy forces
FB=0.5*RA*PR*(TEMP(I,J,K)+TEMP(I,J+1,K)) ! y-dir

!----- Preliminary Velocities -----
IF ((FNP(I,J,K).EQ.1).OR.(FNP(I+1,J,K).EQ.1)) THEN
    DELTAU(I,J,K) = DELT*(-DPX*RDY - FUX - FUY - FUZ + PR*VISU)
ELSE
    DELTAU(I,J,K)=0.0
    U(I,J,K)=0.0
ENDIF

IF ((FNP(I,J,K).EQ.1).OR.(FNP(I,J+1,K).EQ.1)) THEN
    DELTAV(I,J,K)=DELT*(-DPY*RDY - FVX - FVY - FVZ + PR*VISV + FB)
ELSE
    DELTAV(I,J,K)=0.0
    V(I,J,K)=0.0
ENDIF

IF ((FNP(I,J,K).EQ.1).OR.(FNP(I,J,K+1).EQ.1)) THEN
    DELTAW(I,J,K)=DELT*(-DPZ*RDZ - FWX - FWY - FWZ + PR*VISW)
ELSE
    DELTAW(I,J,K)=0.0
    W(I,J,K)=0.0
ENDIF

ENDDO
ENDDO
ENDDO

```

```

U(:, :, :) = U(:, :, :) + DELTAU(:, :, :)
V(:, :, :) = V(:, :, :) + DELTAV(:, :, :)
W(:, :, :) = W(:, :, :) + DELTAW(:, :, :)

```

```

RETURN
END

```

## A.2 Velocity Boundary Conditions

```

SUBROUTINE BCVEL
!   purpose:
!   To give boundary conditions to
!   the velocities around the domain
USE FIELD
IMPLICIT NONE

DO I=1, IM1
  DO J=1, JM1
    DO K=1, KM1

      IF ((FNP(I, J, K).EQ.0).OR.(FNP(I+1, J, K).EQ.0)) THEN
        U(I, J, K) = 0.0
      ENDIF

      IF ((FNP(I, J, K).EQ.0).OR.(FNP(I, J+1, K).EQ.0)) THEN
        V(I, J, K) = 0.0
      ENDIF

      IF ((FNP(I, J, K).EQ.0).OR.(FNP(I, J, K+1).EQ.0)) THEN
        W(I, J, K) = 0.0
      ENDIF

      !At Boundary walls no slip, no pen
      IF (I.EQ.(IM1)) THEN !East wall
        U(I, J, K) = 0.0
        V(I+1, J, K) = -V(I, J, K)
        W(I+1, J, K) = -W(I, J, K)
      ENDIF
      IF (I.EQ.2) THEN !South wall
        U(I-1, J, K) = 0.0
        V(I-1, J, K) = -V(I, J, K)
        W(I-1, J, K) = -W(I, J, K)
      ENDIF
      IF (J.EQ.(JM1)) THEN !North wall
        U(I, J+1, K) = -U(I, J, K)
        V(I, J, K) = 0.0
        W(I, J+1, K) = -W(I, J, K)
      ENDIF
      IF (J.EQ.2) THEN !South wall
        U(I, J-1, K) = -U(I, J, K)
        V(I, J-1, K) = 0.0
        W(I, J-1, K) = -W(I, J, K)
      ENDIF
      IF (K.EQ.(KM1)) THEN !Front wall
        U(I, J, K+1) = -U(I, J, K)
        V(I, J, K+1) = -V(I, J, K)
        W(I, J, K) = 0.0
      ENDIF
      IF (K.EQ.2) THEN !Back wall
        U(I, J, K-1) = -U(I, J, K)
        V(I, J, K-1) = -V(I, J, K)
        W(I, J, K-1) = 0.0
      ENDIF

    ENDDO
  ENDDO
ENDDO
RETURN
END

```

## A.3 Solve the Fluid Equations

```

SUBROUTINE PITER
!   Purpose: To solve the Poissons equation of pressure
!   and adjust velocity and pressure fields so that
!   continuity is maintained for every time step.
USE FIELD
USE SOLVEFIELD
IMPLICIT NONE

```

```

!Calculate tentative velocity field
CALL VEL

! impose BC
CALL BCVEL

! main Momentum loop (not running if there are no fluid nodes)
IF (UCOUNT.GT.0) THEN

!Generate A matrix in Ax=b
CALL GEN_A_MAT(UCOUNT)

!Generate b vector in Ax=b
CALL GEN_B_VEC

!Solve system
CALL CGSOLVER(AVAL, AI, AJ, B, DELP, UCOUNT, ENTRIESA, ITER_P, &
RES_P, EPSI_P, ITMAX_P)

DO I=2, IM1-1 ! Updating U-components, if fluid node
DO J=2, JM1
DO K=2, KM1
IF ((FNP(I+1, J, K).EQ.1).AND.(FNP(I, J, K).EQ.1)) THEN
U(I, J, K) = U(I, J, K) + DELT*(DELP(NN(I, J, K)) &
- DELP(NN(I+1, J, K)))/DELX
ENDIF
ENDDO
ENDDO
ENDDO

DO I=2, IM1 ! Updating V-components, if fluid node
DO J=2, JM1-1
DO K=2, KM1
IF ((FNP(I, J+1, K).EQ.1).AND.(FNP(I, J, K).EQ.1)) THEN
V(I, J, K) = V(I, J, K) + DELT*(DELP(NN(I, J, K)) &
- DELP(NN(I, J+1, K)))/DELY
ENDIF
ENDDO
ENDDO
ENDDO

DO I=2, IM1 ! Updating W-components, if fluid node
DO J=2, JM1
DO K=2, KM1-1
IF ((FNP(I, J, K+1).EQ.1).AND.(FNP(I, J, K).EQ.1)) THEN
W(I, J, K) = W(I, J, K) + DELT*(DELP(NN(I, J, K)) &
- DELP(NN(I, J, K+1)))/DELZ
ENDIF
ENDDO
ENDDO
ENDDO

DO I=2, IM1 ! Updating pressure field, if fluid node
DO J=2, JM1
DO K=2, KM1
IF (FNP(I, J, K).EQ.1) THEN
P(I, J, K) = P(I, J, K) + DELP(NN(I, J, K))
ENDIF
ENDDO
ENDDO
ENDDO

! impose BC
CALL BCVEL

! Deallocate (size is changing)
DEALLOCATE(AVAL, AI, AJ)
DEALLOCATE(B, DELP)

ENDIF
RETURN
END

```

## A.4 Generate the Pressure Correction Matrix System

```

SUBROUTINE GEN_A_MAT(UNKNOWNNS)
! purpose:
! To construct the sparse coefficient matrix A
USE FIELD
IMPLICIT NONE

INTEGER :: UNKNOWNNS, N

```



```

DOUBLE PRECISION :: AP,AN,AS,AE,AW,AF,AB

! Find entries in A so that A can be allocated
N = 0

DO I=2,IM1
  DO J=2,JM1
    DO K=2,KM1

      !current cell is a fluid cell
      IF (FNP(I,J,K).EQ.1) THEN
        N = N + 1                                ! Add Center node

        IF(FNP(I+1,J,K).EQ.1) THEN              !fluid cell to the east
          N = N + 1                                ! Add East node
        ENDIF
        IF(FNP(I-1,J,K).EQ.1) THEN              !fluid cell to the west
          N = N + 1                                ! Add West node
        ENDIF
        IF(FNP(I,J+1,K).EQ.1) THEN              !fluid cell to the north
          N = N + 1                                ! Add North node
        ENDIF
        IF(FNP(I,J-1,K).EQ.1) THEN              !fluid cell to the south
          N = N + 1                                ! Add South node
        ENDIF
        IF(FNP(I,J,K+1).EQ.1) THEN              !fluid cell to the front
          N = N + 1                                ! Add Front node
        ENDIF
        IF(FNP(I,J,K-1).EQ.1) THEN              !fluid cell to the back
          N = N + 1                                ! Add Back node
        ENDIF

      ENDIF
    ENDDO
  ENDDO
ENDDO

ENTRIESA = N
ALLOCATE(AVAL(ENTRIESA),AI(ENTRIESA),AJ(ENTRIESA))
ALLOCATE(B(UNKNOWNNS),DELP(UNKNOWNNS))
DELP(:) = 0.0

N = 0

AE = 1/(DELX**2)
AW = 1/(DELX**2)
AN = 1/(DELY**2)
AS = 1/(DELY**2)
AF = 1/(DELZ**2)
AB = 1/(DELZ**2)
AP = -(AE+AW+AN+AS+AF+AB)

DO I=2,IM1
  DO J=2,JM1
    DO K=2,KM1
      !Current node is a fluid node
      IF (FNP(I,J,K).EQ.1) THEN
        !center node
        N = N + 1
        AVAL(N) = AP
        AI(N) = NN(I,J,K)
        AJ(N) = NN(I,J,K)

        !At Boundary wall/s-l interphase, grad(P)=0
        IF (FNP(I+1,J,K).EQ.0) THEN !East node
          AVAL(N) = AVAL(N) + AE
        ENDIF
        IF (FNP(I-1,J,K).EQ.0) THEN !West node
          AVAL(N) = AVAL(N) + AW
        ENDIF
        IF (FNP(I,J+1,K).EQ.0) THEN !North node
          AVAL(N) = AVAL(N) + AN
        ENDIF
        IF (FNP(I,J-1,K).EQ.0) THEN !South node
          AVAL(N) = AVAL(N) + AS
        ENDIF
        IF (FNP(I,J,K+1).EQ.0) THEN !Front node
          AVAL(N) = AVAL(N) + AF
        ENDIF
        IF (FNP(I,J,K-1).EQ.0) THEN !Back node
          AVAL(N) = AVAL(N) + AB
        ENDIF

        ! Neighbour nodes are fluid nodes
        IF(FNP(I+1,J,K).EQ.1) THEN !East Node

```

```

        N = N + 1
        Aval(N) = AE
        Ai(N) = NN(I,J,K)
        Aj(N) = NN(I+1,J,K)
    ENDIF

    IF (FNP(I-1,J,K).EQ.1) THEN !West Node
        N = N + 1
        Aval(N) = AW
        Ai(N) = NN(I,J,K)
        Aj(N) = NN(I-1,J,K)
    ENDIF

    IF (FNP(I,J+1,K).EQ.1) THEN !North Node
        N = N + 1
        Aval(N) = AN
        Ai(N) = NN(I,J,K)
        Aj(N) = NN(I,J+1,K)
    ENDIF

    IF (FNP(I,J-1,K).EQ.1) THEN !South Node
        N = N + 1
        Aval(N) = AS
        Ai(N) = NN(I,J,K)
        Aj(N) = NN(I,J-1,K)
    ENDIF

    IF (FNP(I,J,K+1).EQ.1) THEN !Front Node
        N = N + 1
        Aval(N) = AF
        Ai(N) = NN(I,J,K)
        Aj(N) = NN(I,J,K+1)
    ENDIF

    IF (FNP(I,J,K-1).EQ.1) THEN !Back Node
        N = N + 1
        Aval(N) = AB
        Ai(N) = NN(I,J,K)
        Aj(N) = NN(I,J,K-1)
    ENDIF

    ENDIF
ENDDO
ENDDO
ENDDO

! Setting the node at which the pressure will always be zero
PREFNODE = 1

DO N=1,ENTRIESA ! Setting reference pressure to zero
    IF((Ai(N).EQ.PREFNODE).AND.(Aj(N).NE.PREFNODE)) THEN
        Aval(N) = 0.0
    ENDIF
    IF((Ai(N).EQ.PREFNODE).AND.(Aj(N).EQ.PREFNODE)) THEN
        Aval(N) = -1.0*10**8
    ENDIF
ENDDO

RETURN
END

!-----
SUBROUTINE GEN_B_VEC
!   purpose:
!   To construct b in Ax=b
USE FIELD
IMPLICIT NONE

DOUBLE PRECISION :: DIV
INTEGER :: N

N=0

DO I=2,IM1
    DO J=2,JM1
        DO K=2,KM1

            ! Current node is a fluid node
            IF (FNP(I,J,K).EQ.1) THEN
                N=N+1
                DIV = RDY*(U(I,J,K)-U(I-1,J,K)) + RDY*(V(I,J,K)-V(I,J-1,K)) &
                    + RDZ*(W(I,J,K)-W(I,J,K-1))
                B(NN(I,J,K))=DIV/(DELT)
            ENDIF
        
```

```

        ENDDO
    ENDDO
ENDDO

! Assigning reference pressure 0 to pressure reference cell
B(PREFNODE) = 0.0

RETURN
END

```

## A.5 Solve the Thermal Equations

```

SUBROUTINE TITER
!   Purpose: Calculate temperature field
!   at current time level
USE FIELD
USE SOLVEFIELD
IMPLICIT NONE

!Store previous time step thermal fields
TEMP_P=TEMP
FS_P=FS

! second starting guess of secant method
FS_N=FS+0.1

!Generate system matrix
CALL GEN_T_MAT

!generate b-Vector
CALL GEN_TB_VEC

!Calculate tentative Temperature field
CALL CGSOLVER(TAVAL, TI, TJ, TB, TEMP_VEC, UCOUNT_T, &
              ENTRIESA_T, ITER_T, RES_T, EPSI_T, ITMAX_T)

!Update tentative temperature field
DO I=2, IM1
  DO J=2, JM1
    DO K=2, KM1
      TEMP(I,J,K) = TEMP_VEC(NNT(I,J,K))
    ENDDO
  ENDDO
ENDDO

!Solid fraction Correction
CALL CORR_FS

!Correct Temperature field
DO I=2, IM1
  DO J=2, JM1
    DO K=2, KM1
      TEMP(I,J,K) = TEMP(I,J,K) + (FS(I,J,K) - FS_P(I,J,K))/STE
    ENDDO
  ENDDO
ENDDO

! Impose BC
CALL BCTEMP

RETURN
END

```

## A.6 Generate the Temperature Matrix System

```

SUBROUTINE GEN_T_MAT
!   purpose:
!   To construct the sparse coefficient matrix A_T
!   for solving temperature
USE FIELD
IMPLICIT NONE
INTEGER :: N
DOUBLE PRECISION :: APO, AP, AN, AS, AE, AW, AF, AB

N = 0

DO I=2, IM1
  DO J=2, JM1
    DO K=2, KM1

      APO = 1.0/DELT
      AE = 0.5*U(I,J,K)/DELX - KON(I,J,K)/DELX**2 &

```

```

      - (KON(I+1,J,K)-KON(I-1,J,K))/(4*DELX**2)
AW = -0.5*U(I-1,J,K)/DELX - KON(I,J,K)/DELX**2 &
      + (KON(I+1,J,K)-KON(I-1,J,K))/(4*DELX**2)
AN = 0.5*V(I,J,K)/DELY - KON(I,J,K)/DELY**2 &
      - (KON(I,J+1,K)-KON(I,J-1,K))/(4*DELY**2)
AS = -0.5*V(I,J-1,K)/DELY - KON(I,J,K)/DELY**2 &
      + (KON(I,J+1,K)-KON(I,J-1,K))/(4*DELY**2)
AF = 0.5*W(I,J,K)/DELZ - KON(I,J,K)/DELZ**2 &
      - (KON(I,J,K+1)-KON(I,J,K-1))/(4*DELZ**2)
AB = -0.5*W(I,J,K-1)/DELZ - KON(I,J,K)/DELZ**2 &
      + (KON(I,J,K+1)-KON(I,J,K-1))/(4*DELZ**2)
AP = 2*KON(I,J,K)*(1.0/DELX**2 + 1.0/DELY**2 + 1.0/DELZ**2)

! center node
N = N + 1
TAVAL(N) = APO + AE + AW + AN + AS + AF + AB + 2*AP
TI(N) = NNT(I,J,K)
TJ(N) = NNT(I,J,K)

!At Boundary walls, grad(T)=0
IF (J.EQ.(JM1)) THEN !North node
  TAVAL(N) = TAVAL(N) + AN
ENDIF
IF (J.EQ.2) THEN !South node
  TAVAL(N) = TAVAL(N) + AS
ENDIF
IF (K.EQ.(KM1)) THEN !Front node
  TAVAL(N) = TAVAL(N) + AF
ENDIF
IF (K.EQ.2) THEN !Back node
  TAVAL(N) = TAVAL(N) + AB
ENDIF

!At Boundary walls, T=Known
IF (I.EQ.(IM1)) THEN !East node
  TAVAL(N) = TAVAL(N) - AE
ENDIF
IF (I.EQ.2) THEN !West node
  TAVAL(N) = TAVAL(N) - AW
ENDIF

! Neighbour nodes in domain
IF(I.LT.IM1) THEN !East Node
  N = N + 1
  TAVAL(N) = AE
  Ti(N) = NNT(I,J,K)
  Tj(N) = NNT(I+1,J,K)
ENDIF

IF(I.GT.2) THEN !West Node
  N = N + 1
  TAVAL(N) = AW
  Ti(N) = NNT(I,J,K)
  Tj(N) = NNT(I-1,J,K)
ENDIF

IF(J.LT.JM1) THEN !North Node
  N = N + 1
  TAVAL(N) = AN
  Ti(N) = NNT(I,J,K)
  Tj(N) = NNT(I,J+1,K)
ENDIF

IF(J.GT.2) THEN !South Node
  N = N + 1
  TAVAL(N) = AS
  Ti(N) = NNT(I,J,K)
  Tj(N) = NNT(I,J-1,K)
ENDIF

IF(K.LT.KM1) THEN !Front Node
  N = N + 1
  TAVAL(N) = AF
  Ti(N) = NNT(I,J,K)
  Tj(N) = NNT(I,J,K+1)
ENDIF

IF(K.GT.2) THEN !Back Node
  N = N + 1
  TAVAL(N) = AB
  Ti(N) = NNT(I,J,K)
  Tj(N) = NNT(I,J,K-1)
ENDIF

TEMP_VEC(NNT(I,J,K))=TEMP(I,J,K)

ENDDO

```

```

ENDDO
ENDDO

RETURN
END

!-----

SUBROUTINE GEN_TB_VEC
!   purpose:
!   To construct b in Ax=b
USE FIELD
IMPLICIT NONE

DOUBLE PRECISION :: APO,AE,AW

TB(:)=0.0
APO = 1.0/DELT

DO I=2,IM1
  DO J=2,JM1
    DO K=2,KM1
      TB(NNT(I,J,K))= APO*TEMP_P(I,J,K)

      !At Boundary walls, T=known
      IF (I.EQ.IM1) THEN !East node
        AE = 0.5*U(I,J,K)/DELX - KON(I,J,K)/DELX**2 &
          - (KON(I+1,J,K)-KON(I-1,J,K))/(4*DELX**2)
        TB(NNT(I,J,K)) = TB(NNT(I,J,K)) - 2*AE*TC

      ENDIF
      IF (I.EQ.2) THEN !West node
        AW = -0.5*U(I-1,J,K)/DELX - KON(I,J,K)/DELX**2 &
          + (KON(I+1,J,K)-KON(I-1,J,K))/(4*DELX**2)
        TB(NNT(I,J,K)) = TB(NNT(I,J,K)) - 2*AW*TH

      ENDIF
    ENDDO
  ENDDO
ENDDO

RETURN
END

```

## A.7 Solid Fraction Correction

```

SUBROUTINE CORR_FS
!   purpose:
!   Correct solid fraction
USE FIELD
USE SOLVEFIELD
IMPLICIT NONE

DOUBLE PRECISION :: VAL1, VAL2, F_INV_PURE

! initial guesses for Xi
DO I=2,IM1
  DO J=2,JM1
    DO K=2,KM1
      XI(I,J,K) = FS(I,J,K) - FS_P(I,J,K) &
        - STE*(F_INV_PURE(FS(I,J,K),TM) - TEMP(I,J,K))
      XI_N(I,J,K) = FS_N(I,J,K) - FS_P(I,J,K) &
        - STE*(F_INV_PURE(FS_N(I,J,K), TM) - TEMP(I,J,K))

    ENDDO
  ENDDO
ENDDO

! Secant method
DO ITER_TT=1,ITMAX_TT
  ! reset residual
  RES_TT=0.0

  DO I=2,IM1
    DO J=2,JM1
      DO K=2,KM1
        ! Hold values for next iteration
        VAL1=FS(I,J,K)
        VAL2=XI(I,J,K)

        ! next value
        FS(I,J,K) = FS(I,J,K) - XI(I,J,K)*&
          (FS(I,J,K)-FS_N(I,J,K))/(XI(I,J,K)-XI_N(I,J,K))
        XI(I,J,K) = FS(I,J,K) - FS_P(I,J,K) &
          - STE*(F_INV_PURE(FS(I,J,K),TM) - TEMP(I,J,K))

      ENDDO
    ENDDO
  ENDDO
ENDDO

```

```

! Replace old values
FS_N(I,J,K) = VAL1
XI_N(I,J,K) = VAL2

! residual
RES_TT = RES_TT + ABS(XI(I,J,K)-XI_N(I,J,K))
ENDDO
ENDDO
ENDDO

! Convergence test
IF (RES_TT.LT.EPSI_TT) THEN
EXIT
ENDIF

ENDDO

! limit Solid Fraction
DO I=2,IM1
DO J=2,JM1
DO K=2,KM1
FS(I,J,K) = MIN( 1.0 , MAX( 0.0 , FS(I,J,K) ) )
ENDDO
ENDDO
ENDDO

RETURN
END

```

## A.8 Thermal Boundary Conditions

```

SUBROUTINE BCTEMP
! purpose:
! To give boundary conditions to the temp and fs
USE FIELD
IMPLICIT NONE

!Impose boundary condition on X-planes
DO J=1,JMAX
DO K=1,KMAX
!Left
TEMP(1,J,K) = 2*TH - TEMP(2,J,K)
FS(1,J,K) = FS(2,J,K)
!Right
TEMP(IMAX,J,K) = 2*TC - TEMP(IMAX-1,J,K)
FS(IMAX,J,K)=FS(IM1,J,K)
ENDDO
ENDDO

!Impose boundary condition on Y-planes
DO I=1,IMAX
DO K=1,KMAX
!Bottom
TEMP(I,1,K)=TEMP(I,2,K)
FS(I,1,K)=FS(I,2,K)
!Top
TEMP(I,JMAX,K)=TEMP(I,JM1,K)
FS(I,JMAX,K)=FS(I,JM1,K)
ENDDO
ENDDO

!Impose boundary condition on Z-planes
DO I=1,IMAX
DO J=1,JMAX
!Back
TEMP(I,J,1)=TEMP(I,J,2)
FS(I,J,1)=FS(I,J,2)
!Front
TEMP(I,J,KMAX)=TEMP(I,J,KM1)
FS(I,J,KMAX)=FS(I,J,KM1)
ENDDO
ENDDO

RETURN
END

```

## A.9 Conjugated Gradient

```

SUBROUTINE CGSOLVER(Aval,Ai,Aj,b,x,unknowns,&
entriesA,iter,res,epsi,itmax)
! Solves the system Ax=b with the conjugate gradients method

IMPLICIT NONE

```

```

INTEGER :: unknowns, entriesA, itmax, iter
DOUBLE PRECISION, DIMENSION (entriesA) :: Aval
INTEGER, DIMENSION (entriesA) :: Ai, Aj
DOUBLE PRECISION, DIMENSION (unknowns) :: tempvec, x, r, d, b

DOUBLE PRECISION :: alpha, beta, tempscal1, tempscal2, epsi, res

iter=0

x(:) = 0.0
r(:) = b(:)
d(:) = r(:)
res = sum(abs(r))

! Convergence test
IF(res.LT.epsi) THEN
    RETURN
ENDIF

DO iter=1, itmax
    ! (r'*r)
    CALL VECVEC(r,r,tempscal1,unknowns)

    ! (A*d)
    CALL MATVEC(Aval,Ai,Aj,d,tempvec,unknowns,entriesA)

    ! (d'*A*d)
    CALL VECVEC(d,tempvec,tempscal2,unknowns)

    ! alpha = (r'*r)/(d'*A*d)
    alpha = tempscal1/tempscal2

    ! x = x + alpha*d
    x(:) = x(:) + alpha*d(:)

    ! r_new = r - alpha*A*d
    tempvec(:) = r(:) - alpha*tempvec(:)

    ! (r_new'*r_new)
    CALL VECVEC(tempvec,tempvec,tempscal2,unknowns)

    ! beta = (r_new'*r_new)/(r'*r)
    beta = tempscal2/tempscal1

    ! r = r_new
    r(:) = tempvec(:)
    ! d = r + beta*d
    d(:) = r(:) + beta*d(:)

    ! Convergence test
    res = sum(abs(r))
    IF(res.LT.epsi) THEN
        EXIT
    ENDIF
ENDDO

RETURN
END

```

!-----

```

SUBROUTINE MATVEC(Aval,Ai,Aj,u,x,m,n)
! Matrix-vector multiplication, x=Au
! A: matrix
! u: vector
! x: return vector
! m: length of u
! n: entries in A
IMPLICIT NONE
INTEGER :: i, m, n
DOUBLE PRECISION, DIMENSION(n) :: Aval
INTEGER, DIMENSION(n) :: Ai, Aj
DOUBLE PRECISION, DIMENSION(m) :: u,x

x(:) = 0
DO i=1,n
    x(Ai(i)) = x(Ai(i)) + Aval(i)*u(Aj(i))
ENDDO

```

```

RETURN
END

```

!-----

```

SUBROUTINE VECVEC(u,v,x,m)

```

```

! Vector-vector multiplication, x=uv
! u: vector1
! v: vector2
! x: return scalar
! m: size of u and v
IMPLICIT NONE
INTEGER :: i, m
DOUBLE PRECISION, DIMENSION(m) :: u, v
DOUBLE PRECISION :: x

x = 0.0
DO i=1,m
  x = x + u(i)*v(i)
ENDDO

RETURN
END

```

## A.10 Solid Fraction Temperature Relations

```

FUNCTION F_INV(x,ys,y1)
! Inverse of solid fraction temperature relation for substance
! Here: linear
! x = Solid fraction
! y = Melting temperature (s-solidus, l-liquidus)

DOUBLE PRECISION :: F_INV,x,ys,y1

F_INV = y1 + (ys - y1)*x

RETURN
END FUNCTION F_INV

```

```

-----
FUNCTION F_INV(x,xs,ys,y1)
! Inverse of solid fraction temperature relation for substance
! Here: linear eutectic
! x = Solid fraction, (e-eutectic)
! y = Melting temperature (s-solidus, l-liquidus)

DOUBLE PRECISION :: F_INV,x,xs,ys,y1

F_INV = max(y1-x/xs*(y1-ys),ys)

RETURN
END FUNCTION F_INV

```

```

-----
FUNCTION F_INV(x,xs,ys,y1)
! Inverse of solid fraction temperature relation for substance
! Here: exponential
! x = Solid fraction
! y = Melting temperature (s-solidus, l-liquidus)
! eul = euler's number
! del = phase transition parameter
REAL, PARAMETER :: eul=2.71828182845904, del=-3.0
DOUBLE PRECISION :: F_INV,x,xs,ys,y1

F_INV = y1 + (ys - y1)*(eul**(del*x) - 1.0)/(eul**(del) - 1.0)

RETURN
END FUNCTION F_INV

```

```

-----
FUNCTION F_INV_PURE(x,ym)
! inverse of solid fraction temperature relation for pure substance
! x = Solid fraction
! ym = Melting temperature
DOUBLE PRECISION :: F_INV_PURE,x,ym

F_INV_PURE = ym

RETURN
END FUNCTION F_INV_PURE

```