Master thesis : Implementation of a melting/solidification process with the enthalpy method

Auteur : Lopez Sahuquillo, Jaime
Promoteur(s) : Terrapon, Vincent
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil en aérospatiale, à finalité spécialisée en "aerospace engineering"
Année académique : 2016-2017
URI/URL : http://hdl.handle.net/2268.2/3274

University of Liege

Faculty of Applied Sciences

Master Thesis

Graduation Studies conducted for obtaining the Master's degree in Aerospace Engineering by JAIME LÓPEZ

Implementation in OpenFOAM of a Melting/Solidification Process with the Enthalpy Method

Industrial Tutor: Salvador Lucas

Academic advisor: Terrapon Vincent

Jury members: Ponthot Jean-Philipe, Terrapon Vincent, Salvador Lucas, Dewallef Pierre

Year 2016-2017

Dedicado a
mi familia

Ackowledgements

First, I would like to thank both, my tutor in the company, Lucas Salvador, and my academic advisor at the university Vincent Terrapon. To Lucas Salvador, I would like to acknowledge his commitment, the really long chats, corrections and advising meetings that we had and, in definitely, his always helping hand attitude. On the other hand, I would like to thank Mr. Terrrapon for his advice and his wisdom that I hope I manage to reflect in this project.

Furthermore, I would like to say thank you to all the friends and people that I met here in this two years in Belgium. A special mention to Jon for all his help and explanation on the OpenFOAM software and in general CFD aspects.

At last, I would like to give a special mention to my parents, who had given me everything. And to my grandfather, that at his age, he was still able to read my Thesis, correct it, and give lessons about a subject that is not even of his field of expertise...

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11.

Abstract

Phase change materials (PCMs) are of high interest in thermal storage and thermal management applications for the earth and for space environments. Nevertheless, their functionality is intrinsically attached to phase change processes, which, from experience, it is known that they are computationally challenging. The present project arises with the intention to give a numerical solution to this problematic.

A solver based on the enthalpy-porosity technique, capable to deal with diffusive-convective phase change has been adapted for `OpenFOAM 4.1`. For the implementation of the enthalpy technique, the work of Voller [1] has been closely followed, and a detailed explanation of the equations employed and the assumptions that support them is given. Furthermore, the numerical approach is also specified, with a close attention to the discretization process based on the Finite Volume Method (FVM). The solver algorithm is provided with a deep explanation of its implementation in `OpenFOAM`. Furthermore, an analysis of the convergence of the numerical solution is provided.

Moreover, the works of several authors [2][3][4], have been employed to help in some aspects of the implementation and validation of the solver. As part of this validation, the controversial case of the melting of pure Gallium in a rectangular cavity is computed with the `OpenFOAM` solver. The author gives some discussion about the results obtained and compares them with the existing literature in order to assess the accuracy of the mathematical model employed.

The last part of the project employs the customize solver to analyze the thermal behaviour of a PCM during melting. Three different cases are proposed and tested for two different geometries: one under gravity conditions, where natural convection is part of the heat transfer process, and another two independent of gravity or proper of micro-gravity environments: a pure conductive case and a case with Bénard-Marangoni convection.

Keywords: Enthalpy-porosity technique, solver, OpenFOAM, Finite-Volume method PCMs, Marangoni convection.

Contents

Nomenclature

Parameters

$\vec{g}$	gravity
$\vec{S}$	Surface vector
S	Surface module (scalar)
$\vec{u}$	Velocity
a_p	Coefficient depending on the convective and diffusive fluxes
b_p	Coefficient depending on the ϕ in previous iterations
c_p	Specific heat
f	Liquid fraction
H	Total enthalpy (sensible plus latent)
h	Sensible heat
$\Delta H = L$	Latent heat
L_c	Characteristic length
u_c	Characteristic velocity
t_c	Characteristic time
m	Mass
P	Static pressure
p	pressure per unit of density
Q	Heat
T	Temperature
T_m	Melting point
V_P	Computational volume
S_h, S_u, S_b	Source terms
A	Porosity function
$\vec{x}, \vec{y}, \vec{z}$	Vectors of the Cartesian space

Greek symbols

α	Thermal diffusivity
β	Thermal coefficient of expansion
ϵ	Half mushy range
λ	Porosity
μ	Kinematic viscosity
ν	Dinamic viscosity
ρ	Density
κ	Thermal conductivity
ϕ	Fluid variable in FVM
Γ_ϕ	Diffusion coefficient of ϕ
σ	Surface stress

Dimensionless numbers

Bo	Bond number
Gr	Grashof number
Ma	Marangoni number
Nu	Nusselt number
Pr	Prandtl number
Ra	Rayleigh number
St	Strouhal number
Ste	Stefan number
Re	Reynolds number

Abbreviations

PCM	Phase Change Material
HTF	Heat Transfer Fluid
CFD	Computational Fluid Dynamics
FVM	Finite Volume Method
FEM	Finite Element Method
FDM	Finite Difference Method
CV	Control Volume
TES	Thermal Storage System
SHTS	Sensible Heat Storage System
LHTS	Latent Heat Thermal Storage System
DT	Darcy Term

Subscripts

l	Property of the fluid in liquid phase
s	Property of the fluid in solid phase
nb	neighbour value
ref	reference value
f	Value at the face of the cell
p	value at the center of the cell

1 Introduction

1.1 Objectives and Motivation

During the last few decades, the use of phase change materials (PCMs) in the industry has been growing either in earth applications as in space ones. This is mainly due to the high latent heat that they perform and their ability to release heat at a constant temperature [13], what makes these materials prone to be used as heat/energy storage or as a system to protect other devices from extreme temperatures that could compromise their correct functionality. However, since the PCMs are meant to be used at melting (or freezing) temperatures in order to take advantage of their thermal capabilities, to get accurate results in numerical simulations, the phase change and its corresponding moving boundary, should be taken into account.

The first time moving interface problems were referred, was in the late 19^{th} century by J.Stefan. Stefan was an Austrian-Slovenian physicist who formulated the problem of obtaining the temperature distribution, and the progress of the freezing front, of a solidifying portion of water [14]. After him, and increasingly during the last thirty years, moving boundary problems and its challenging computation has been studied by a large number of scientists and engineers delivering a great number of techniques, many of them summarized by Crank [15]. Nevertheless, the majority of these techniques are suited for conduction driving problems, while in problems with a liquid-solid interface, convection also plays an important role in the heat transfer. As such, the problem combining both heat transfer phenomena has received deeper attention in the last decades.

In the year 1987, V.R. Voller was one of the first in describing an enthalpy-porosity method to accomplish numerical calculation for convective-conductive moving boundary problems [2]. This enthalpy-porosity method, although still rather complex, emerge as one of the simplest and more effective methods to account this kind of problems and it is the technique employed during the present project to make the different calculations.

Once the general frame of the present project has been introduced, show the objectives pursued on it is the next step. Two main subjects are the aim of this project:

1. **Develop and validate a solver base on the enthalpy porosity technique for the software** `OpenFOAM`. The solver should be capable of deliver accurate solutions for conductive phase changes, proper of micro-gravity environments , and convective cases, more often found in the earth.

2. **Study and discuss melting/solidification processes** in different geometries and under different physical circumstances. And also understand the heat transfer mechanisms involved in each case.

Beyond this, the project also intends to reach some other objectives, which, although more academical, they still play an important role in the project:

3. **Understand what is a PCM and what are the potential applications**, both in space and earth. And also understand the relation of this materials with the need of developing a solver for phase change problems.

4. **Understand the fundamentals of the enthalpy-posority-technique** including the underlying equations and their treatment in order to model phase change processes.

5. **Understand the discretization process and the numerical methods employed** to obtain the solution of the physical problem. A big importance is given to the solving algorithm

1.2 Project organization

This Master thesis is composed by a total amount of seven chapters plus appendix and bibliography. Likewise, these seven chapters are grouped in three big blocks. The first block, formed by this chapter and the second one, deal with theoretical background and the framework of the project, introducing the PCMs, the moving boundary problem and the relationship among them. The second block is composed of chapters three and four where mathematical and numerical aspects are discussed including some key developments in the understanding of the project. Then, the third block is formed by chapters six and seven were the results, discussion and conclusion of the computations is performed.
Subsequently, a brief description of all the chapters is presented, noticing that their organization matches, in most cases, with the objectives planned:

- **Chapter one,** the present one, works as an introduction giving the frame of the project along with the organization.

- **Chapter two,** deals with the materials studied on the project, PCMs, and gives information about applications, types and performance.

- **Chapter three,** gives information about the approach followed to model the problem and formal background of the physics involved in form of equations. Some key concepts are explained here, as it is the enthalpy-porosity method for phase change processes.

- **Chapter four,** deals with all the discretization process of the equations presented in chapter three, and the numerical methods employed to solve the melting/solidification process. Also details deeply the implementation of the problem and the customized solver in `OpenFOAM`, and a description of the solver algorithm. Moreover, a thorough study on the solution convergence is presented.

- **Chapter five,** this chapter contains simulations accomplished with the solver and compared with the literature in order to asses the validity of the presented solver. Furthermore, a deeper understanding of the melting/solidification process is reached from the results.

- **Chapter six,** where after the validation of the solver, simulations are performed to obtain information and results on the influence of the melting process for different geometries and taking into account different heat transfer mechanisms.

- **Chapter seven,** presents the conclusion of the project in relation to the objectives. Furthermore, it shows future lines of work in forthcoming projects.

2 Phase Change Materials

The term phase-change material (PCM), usually refers to materials with a high latent heat value at melting or freezing temperature. This feature allows them to release and absorb heat at almost constant temperature when they are in transition from one phase to another. Thus, when they are freezing, they release a high amount of heat almost isothermally, and when they are melting, they absorb the same amount of heat, again with low variations in temperature. This property makes them very suitable as heat reservoirs, being able to store from five to fourteen times what a conventional heat storing method can handle. This kind of materials has gained remarkable attention during the last years as LHTS (Latent heat thermal storage) materials for space and earth applications.

As part of this chapter, a review of the state of the art of thermal storage systems is provided. Then, a special emphasis on PCMs is made, enumerating: the different kinds of PCMs available in the market, the advantages and disadvantages over other thermal storage and thermal control systems, and the applications of these materials with a special focus on the space.

2.1 Thermal storage systems

As part of the thermal control of a spacecraft, thermal storage systems (TES) play a key role due to their capacity of supplying energy under request. Moreover, TES also present a lot of applications and usefulness on the earth (i.e.: conditioning of buildings). Three main types of TES are available depending on their use of latent, sensible or chemical heat [16].

2.1.1 Sensible heat storage systems

Sensible heat storage systems (SHTS) are those who take profit of their high specific heat, c_p, to function as a heat reservoir. Thus, thanks to the high value of the energy density, they are capable to store highers amounts of heat, than other materials, with lower gradients of temperature and saving space. Sensible heat is defined as follows:

$$Q = \int_{T_1}^{T_2} mc_p dT = mc_p(T_2 - T_1) \ [J] \tag{2.1}$$

where T_1 and T_2 are respectively the initial and the final temperature of the material. Therefore, from equation 2.1, it is deduced that: the larger the density ρ and the specific heat c_p of a material, the better will be for sensible thermal storage purposes.

2.1.2 Latent heat storage systems

Latent heat storage systems (LHTS) work in a similar way than SHTS, however, they also use the energy needed to change the phase of a given material (i.e.: the energy needed to transform ice at 0º into water at 0º), also known as latent heat (L), to store thermal energy. The material will follow a cycle of charge-discharge processes every time heat is taken and given from the PCM, keeping temperature almost constant as long as the only term that intervene on the process is the latent heat. The energy density for LHTS writes: The thermal heat storage of a LHTS reads:

$$Q = \int_{T_1}^{T_m} m_s c_{ps} dT + \int_{T_m}^{T_2} m_l c_{pl} dT + mfL = m_s c_{ps}(T_m - T_1) + m_l c_{pl}(T_2 - T_m) + mfL \ [J] \tag{2.2}$$

where c_{pl} and c_{ps} are, respectively, the specific heat of the liquid and the solid phase, m_l and m_s are, respectively the mash of liquid and the mass of solid, f is the liquid fraction and T_m is the melting point (or melting temperature) of the material.

Comparing equation 2.2 with equation 2.1, and taking a look at figure 1 one can easily observe that LHTS perform much higher energy density than SHTS, and with the added of store it even isothermaly. However, LHTS have some inconveniences that will be addressed in following sections.

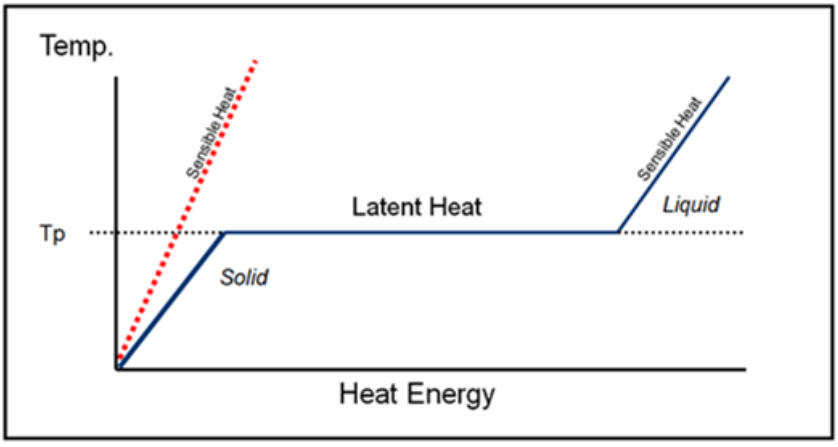


Figure 1: Entalpy-temperature curve for LHTS (in blue) and SHTS(in red). Source [17]

PCMs, are the most representative materials of this kind of thermal storage (mainly in the liquid-solid transition) and as already mentioned, this technology is one of the core points of the present project.

2.1.3 Chemical heat thermal storage

In a similar way than LHTS, endothermic and exothermic chemical reactions take place at specific temperatures absorbing or releasing heat in the process. For this technology to be feasible as heat storage systems, the chemical reactions must be reversible, in order to, again similarly to LHTS, be able to perform charge and discharge cycles. Some benefits of this technology are the high thermal densities, the very large storage times at ambient temperature, and their heat pumping-capabilities. Nevertheless, this technology is still in a very raw state and foster the development of this technology must be a priority for the aim of a sustainable and renewable energy-based future.

2.2 Types of PCMs [5]

Due to the generality of the PCM definition, many different materials can be considered as PCMs. Thus, despite the multiple kinds, most of them fall in one of these three categories: organics, inorganics, and liquid metals.

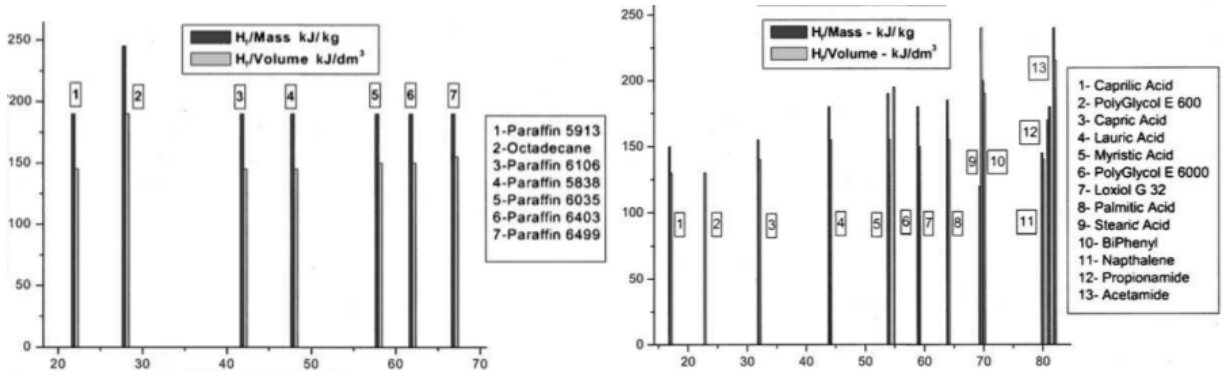
2.2.1 Organic PCMs

Probably the most popular type of PCM, mainly due to their high availability, their low cost and the simplicity of working with them. Between all the organic PCMs, a distinction is made between: the alkane or paraffin family (C_nH_{2n+2}) and the fatty acids family ($CH_3(CH_2)_{2n}COOH$). All of these organic PCMs have similar characteristics, therefore they are all used for alike kinds of applications.

One of the more remarkable characteristics of organic PCMs is that the change of phase occurs smoothly between a range of temperatures and not like in other PCMs where the phase change is isothermal. Another remarkable feature is the common melting point of these PCMs, between 35ºC and 75ºC, what makes them perfect as a thermal protector for electronics (which maximum operational temperature is around 85ºC). Furthermore, these materials perform chemical and physical stability, long durability and low conductivity, what it is usually a disadvantage for thermal storage and management applications.

2.2.2 Inorganic PCMs

Inorganic PCMs are mainly compound by salt and salt hydrates, such as $MgCl_2 \cdot 6H_2O$, KNO_3, $MgCl_2$,$CaCl_2 \cdot 6H_2O$, and the characteristics of both are similar. Furthermore,

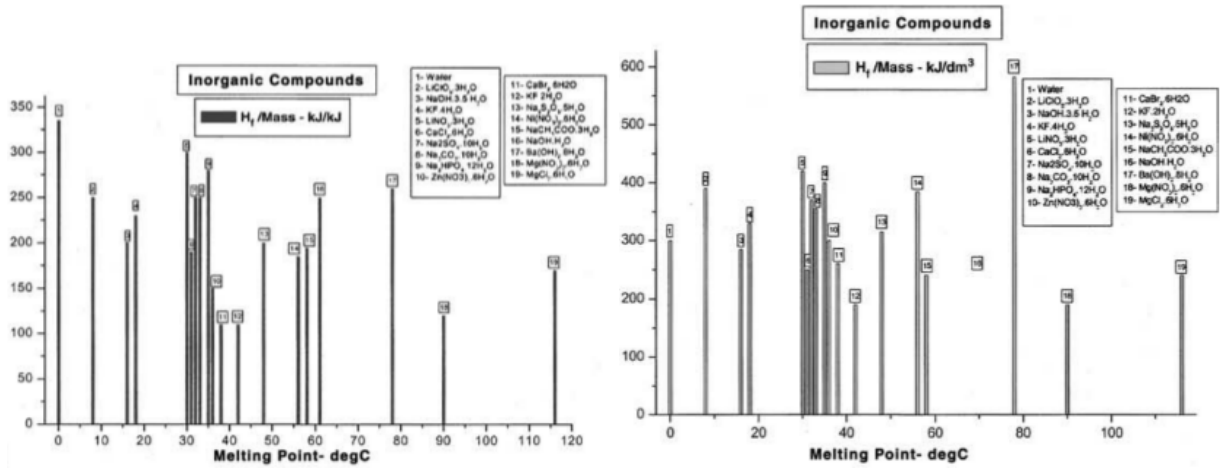


(a) Paraffins (b) Acids

Figure 2: Organic PCMs.The x-label is the temperature of the melting point and the y-label is the latent heat of different organic PCMs. Source [18]

all of the salts and salt hydrates have a well-defined crystalline structure.

General characteristics of these compounds are high thermal conductivity compared to paraffins, almost double, high latent heat, small changes in volume, no corrosiveness and low toxicity. Also, they perform a wide range of melting points, between 10°C and 900°C. In the lower range of temperatures, where they overlap with organic PCMs, these are preferred over the salts. However, for applications with high melting points requirements, such as solar energy applications, salts and salts hydrates usually are the first choice.



(a) (b)

Figure 3: Inorganic PCMs. The x-label is the temperature of the melting point and the y-label is the latent heat of different inorganic PCMs. Source Source [18]

2.2.3 Metallic PCMs

This type of PCMs is composed of pure metals and metal alloys, and is probably the less commonly found in human applications, mainly because their low latent heat compared with other PCMs. Nevertheless, metal and metals alloys still perform promising features that make them useful for certain LHTS applications. Some of their more characteristic qualities are the easiness to work with them, their physical and chemical stability, at high temperature, and their wide range of melting points that they perform. From all the characteristic of these PCMs, the more unique and remarkable is, contrary to all other kinds of PCMs, the high thermal conductivity that they perform. This quality makes them a good choice when the need for a low charge/discharge rate of the PCM is needed.

2.3 Advantages and disadvantages of PCMs [5]

In this section, a brief incise will be made on the advantages and disadvantages of the PCMs as thermal storage systems. Nevertheless, is important to keep in mind that some of the disadvantages sometimes can be an advantage, and the other way around, depending on the desired application. For instance, usually, a high thermal conductivity is considered as a desired quality on a PCM, because it allows fast charge/discharge cycles, nevertheless, some specific missions, where the exposition of the spacecraft to a heat source is long and there are some electronic devices that should be protected from high temperatures, would benefit of a lower conductivity because it will work as thermal insulator. Thus, the classification made here is more a generality, based on the more common applications of these materials, than a rigorous statement.

2.3.1 Advantages

The principal benefits of PCMs as thermal management and thermal energy storage systems are:

- **Fixed temperature:** capacity of maintaining a fixed temperature, or almost fixed, during the phase change process regardless of the incoming heat flux.
- **Self-sufficient:** since PCMs don't need of any other component but themselves to fulfil their function as a thermal storage system. This quality carries other beneficial consequences with it, like lightness, portability and high reliability of the system.
- **No energy consumption:** because they just release and absorb heat following

the thermodynamic laws, there is no need of any additional power source to make them work.

- **High thermal density:** Thanks to the latent heat storage that they are based on. This quality allows to reduce the mass of the thermal system decreasing as well the dimension of the containers and thus, saving room.

2.3.2 Disadvantages

The PCMs, although powerful systems for thermal control, present some drawbacks that can prejudice their suitability for certain missions:

- **Low thermal conductivity:** this feature together with the absence of gravity on space environments, therefore there is no natural convection, prevent the use of this systems on missions where the temperature oscillates, from high to low values, rapidly. One of the proposed solutions for this problem is the use of metal nanoparticles submerged on the PCM , by this way the effective conductivity will increase, shorting the charge and discharge times of the PCM. Another proposed solution is to take advantage of the Marangoni convection by putting one of the sides of the PCM in contact with a fluid, by this way the melting/freezing time will decrease. During this work, this last option will be further studied.

- **Compatibility of the PCM with the container:** since PCMs are usually formed by a liquid phase, as well as solid phase, the solubility and permeability of the chosen material and the container must be studied. Wrong combinations between the PCM and the container material can lead to leaks of the PCM or dissolution of the container, spoiling like this the performance of the system.

- **Limited life:** Depending on each kind of PCM, the process of charging-discharging degrades the material, hence losing effectiveness after a given number of cycles.

2.4 Applications

Although the main application considered in this project, is the one as latent heat storage systems, the qualities of the PCMs have a wide field of applications. Therefore, three main categories of applications are identified in function of the properties of PCMs they take advantage of [19]:

- **Use of latent heat related to the phase transition:** this kind of applications take advantage of the heat released and absorbed during the phase change to increase or decrease the temperature of the surrounding systems depending on the needs.

- **Use of the state change produced in the phase transition:** this kind of applications takes advantage of the change in the properties, such as conductivity and porosity, of the PCM during the phase transition.

- **Use of the fixed temperature during the phase change process:** this kind of application take advantage of the constant temperature maintained on the PCMs during the phase change despite the heat absorbed or released in the process.

Moreover, and since this project is mainly focused on PCMs for the space environment, the last section is dedicated to space applications.

2.4.1 Use of the latent heat related to the phase transition

Two kinds of applications can be noticed:

- **Latent heat storage.** The use of latent heat during phase change, which is large with respect the total heat capacity of the material, appears as promising for thermal energy storage, performing much higher storage density, and lower temperature ranges, than sensible heat storage devices. Depending on the use, PCMs are selected regarding its melting temperature. For instance, PCMs with melting points below 15'8^{o}C are used for air conditioning applications, while PCMs over 90,8^{o}C are employed as for absorption refrigeration. PCMs between this two points are mainly used for solar energy storage applications. Among them, passive methods, frequently coupled with active systems, for heat control of satellites [20] and buildings [21] can be found. These methods provide multiple economic, room and energy-saving benefits.

- **Heat sink.** High heat flux releases come from high-power lasers, electronic devices, engines, etc... Therefore heat sinks are the passive heat exchangers with the capacity of removing the scattered heat, enhancing like this the performance, durability and safety of these devices. For this applications, many different techniques have been applied, including heat transfer fluids (HTF), fluids with metal nanoparticles to increase the thermal conduction...However, the heat transfer performance of pure fluids, even with conductive nanoparticles, is rather small. Nanoparticles of PCMs

submerged in the fluid have emerged as a solution [22]. Thanks to the high specific heat of the melted parts and the high latent heat, the fluid is capable of remove much more amount of heat to protect the different devices from extremely high temperatures.

2.4.2 Use of the state change produced in the phase transition

Two kinds of applications can be noticed:

- **Control of diffusivity and drug release.** Drugs that response to a thermal stimulus is a recent pharmacological and medical area since the body temperature is prone to vary depending on diseases and ambient conditions. PCMs appear as a new kind of thermosensitive materials that can regulate the dose of a drug in relation to the temperature. Thus, and because of the solid liquid transition, a PCM in function of the temperature can regulate the diffusivity of a given drug [23]. Thus, for instance if there is a need to release a certain medicament when the local body temperature reaches 38 °C, a PCM with the same melting point should be employed, like this, below 38°, the PCM is solid blocking the supplying of the drug, on the contrary, when the temperature of the body is 38 °C or above the drug will be released in function of the temperature, being the higher the dose when the higher is the temperature. For this kind of applications, organic PCMs are used, such as 1-tetradecanol, fatty alcohols and acids, and other inorganics, like gold nanocages (AuNCs).

- **Phase change memories** A currently in development application for PCMs is their use as information storage and memory devices. The aim of this technology, named phase change memory, is to overcome the limitations of the current flash based technology. The basis of this technology is to apply an electric current or a light pulse (laser) to a heating material that rapidly heats up the PCM switching it from a crystal form into an amorphous state, like this the PCM will be able to change its reflectivity and electrical conductance. It is a remarkable feature that since a PCM reaches several states depending on the temperature, a PCM is able to codify as many bits as different states can reach, meaning that their memory capabilities are much higher than conventional memories. Furthermore, the PCMs are able to reach again their initial state if they are cool down, what means that the memory is rewritable [24]. However, there is still a lot of problems in programming cells in this way, what have caused fewer applications of this technology than expected. Currently, scientists are searching for alternative PCM solutions, to overcome these

drawbacks, with mixed success.

2.4.3 Use of fixed temperature during the phase transition

Two kinds of applications can be noticed:

- **Sensing and detection.** As a recent application, PCMs (indium, bismuth oxide, n-propyltriethoxysilane) are being used as a detector of biomarkers in cancer diagnosis. Current biomarker detectors have several lacks, including the low reliability to distinguish between lethal and benign cancer and different range of biomarker concentrations. The use of liquid and solid phase-change nanoparticles is delivering good results as detectors that simultaneously can recognise the different size of biomarkers. The melting peak featured by PCM, together with the fusion enthalpy determine, respectively, the type and concentration of biomarkers [25].

- **Barcoding.** Nowadays, the process of barcoding with the use of optical bar codes, micro fibres and RDIF chips do not allow to massively barcode objects at a competitive price. PCM technology is being employed as a solution to this issue. For this aim, the melting properties of different metallic PCMs (Pb-Sn alloy, Sn and Bi) in a way that consists in encoding by taking advantage of the different melting peaks given by the mixture of nanoparticles [26].

2.4.4 Space applications

One of the eldest use of PCMs is the application on space systems, NASA was already performing studies on these materials during the 70's to assess their potential as thermal control devices [27]. The extreme thermal environment that is the space, together with the high oscillating temperatures founded there (for instance when a spacecraft transit from eclipse to the sun light) demands a high thermal control performance, thus the passive and reliable functioning of PCMs make them fit for this kind of applications. Plenty of spacecraft and other space systems employed PCMs to fix a constant temperature (for instance: Apolo 15,16 and 17). Nevertheless, there are still some issues to be solved, the absence of natural convection in zero-gravity environments, like in space, make the melting/solidification process very slow, decreasing their functionality as thermal control devices. And also, phenomena like the Bénard-Marangoni convection take a high relevance in space environments and must be taken in to account in the design of PCMs.

3 Mathematical Formulation. The Enthalpy-Porosity Method for Stefan Problems

In the previous chapter, it was mentioned that PCMs are materials that take advantage of the latent heat characteristical of phase changes, to undertake multiple functions mainly related to thermal management. Therefore, to fully take advantage of the PCMs properties, a fluid-thermal analysis of the phase change process becomes of main relevance.

This chapter aims to describe the equations involved in heat transfer processes in a material undertaking phase transition, what commonly has been referred in the literature as the Stefan problem or Stefan-like problems [14]. The equations described are conformed under the enthalpy-porosity approach, which is one of the most popular for this kind of problems. The chapter ends by giving further insight to the different forms of heat transfer that appear in the melting processes studied in this thesis.

3.1 Governing equations in fluid dynamics [6]

Like in every computation where a continuous fluid is under study, the conservative fluid dynamic equations are usually employed to describe the properties of the fluid along time and space. These equations are the continuity equation, that is based on the conservation of mass, the momentum equation on each of the directions under study, also known as Navier-Stokes equations, that is the second law of Newton applied to fluid dynamics, and the heat equation that involves the heat transfer processes that take place in the fluid. All of them form a set of three non-linear partial differential equations that, in the most general formulation, have the following shape:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \vec{u}) \tag{3.1}$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = -\nabla P + \rho \vec{g} + \nabla \cdot (\mu \nabla \vec{u}) \tag{3.2}$$

$$\frac{\partial H}{\partial t} + \nabla \cdot (\vec{u} H) = \nabla \cdot (\kappa \nabla T) \tag{3.3}$$

Where equation 3.1 is the continuity equation, equation 3.2 is the momentum equation in n-dimensions and equation 3.3 is the energy equation. Regarding the variables, ρ is the density of the fluid, $\vec{u}$ is the velocity vector in n-dimensions, P is the static pressure,

$\vec{g}$ is the gravity acceleration, μ is the kinematic viscosity, H is the enthalpy per volume unit, kappa is the thermal fluid conductivity and T is the fluid temperature.

3.2 Simplifications and Assumptions [7]

Due to the specific problem solved in this case, a series of assumptions are considered in order to simplify the resolution of the fluid equations.

- **Laminar regime:** since the Reynolds number does not have high enough values to consider turbulent effects.

$$Re_L = \frac{\rho U L_c}{\mu} \leq Re_{L,crit} \tag{3.4}$$

  where L_c is the characteristic length.

- **Constant thermophysical properties:** thermal conductivity (κ), specific heat (c_p), and the thermal expansion coefficient (β) are considered independent of temperature for simplifying the computations and because of the low impact on the results in problems with small ranges of temperature.

- **Constant density**, since the fluid part of the material, is always in liquid phase and at low Mach numbers, the density is considered independent of velocity, pressure or temperature. Only temperature action on the density will be accounted for buoyancy effects through the, later explained, Boussinesq approximation.

- **Newtonian fluid**, meaning that the viscosity of the fluid is constant, $\frac{\partial \mu}{\partial t} = \nabla \mu = 0$.

The previous assumptions lead to a new form of the equations.

$$\nabla \cdot \vec{u} = 0 \tag{3.5}$$

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot (\vec{u}\vec{u}) = -\nabla P + \vec{g} + \nu \Delta \vec{u} \tag{3.6}$$

$$\frac{\partial H}{\partial t} + \nabla \cdot (\vec{u} H) = \nabla \cdot (\kappa \nabla T) \tag{3.7}$$

where ν is the dynamic viscosity and all the other parameters are already defined.

3.3 The enthalpy-porosity method

Historically, the melting/solidification processes, also known as Stefan-like problems, had been rather challenging to compute due to the moving interface between the solid and the liquid phase. This inconvenience made very complex the computation of the heat transfer due to the difficult modelling of the velocity field. During years, different approaches had been made with different grades of success. According to *Brent et al.* [28], there are three groups of methodologies developed for this kind of problems: empirical, classical and enthalpy technique. The enthalpy-porosity method forms part of the last kind of approaches and represents a simple, direct and physical way of dealing with this kind of problems. During all this section, the approach made by Voller for the enthalpy-porosity technique is the major reference [1].

3.3.1 General description

The success of this method lies on the simplicity of tracking the phase front thanks to the fixed grid enthalpy-porosity formulation, that presents the enthalpy in the following way.

$$H = h + \Delta H \tag{3.8}$$

where $h = cT$ is the sum of the sensible heat and ΔH is the latent heat, which as showed below, is function of the temperature and can take the following values.

$$H = f(T) = \begin{cases} L & T \geq T_l \\ L \cdot f & T_l > T \geq T_s \\ 0 & T < T_s \end{cases} \tag{3.9}$$

Where f is the local liquid fraction, T_l is the coldest temperature at which the solid fraction is still zero and T_s is the hottest temperature at which the solid fraction is equal to one.

Thus, from the value of the latent heat, it is easy to deduce the state of matter at a given point and time. Equation 3.9, differentiates the three phases developed during the melting/solidification process, which are, the solid phase ($\Delta H = 0$), the liquid phase ($\Delta H = L$) and the mushy phase($0 < \Delta H < L$), introduced for the first time by Voller and Prakash[1], which is the transition zone between the solid and the liquid phase.

The last important aspect taken into account is the modelling of the velocity field,

which of course will be influenced by the state of the matter.

$$u = \begin{cases} u_l & T \geq T_l \\ u_l \cdot f & T_l > T \geq T_s \\ 0 & T < T_s \end{cases} \tag{3.10}$$

where u_l is the actual fluid velocity.

Although the main characteristics of the enthalpy method have already been described, source terms are needed to describe in detail the conservation of momentum and the conservation of energy in this process.

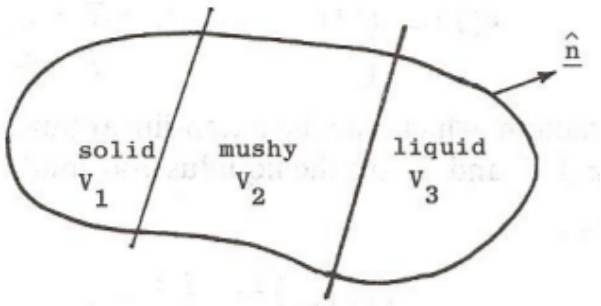


Figure 4: Phase change volume. Source [2]

3.3.2 The conservation of momentum

Following with the enthalpy-porosity definition, some modifications are introduced in the momentum equation. These modifications are presented in the form of two source terms that model two different phenomena.

3.3.2.1 The Boussinesq approximation

This approximation is made with the aim of considering buoyancy effects due to the change of density in the fluid caused by the temperature gradients, without tanking these density variations into account on the other terms of the Navier-Stokes equations. This simplification stands under the consideration that, the change of density in the fluid has a big enough impact on gravity terms to neglect the effect of this same change of density in the inertia terms.

Consider a fluid with changing density in function of the temperature. From a Taylor

expansion the density function can be written as follows:

$$\rho(T) = \rho_0 + \frac{\partial \rho}{\partial T}\Delta T \tag{3.11}$$

Likewise, the density function can be written in function of the thermal expansion coefficient.

$$\rho = \rho_0(1 - (\beta(T - T_m)) \tag{3.12}$$

where T is the temperature at the computational domain, T_m is a reference temperature and β is the thermal expansion coefficient, that can be written as:

$$\beta = -\frac{1}{\rho}\frac{\partial \rho}{\partial T} \tag{3.13}$$

Once the Boussinesq approximation is explained, it can be introduce in the gravitational term of the momentum equation obtaining the Boussinesq source term:

$$\vec{S_b} = -\vec{g}\beta(T - T_m) \tag{3.14}$$

As described in the introduction of this section, this term will take into account the natural convection.

3.3.2.2 Darcy's law Source Term

The Darcy's term models the influence of the solid and the mushy region in the velocity field. The Darcy law in a mushy region has the following shape:

$$\vec{u} = -(K/\mu) \cdot \nabla P \tag{3.15}$$

where K stands for the permeability and is a function of the solid fraction. The Darcy source term can be defined by the following law:

$$S_u = -A\vec{u} \tag{3.16}$$

where A is a parameter that models the influence of the mushy region on the velocity field. This parameter A is a function of porosity or liquid fraction. A derived form of the Darcy law is the well-known Carman-Konseny equation, which is useful for model mushy regions for the non-isothermal case (a case where the change of phase doesn't occur at a constant temperature, thus the mushy region appears). This equation is able

to determine the shape of the parameter A.

$$A = -C\frac{(1-\lambda^2)}{(\lambda^3+q)} \tag{3.17}$$

where C is a constant depending on the morphology of the mushy region that usually oscillates between 10^4 and 10^7, λ is porosity or liquid fraction ($\lambda = f$), and finally q is a constant usually around 0.001 introduced to avoid a zero-value denominator when porosity tends to zero. Being the parameter A known, the source term that describes the influence of the mushy region on the momentum equation is as follows:

$$S_u = -C\frac{(1-\lambda^2)}{(\lambda^3+q)} \cdot \vec{u} \tag{3.18}$$

Thus from equation 3.18, the velocity field on the mushy region will be an increasing function with the porosity. Whereas the porosity tends to zero the velocity field will be fully developed, while when the porosity tends to one, the Darcy Term will take a very high value that enforces the velocity field to zero.

3.3.3 The conservation of energy. The enthalpy source

In the case of the energy equation, further changes must be done to take into account the heat transfer produced during the phase change or likewise the latent heat component present in this kind of processes.
The simplest way to account for this component, it is the already described by the equation 3.8, where a distinction is made between the latent heat component of the enthalpy (ΔH) and the sensible heat component ($h = c_p T$). Nevertheless, equation 3.8 can take a more complex form if the energy exchange is depicted deeply.

$$H = \rho(1-f)\int_{T_{ref}}^{T} c_s(T^*)dT^* + \rho f \int_{T_{ref}}^{T} c_l(T^*)dT^* + \rho f L \tag{3.19}$$

Since c_l and c_s are considered constant during the project, they can be taken out of the integral and used for define a weighted specific heat in function of the liquid fraction.

$$c_p = (1-f)c_s + f c_l \tag{3.20}$$

Now, the expression given by equation 3.20 and 3.19 can be introduced in equation 3.3,

reaching the following expression:

$$\frac{\partial \rho c_p T}{\partial t} + \nabla \cdot (\vec{u} \rho c_p T) = \nabla \cdot (\kappa \nabla T) - \rho L \Big(\frac{\partial f}{\partial t} + \vec{u} \cdot \nabla f \Big) \tag{3.21}$$

A new source term appears in the energy equation when introducing the enthalpy expression. In this term, a spatial and a time derivative of the liquid fraction can be distinguished. This is occasioned due to the transient behaviour of the liquid fraction in the whole computational domain, being prone to vary in space and time. To simplify the computations, the hypothesis that the convective terms of the liquid fraction are negligible in front of the temporal terms is considered to be true [29].

$$\frac{\frac{\partial f}{\partial t}}{\vec{u} \cdot \nabla f} \sim \frac{L_{boundary}}{t_c u_c} \sim St \gg 1 \tag{3.22}$$

where $L_{boundary}$ stands for the characteristic length at the boundary, t_c and u_c are respectively the characteristic time and velocity and St is the Strouhal number. Taking into account all of the hypothesis mentioned in this section and the continuity equation for incompressible fluids explained at the beginning of this chapter (equation 3.5), it can be combined with the energy equation to get the shape of the source term.

$$S_h = -\frac{\delta H^*}{\rho} \frac{\partial f}{\partial t} = -\delta H \frac{\partial f}{\partial t} \tag{3.23}$$

where the term $\frac{\partial f}{\partial t}$ needs of discretization to be implemented. For the treatment of the variation of the liquid fraction in the latent source term, the approach made by *Voller and Swaminathan* [30] is the one followed in this thesis.

The energy equation can be written in the form of an iterative system, since the intrinsic non-linearity due the dependency of the enthalpy with the temperature, based in an implicit Euler backward scheme for the time discretization:

$$a_p T_p^{m+1} = \sum_{nb} a_{nb} T_{nb}^{m+1} + d_p [H_p^m - H_p^{m+1}] + b \tag{3.24}$$

where a b and d are coefficients of the discretization equation. The subscripts and superscripts p, nb and m refer respectively, to the current cell, the neighbour cells, and the value at the previous iteration. From equation 3.24, the enthalpy source term is written as follows:

$$S_h = d_p [H_p^m - H_p^{m+1}] \tag{3.25}$$

taking equation 3.9, the source term can be written as function of the liquid fraction.

$$S_h = d_p[f_p^m - f_p^{m+1}] \tag{3.26}$$

Developing the Taylor equation for f_p^{m+1}, the liquid fraction reads:

$$f_p^{m+1} = f_p^m + \frac{df}{dT}|_{f_p^m}[T_p^{m+1} - F^{-1}(f_p^m)] \tag{3.27}$$

where $F^{-1}(f_p^m)$ is the inverse liquid fraction function, that is the temperature correspondent to the liquid fraction. Using this linearization, now equation 3.26 reads:

$$S_h = S_p T_p^{m+1} + S_c \tag{3.28}$$

where

$$S_p = -d_p \frac{df}{dT}|_{f_p^m}[T_p^{m+1} \tag{3.29}$$

and

$$S_c = d_p[f_p^m - f_p^{m+1}] + S_p F^{-1}(f_p^m) \tag{3.30}$$

Thus, the iterative scheme now becomes:

$$(a_p - S_p)T_p^{m+1} = \sum_{nb} a_{nb} T_{nb}^{m+1} + S_c + b \tag{3.31}$$

Following equation 3.31, the iterative procedure for a single time step is described:

1. The first step is make the guess of considering the liquid fraction of the present time step equal to the liquid fraction of the previous time step $f^{t+\Delta t} = f^t$

2. The temperature field T_{m+1} is calculated from equation 3.31 with the values of the coefficients, S_p, S_c and $\frac{df^m}{dT}$.

3. With the value of temperature computed T^{m+1}, the liquid fraction is updated with help of a relaxation factor.

$$f_p^{m+1} = f_p^m + \gamma[F(T_p^{m+1}) - f_p^m] \tag{3.32}$$

   where γ is the under-relaxation parameter with value $0 \leq \gamma \leq 1$.

Steps 2 and 3 are repeated until convergence, when the current temperature field is

consistent with current liquid fraction.

At last, the dependency of the liquid fraction with the temperature is outlined. During the computations performed in this project, a linear form of the liquid fraction f which only depends on temperature has been used for simplicity [1].

$$f(T) = \begin{cases} 0 & T \geq T_l \\ (\frac{T-T_s}{T_l-T_s})^\beta & T_l > T \geq T_s \\ 1 & T < T_s \end{cases} \tag{3.33}$$

where in this case β=1 due it is the value more often used in the literature and at the same time avoid non-linearities. Nevertheless, and depending on the material it is being dealt with, the value of β changes, getting like this different functions of the liquid fraction.

3.4 Final form of the fluid dynamic equations

After all the discussion made on previous sections around the form of the equations that describe the present physical problem, a final form of the fluid dynamic equations, taking into account the enthalpy porosity formulation, can be written as follows:

$$\nabla \cdot \vec{u} = 0 \tag{3.34}$$

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot (\vec{u}\vec{u}) = -\nabla p_m + \nu \Delta \vec{u} + \vec{S_b} + \vec{u} S_u \tag{3.35}$$

$$\frac{\partial c_p T}{\partial t} + \vec{u} \cdot \nabla (c_p T) = \nabla \cdot (\frac{\kappa}{\rho} \nabla T) + S_h \tag{3.36}$$

Later, in section 4, the discretization of this final form of the fluid equations for the enthalpy method is depicted.

3.5 Heat transfer in Stefan problems

This section arises with the intention of clarifying the heat transfer mechanisms involved in the melting/solidification processes studied on this project. Moreover, each of this heat transfer mechanisms is the main heating force on each of the cases performed in chapter

6. Therefore, a summary of this heat transfer processes is done to provide a deeper understanding of the phase/change process. Furthermore, several dimensionless numbers are defined and studied because of their quality of giving orders of magnitude and a qualitative description of the problem, without going too deep into the mathematics.

As a starting point, the most typical dimensionless number, which presence is of main importance in phase change processes[31], is the Stefan number, that reads:

$$Ste = \frac{c_p \Delta T}{L} \tag{3.37}$$

which yields the ratio between specific heat and latent heat present in a phase change process. In problems with a high Stefan number, the heat transfer will be dominated by the sensible heat, whereas in problems where the Stefan number is low, the presence of latent heat will have a high impact in the solution. PCMs usually perform really high latent heat values, yielding to Stefan numbers of the order of 10^{-1} and below, depending on the studied case.

3.5.1 Conduction [8]

Conduction is a heat transfer process that, in more or less magnitude, it is always present in heat transfer problems, since it only needs the presence of a physical media in order to actuate.

Thermal conduction is the process where heat is transmitted through collisions of particles between one or several bodies. The collision of particles transmits, in a disorganized way, kinetic and potential energy, which together are known as internal energy.

The thermal conductivity is directly linked with the diffusive term of the transport equation. Thus, a pure conduction process can be defined by the diffusive term of the heat equation.

$$\frac{\partial Q}{\partial t} = -\kappa S \frac{\partial T}{\partial x} \tag{3.38}$$

where κ is the thermal conduction coefficient, that relates the amount of heat transferred per unit of time $\frac{\partial Q}{\partial t}$ through a section S due a gradient $\frac{\partial T}{\partial x}$. If equation 3.38 is reformulated, the classical Fourier's law is obtained:

$$\vec{q} = -\kappa \nabla T \tag{3.39}$$

Furthermore, if transitory effects are willed to be accounted, the heat diffusion equation

in three dimensions reads:

$$\frac{\partial c_p T}{\partial t} = \nabla(\frac{\kappa}{\rho}\nabla T) + \dot{q} \tag{3.40}$$

Notice that this equation matches with the aforementioned equation 3.36, but neglecting the convective term.

3.5.2 Rayleight-Bénard convection [9]

Rayleigh-Bénard convection, also known as natural convection, is the movement of fluid as a consequence of gradients of temperature under the effects of gravity [8]. The temperature has a direct impact on the density: an increase in the temperature decrease the density while a decrease in the temperature produces the opposite effect. The difference in density will produce buoyancy forces that will cause the movement of the fluid, accelerating the heat transfer processes.

The thermal convection, as its own name explains, is directly related to the convective term of the heat transfer equation. However, since in convection, the velocity field also plays a role in the heat transfer, all of the three transport equations together with the Boussinesq formulation must be taken into account.

$$\nabla \cdot \vec{u} = 0 \tag{3.41}$$

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot (\vec{u}\vec{u}) = -\nabla p_m + \nu \Delta \vec{u} - \vec{g}\beta(T - T_m) \tag{3.42}$$

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T = \nabla(\alpha \nabla T) \tag{3.43}$$

where α stands for thermal diffusivity of the media and all the other variables have already been described. An important dimensionless number coming from the equation above is the Grashoff number (Gr) that gives the coefficient of natural convection. Moreover, to define this number the conditions needed to apply the Boussinesq approximation must be satisfied:

$$Gr = \frac{g\beta\Delta T d^3}{\nu^2} \geq 1 \tag{3.44}$$

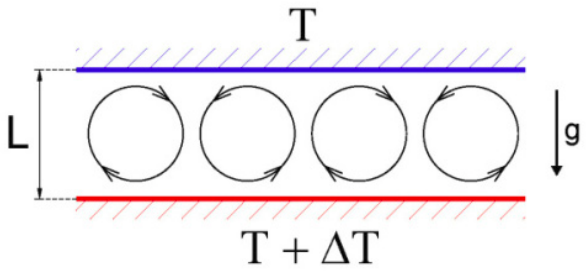


Figure 5: Explicative scheme of Rayleigh-Bénard convection.

Another dimensionless number named Rayleigh number (Ra) can be defined:

$$Ra = \frac{Buoyancyforces}{Viscousforces} = \frac{g\beta\Delta T}{\nu\Delta u} = \frac{g\beta\Delta T d^3}{\nu\alpha} = GrPr \tag{3.45}$$

where d is the thickness of the liquid layer nd the other paramters have been already defined. As expressed in equation 3.45, the Rayleigh number gives the relation between the buoyancy forces with the viscous forces [9]. Thus, from this number, it is possible to determine the influence of natural convection in the heat transfer process. By this means, one can determine in a process if the dominant heat transfer is by conduction, at low Rayleigh numbers, or by convection at high Rayleigh numbers. As a result of a sufficiently low Rayleigh number, convective terms could be neglected of the heat equation without sensible changes on the results of the process.

Looking closely to equation 3.45, an undefined dimensionless number, with high relevance in convective processes, is perceived, which is the Prandtl number:

$$Pr = \frac{\mu c}{\kappa} \tag{3.46}$$

The Prandtl number represents the ratio between the momentum diffusivity and the thermal diffusivity. Where a high Prandtl number stands for a process where the transmission of energy due momentum is higher than the transmission of energy by thermal conduction (ie. the velocity boundary layer thickness is higher than the thermal boundary layer thickness) whereas a low Prandtl number implies exactly the opposite, a dominant thermal conductivity energy transmission and therefore thicker thermal boundary layer with respect the velocity boundary layer [9].

3.5.3 Bénard-Marangoni convection

Bénard-Marangoni convection was first observed by *Henri Bénard* at the beginning of the 20^{th} century [32], however, he wrongly attribute the movement of the fluid to buoyant

forces. Later, in the year 1956, *J.M. Block* was the first in demonstrating that the formation of convective cells that Bénard observed in a fluid were due to gradients in surface tension [33].

Whenever a surface of a system is in contact with a gas, and the system is submitted to a gradient of temperature, the so named Bénard-Marangoni convection appears [34]. The main difference between the Marangoni convection and the natural convection is that this first one takes place due the action of surface tension, being present even without gravity force. In this cases, the boundary condition of the surface in contact with a gas, often referred as "free surface", present a modification of the surface tension given by the following equation:

$$\sigma = \sigma_0 + \frac{\partial \sigma}{\partial T} \Delta T \tag{3.47}$$

This equation describes the relationship of the surface tension σ with the gradient of temperature, where $\frac{\partial \sigma}{\partial T}$ is the Marangoni coefficient which value is usually negative, since in liquids an increase in temperature often means a decrease in surface tension. Thus, Marangoni convection will occur when, in a liquid in contact with a gas, a gradient of temperature appears, producing a gradient in the surface tension that will cause a movement of the fluid in the attempt to restore the equilibrium.

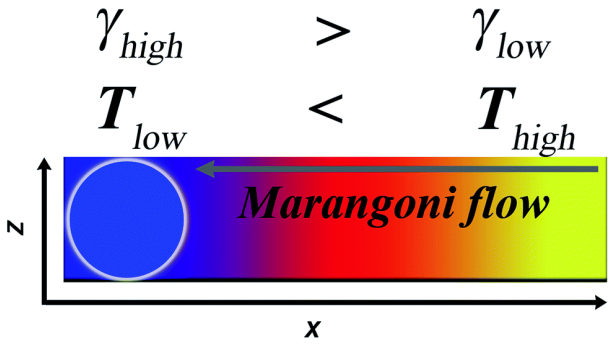


Figure 6: Explicative scheme of Bénard-Marangoni convection. Source [35]

The Marangoni dimensionless number is defined as:

$$Ma = \frac{\partial \sigma}{\partial T} \frac{\Delta T d}{\mu \alpha} = \frac{Surface forces}{Viscous fores} \tag{3.48}$$

The Marangoni number and provides a ratio between the convective forces created by the Marangoni convection and the viscous forces. In a similar way than the Rayleigh number in the natural convection, this number will give the influence of the Marangoni convection in the heat transfer.

Lately, in 1964, Nield [36] developed a theory that comprised both destabilize forces, buoyancy and surface tension. Using linear analysis, he found the following relationship between the Rayleigh number and the Marangoni number:

$$\frac{Ra}{Ra_c} + \frac{Ma}{Ma_c} \simeq 1 \tag{3.49}$$

where Ra_c=680 is the critical Rayleigh number when the surface tension is zero ($\gamma = 0$), and $M_c = 81$ is the critical value of the Marangoni number in micro-gravity (g=0)[34]. This relationship has been represented for the extreme cases of a Biot number of value zero and infinite in the figure 7.

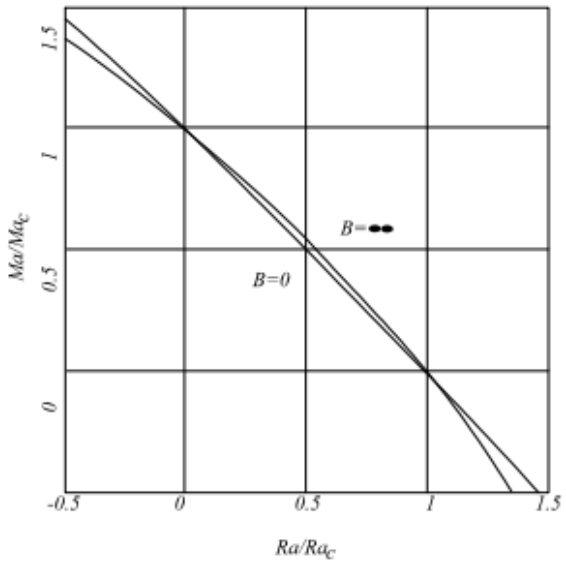


Figure 7: Diagram of Nield stability for the convection driven by buoyancy and surface tension. Letter B represent the Biot number. Source [37]

Notice that Bi, is the Biot number that is the quotient, in a body, between the heat transferred by convection on the surface and the heat transferred by conduction through the body.

$$Bi = \frac{hL_c}{\kappa} \tag{3.50}$$

where h is the heat transfer coefficient on the surface, L is the characteristic length and κ stands for the thermal conductivity. Another important relationship between both

numbers is the Bond number:

$$Bo = \frac{Ra}{Ma} = \frac{-\rho g \beta d^2}{\frac{\partial \sigma}{\partial T}} = \frac{Buoyant forces}{Surface forces} \tag{3.51}$$

This Bond number, as described in equation 3.51, compares the influence of the two convective forces, giving in function of the value, which of them prevail over the other.

As a last remark, it is important to realize that both convection phenomena, Rayleigh-Bénard and Bénard-Marangoni, are independent of each other. While the first one depends on the gravity force to produce the movement of fluid, the second one depends on the gradient of surface tension to produce it, no matter if there is gravity acting on the system or not. This fact is translated in that Marangoni convection can appear in micro-gravity environments, accelerating the melting/freezing processes.

4 Numerical Methods & Solver Description

The fluid equations, even after some hypothesis and simplifications, usually have the shape of non-linear partial differential equations, what means that no analytical solution can be obtained from them. For this reason, different numerical methods are used to arrive at the solution of this kind of problems. However, to be able to apply those, it is first needed to discretize the computational domain where the solution is going to be computed. Nowadays, three discretization methods are mainly used to solve the fluid dynamic equations:

- **Finite Volume Method (FVM).** It is a discretization technique where the integral formulation of the conservation laws is discretized directly in the physical space. The domain is divided into volumes or cells where the average value, in the cell, of the fluid variables are storage. It is the most widely applied method today in CFD[38] and the one that uses `OpenFOAM`, thus, the main focus will be made on it in following sections.

- **Finite Element Method (FEM).** Likewise in the FVM, the computational domain is divided into elements and it requires of the integral formulation as a starting point. Besides these similarities, the FEM has other particularities, as the usual unstructured grid choice and the parametric representation of the unknowns in each element. On its early years, it was a method mainly used for solving problems in the field of linear and non-linear structural mechanics. Nevertheless, soon it appeared to be a method with use in solving continuous field problems [39].

- **Finite Difference Method (FDM).** It is a discretization method based on the development of Taylor series in order to translate the derivatives into algebraic relations that are easy to handle for a computer. In opposite to FVM and FEM, the domain is divided into a set of points where each of the different fluid dynamic variables are storage. It is the easiest method to implement and particularly useful in uniform meshes.

4.1 The Finite Volume method

Although the FVM was already introduced in the previous section, since it is the method used during all the present project, a further description of how it works, was consider needed. The main strength of the FVM is the direct connection to the physical properties, this comes from the discretization of the integral form of the conservative equation, what is translated in that, the FVM, satisfy automatically the conservative laws.

In general, when a method is used to discretize a problem, there are two main aspects that shall be dealt with: the discretization of the computational domain or how the physical domain is transformed into a geometrical/numerical domain, and the discretization of the equations that govern the physical problem or how the non-algebraical terms are transformed into algebraic terms keeping consistency with the original equation and mimicking, as close as possible, the physics.

4.1.1 Discretization of the computational domain [10]

Due to its generality, the computational domain is converted in volume cells, called control volumes CV, that can form either irregular or regular grids as long as there is no overlap between cells. Each volume cell has a centroid or computational point p where the solution is obtained. This point P satisfies the equation below.

$$\int_{V_p} (\vec{x} - \vec{x_p})\, dV \tag{4.1}$$

where V_p is the volume of the CV. This CV is limited by a number of faces, each of them shared only by another CV. These faces are classified in internal faces, those who are the separation of two CV, and boundary faces, that are those who just belong to one CV because they are in contact with one of the boundaries of the domain. There is still one degree of freedom in the design of the mesh for the FVM. Depending on how the grid is related to the control volumes it can be defined:

- **A cell-centered approach,** where the unknowns are defined at the centres of the mesh cells and the volumes and surfaces are defined by the grid lines. The value of the flow variables are defined as the average of this variable in the cell, the later appearing equation 4.3, will describe the computation of this average.
- **A vertex-centered approach,** where in this case, unknowns are considered at the vertex of the mesh. It has the advantage of allows more flexible geometries.

In this project, a cell-centered approach is the cell type selected, since is the one used by the working software OpenFOAM.

For a better understanding of all the discretization process, the different elements of a control volume cell (V_P) are presented:

- The centroid of the control volume V_P is **p**.

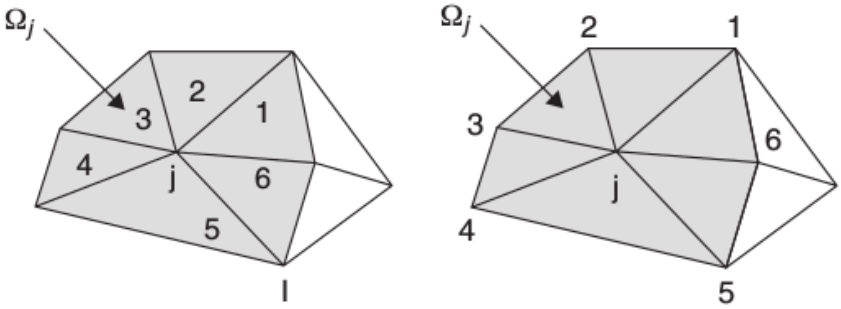


Figure 8: On the left, a cell-centered grid and on the right a vertex-centered grid [38].

- The vector that links the centroid of a control volume (e.g.: **p**) with the centroid of a contiguous control volume (e.g.: **N**) is named **d**.
- The control volume faces are referred under the letter **f**, which is also the face centre.
- The intersection of the vector **d** with a face, is labeled $\mathbf{f}_i$.
- The face area vector $\mathbf{S}_f$ is located at each face centroid and point outwards the control volume, has a value equal to the area of the surface and is perpendicular to it.
- The vector that goes from the point **p** to the point **f** is labeled **pf**

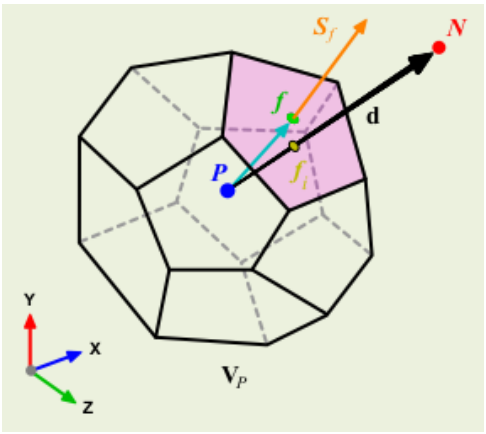


Figure 9: Finite volume cell with a cell-centered approach [10].

4.1.2 Discretization of the fluid dynamic equations [11][10]

The Navier Stokes equations, the heat equation and the continuity equation, shown in section 3.1, can be summarized, all together, with the use of the Reynolds transport

Equation	ϕ	Γ_ϕ	S_ϕ
Continuity	1	0	0
Momentum	$\vec{u}u$	ν	-∇p
Energy	c_pT	κ	0

Table 1: Parameters for the Transport equation

equation which is derived from the conservative law. The transport equation reads:

$$\underbrace{\int_{V_P} \frac{\partial \rho\phi}{\partial t}\, dV}_{Temporal\ term} + \underbrace{\int_{V_P} \nabla \cdot (\rho\vec{u}\phi)\, dV}_{Convective\ term} = \underbrace{\int_{V_P} \nabla \cdot (\rho\Gamma_\phi \nabla\phi)\, dV}_{Diffusive\ term} + \underbrace{\int_{V_P} S_\phi\, dV}_{Source\ term} \tag{4.2}$$

where ϕ can be any kind of scalar or vectorial variable of a continuous media, Γ_ϕ is the diffusivity of the variable in the media, and S_ϕ is a source term.

Table 1 shows the shape of the transport equation in order to get the continuity, momentum and heat equations. It is important to underline that on the FVM, the transported fluid property ϕ is defined in the control volume as follows:

$$\phi = \phi_P = \frac{1}{V_p} \int_{V_P} \phi(x) dV \tag{4.3}$$

As it can be observed on equation 4.3, on the finite volume method, a fluid property is defined in the center of the cell as a ratio of the value of this property integrated along the cell.

Moreover, to obtain a completely discretized form of the previous terms, a discretization of the following terms is needed.

- **Interpolation of the cell-center value ϕ.** The most common and the one chose during this project is the linear interpolation, computed as a linear Taylor expansion, that writes as follows:

$$\phi_p = \phi_f + (\vec{x} - \vec{x_f})\nabla\phi_f \tag{4.4}$$

  Basically, equation 4.4 tells how the value at the centroid of the cell is obtained from the value at the centroid of the faces. Notice that, in order to have a fully discretize scheme, a discretization of the term $\nabla\phi_f$ is needed.

- **Discretization of face gradient $\nabla\phi_f$.** On the simulations a orthogonal correction

scheme is selected.

$$\nabla\phi_f = f_x\nabla\phi_P + (1 - f_x)\nabla\phi_N \tag{4.5}$$

Notice that in order to have a fully discretize scheme, a discretization shceme for $\nabla\phi_P$ and $\nabla\phi_N$ is needed.

- **Discretization of the center-cell gradient $\nabla\phi$.** The most common discretization, and the one used on the simulations, is through the Gauss theorem, that reads:

$$\int_{V_P} \nabla \cdot \vec{E}\ dV = \int_{S_f} \vec{E} \cdot d\vec{S} \tag{4.6}$$

where $\vec{E}$ is a vector field that in fluid dynamics can be either the convective flux ($\rho\vec{u}\phi$) or the diffusive flux ($\rho\Gamma_\phi\nabla\phi$).

Notice that the schemes selected for the interpolation, $\nabla\phi_f$ and $\nabla\phi$ only have influence in the general discretization and may be different for the definition of the convective and diffusive fluxes.

Once the four terms of the transport equation are presented, and the interpolation and the gradients discretizations are specified, the treatment for each of the terms can be fully described.

4.1.2.1 Temporal term

The discretization chosen for the temporal derivatives is as follows:

$$\left(\frac{\partial\rho\phi}{\partial t}\right)_p = \frac{\rho_p^{n+1}\phi_p^{n+1} - \rho_p^n\phi_p^n}{\Delta t} \tag{4.7}$$

where n and n+1 are two consecutively time-steps. This is an implicit Euler scheme, that is first order accurate, but avoid oscillations common in melting problems due to the restrictive stability limitations [3]. Respect to the time integrals, the scheme selected is as follows:

$$\int_t^{t+\Delta t} \phi(t)dt = \theta\Delta t\phi(t + \Delta t) + (1 - \theta)\phi(t)\Delta t \tag{4.8}$$

where $\phi(t)$ is the temporal function of a fluid variable, and θ is an integration parameter that for a value of one delivers a fully implicit scheme and for a value of zero is a fully

explicit scheme. In this project $\theta = 1/2$, yielding to an hybrid scheme.

$$\int_t^{t+\Delta t} \phi(t)dt = \frac{1}{2}(\phi^{t+\Delta t} + \phi^t)\Delta t \tag{4.9}$$

This temporal scheme is known as Crank-Nicholson, and it is second order accurate but oscillatory. However, oscillations did not appear on the results in this case, reason why this scheme was selected.

4.1.2.2 Convective term

From the discretization of the cell center gradient described in equation 4.6, the convective term is transformed as follows:

$$\int_{V_p} \nabla \cdot (\rho\vec{u}\phi)\, dV = \int_S (\rho\vec{u}\phi)\, dS \tag{4.10}$$

By this, the volume integral that is the convective term is transformed into a surface integral . Then, the centroid of the control volume surfaces are defined as:

$$\int_S (\vec{x} - \vec{x_f})\, d\vec{S} = 0 \tag{4.11}$$

Then the interpolation described in equation 4.4 is employed to get the value of the cell centroids ϕ from the values at the faces ϕ_f. This interpolation, together with the above definition of centroid, allows to fully discretize the convective term without the need of a discretization of the $\nabla\phi_f$ term.

$$\int_S \rho\vec{u}\phi\, d\vec{S} = \sum_f \int_f \rho\vec{u}\phi\, d\vec{S} = \sum_f (\rho\vec{u}\phi)_f \vec{S_f} \tag{4.12}$$

The term $(\rho\vec{u}\phi)_f$ is the convective flux F_c of a fluid property going through a face. However, to compute the convective flux, the value of the variable ϕ_f over all the faces must be computed. For this aim, an interpolation scheme to obtain the value of the fluid property at the faces ϕ_f from the value of the fluid property at the centroid ϕ of the cell is employed. The scheme selected on the computations is the linear upwind, that is bounded and first order accurate.

$$\phi_f = \phi_{PN} = \begin{cases} \phi_P & if \;\; F_c \geq 0 \\ \phi_N & if \;\; F_c < 0 \end{cases} \tag{4.13}$$

4.1.2.3 Diffusive term

As happened with the convective term, Gauss theorem together with the interpolation scheme defined in equation 4.4, and the centroid definition is applied on the diffusive term of the transport equation:

$$\int_{V_P} \nabla \cdot (\rho\Gamma_\phi \nabla\phi)\, dV = \int_S (\rho\Gamma_\phi \nabla\phi)\, d\vec{S} = \sum (\rho\Gamma_\phi \nabla\phi)_f \vec{S_f} \tag{4.14}$$

Where the term $(\rho\Gamma_\phi \nabla\phi)_f$, is the diffusive flux F_D. To obtain the value of the diffusive flux in each face, a discretization of the divergence of the fluid variable $\nabla\phi_f$ at the boundary of the cells is needed. For this aim, first the discretization of the fluid property gradient at the centroid faces $\nabla\phi_f$ used in the simulations is applied:

$$\nabla\phi_f = f_x(\nabla\phi)_P + (1 - f_x)(\nabla\phi)_N \tag{4.15}$$

where $f_x = \frac{||f\vec{N}||}{||P\vec{N}||}$. This surface gradient scheme is known as orthogonal correction, that matches with the general surface gradient scheme. Then, a Gauss discretization is chosen for the gradient at the centroid of the cells $\nabla\phi_P$ and $\nabla\phi_N$, that together with a linear interpolation scheme delivers the fully discretize form of the gradient ϕ_f on the diffusive term:

$$(\nabla\phi)_f = \frac{\phi_N - \phi_P}{||P\vec{N}||} \tag{4.16}$$

For all this considerations, an orthogonal and equidistant mesh, like the one on the simulations, have been considered.

4.1.2.4 Source term

The value of the source terms where obtained by applying the mean value theorem:

$$\int_{V_P} S_\phi\, dV \approx S_{\phi,P} V \tag{4.17}$$

4.1.2.5 Final form of the discretized transport equation

If all the discretizations explained above are applied for each one of the terms of the transport equation (4.2), the "semi-discretized" form of the transport equation is obtained [38]:

$$\int_t^{t+\Delta t}\left[\int_{V_P}\left(\frac{\partial\rho\phi}{\partial t}\right)_P dV_p + \sum_f(\rho\vec{u}\phi)_f\vec{S_f}\right]dt = \int_t^{t+\Delta t}\left[\sum_f(\rho\Gamma_\phi\nabla\phi)_f\vec{S_f} + S_{\phi,P}V_P\right]dt \tag{4.18}$$

Then, consider constant density and diffusivity in time, the temporal scheme selected is applied:

$$\begin{aligned}\frac{\rho_p\phi_P^{n+1}-\rho_p\phi_P^n}{\Delta t}V_P + \Big(\sum_f(\rho\vec{u}\phi)_f^{n+1}\cdot\vec{S_f} + (\rho\Gamma_\phi\nabla\phi)_f^{n+1}\vec{S_f}\Big) = \\ \frac{1}{2}\Big(\sum_f(\rho\vec{u}\phi)_f^{n}\cdot\vec{S_f} + \sum_f(\rho\Gamma_\phi\nabla\phi)_f^{n+1}\vec{S_f}\Big) + S_{\phi,P}V_P\end{aligned} \tag{4.19}$$

From this last equation, the solution for the problem ϕ_P at time t_{n+1} can be obtained. For this aim a fully implicit equation system, depending on the values of the fluid variable and its value at the boundaries, is solved. This equation system reads as follows:

$$a_P\phi_P^{n+1} + \sum_{nc}a_{nc}\phi_{nc}^{n+1} = b_\phi \tag{4.20}$$

where the $a's$ are coefficient depending on the convective and diffusive fluxes in the control volume P, the suffix nb refers to the values at the centroid of the neighbour cells and b_ϕ stands for all the terms considered as known (calculated in previous iterations). This equation system works for every cell centroid of every cell, therefore for every CV, the unknown ϕ_P^{n+1} depends of the solution in adjacent cells. A system of equations that allows to obtain the value of ϕ_P^{n+1}:

$$\mathbf{A}\vec{x} = \vec{b} \tag{4.21}$$

Where **A** is the matrix of the a's coefficients. In this project the solution to this lineal system of equations is obtained using a GAMG (Geometric algebraic multi-grid) method with a DILU (Diagonal-based Incomplete Lower-Upper) preconditioner to help convergence.

Furthermore, to ensure stability, the different fields and equations are subjected to

under-relaxation. The under-relaxation of the solution, consist in bounding the maximum change that a variable can suffer in a time-step or iteration. Like this, destabilization of the solution is avoided, helping to ensure that the solution obtained is an exact solution of the numerical scheme within a tolerance. To relax a solution the following formula should be applied:

$$\phi^{n+1} = \gamma\phi^{*} + (1-\gamma)\phi^{n} \tag{4.22}$$

where ϕ^{*} is the value of the computed variable at iteration n+1 before under-relaxation, ϕ is the value of the variable at iteration n and ϕ^{n+1} is the value of the variable at iteration n+1 after under-relaxation. The higher is the under-relaxation factor γ, the more stable will be the numerical solution. On the other hand, lower values of the under-relaxation factor ensure a faster convergence. Thus, at the end, the value of the under-relaxation factor is a compromise between convergence velocity and stability.

4.2 OpenFOAM: A finite volume method environment

For the computations performed during the project, the software `OpenFOAM` was the tool chosen. `OpenFOAM` is a free open-source software conceived mainly to perform computational fluid dynamics (CFD) problems based on a finite volume discretization, though it is useful for other fields of engineering and science like structural dynamics, electromagnetism, etc...

In this section, a brief explanation on the functioning of the `OpenFOAM` software is done, including details of the implementation of the case in the present project.

4.2.1 Structure of a case in OpenFOAM [12]

In order to be able to run a simulation, `OpenFOAM` depends on a serial of files that will set the different parameters of each specific case like geometry, boundary conditions, initial conditions, discretization schemes, etc...The figure below describes a general set up, with the pertinent files, for a heat transfer problem like the one studied during this project. A sample of the files used in a case of this project, where it can be seen in detail each of the parameters selected on the different dictionaries, is provided in Appendix D.

Notice that **0**, **constant**, and **system** are directories, while the other elements are files. Also, notice that some files have an "*" symbol alongside its name, this is to remark that these files are required always in `OpenFOAM` no matter what kind of problem one is trying to solve. Nevertheless, while the directory 0 is mandatory, the need of the different

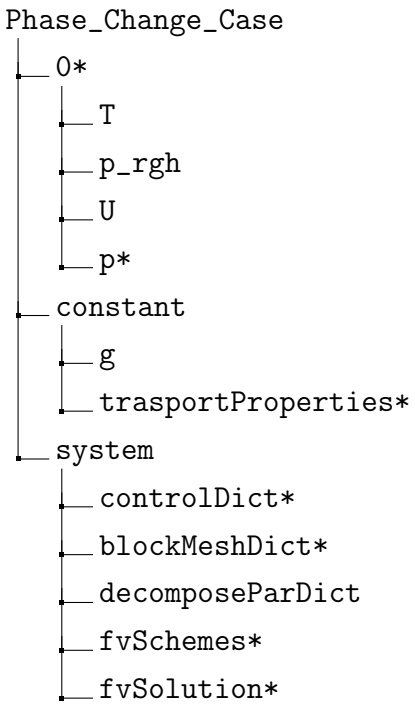


Figure 10: General structure of an OpenFoam case

files that should be supplied depends on the equations solved in each specific case.

4.2.1.1 The 0 directory

The "0" directory contents all the information regarding the boundary conditions at time zero and the initial conditions of the case. In these files, one should specify everything that is known about the fields (temperature, velocity...) beforehand. Similar to this directory, there are the time directories, that once the program has been run, contain information about the fields at different times.

4.2.1.2 The constant directory

Like the own name address, contains information about the fields that are constant through all the simulation (e.g.: gravity) and physical properties for the application concerned. Furthermore, once the mesh is created, with the command `blockMesh`, the directory **polyMesh** appears containing all the information relevant to the mesh (points, faces...).

4.2.1.3 The system directory

This directory contains all the files with parameters associated with the solution itself and the control over it. A brief description of the files employed in this project is provided:

- **controlDict:** in this file all the parameters concerning time in the computation are established. Referring, among others, the time step, the computational time, the time directories created, etc...

- **blockMeshDict:** in this file, the computational domain and all the concerning parameters are configured. This include, the size and shape of the computational domain and the mesh, and the definition of the boundaries. Once the `blockMeshDict` dictionary is set properly, with the command `blockMesh` the mesh is created and storage in the directory **polyMesh** inside **constant**.

- `decomposeParDict:` since during this project, parallel computation techniques have been employed, with the help of a cluster of computers, the `decomposeParDict` dictionary is required. In this file, the parameters regarding the decomposition of the mesh and the fields to be computed separately, are defined. The decomposition should be performed right before the starting of the computations, once the mesh and additional fields are set, with the command `decomposePar`. Once the computations are done, the solution fields and the mesh is reconstructed with the `reconstructPar` command.

- **fvSchemes:** in this dictionary the schemes selected for the discretization of the derivative terms are defined. A scheme should be precise for each of the following terms:

  - Time schemes, labeled in `OpenFOAM` as `ddtSchemes`, discretize the temporal derivatives $\frac{d}{dt}, \frac{d^2}{dt^2}$.
  - Gradient schemes, labeled as `gradSchemes`, discretize the gradient terms $\nabla\phi_P$.
  - Divergent schemes, labeled as `divSchemes`, discretize the convective terms $\nabla \cdot (U\phi)$
  - Laplacian schemes, labeled as `laplacianSchemes`, discretize the diffusive terms $\nabla \cdot (\Gamma_\phi \nabla\phi)$
  - Interpolation schemes, labelled as `interpolationSchemes`, contains the information of values at the faces are obtained from the values at the centres. The most common is the linear interpolation.

$$\phi_f = f_x \phi_P + (1 - f_x)\phi_N \tag{4.23}$$

where,

$$f_x = \frac{fN}{PN} = \frac{|x_f - x_N|}{|d|} \tag{4.24}$$

  - snGradSchemes, labeled `snGradSchemes`, discretize the surface normal gradients at the faces $\vec{n}_f \cdot \nabla\phi_f$.

- **fvSolution:** contains several sub-dictionaries that control the solver parameters. There are three main sub-dictionaries:

  - Linear solver control: the control of the linear solver parameters is specified on sub-dictionary `solvers`. It is important to distinguish between the linear solver, that describes the method of number crunching employed for solve the transport equations, with application solver that encompasses all the equations and algorithms employed to solve the problem. Here parameters as tolerance, the preconditioner, etc, can be chosen.
  - Relaxation factors: labelled as `relaxationFactors`, this sub-dictionary controls under-relaxation, a technique already described in this chapter, that improves stability on the solution. Different factors of relaxation can be chosen for the different variables of the solution.
  - Algorithm selection: here the method used to coupled the transport equations between them is selected among SIMPLE, PIMPLE and PISO. Beyond this, other parameters regarding the control of the solution can be provided here.

  In the `fvSolution` dictionary other parameters, even beyond the resolution of the transport equation, can be provided in order to control the solvers and the algorithms. For example in this project, if natural convection effects want to be neglected in the solution, the `onlyConduction` parameter will be set to **1**. By this, the application solver will call the file `TEqn_onlyConduction.H` where the equation solved is the pure diffusive heat equation.

4.3 Description of the solver

In the soul of enthalpy-porosity methods, a specific `OpenFOAM` solver have been developed to solve the phase change problematic proposed in this project. This section aims to give

light on the algorithm used during the simulations and the implementation of itself in `OpenFOAM`, so anybody who wishes to implement a phase-change solver can find a solid guide in this document.

The solver in question is an adaptation to `OpenFOAM 4.1` of a previous solver for the version 2.3. Major changes have been produced to achieve the actual state of the solver:

1. Adaptation of the `OpenFOAM 2.3` environment to a environment suitable with `OpenFOAM 4.1`. Including major changes in the definition of the functions, variables and operations.

2. Arrangement of the solver to compute phase change on materials without inclusions. The original solver was set with a model that was only able to work with **materials with nanoparticles of metal encrusted** to enhance the conduction. This model has been removed and adapted to be able to compute phase changes on materials without any kind of inclusions. This included major changes in several files where the nanoparticles had an impact in different fields related to the heat transfer (i.e κ c_p, f...)

3. Remove of the scalar field `x_grad`. That was a field created to compute the centre of the cells to help the computation of the Marangoni effect. Nevertheless, this field made impossible the computation in parallel, because the `descomposeParDict` dictionary was not able to split this field into several domains This is the reason why this field was removed and a new Marangoni boundary condition was adapted.

4.3.1 General algorithm

One of the most important aspects of the simulations performed in this project is the explanation of the solution algorithm. Algorithm refers to the set and order, of the operations performed over the computational domain to obtain, from the boundary and initial conditions, the physical state of the studied computational domain in a moment forward in time. In summary, is the iterative process, that based in the equations presented in chapter 3 and with the backup of the numerical methods explained in sections before, allow to compute the variables of interest on the thermal-fluid problem in space and in time.

Once again, it is underlined not to confuse the solution algorithm, explained in this section, with the linear solver, that is the iterative number-crunching method employed to solve the transport equations contained in the algorithm (i.e. energy, momentum and continuity equations). Figure 11 shows a general, and simplified, scheme of the algorithm

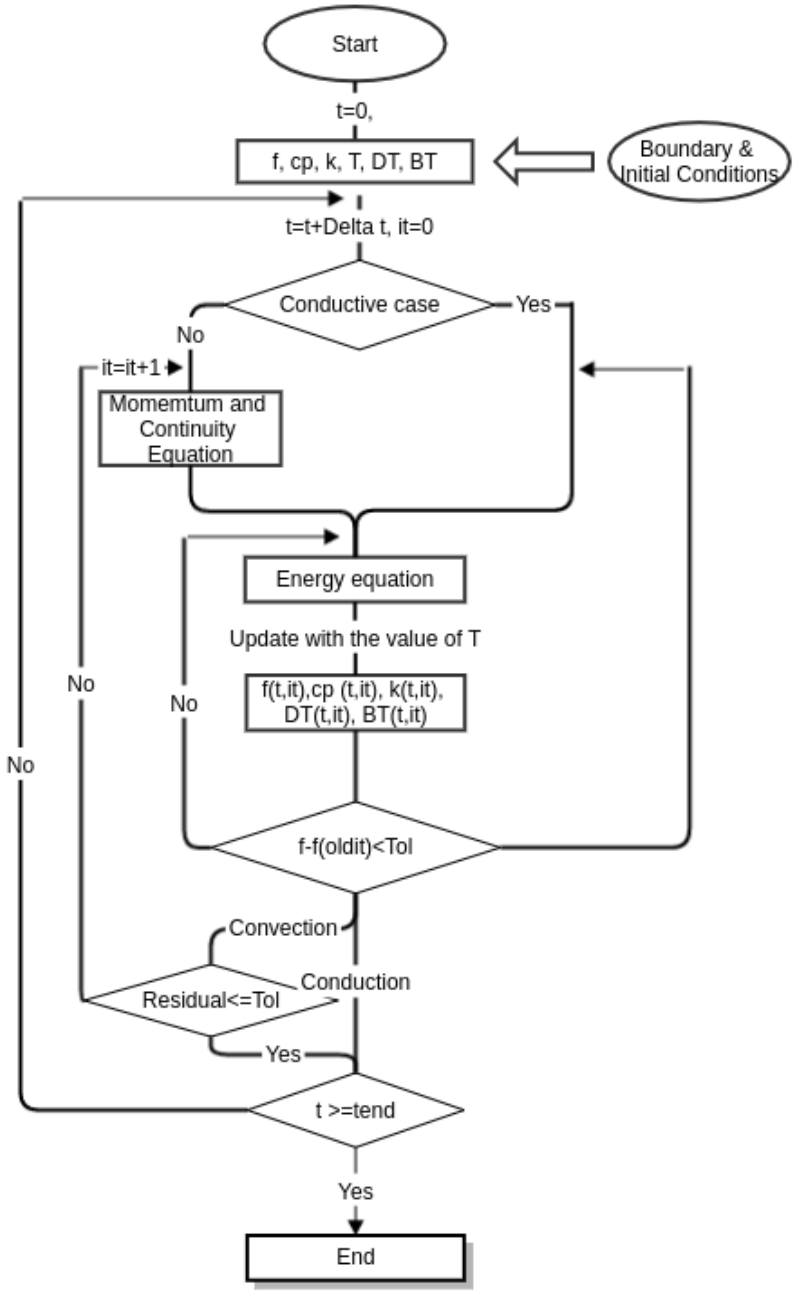


Figure 11: Flowchart of the numerical algorithm

implemented to accomplish for melting/solidification processes with the enthalpy-porosity method. For a better understanding of the process, the developments made in chapter 3 shall be present in the reader.

The algorithm starts with the computation of the initial conditions in all the computa-

tional domain from the boundary and initial conditions that are known. These parameters are the temperature T, with which the liquid fraction f can be computed. Then, with the liquid fraction, the Darcy term (DT) and buoyancy/Boussinesq term (BT), which only apply to the convective case, and the physical properties of the material are obtained.

Once the initial conditions are all set, the algorithm moves forward into the first time step to compute the correspondent solutions at this time. First, the algorithm will take one out of two paths depending on if the case of study is only conductive ($g = 0$) or if it also considers heat transfer by convection due to buoyancy effects ($g \neq 0$).

- **Conductive Case:** when the effects of gravity on the fluid are neglected, the solution will be obtained only using the diffusive heat equation, with the source term that takes into account the latent heat enthalpy, as described in section 3.5.1.

  To solve the conductive heat equation, the algorithm enters an iterative loop depending on the average change of the liquid fraction with respect its previous value. In detail, the algorithm starts solving the heat equation with the number-crunching method chosen, GAMG in this case, until reaches a value within a fixed tolerance. When this happens, the computed value of the temperature field is employed to calculate the liquid fraction at every cell and with it, all the material properties (c_p, κ...) are recalculated for the new f. If the value of the computed liquid fraction f^{n+1} minus the value of the liquid fraction at the previous iteration f^n (or at the previous time-step if it is the first iteration), is equal or less than a tolerance value, the solution is considered converged. If not, the heat equation is solved again with the updated fields of properties and the liquid fraction, and the process is repeated until convergence is reached. Once the solution is converged, the algorithm advances to the next time step. The whole process is repeated, for each time-step until the algorithm reaches the simulation time fixed as the limit.

- **Convective Case:** when the heat transferred by convective effects are taken into account in the problem, the buoyancy term because of gravity forces, can not be neglected. Therefore, since the presence of a moving fluid shall be accounted, the whole set of the transport equations (momentum, continuity and energy) shall be solved, as specified in section 3.5.2. For solving the linear algebraic system of equations an algorithm base on the SIMPLE algorithm presented by Patankar [40] is used.

  The algorithm starts by solving the velocity equation with the values of the variables at the previous iteration/time and the uncorrected mass and pressure fluxes are computed. Then the corrected continuity equation is solved and velocity com-

ponents, pressure and mass fluxes are corrected. Right after, the algorithm enters into a loop depending on the liquid fraction and solves the heat equation, which is coupled through buoyancy with the momentum equation. Then exactly as in the heat equation, this process is repeated, updating the different fields and source terms (DT, BT) at each iteration, until the fields of temperature and liquid fraction are consistent. Nevertheless, in this case, once the value of the liquid fraction has converged, the sum of the absolute residuals in the momentum and continuity equation is checked. If they are above the tolerance, the algorithm starts all over again from the momentum equation with the updates values. On the other hand, if the residuals are under the tolerance value, the algorithm goes to the next time step and repeat the process until the simulation time fixed as the limit is reached.

4.3.2 Implementation of `PCMsolver 4.1` in `OpenFOAM 4.1`

`PCMsolver` is the name of the solver developed by third parties for `OpenFOAM 2.3` capable to perform phase change simulations. However, as explained at the beginning of section 4.3, this solver has been rearranged by the author, to fulfil the requirements needed during the simulations of this project. From now on, and for avoiding any confusion, this rearranged version of the solver will be called `PCMsolver 4.1`.

Although `OpenFOAM` has several solvers capable of dealing with heat transfer processes driven by buoyancy, none of them contemplates phase change processes with Darcy and enthalpy source terms. Thus, the original solver (`PCMsolver`), was developed based in `buoyantBoussinesqPimpleFoam` of `OpenFOAM 2.3`, that is a transient solver for buoyant, incompressible and turbulent flow already distributed with the `OpenFOAM` software. Knowing this, it can be explained how to obtain the modified solver `PCMsolver 4.1` from the `buoyantBoussinesqPimpleFoam` of `OpenFOAM 4.1`.

4.3.2.1 Modifications on createFields.H

Most of the modifications on this file are linked to two aspects: remove turbulent related fields that, since turbulence is not accounted by the solver, are not needed, and include fields related to the liquid fraction, the enthalpy source term and the Darcy term. The most important modifications are:

- **Remove of the incompressible turbulence model and the turbulence kinematic thermal diffusivity,** since as mentioned above, the solver does not account for turbulent cases. However, this may not be erased if the solution aims to account

for turbulence in the convective flow, as the model presented by Wiwatanapataphee [41].

- **Introduce the liquid fraction term,** that in function of the temperature, as specified in chapter 3 will compute the value of the liquid fraction on each cell.
- **Introduce c_p and κ as fields,** instead of constant values since they will take a ponderate value between the solid and liquid values depending on the liquid fraction.
- **Remove the Moving reference frame,** because as explained in precedent chapters, the enthalpy method uses a fixed grid.

The modifications in this file shall be accompanied with the consequent modifications on the `readTrasnportproperties.H` file.

4.3.2.2 Modifications on UEqn.H

The modifications in this file correspond to the elimination of the turbulence term and the addition of the Darcy source term in the momentum equation. Moreover. all the implementations related to the moving reference frame shall be removed.

4.3.2.3 Modifications on pEqn.H

In this file, there are no further changes beyond removing options and terms related to turbulence and the moving reference frame.

4.3.2.4 Modifications TEqn.H

The most relevant changes are produced in this file, due to the implementation of the enthalpy source term, and the mathematical complexity of the attached liquid fraction computation, that requires of an outer iterative loop to satisfy its continuity. Furthermore, since the customized solver is able to deal with both, conductive and convective problems, the conformation of two `TEqn.H` files, one for each kind of problem is needed. These two files are basically equal except for the convective term that is neglected on the file that deals with conduction. Beyond that, the changes produced in the `TEqn.H` file are as follows:

- **Remove all turbulent and radiant heat transfer parameters,** since once again, these effects are not considered on the present solver.

Then a loop, depending on the liquid fraction residual, must be implemented containing:

- **Computation of the enthalpy source term,** as described in section 3.3.3, that should be implemented before the energy equation, because is part of the energy equation itself.
- **The energy equation** that will include a term for the enthalpy source, and a term for the convective transfer in case it shall be accounted.
- **Actualization of the liquid fraction,** with the new value of T computed. Also, the c_p and κ are updated with the new value of the liquid fraction. Once out of the loop, Darcy and buoyancy terms are updated for the momentum equation of the next time step.

4.4 Boundary conditions

A fundamental information for the resolution of a problem are the boundary and initial conditions (respectively shorten BC and IC). These conditions give a starting point to compute, from the discretization equations, the solution of a problem in space and in time.

In this section, the boundary conditions employed in chapters 5 and 6 are described. Following the distinction made by H.Jasak, two kinds of boundary conditions are distinguished in the definition of the problem: physical and numerical [11].

4.4.1 Physical boundary conditions

During the simulations, melting/solidification processes have been studied for two different geometries:

1. A square of surface $2cm \cdot 2cm = 4cm^2$
2. A rectangle of surface $1 \cdot 4 = 4cm^2$

Notice that the area of both geometries is the same in order to compare properly the influence of the shape in the heat transfer. Moreover, also notice that the geometries considered are surfaces, thus, a plane of symmetry is considered on the z direction neglecting all possible 3D effects.

From figure 12, five boundaries of the geometry can be differentiated:

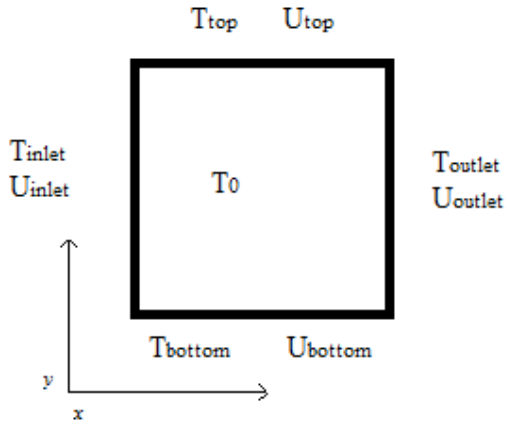


Figure 12: Configuration of the computational domain

- **Inlet:** is considered to be a regular wall, therefore, the no-slip condition is applied.
- **Outlet:** consider an adiabatic wall, therefore, not heat exchange and non-slip conditions are assumed.
- **Top:** is the only boundary that changes depending on the case. On one hand, for the conductive and convective cases it is considered to be an adiabatic wall, no heat transfer through the wall and non-slip condition, on the other hand, for the Marangoni case, a free surface in contact with a gas is considered, what is translated into a gradient of velocity.
- **Bottom:** it is an adiabatic wall, therefore, no heat transmission and non-slip conditions are applied.
- **Back&Front:** because the problem is solved for a 2D geometry, a symmetry plane is considered in the z direction. Thus, no more specifications on the boundary conditions are needed for this boundaries.

4.4.2 Numerical boundary conditions

There are two basic types present in this project [11]:

- **Dirichlet boundary condition:** that fixes a value of a variable in the boundary.
- **Von Neumann boundary condition:** that fixes a gradient of a variable in the boundary.

The table below shows the general form of the boundary conditions in each of the studied cases. A distinction, referring the boundary conditions, is made between cases with presence of Marangoni convection and cases without the presence of this phenomena.

	Conduction/Convection	Marangoni
Inlet	$T = T_{inlet}$ $\vec{u} = 0$	$T = T_{inlet}$ $\vec{u} = 0$
Outlet	$\frac{\partial T}{\partial y} = 0$ $\vec{u} = 0$	$\frac{\partial T}{\partial y} = 0$ $\vec{u} = 0$
Top	$\frac{\partial T}{\partial y} = 0$ $\vec{u} = 0$	$\frac{\partial T}{\partial y} = 0$ $\nabla \vec{u} = -\frac{\partial \sigma}{\partial T}\frac{\nabla T}{\mu}\vec{x}$
Bottom	$\frac{\partial T}{\partial y} = 0$ $\vec{u} = 0$	$\frac{\partial T}{\partial y} = 0$ $\vec{u} = 0$

Table 2: Boundary conditions for cases in chapters 5 and 6.

.

The term $\nabla \vec{u} = -\frac{\partial \sigma}{\partial T}\frac{\nabla T}{\mu}\vec{y}$, on the top boundary in cases with Bénard-Marangoni convection, comes from the influence of a free surface, in contact with a fluid, on the flow as explained in section 3.5.3. The gradients of temperature produce a difference in surface tension that causes the movement of parts of the fluid with lower surface tension into parts of the fluid with higher surface tension.

For the concrete values of the boundary conditions, one should address to the table of the case of interest in the Appendix B.

4.5 Convergence Study

In this section, the convergence study of the numerical solution with the exact solution of the differential equation is discussed. If the problem converges, this ensures that the solution obtained with the discretized equations is a real solution of the differential equations that govern the problem [38].

The Lax theorem reads: "*for a well-posed initial value problem and a consistent discretization scheme, stability is the necessary and sufficient condition for convergence*"[38]. Figure 13 gives a schematic view of this theorem and the relationship between the principal terms in convergence studies. To explain better this theorem about solution convergence, some terms shall be defined:

- **Consistency:** it is bounded to the numerical discretization. A numerical equation is considered to be consistent when it tends to the differential equation with the

decreasing value of the time and space steps. The parameter that measures the error between the numerical equation and the differential equation is the truncation error ϵ_T, which for a transient 2-D problem as it is the case, reads:

$$\epsilon_T = O(\Delta t^q, \Delta x^p, \Delta y^r) \tag{4.25}$$

where Δx, Δy, and Δt stand, respectively, for the length of a cell in the x-direction and the y-direction, and the time-step, and q,p and r are the order of the error in time and in the two space dimensions. Thus, following the description of consistency and applying it to equation 4.25 yields:

$$\lim_{\Delta x, \Delta y, \Delta t \to 0} \epsilon_T = 0 \tag{4.26}$$

what ensures, that a numerical scheme is consistent when the truncation error, and therefore space and time-steps, tend to zero.

- **Stability:** it is a property of the numerical solution, thus independent of the differential equation. A solution of the numerical scheme is said to be stable when the error between this solution and the exact solution of the numerical scheme is bounded during all the iterative process.

- **Convergence:** is a property of the numerical solution. A solution is considered converged, when is an exact solution of the differential equation within a given tolerance.

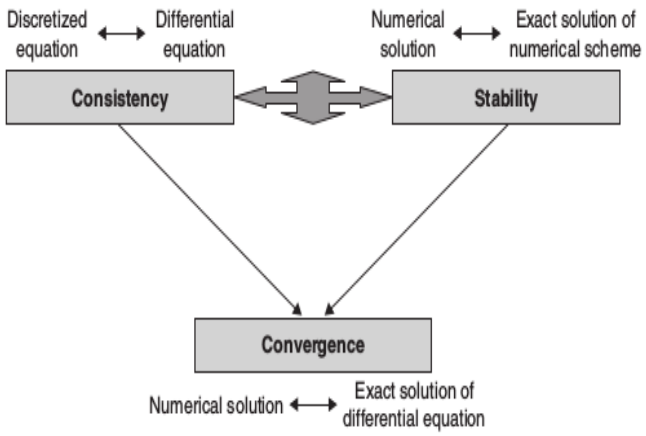


Figure 13: Scheme of the relations between consistency, stability and convergence. Source [38].

Based on this definitions and on the Lax theorem, different studies have been performed to enssure the convergence of the numerical equation solution with the solution

of the differential equation. The first step was to choose an adequate and consistent discretization, what was already done in this chapter in section 4.1.2. The next steps, which follow this introduction, are: a mesh and time convergence study, to ensure the independence of the solution with the time and space discretization (i.e. $\epsilon_T \rightarrow 0$), and a stability analysis by controlling that the solution reaches a steady state after certain simulation time.

Notice that the whole study is done for a homonym case to the one presented in section 6 for a square geometry with natural convection.

4.5.1 Mesh Convergence

The definition of the mesh is a step of high importance in the resolution of a numerical case. If the mesh is not set properly (i.e. not fine enough), the solution may never converge or, if it does, the solution may vary too much from the exact solution. Furthermore, the finer is a mesh, the geometry will be divided into more cells, what means more computations are performed per time-step increasing the time that will take a simulation to run. Knowing this, the study of convergence is done with the objective to find a mesh that without losing relevant accuracy, allows computing the solution with the lowest possible computational cost.

It is known from the literature, that the size of the grid has a high impact on the solution of melting and solidification processes. Although many authors have employed coarse grids for this kind of process [2] [28], around 40x40 cells, later studies have shown that what was considered mesh convergence, was indeed, far from converging [4]. For this reason, the analysis for mesh convergence shown here arrives at values of Δx and Δy several orders of magnitude below of the common values. Nevertheless, with the help of a computer cluster, as the one employed during this project, a very high-resolution mesh can be used, within acceptable computational times (15-20 hours for the chosen grid size after this convergence study).

Due to the method employed (i.e.: enthalpy-porosity technique), the problem is solved with a fixed grid. Furthermore, since the geometries employed in the simulations are simple regular polygons, a square and a rectangle, a structured grid seems to be the best option.

4.5.1.1 Mesh convergence on a square geometry

The study of the mesh convergence is done for a convective case with fixed time step and with the same parameters as the ones given in table 7 in the Appendix B. Grids of 200x200, 300x300, 400x400, 500x500 cells have been tested. The evolution of the liquid fraction in time is proposed as the variable to test the accuracy of each of the meshes. The case employed for the study is exactly the same solved in section 6.2 for a square geometry of $2 \cdot 2 = 4cm^2$.

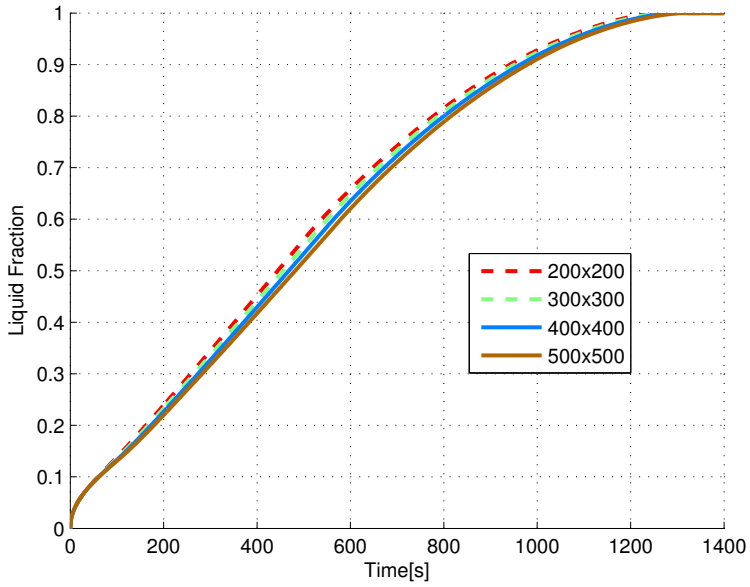


Figure 14: Mesh convergence analysis for a square geometry

Figure 14 shows the evolution of the liquid fraction in time for different cell densities. It can be noticed that the finer is the mesh, the larger is the melting time. The reason is that cells that for a coarse grid were melted when the mesh is refined, increasing like this the accuracy, they are not melted anymore.

From all the meshes studied, the one with 400x400 cells is chosen for the simulations. The reason is that the accuracy presented by this mesh is almost the same as the finest mesh (the difference in melting time is around 1.5%), while the computational time is sensibly lower. Computations in the cluster with the 400x400 grid take around 20 hours with 25 processors, while a 500x500 grid takes around 45 hours with 36 processors.

The chosen grid yield to a value of Δx and Δy as follows:

$$\Delta x = \frac{L_x}{N^\circ \, Cells \; in \; \vec{x}} = \frac{0.02m}{400} = 5 \cdot 10^{-5} m$$

$$\Delta y = \frac{L_y}{N^{\circ}\ Cells\ in\ \vec{y}} = \frac{0.02m}{400} = 5 \cdot 10^{-5} m$$

where L_x and L_y are respectively the length of the geometry in the `x` and `y`-directions. The importance of fixing these parameters rely on the fact that, if the simulation shall be performed on a square geometry of different size, rather than perform another convergence analysis, the same Δx and Δy can be kept to ensure convergence.

4.5.1.2 Mesh convergence on a rectangular geometry

The study of the mesh convergence for a rectangular geometry is done for grids of 300x200, 400x200, 500x200, 600x200 and 700x200. Notice that the length of the step on the `y`-direction is fixed with the information extracted from the convergence of the square geometry.

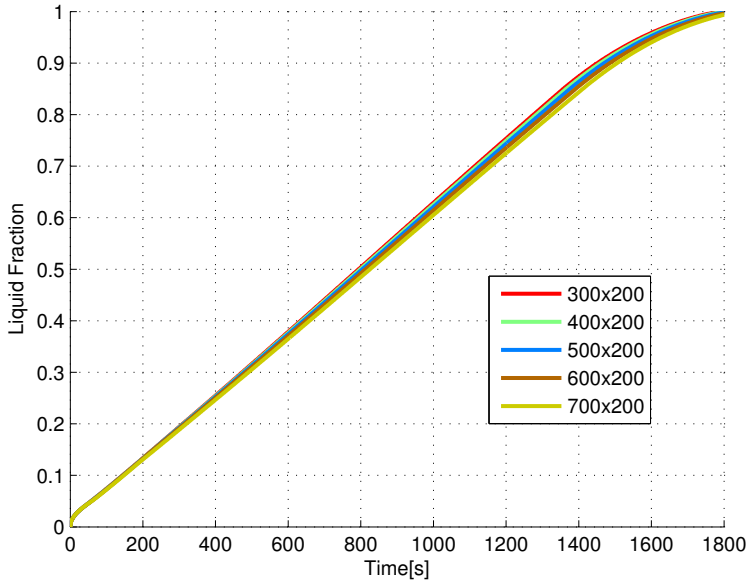


Figure 15: Mesh convergence analysis for a rectangular geometry

Figure 15 shows the liquid fraction in time for all the meshed studied. Again, as expected, the melting time becomes higher with the increase of the number of cells. This time, the mesh of 600x200 cells is considered due to the nice accuracy-computational time ratio that it performs. Then, using the same formula for compute Δx and Δy yields:

$$\Delta x = 6.67 \cdot 10^{-5} m$$

$$\Delta y = 5 \cdot 10^{-5} m$$

4.5.2 Time Convergence

In the same manner and for the same reasons than the grid refinement study, a time convergence study is performed. The aim, once again, is to choose a time-step that captures with accuracy the physics without compromising in excess the computational time. The convergence analysis shown in figure 16 has been done for a square geometry with a mesh size of 400x400 in order to assure that the solution is not grid size dependent and evaluate properly the error with the decreasing of the time step.

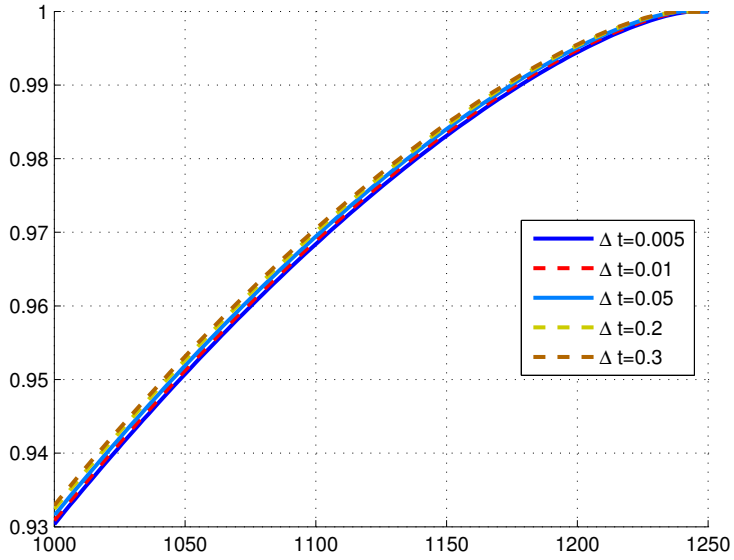


Figure 16: Time convergence for a square geometry

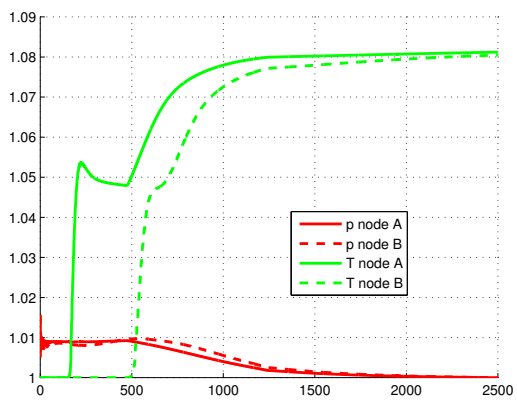
When analysing figure 16, it is observed that the rate of solidification decrease with the decreasing time-step. The error between the highest time-step to the lowest is very small, reason why only a small part of the plot is available to distinguish the progress of the error.

In spite of this time convergence analysis, the final choice for the time-step was an automatic time-step which is capable to increase or decrease the time step when needed in order to assure accuracy and stability.

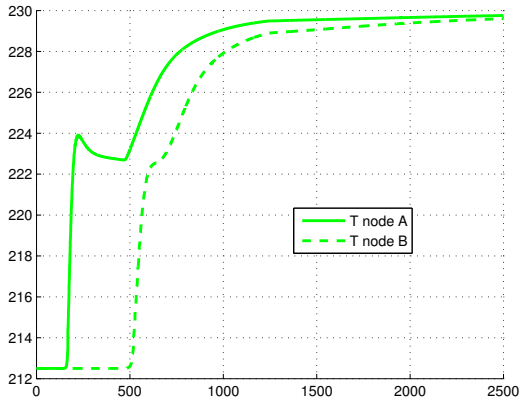
Now, that with this convergence analysis, an admissible space and time-step independence have been reached, it can be assured that the numerical equation is equivalent to the differential equation, and it just remains to check the stability of the solution.

4.5.3 Study on the convergence of the fields

The last step to assure, that the solution obtained by numerical methods is indeed a correct solution of the differential equation that governs the problem, is to control that during the iterative process the solution obtained does not deviates from the exact solution of the numerical scheme. Usually, in steady simulations, this would be proven by studying the residuals and checking that they decrease at each iteration. However, phase change processes are a transient phenomenon, thus a study on different fields along time will give better information to assess the convergence state of the solution.



(a) Normalized temperature and pressure along time for nodes A and B



(b) Value of T in node A and B along time

Figure 17: Field convergence study

In other words, once it is known that the numerical scheme is consistent with the differential equation, and once the truncation error has been minimized thanks to the mesh and time convergence analysis, the only step that remains to assure that the solution obtained, in a transient simulation, converges the exact solution of the numerical scheme, is by controlling the convergence of the fields to a value after a certain simulation time. This affirmation become obvious when one thinks about the nature of the problem of study, being not hard to guess that after a certain simulation time, a material that is submitted to a fixed temperature (surrounded by adiabatic walls) will reach a point at which the temperature in all the body will be equal to this fixed temperature inducing all the other fields to settle down.

This study is shown in figure 17a, where two of the most representative variables in the iterative process (remember that the velocity field is coupled to the temperature field) are monitored over time at two different points until the steady state is reached after around 2000 seconds. Where the node A is situated at $x = 0.49cm$, $y = 1.49cm$ and node B is at $x = 0.99cm$, $y = 0.99cm$. Here, it can observe how indeed, the temperature and the pressure field converge to a value after a given simulation time. Furthermore, the value at which convergence is reached for the temperature field (230K) is the same value as the fixed temperature of the wall, what gives physical consistency to the results obtained (figure 17b).

5 Validation of the Solver

In this chapter, the developed OpenFOAM solver, based on the enthalpy-porosity technique, is validated for heat transfer processes in phase transition. For this aim, different cases from the literature have been taken and replicated with the customize solver (`PCMsolver 4.1`. The results obtained are compared and discussed in order to assess the efficacy of the solver under different circumstances.

The chapter is divided into two sections, one where the solver performance is evaluated for various conductive cases, and another one where the solver is evaluated for a convective case. To assess the validity of the solver, the conductive case is compared with the analytical solution of the Stefan problem, whereas to assess it on convective cases, experimental data available in the literature has been used. In total, three cases are proposed, two for conduction and one for convection.

5.1 A comment on isothermal phase change

In general two kinds of phase change are considered in function of the way the latent heat is released (or absorbed) during freeze or melt:

1. **Non-isothermal:** it is the most common in phase change processes.[30] In this cases, the phase change occurs in a temperature range $T_m \pm \epsilon$, allowing the formation of the mushy region, which is the transition between the liquid and the solid phase, with a thickness depending on the range of temperatures where occur the phase change. It is Proper of alloys, and non-pure materials.

2. **Isothermal:** the phase change, and therefore the release (or absorption) of the latent heat occurs instantaneously at the melting temperature T_m. Thus, there is no mushy region, since the boundary between the solid and the liquid phase is a line without any thickness associated. However, when performing numerical computations, this line will have a small thickness coming from the discretization of the computational domain [28].

Most of the cases available in the literature, and the ones used for this validation deal with isothermal phase changes. However, from the definition of the liquid fraction given in section 3.3.3, the following issue is found when solving isothermal cases.

As commented before in isothermal phase change, the latent heat is all released at once at the melting temperature, avoiding the development of the mushy region. Therefore,

$$T_m = T_s = T_l \tag{5.1}$$

from the equation above, the liquid fractions reads

$$f(T) = \frac{T - T_s}{T_l - T_s} \rightarrow \infty \tag{5.2}$$

evidently, a liquid fraction of infinite value has no physical meaning. As a solution, the problem in `OpenFOAM` is solved for a change of phase produced within different ranges of temperature. The ranges selected, are formed around the melting temperature in the following manner:

$$T_s = T_m - \epsilon \tag{5.3}$$

$$T_l = T_m + \epsilon \tag{5.4}$$

This approach allows to quantify the influence of the size of the mushy region in the solution, and furthermore allows assessing the more suitable size of the mushy region (or temperature range) for simulate isothermal cases.

5.2 Validation of the solver for conductive problems.

From the description of conduction in chapters before, it is known that, in pure conduction processes the only equation involved is the energy conservation, neglecting convective terms ($\vec{u} = 0$) and taking into account the source term coming from the latent heat.

$$\frac{\partial c_p T}{\partial t} = \nabla(\frac{\kappa}{\rho}\nabla T) + S_h \tag{5.5}$$

This equation, and therefore this problem, is the only one of the phase change problems that has an analytical solution.[7] This analytical solution correspond to the Stefan problem, which among several other assumptions (refer to the book written by V. Alexiades and A.D Solomon[7] for specific information), considered a the isothermal melting of a semi-infinite slab of ice (or any solid material) with a fixed temperature at the boundary above the melting point.

After some developing, the Neumann solution for the location of the front reads:

$$X(t) = 2\lambda\sqrt{\alpha_L t} \tag{5.6}$$

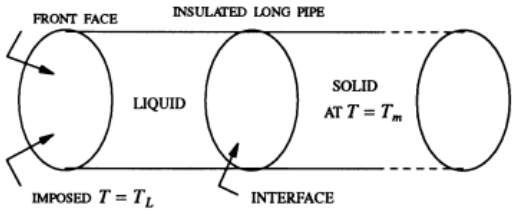


Figure 18: Configuration of the Stefan problem after some melting time

Where α_L is the thermal diffusivity of the liquid phase, and λ is the solution of a trascendental equation depending on the Stefan number (equation 3.37). As a last remark, notice that with the advance of the melting front, the liquid fraction can be easely computed as the average advance of the melting front

All this information has been used to develop a code in `MATLAB`, able to compute analytical solutions of the equation 5.5 under the assumptions of the Stefan problem. The code can be found in Appendix C.

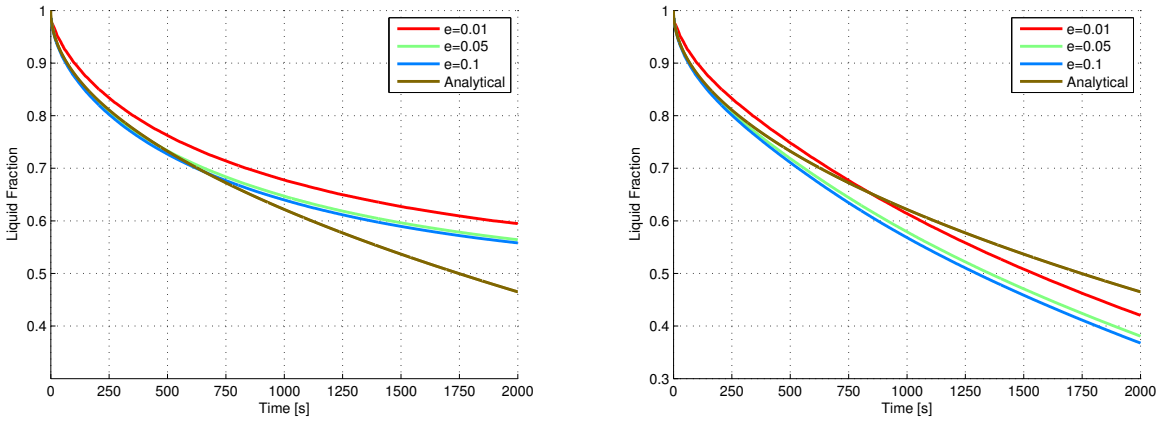
Two different cases are employed to evaluate the performance on conductive problems.

5.2.1 Case 1

The solver is tested for a conductive problem proposed by Voller in 1987[2]. In this scientific paper, the freezing process of a square geometry, considering purely diffusive phase change, is studied. With respect the data used for the computations, table 6 summarize the different parameters. The boundary conditions match with the ones presented in chapter 4.4 for conductive and convective cases, except for the temperature at the outlet where instead of an isothermal wall there is a fixed temperature. Regarding the discretization schemes, the mesh and the overall computational method, one shall refer to the development made in previous chapters. For further details on the method used by `Voller`, the aforementioned paper gives this information.

The analytical results (obtained with the `MATLAB` code, along with the results computed with the `PCMsolver 4.1` are shown in figure 19a for different ranges of temperature. When comparing the values obtained, the results are quite similar before t=1000s, where the maximum error is under 9% in the worst case ($\epsilon = 0.01$). After this point, the error begins to be non-acceptable, where the case closest to the analytical solution is around 20% at t=2000s.

In general, there seems to be some kind of delay in the moving of the freezing front, that is corrected with the increasing size of the mushy region. The reason for this delay may be found in that on the analytical solution, an adiabatic wall is considered at the



(a) Outlet wall at $T_{outlet} = 237.5$

(b) Adiabatic outlet wall

Figure 19: Rate of solidification computed with OpenFOAM and analytical methods

outlet, while in the computed solution, the wall is considered to be at a fixed temperature above the melting point. This produces that as the front approaches the outlet, the temperature of this boundary interferes in the advance of the front slowing down the advance of the front.

With respect the unexpected fact that the solution is more accurate as the size of the mushy region is larger, the reason is that, as the range of temperature, where the phase change occurs, is increased, the energy that needs the material to melt is lower (the latent heat for melting is still the same, but the sensible heat decreases in both phases as the mushy region becomes wider), what is translated into a faster advance of the melting front. Nevertheless, this correction of the velocity of the melting front with the size of the mushy region is not physical at all, and therefore, further evaluations of the solver must be done.

For this aim, a similar case is plotted in figure 19b, but this case considering an adiabatic wall at the outlet. This time, the results showed match with the expectations. The finer is the melting front considered, the better match the results with the analytical solution. However, the prediction made by the case that better adapts to the analytical solution ($\epsilon = 0.01$), still present some important discrepancies, making the solution not very physical. The reason of this may be found in that, in cases such as this one, where that small amount of latent and sensible heat is involved, the size of the mushy region, and so the range of temperature selected, have a great impact on the solution, making difficult to compute with accuracy isothermal phase changes for long periods of time. Thus, another case is presented with a higher amount of heat exchanged to prove the validity of this hypothesis.

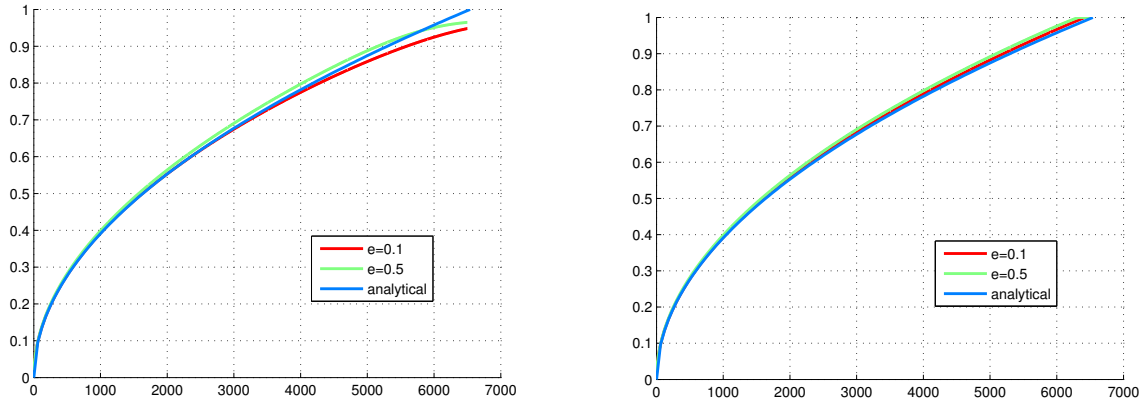
	Adiabatic case		Nonadiabatic case	
	Liquid fraction	Relative Error	Liquid fraction	Relative Error
$\epsilon = 0.01$	0.6139	1.25%	0.677	8.99%
$\epsilon = 0.05$	0.5792	6.83%	0.647	4.07%
$\epsilon = 0.1$	0.5682	8.6%	0.6398	2.9%
Anlytical	**0.6217**	**-**	**0.6217**	**-**

Table 3: Error of the case with respect analytical values at t=1000s.

5.2.2 Case 2

The solver is tested for a case with a greater heat exchange component. The parameters of this case are summarized in table 7 in the Appendix B, nevertheless, the solution has been computed for two ranges of temperature, around the melting temperature $T_m = 212.5$, computing T_l and T_s following equation 5.4 and 5.3with the values $\epsilon = 0.1, 0.5$.

Again computations have been made under the consideration of an outlet at a fixed temperature ($T_{outlet} = 212.5$) and considering this outlet an adiabatic wall. The analytical solution is employed as a reference value.



(a) Outlet wall at $T_{outlet} = 212.5$

(b) Adiabatic outlet wall

Figure 20: Rate of solidification computed with OpenFOAM and analytical methods

Figure 20 corroborates what was predicted in the case before: the solver gains in accuracy when the size of the mushy region does not affect, in a sensitive way, the amount of heat exchanged. Furthermore, the results obtained show a really high accuracy even for large periods of computation. Only a small discrepancy is shown in the case with a nonadiabatic wall, for the already commented influence of the outlet temperature.

Also notice, that in this case the properties in the solid and liquid phase are considered equal, while usually, this is not the case. Further influence of the mushy size, in the

process, would be expected with changing properties of the material with the phase.

	Adiabatic case		Nonadiabatic case	
	Liquid fraction	Relative Error	Liquid fraction	Relative Error
$\epsilon = 0.1$	0.9669	0.96%	0.9247	3.4%%
$\epsilon = 0.5$	0.976	1.95%	0.9508	0.72%
Anlytical	**0.9577**	**-**	**0.9577**	**-**

Table 4: Error of the case with respect analytical values at t=6000s

5.3 Validation of the solver for convective problems

The solver is tested for a melting process where the buoyancy forces due the effect of gravity are accounted. Since fluid movement interferes with the heat transfer ($\vec{u} \neq 0$), the full set of the fluid equations, shown in section 3.4, shall be employed to perform the simulation.

The case employed to assess the performance of the solver on a convective problem is one recursively solved in the literature: the melting of pure gallium. Several amounts of information are employed for this case: the experimental results obtained by *Gau and Viskanta* [42] and the numerical results from *Jana et al.* [3], for the melting front. And also results of the streamlines obtained by *Hannoun et al.* [4]. The reason to compare the obtained results with that amount of literature is due to the disagreement concerning the melting of pure gallium after *J.A. Dantzig* [43], contrary to the experimental results cited above and previous numerical results, obtained a multicellular solution (i.e.: several convective cells) in the flow during melting of Gallium , instead of the expected monocellular solution (i.e.: only one convective cell). Since then, this debate has been opened, and many scientists support the multicellular solution of the flow claiming that the experimental results obtained by *Gau and Visanka*, were not representative of the actual phenomena occurring in the melting of Gallium [44].

5.3.1 Case 3

The properties of the material are summarized in table 8 at the Appendix B. The computational grid is uniform with a size of 400x300 cells, which does not match with the standards established in section 4.5.1 because of the limited available time to perform this study. The time-step was fixed to automatic time-step. The boundary conditions are the usual ones, saving for the outlet where a fixed temperature T_{outlet} is considered.

Again, the problem is isothermal, so a fine interval of temperature $\epsilon = 0.05$ has been employed for the simulations.

In figure 21, the melting front of the obtained results is compared with the experimental results from *Gau and Visanka* [42] and with the numerical results of *Jana et al.* for the same case of Gallium melting in a rectangular cavity.

With respect the experimental results: at an early time, the present results show a different front morphology (more curved). These differences become bigger during the first 6 minutes of simulation, where the solution obtained in this project is characterized by a more irregular front with the presence of bulges. However, later in time, after 10 minutes, the morphology of the front becomes to converge with the morphology of the experimental results. After 17 min, is appreciated, how indeed the morphology of both fronts get closer with time, presenting the same general morphology despite the moderate difference in position.

When comparing the present solution with the results obtained by *Jana et al.* [3], the similarity is high at 6 min and with some discrepancies at 10 minutes.

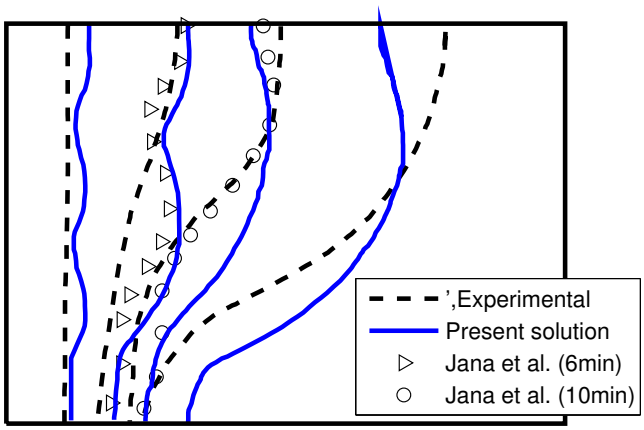


Figure 21: Position of the melting front after 2 min, 6 min, 10 min and 17 min for experimental values and present values. Values from *Jana et al.* at 6 min and 10 min all also provided

To find an explanation of these differences in the melting front, at early times, between the results of *Jana et al.* [3] and the present results, with the results obtained in the

experiment performed by *Gau and Visanka* [42], one shall refer to the literature, where a big debate is found concerning this issue directly linked to the shape of the flow. A part of the scientific community supports the experiment by *Gau and Visanka* [42] and provide numerical results that match with the idea of a unique convective cell in the flow that will produce this one-bulge shape in the melting front of the Gallium experiment [45][46]. However, another part, supported mainly by numerical results [4][3], defend a multicellular flow, where several convective cells are formed at early melting times producing this multiple-bulge melting front shape.

Regarding the solution of the flow obtained in the present simulation, figure 23 depicts the streamlines at different times. These results show clearly, that with the solver employed in this project, the shape of the flow matches with the theory of a multicellular flow. When these results are compared with the obtained by *Jana et al.* available in this reference [3], or the obtained by *Hannoun et al*, shown in figure 22, it can be noticed a high similarity in the quality of the flow and in the number of convective cells. Some discrepancies are expected, mainly in the number of convective cells at early simulation times, because, as explained in the study made by `Hannoun et al.` [4], the number of convective cells detected depends on the refinement of the mesh, because, mainly at early times, they are very small and require high accuracy in order to be detected. This study by `Hannoun et al.` used a grid of 840x600 cells (except at time 280s where the grid is of 560x400 cells), while in the present study the grid employed has 400x300 cells. Nevertheless, the results presented seem to be in high agreement with the ones obtained by *Hannoun et al.* [4] and `Jana et al`[3] and even higher agreement is expected with the refinement of the mesh.

At last, after all the discussion about the shape of the flow, it is important to outline, once again, that the discrepancies between the experimental results and the results obtained in this project regarding the front interface shown in figure 21 are fully linked to this disagreement in the number of convective cells, that are the ones that causes this bulges in the melting front, and therefore the unequal results. Nevertheless, as already discussed, for later time simulations, the present results get closer to the results from the experiment. This may find an explanation in figure 23 where it can be observe that the number of cells decrease until a certain time, where the author have obtained an unique convective cell, like in the experimental solution. Thus, similarities on the shape of the front arise because of the alike flow patter (eith respect the experimental results) after certain simulation time.

Furthermore, 3D effects and measurement errors shall be considered as another source of discrepancies between numerical and experimental results.

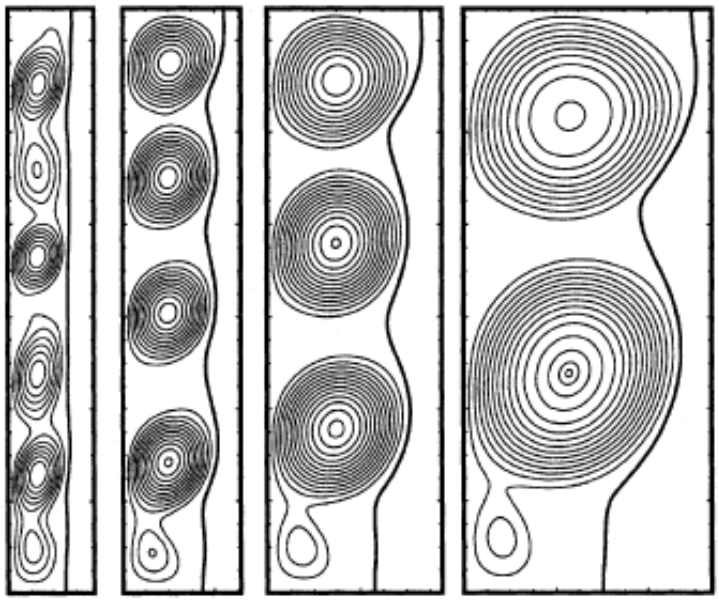


Figure 22: Stream function for the melting of Gallium after 42s, 85s, 155s and 280s obtained by Hannoun with a grid of 1120x800 cells.

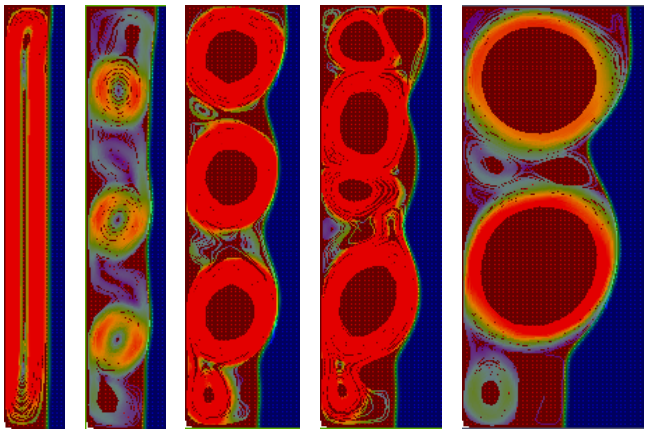


Figure 23: Stream function for the melting of Gallium after 40s, 80s, 140s, 160s, 280s obtained in simulations with the `PCMsolver 4.1, with a 400x300 grid.`

5.4 Conclusions

In this section, the conclusions regarding this validation chapter are presented. Three points of relevance are discussed in this chapter: the quality of the solutions obtained for conductive cases, the quality of the solutions fr a convective case, and the nature of the flow in the melting of a Gallium cavity. All of them are outlined below.

1. **Assesment of the solver,** `PCMsolver 4.1` **for conductive cases:** from the comparisons made in the section regarding the validation of the solver for conductive

cases, it can be determined, that this solver presents a reliable an solid tool for performing simulations of a pure diffusive phase change. Furthermore, in this section was also shown that the solver is capable to deal with both, solidification and melting processes. Nevertheless, the solver presents some weakness in conductive cases. Simulations performed during long times can yield to inaccurate solutions due to the impossibility of the solver to simulate pure isothermal processes. The error produced when an interval of temperature is assumed in isothermal cases is not neglectable when the heat exchanged during the melting process is rather low, affecting in a sensitive way the solution

2. **Assesment of the solver,** `PCMsolver 4.1`**, for convective cases:** In spite of the real shape of the flow in the melting of Gallium, the solver seems to deliver rather suitable results for convective cases when compared with literature. It is true, that further validations, mainly with non-isothermal cases, shall be performed in order to give a definitive answer about the capabilities of the solver on convective problems. Beyond that, so far, the results obtained with this solver, look quite promising.

3. **Respect the nature of the flow in the melting of Gallium:** in spite of all the literature revised by this author, no scientific paper was found that gave a definite answer regarding the number of convective cells formed during the melting of Gallium. It is true that there are experiments [42] that support the theory of a monocellular flow, however, the methods employed in the performance of this experiments have been disclaimed by some authors [44]. One of the most interesting papers, regarding this topic, is the one presented by *Hannoun et al.* [4], where a thorough analysis of the case is performed, claiming that the reason why some authors do not find more than one convection cell, is because they did not refine enough the mesh yielding that their solution could not capture the small convective cells, proper of early simulation times, what affected to the simulation results at later times.

   As the last comment, from all the literature reviewed by this author, all the authors that have obtained monocellular solutions where employing a coarse grid of the order of 30x30 or 40x40 cells. This does not imply directly that the monocellular solution is wrong, but it sustains the possibility of the multicellular solution to be true. Furthermore, an interesting phenomenon is observed when comparing the form of the flow in the Gallium (multicellular) with the flow of the PCM studied in chapter 6, which is monocellular, this issue is discussed further on the appendix A.

6 Results and Discussion

Once in chapter 5, the developed solver has been proven to be a reliable and fair accurate tool for computing thermal processes in phase transition, it can be employed to get further insight on the topic. Therefore, in this chapter the different results from the computations performed, with the use of the `OpenFOAM` solver, are presented and discussed. A special focus is made on melting times for each heat transfer mechanism and geometry, and the causes of their divergences.

Thus, the next computations serve as the highlight of all the theory and background developed during the earlier chapters of the project.

The melting process is studied for three different cases:

1. **Conductive melting of a PCM:** where the effect of gravity is neglected (i.e.: in space environment) and as so, the convective term of the heat equation because no other external force is considered.

2. **Convective melting of a PCM with RAyleigh-Bénard convection:** under the presence of gravity forces($g \neq 0$), where both conductive and convective heat transfer appears, since convection due to buoyancy forces is considered.

3. **Melting of a PCM with Bénard-Marangoni convection:** in micro-gravity environment ($g = 0$), thus natural convection is not present. Diffusive heat transfer (i.e.: conduction) is also present.

Furthermore, all cases are tested in two different geometries, a square and a rectangular one. Like in chapter 5, a summary of the parameters of the cases is disposed of appendix B. The parameters of the calculations and the data of the material, all were provided by the CSL.

Regarding the computational domain, the boundary conditions and the mesh and time-step selected on each case, they were already set, respectively, in sections 4.4.1, 4.4.2, 4.5.2 and 4.5.1.

6.1 Melting of a PCM with conduction

This section deals with the conductive melting of a PCM. This case is of interest because in many situations conduction is the only relevant heat transfer mechanism present in the melting process. Furthermore, takes an even higher relevance in space applications, where micro-gravity environments avoid the presence of natural convection.

6.1.1 Melting on a square geometry

A simulation of the pure conductive case has been made for a square geometry. Figure 24 shows the progress of the boundary at different times (250s, 1500s, 4000s).

One of the first things perceived is that the melting, and therefore the advance of the front, is completely parallel to the heating boundary, from where it can be deduced that the field of temperatures, and thus the heat transferred, is independent of the dimension parallel to the heating boundary (y-axis). This fact confirms what it was already known from the literature, that pure conductive phase change is a 1-D problem. It should be also

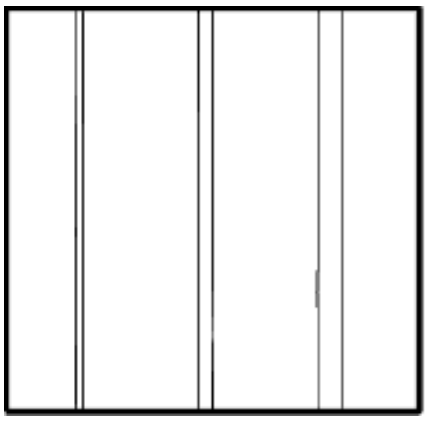


Figure 24: Solid-liquid interface at t= 250s, 1500s and 4000s. The front at each time is formed by two lines that define the length of the mushy region.

remarked that each solid-liquid interface in time is formed by two lines. The closest to the hot boundary, corresponds with the interface between the liquid phase and the mushy region, while the closest to the cold boundary corresponds to the mushy-solid interface. It is important to notice that this is linked straightforward with the temperature field: before the liquid-mushy interface, there is a gradient of temperature that goes from 213.5 K at the interface to 230 K at further points from the front. In the mushy region, there is a temperature gradient that goes from 213.5 at the liquid-mushy interface, to 212.5 at the mushy-solid interface. Finally, all of the PCM that falls after the mushy-solid interface is at a uniform temperature of 212.5 K.

Two last related and remarkable facts are observed. First that the velocity of the front, and therefore the melting process, gets slower with time (table 5 gives concrete data on this fact). The further is the front from the hot boundary the lower will be the temperature gradients, because the PCM is closer from a thermal equilibrium, and therefore less heat will reach the boundary slowing down the melting process. The second fact is that the mushy region increases its length with time. This is caused by the already mentioned smaller gradients of temperature, and the lower heat transfer caused by them.

The gradient of temperature in the mushy region will be lower as well, producing the need of having a larger transition between the solid and the liquid interface.

6.1.2 Melting of a rectangular geometry

A simulation of the pure conductive case have been made for a rectangular geometry. The rectangular conduction-driven melting process, is the closest one to the original Stefan problem that considered a semi-infinite slab. Figure 25 shows the progress of the melting front at after 100s, 6000s, and 18000s.

Figure 25: Solid-liquid interface at t= 100s, 6000s and 18000s. Isothermal lines are parallel to the front.

The rectangular geometry shows the same qualitative behaviour than the melting of a square geometry shown in figure 24. However, in this case, the length of the geometry is four times the height of the hot wall, when on the square both longitudes are the same. Therefore the incoming heat on the geometry is two times less, while the surface that must be melted is still the same. This is translated in a delay of the melting process and smoother gradients of temperature, what produces even larger mushy regions that on the previous case.

6.2 Melting of a PCM with Rayleigh-Bénard convection

The melting of both geometries is studied in an environment with the presence of gravity, therefore natural convection effects are considered. For the simulations, a value of gravity $g = 9.8$ has been used.

6.2.1 Melting of a square geometry

A simulation of the melting process of a square geometry under gravity effects has been performed. Results of the simulation after 100s, 450s and 800s are shown in figure 26.

When comparing with the conductive case, the main difference is that this time, the front line, and therefore the isothermal lines, are not parallel to the heating boundary.

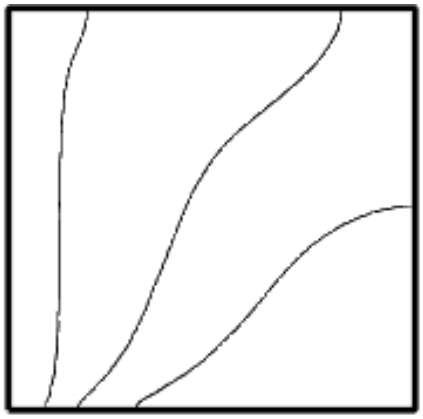


Figure 26: Solid-liquid interface at the midpoint of the mushy region t= 100s, 450s and 800s. Notice once more that the isothermal lines are parallel to the front, ranging from 212.5K on the solid side to up to 230K on the closest parts of the hot surface

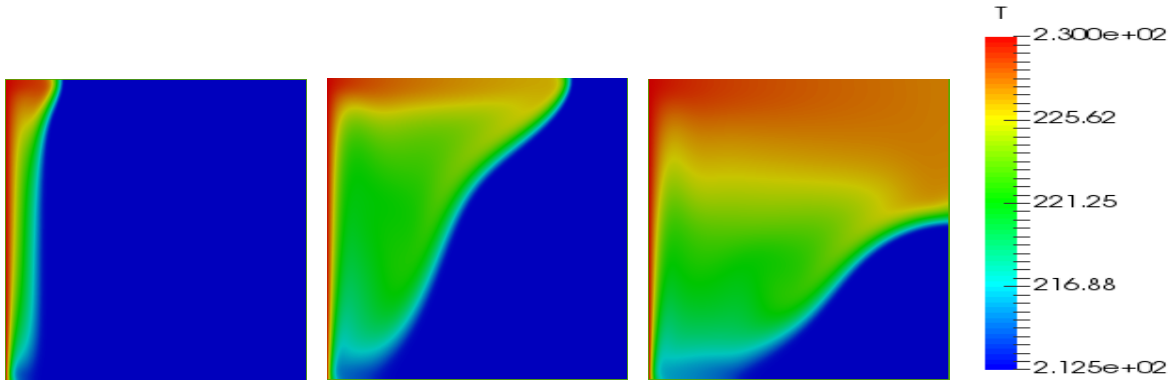


Figure 27: Temperature field at 100s, 450s and 800s for the melting of a PCM with Rayleigh-Bénard convection.

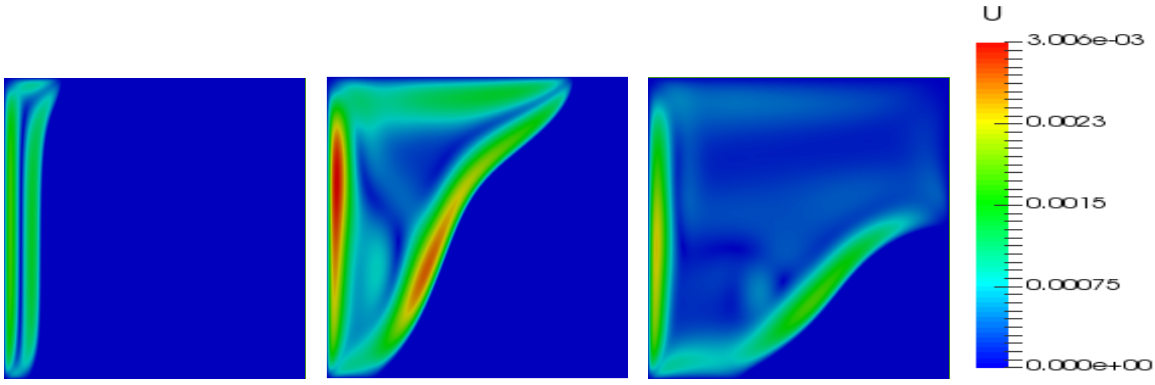


Figure 28: Module of the velocity field at 100s, 450s and 800s for the melting of a PCM with Rayleigh-Bénard convection.

The cause of this phenomena is quite easy to deduce: the gradient of temperature between the hot wall and the solid phase translates into different densities in the fluid phase. This,

in presence of gravity, activates buoyancy effects generating natural convection. In this manner, the fluid next to the hot wall increase temperature ascending to the upper part of the PCM, pushing the colder fluid, next to the solid wall, to the bottom. Then, the fluid that is now on the bottom and close to the hot wall, will increase in temperature and the cycle will start again forming a convective cell, that is how it is called the cyclical loop-movement of the fluid due convective effects (which is in the clockwise direction, in this case, due to the position of the heating boundary). Thus, since the hot liquid is on the upper part of the PCM, the heat transferred is higher here, producing a faster melting on the upper part of the PCM. Furthermore, the energy transferred by the friction of the moving fluid (or likewise by the effect of the convective terms) with the liquid-solid interface helps to the faster melting of the PCM in all its domain.

Figures 27 and 28 helps to understand the convective cell functioning by showing the relationship between the temperature with the magnitude of the velocity field. Notice how the velocity field describes the movement of the aforementioned convective cell. The largest velocities are founded towards the hot boundary and the melting front, because there, is where the temperature gradients are higher.

Figures 27 and 28 also allow to see the stages of the melting process. Where three phases are distinguished in function of the thermal stability and the kind of heat involved (sensible or latent). First, at an early stage, there is a phase of destabilization, where the incoming heat on the PCM is mainly destined to increase the latent heat of the unmelted part producing an increase of the temperature gradients and therefore an increase of convection. Then, the second state is when this destabilization reaches a peak, performing the highest gradients of temperature and therefore the highest values of heat transferred by convection. After this peak, the third stage starts, where most part of the PCM is already melted and the heat is mainly destined to increase the sensible heat. This translates in a stabilization of the temperature gradients, lowering down, progressively, convective effects until the temperature of the PCM is completely uniform at 230K.

As the last remark, one should consider that the fronts observed in figure 26, are situated at the midpoint of the mushy region. This is because, in this case, the mushy region is rather small and no increase in its length with time is observed. Therefore, despite that `Paraview` (the software available with `OpenFOAM` that plots the results), is able to show the whole and rather small size of the mushy region, the results incorporated here have considered the front as a line for the sake of clarity.

6.2.2 Melting of a rectangular geometry

A simulation of the melting process of a rectangular geometry under gravity effects has been performed. Results of the simulation after 100s, 800s and 1400s are shown in figure 29.

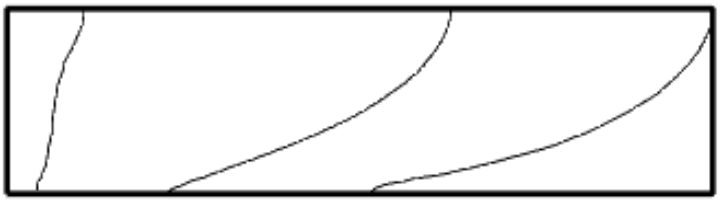


Figure 29: Solid-liquid interface at the midpoint of the mushy region t= 100s, 800s and 1400s. Notice once more that the isothermal lines are parallel to the front, ranging from 212.5K on the solid side to up to 230K on the closest parts of the hot surface

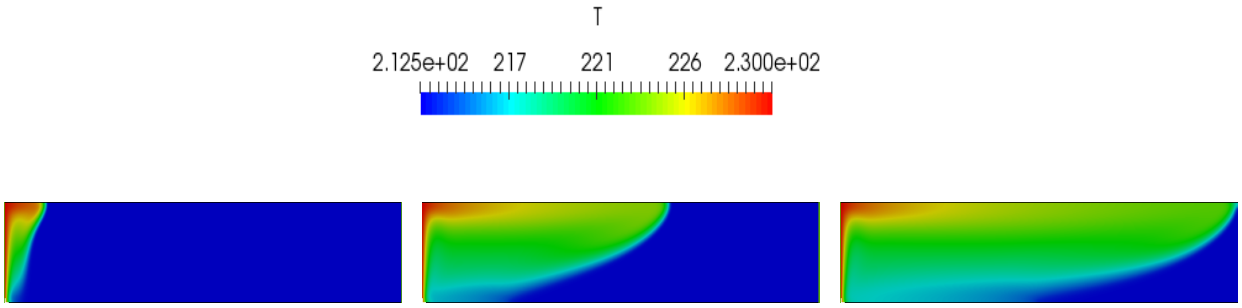


Figure 30: Temperature field at 100s, 800s and 1400s for the melting of a PCM with Rayleigh-Bénard convection

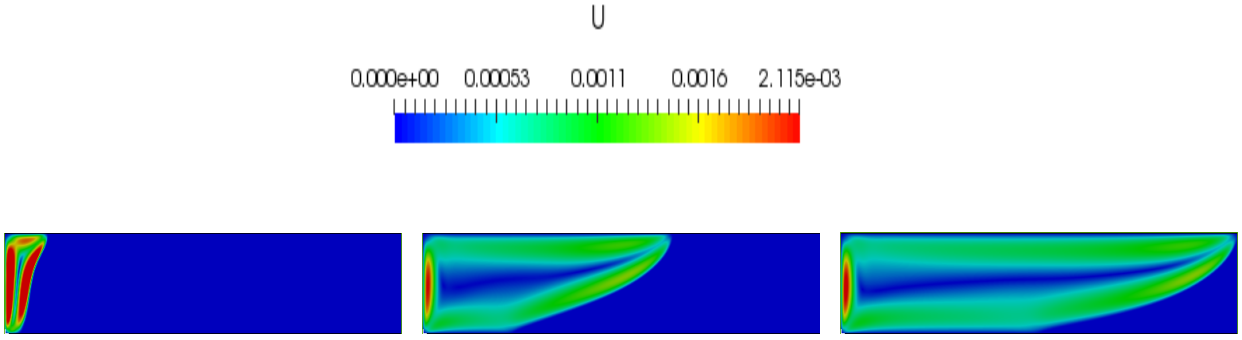


Figure 31: Module of the velocity field at 100s, 800s and 1400s for the melting of a PCM with Rayleigh-Bénard convection

Again, like in the conductive case, the larger horizontal length of the rectangular geometry is translated into a larger melting time. However table 5 and figure 29 shows a higher improvement, with respect the conductive case, for the rectangular geometry than for the square. The explanation is that the low transfer in the conductive case is mainly

motivated by the low area of influence of this mechanism on a rectangular geometry, whereas the convective movement will improve the transference in all the geometry, what accelerates in an exponential way, the melting of the PCM. Another way to describe this improvement is that in the rectangle, if one looks closely to figure 30, the overall increase of temperature is remarkably higher between the figure at 800s and 1400s than the overall increase in temperature between the rectangle at 100s and at 800s, although the time transcurred is higher in the second case. The reason is that at later times, the influence of the flow in a rectangle reaches a much higher surface (figure 31), and this influence is also higher with respect the flow on the square geometry at any time (figure 27. Thus, this causes a bigger heat transmission through viscous effects on the liquid-solid interface, what is translated in a remarkable improvement of the melting time. Nevertheless, at happened in the square geometry, at latest simulation times the uniformity of temperature decelerates sensibly the melting velocity.

Beyond what it was described above, in general, the heat transference is homonym to the square geometry. Thus, for a deeper explanation of the phenomena, one shall refer to the section addressed to the square geometry.

6.3 Melting of a PCM with Bénard-Marangoni convection

The melting of a PCM is studied under the presence of Bénard-Marangoni convection in a micro-gravity environment. Therefore, no natural convection takes place in the heat transfer. As explained in section 3.5.3, to have Marangoni convection, a free surface in contact with a fluid is needed. The boundary conditions depicted in section 4.4 will enforce this phenomenon.

Again, the process is studied for a rectangular and a square geometry. Furthermore, a study on the velocity of the free surface and on the Nusselt number gives relevant information about this heat transfer mechanism.

6.3.1 Melting on a square geometry

In figure 32 the position of the melting front at t=100s, 450s and 1000s is shown. The process presents an important progress in the melting time compared to the conductive case, however, it is still slower than the process in the presence of gravity (table 5).

The hot wall, as expected, produce a gradient of temperature along the geometry, taking relevance the gradients along the free surface (disposed on the upper part of the figure), that allow the development of the Marangoni convection. Figures 33 and 34

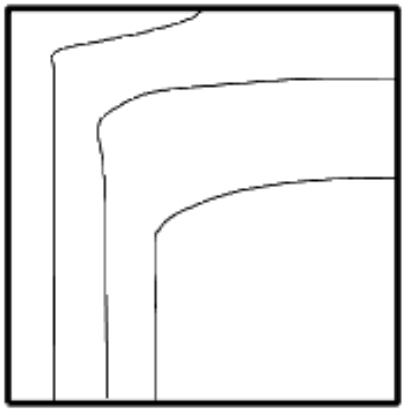


Figure 32: Solid-liquid interface at the midpoint of the mushy region t= 100s, 450s and 1000s. Notice once more that the isothermal lines are parallel to the front, ranging from 212.5K on the solid side to up to 230K on the closest parts of the hot surface

shows the influence of the gradient of temperature on the free surface, producing a faster melting on the top part of the geometry.

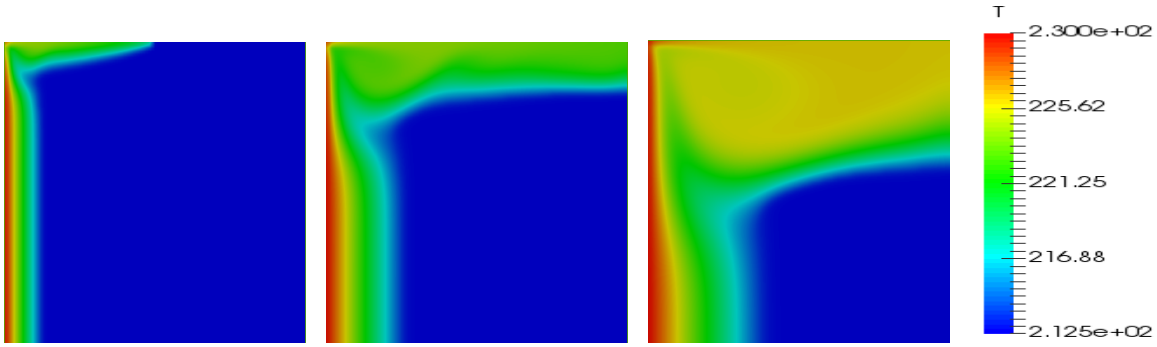


Figure 33: Temperature field at 100s, 450s and 1000s for the melting of a PCM with Bénard-Marangoni convection

It is easy to separate the melting caused by the Marangoni convection which, as already described, produce a melting parallel to the horizontal axis, from the free surface towards the bottom, from the melting produce by the conductive heat transfer. Conduction, as learned from section 6.1, produces a melting of the geometry, and a heat transfer, parallel to the hot wall. This two heat transfer mechanisms together, produce the particular form of the melting front in this process.

Notice, that again, there is an important decrease on the melting velocity of the PCM, when this is almost melted. The uniformity of the temperature field, or what it is the same, the low gradients of temperature, translates into a lower heat transference for both driving mechanisms (conduction and Marangoni convection).

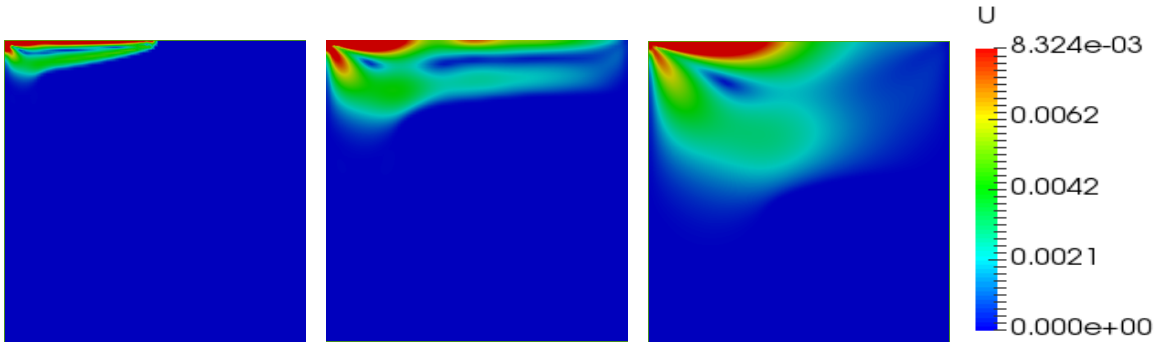


Figure 34: Module of the velocity field at 100s, 450s and 1000s for the melting of a PCM with Bénard-Marangoni convection

6.3.2 Melting on a rectangular geometry

A simulation of the heat transfer under Bénard-Marangoni convection has been done for a rectangular geometry. Figure 35 shows the progress of the melting front after 100s, 800s and 1400s.

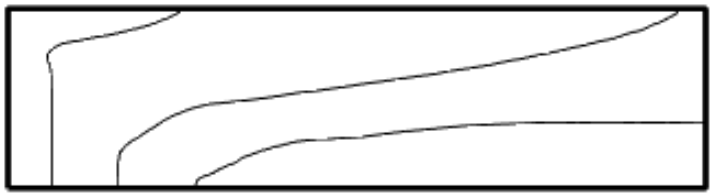


Figure 35: Solid-liquid interface at the midpoint of the mushy region t= 100s, 800s and 1400s. Notice once more that the isothermal lines are parallel to the front, ranging from 212.5K on the solid side to up to 230K on the closest parts of the hot surface

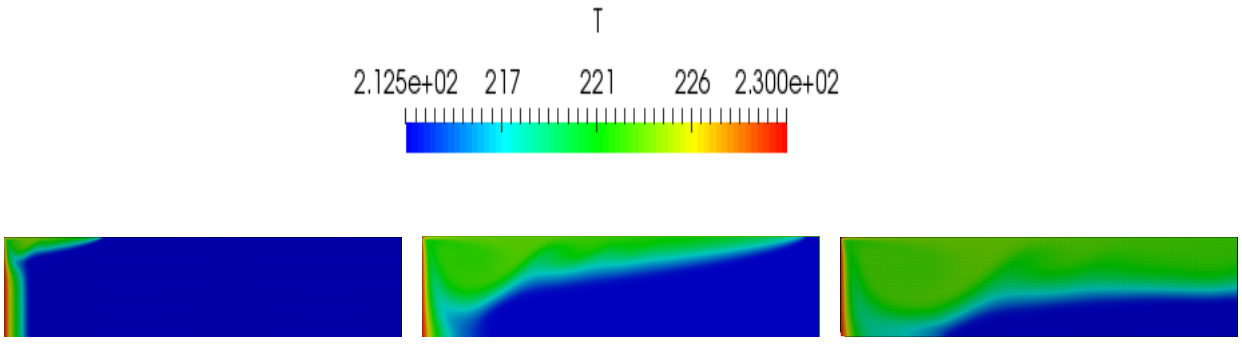


Figure 36: Temperature field at 100s, 450s and 1000s for the melting of a PCM with Bénard-Marangoni convection

The most relevant aspect of this case, it that, by looking at table 5, one can notice that the Bénard-Marangoni convection, contrary to the other heat transfer mechanisms studied, produces shorter melting times for rectangular geometries. The reason can be

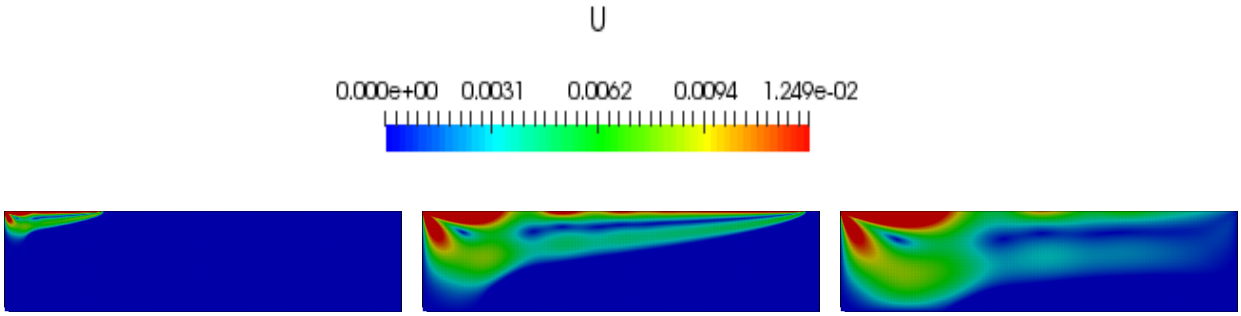


Figure 37: Module of the velocity field at 100s, 450s and 1000s for the melting of a PCM with Bénard-Marangoni convection

find on the larger free surface proper of the rectangular geometry. When comparing figure 34b with figure 37b, which both address the velocity of the fluid after 450s of simulation, the surface influenced by convection on the second one, is sensibly higher. This will accelerate the melting of the PCM. Another look can be taken to the temperature profile shown in figure 36, where it is seen, how indeed the temperature of the PCM raises in a faster way.

Further insight on the differences between the square and the rectangular geometry is given by a study on the Nusselt number.

6.3.3 Study on the Nusselt number

The Marangoni convection is one of the most interesting phenomena studied in this thesis, mainly for its relevance in micro-gravity environments. Thus, a study on the Nusselt number at the inlet and the velocity of the fluid at the free surface was performed to collect more information about the process.

First, to understand this study, the Nusselt number shall be defined and explained. This dimensionless number reads:

$$Nu = \frac{hL_c}{\kappa} \tag{6.1}$$

where h stands for the convective heat transfer coefficient, κ is the thermal conductivity coefficient and L_c is the characteristic length. Thus, the Nusselt number represents the increase in the heat transfer on a fluid surface through convective means, with respect to the heat transferred on the same surface through conduction [47]. A large Nusselt number will stand for a large influence of convective terms in the transfer, while a low Nusselt number will address a more conductive-driven heat transfer.

Therefore, the study of the Nusselt number will give information on the kind of heat

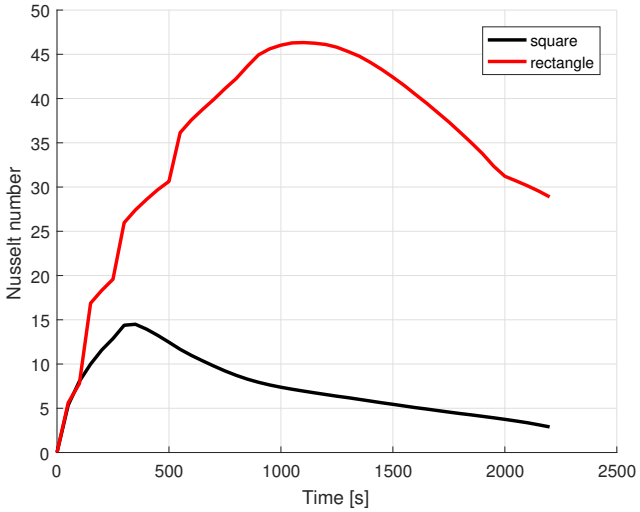


Figure 38: Average Nusselt number in time at the inlet of the square and the rectangular geometry in Marangoni convection

transfer that has more relevance in the simulation. Figure 38, delivers the average Nusselt number in time, on the hot surface, for both geometries. As expected the Nusselt number is far above one in most part of the process, a result that matches with the differences in melting times, observed in table 5, between the cases with Marangoni convection and the cases with only conductive heat transfer. Nevertheless, at an early simulation time, the conduction has a main role in the heat transfer, this is because at the first steps of the melting process the PCM is mainly solid, though the little amount of fluid found in the cavity has almost no velocity due to the shear stresses on the solid boundaries. Taking into account that at an early time, the only fluid part is towards the free surface next to the hot wall, figure 39 supports this aforementioned affirmation.

The last aspect that is worth to be mentioned, is that the Nusselt number takes greater values on the rectangular geometry, which means that convection is far more present the larger is the free surface. This fact helps to explain why the melting time is smaller, in this case, for this kind of geometry.

Figure 39 still yields more interesting information. The velocity on the free surface quickly reaches a peak that coincides with the increase of the Nusselt number in the hot wall. After this point, despite the Nusselt number keeps augmenting at the inlet, because of the increasing presence of fluid with time, the velocity at the free surface decays. This deceleration has an explanation on figure 36. Due to the high convection, the temperature on the free surface is the first one to stabilize, what means that the velocity of the fluid will be very slow due the low gradients of temperature.

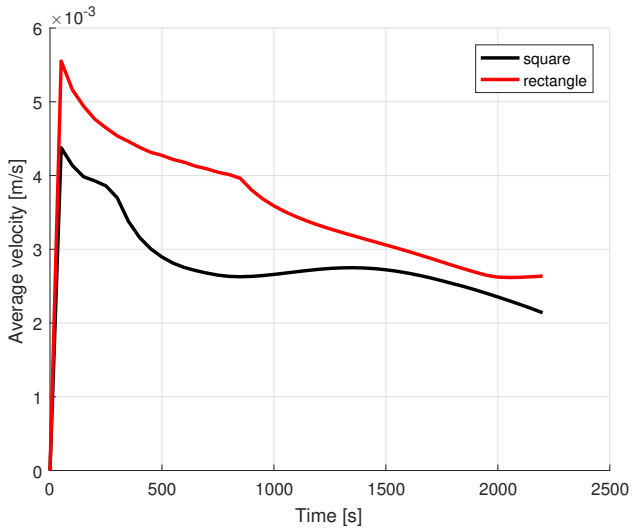


Figure 39: Average velocity on the free surface

6.4 Conclusions

Different conclusions are made for each of the three heat mechanisms studied:

1. **Conduction:** the two more remarkable results concerning conduction are the confirmation, that indeed is a pure 1-D process and only the length of the geometry has an impact on the melting, and that the length of the mushy region increase with time due to the lower gradients of temperature present at this simulation times. Furthermore, the table 5, shows a clear difference in the melting time between one geometry and another. Nevertheless, this differences was somehow expected, due to the smaller heating surface of the rectangular geometry. Another simulation, with the same amount of incoming heat at the inlet for both geometries, could lead to interesting results regarding the melting times.

2. **Rayleigh-Bénard convection:** the main important fact observed in convective cases is the high improvement on the melting time. The action of convective terms accelerates the process by a 495% with respect the conduction in a square geometry and a 1384% with respect the conduction on a rectangular geometry. The higher improvement on the rectangular geometry with respect the square geometry (almost 3 times more) is found on the higher surface of the melting front influenced by the flow in the rectangular geometry, what improves in a remarkable way the convective heat transfer.

3. **Bénard-Marangoni convection:** which most interesting result, is that is the only heat transfer mechanisms in which the rectangular geometry, in spite of its

lower heating source, perform shorter melting times that the square geometry. This is caused because as already explained, the Marangoni convection produces higher flow velocities the larger is the free surface. Furthermore, it is also of interest, the influence of each kind of heat transfer at specific periods of the process, being the conductive heat transfer higher at very early simulation times, and then, for the rest of the process, the convective terms dominate the heat transfer.

	Conduction		Rayleigh-Bénard		Bénard-Marangoni	
	$f = 0.5$	$f = 1$	$f = 0.5$	$f = 1$	$f = 0.5$	$f = 1$
Square	1574s	6326s	469s	1276s	725s	2295s
Rectangle	6290s	25248s	814 s	1823 s	840s	1972s

Table 5: Melting time for different geometries and heat transfer mechanisms.

7 Conclusions and Future Work

7.1 General conclusions of the project

The real aim of this project and the reason why a whole mathematical method, with all the numerical methods involved and the correspondent implementation in `OpenFOAM` has been performed, it was to develop a tool that could help in the design of PCMs for thermal management purposes. This tool is indeed the solver adapted to `OpenFOAM 4.1`, which suppose the main achievement and objective reached in this project along with his correspondent validation. If, certainly, the solver still have limitations (no 3-D effects, no convective term on the latent heat source), and some uncertainties, like the pattern of the flow. It gives a fairly first approach for the design of a PCM and also supposes a starting point for the development of better and more accurate solvers. Moreover, this accomplished objective drives us to the other important aim of this project: the study of melting/solidification processes in different geometries and under different heat transfer mechanisms. Chapters 5 and 6 provides a long and deep analysis of the melting process. For conclusions on this results, one should refer to this chapters. Furthermore, unexpected results were found regarding the pattern of the flow under different Prandtl numbers. This is explained in detail in Appendix A

Besides this main achievements, other goals, although more academical, have been reached during this thesis, giving a wider scope of the subject treated:

- The concept of PCM and a general overview of this technology has been discussed making an emphasis on the unique qualities of these materials and the need of a tool capable to perform a thermal analysis in phase change transitions to reach all the potential of this materials.
- A thorough explanation of the enthalpy-porosity technique has been provided. Including all the assumptions and simplifications made in the model and pointing the consequences of these assumptions and the consequent limitations.
- A deep study of the numerical methods employed in the simulation has been shown and explained. Furthermore, a thorough analysis of the solution quality has been provided.

7.2 Future work

This topic and the approach of this project lead, indeed, to a really high number of paths to continue the work here performed. For this reason, this section will try to summarize the most relevant and critical ones:

1. Regarding the validation of the solver, the next step would be to assess its performance under non-isothermal convective phase change with experimental data from the literature. Optimal results are not expected since the convective part of the latent heat source term was neglected as pointed out in equation 3.22. A study on the functionality of the solver in relation this parameter would be of high interest. Furthermore, a validation of the process with Bénard-Marangoni convection should be performed to test the capacity of the solver in this problem.

2. Regarding the development of the solver there are multiple ways to proceed: adapt the solver to deal with different liquid fractions following the expression of the liquid fraction in equation 3.33, include a solution for isothermal phase changes...however, the most interesting approach probably would be to develop, from this solver and following an enthalpy-porosity approach, a solver capable to deal with 3-D phase changes accounting for the remaining equation of the Navier-Stokes equation and other possible 3-D effects present in such cases.

3. Regarding the study of phase changes in PCM, the next logical step would be to try new and more complex geometries to observe how the phase change occurs in such cases. Furthermore, a preliminary design of a PCM in a real case scenario on a spacecraft, would be an exercise of high interest to understand the challenges of the design of PCMs for space applications and real applications in general.

4. Regarding the difference in the flow patterns shown in the PCM simulations with respect the simulations involving Gallium, further simulations and analysis should be performed to prove if there is any truth in the divagation made in Appendix A.

A Appendix: A word on monocellular and multicellular flow

If the results obtained in the validation chapter regarding the flow pattern in the melting of gallium in a rectangular cavity (figure 23) are compared with the results obtained for the velocity flow in the melting of a PCM with natural convection for a square and rectangular geometries (figure 28 and figure 31), and interesting result is found when observing the number of convective cells in each case. In the case of the Gallium, as already commented, there are multiple cells at early simulation times and slowly, they start to unify forming bigger rolls until there is only one. However, for the PCM there is only one roll in all the melting process. Then, one could question why, with the same solver, the solution obtained for the pattern of the flow is monocellular in one case and multicellular in another? The explanation founded, is that if a closer attention is paid on the parameters of each case, the PCM has a Prandtl number of $Pr = 23.72$ while the Gallium has a Prandtl number of $Pr = 0.0216$. Recalling the formula of the Prandtl number and its definition as the quotient between the diffusion of momentum with the heat diffusion, this is translated in that, in the case of the Gallium, the diffusion of heat is a couple of orders of magnitude above the diffusion of momentum, what is translated in that at early simulation time, the convective terms will not have a high influence in the melting of the Gallium. On the other hand, conductive terms will have more influence and coupled with the small influence of the convective terms will produce the bulge-shape melt front. This shape of the front will induce at the same time the formation of convective cells. At later times, the velocity of the fluid becomes higher because the increase of the fluid cavity with the melting, and also because of the higher gradients of temperature that increase buoyancy effects. For these reasons, convective effects will become again relevant creating a unique convective cell. With respect the PCM, convective terms will domain the phase change during the whole phase change due to the high Prandtl number, therefore from the very first moment, the low impact of the conductive terms will not be able to create major changes in the boundary avoiding the formation of more than one cell. Moreover, last time results obtained with different values of the Prandtl number and geometries were not conclusive (reason why they were not incorporated), although they suggest that the geometry might have an impact on the pattern of the flow.

Nevertheless, this explanation is just a hypothesis from the results obtained, and although it looks like a possibility, further studies and simulations on the dependency of the flow pattern with the Prandtl number shall be performed in order to give an explanation to this differences between the flow in the PCM and the flow in the Gallium.

B Appendix: Tables of Case Parameters

Case 1

Parameter	Value	Dimension
Density, ρ	1	$kg \cdot m^{-3}$
Specific heat, c_p	1	$J \cdot kg^{-1} \cdot K^{-1}$
Thermal conductivity, κ	0.001	$W \cdot m^{-1} \cdot K^{-1}$
Latent heat, L	5	$J \cdot kg^{-1}$
Melting point, T_m	273	K
T_{inlet}	272.5	K
T_{outlet}	273.5	K
T_0	273.5	K
Viscosity, μ	1	$kg \cdot m^{-1} \cdot s^{-1}$
T_{ref}	273.5	K
ϵ	0.1,0.05,0.01	K

Table 6: Table with the physical parameters of the case posed by Voller

Case 2 and Results

Parameter	Value	Dimension
Density, ρ	1280	$kg \cdot m^{-3}$
Specific heat, c_p	2900	$J \cdot kg^{-1} \cdot K^{-1}$
Thermal conductivity, κ	0.44	$W \cdot m^{-1} \cdot K^{-1}$
Latent heat, L	$172 \cdot 10^3$	$J \cdot kg^{-1}$
Melting point, T_m	213	K
T_{inlet}	230	K
T_{outlet}	Adiabatic	K
T_0	212.5	K
Vol. thermal exp. coef., β	$9.1 \cdot 10^{-4}$	K^{-1}
Viscosity, μ	$3.6 \cdot 10^{-3}$	$kg \cdot m^{-1} \cdot s^{-1}$
T_{ref}	213	K
ϵ	0.5	K
gravity, g	9.8	$m \cdot s^{-2}$
Stefan number, St	0.278	-
Prandtl number, Pr	23.72	-

Table 7: Table with the physical properties of the PCM used in the results chapter and in the validation of conduction

Case 3

Parameter	**Value**	**Dimension**
Density,ρ	6093	$kg \cdot m^{-3}$
Specific heat, c_p	381.5	$J \cdot kg^{-1} \cdot K^{-1}$
Thermal conductivity, κ	32	$W \cdot m^{-1} \cdot K^{-1}$
Latent heat, L	80160	$J \cdot kg^{-1}$
Melting point, T_m	302.78	K
T_{inlet}	311	K
T_{outlet}	301.3	K
T_0	301.3	K
Vol. thermal exp. coef., β	$1.2 \cdot 10^{-4}$	K^{-1}
Viscosity, μ	$1.81 \cdot 10^{-3}$	$kg \cdot m^{-1} \cdot s^{-1}$
T_{ref}	302.78	K
ϵ	0.05	K
gravity, g	9.8	$m \cdot s^{-2}$
Cavity length, l	0.0889	m
Cavity height, h	0.0635	m
Stefan number, St	0.046	-
Prandtl number, Pr	0.0216	-
Rayleigh number, Ra	$7 \cdot 10^5$	-

Table 8: Table with the physical parameters of the Gallium melting for the convective case validation

C Appendix: Matlab code for Stefan problem

```
% Analytical result of the Stefan Problem in one dimesion. Source Mathematical modeli
%-------------------------------------------------------
close all
clear all
clc
%-------------------------------------------------------
% Assumptions
%rho=cte    % Incompressibility
phi=0;    % No dissipation effects
mu=10E-5; % Constant viscosity
v_x=0;      % No convection and 1D problem

%Temperature conditions
Timp=-0.5; %C
Tm=0; % C
Tinit=0.5; %C

% Matirial data
rho=1; %kg/m
c_l=1; %J/kg C
c_s=1; %J/kg C
k_l=0.001; %W/mC
k_s=0.001; %W/mC
L=5; %J/kg
alpha_l=k_l/(rho*c_l);
alpha_s=k_s/(rho*c_s);

% Time and space discretization
n_e=100;   %number of elements
d=1;     %size of the bar[m]
d_x=d/n_e; %delta x
n_s=1000; %number of steps in the time discretization
t=1000; %Calculating time
d_t=t/n_s; %delta t=60s
```

```
%Creating body (Initial and Boundary conditions)
Body_T(n_s,n_e)=zeros; %Temperature of the body in space and time



% Melting process
if Timp>Tm
    alpha_1=alpha_s;
    alpha_2=alpha_l;
    St_s=c_s*(Timp-Tm)/L;
    St_l=c_l*(Tm-Tinit)/L;
    nu=sqrt(alpha_s/alpha_l);
    %solving trascendental equation to get lambda
    lambda=fsolve(@(lambda)St_l/(exp(lambda^2)*erf(lambda))-St_s/(nu*exp(nu^2*lambda^

else %Freezing process
    alpha_1=alpha_l;
    alpha_2=alpha_s;
    St_s=c_s*(Timp-Tm)/-L;
    St_l=c_l*(Tm-Tinit)/-L;
    nu=sqrt(alpha_l/alpha_s);
    %solving the trascendental equation to get lamda
    lambda=fsolve(@(lambda)St_s/(exp(lambda^2)*erf(lambda))-St_l/(nu*exp(nu^2*lambda^
end

%Creatin the vector of position of the boundary solid/liquid
Xj(1,n_s)=zeros;

for i=1:(n_s+1);
    t(i)=d_t*(i-1);
    Xj(i)=2*lambda*sqrt(alpha_1*t(i));
end


figure('units','normalized','OuterPosition',[0.5 0.5 0.5 0.5],'name','Front displacem
hold on
plot(t/3600,Xj,'LineWidth',2)
```

```
hold off
grid on
axis([t(1) t(n_s+1)/3600 Xj(1) Xj(n_s)])
ylabel('Front (m)','FontWeight','bold')
xlabel('Time [hr]','FontWeight','bold')
```

D Appendix: OpenFOAM case

Directory 0

File T

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.1                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      T;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 212.5;

boundaryField
{
   inlet
   {
    type       fixedValue;
    value      uniform 230;
   }
   outlet
   {
    //type       fixedValue;
    //value     uniform 298.15;
```

```
   type       zeroGradient;
   }

  top
   {
    type      zeroGradient;
   }
 bottom
   {
    type      zeroGradient;
   }

   Back&Front
   {
    type     empty;
   }
}

// ************************************************************************* //
```

File U

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{

    /*
    top
    {
      type             marangoni;
      marangonicoeff        -0.0225;
      value            uniform (0 0 0);
    }
    */

    top
    {
      type             fixedValue;
      value            uniform (0 0 0);
    }

    inlet
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

    outlet
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

    bottom
    {
```

```
        type            fixedValue;
        value           uniform (0 0 0);
    }

    Back&Front
    {
type empty;
    }
}

// ************************************************************************* //
```

File p

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
```

```
    {
        type            calculated;
        value           $internalField;
    }

    outlet
    {
        type            calculated;
        value           $internalField;
    }

    top
    {
        type            calculated;
        value           $internalField;
    }
    bottom
    {
        type            calculated;
        value           $internalField;
    }


    Back&Front
    {
type empty;
    }
}

// ************************************************************************* //
```

File p_{rgh}

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
```

```
|   \\  /    A nd           | Web:      www.OpenFOAM.org                    |
|    \\/     M anipulation  |                                               |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p_rgh;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type            fixedFluxPressure;
        rho             rhok;
        value           uniform 0;
    }

    outlet
    {
        type            fixedFluxPressure;
        rho             rhok;
        value           uniform 0;
    }

    top
    {
        type            fixedFluxPressure;
        rho             rhok;
        value           uniform 0;
```

```
    }

    bottom
    {
      type              fixedFluxPressure;
      rho               rhok;
      value             uniform 0;
    }
    Back&Front
    {
type empty;
    }
}

// ************************************************************************* //
```

Directory Constant

File transportProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.1                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
   version     2.0;
   format      ascii;
   class       dictionary;
   object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
rhoLiquid                   rhoLiquid [1 -3 0 0 0 0 0] 1280;
mu                                mu [1 -1 -1 0 0 0 0] 0.0036;
```

```
cpLiquid                      cpLiquid [0 2 -2 -1 0 0 0] 2900;
cpSolid                       cpSolid [0 2 -2 -1 0 0 0] 2900;
kappaLiquid                kappaLiquid [1 1 -3 -1 0 0 0] 0.44;
kappaSolid                 kappaSolid [1 1 -3 -1 0 0 0] 0.44;
betaLiquid                 betaLiquid [0 0 0 -1 0 0 0] 9.1e-4;
TSolid                        TSolid [0 0 0 1 0 0 0] 212.5;
TLiquid                      TLiquid [0 0 0 1 0 0 0] 213.5;
Tref                   Tref [0 0 0 1 0 0 0] 213;
L                                 L [0 2 -2 0 0 0 0] 172e03;


darcyTermConstant             darcyTermConstant [1 -3 -1 0 0 0 0] 1.6e3;
darcyTermEpsilon             darcyTermEpsilon [0 0 0 0 0 0 0] 1e-03;
```

File g

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.1                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
   version     2.0;
   format      ascii;
   class       uniformDimensionedVectorField;
   location    "constant";
   object      g;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 1 -2 0 0 0 0];
//value    ( 0 -9.81 0 );
value      ( 0 0 0 );
```

Directory System

File controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.1                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
//libs ("libmyBCs.so");

application     PCMwittig;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         1800;

deltaT          0.002;

writeControl    adjustableRunTime;

writeInterval   50;
```

```
purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision         6;

runTimeModifiable  true;

adjustTimeStep      yes;

maxCo                 5;

maxDeltaT            0.2;

// ************************************************************************* //
```

File blockMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.1                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
   version     2.0;
   format      ascii;
   class       dictionary;
```

```
   object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
convertToMeters 0.01;

vertices
(
    (0.0 0.0 0.0)    //0
    (2.0 0.0 0.0)    //1
    (2.0 2.0 0.0)    //2
    (0.0 2.0 0.0)    //3
    (0.0 0.0 1.0)    //4
    (2.0 0.0 1.0)    //5
    (2.0 2.0 1.0)    //6
    (0.0 2.0 1.0)    //7
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (400 400 1) simpleGrading (1 1 1)
);

edges
(
);

patches
(
    patch inlet
    (

            (0 3 7 4)

        )


    empty Back&Front
```

```
        (
            (4 5 6 7)
            (0 3 2 1)

        )


    patch top
        (
        (2 3 7 6)
        )

   patch bottom
       (
       (0 1 5 4)
       )


    patch outlet

       (
           (1 5 6 2)
       )

);

mergePatchPairs
(
);

// ************************************************************************* //
```

File decomposeParDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
```

```
|  \\    /   O peration     | Version:  2.2.2                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    note        "mesh decomposition control dictionary";
    object      decomposeParDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

numberOfSubdomains  9;


method          scotch;




simpleCoeffs
{
    n           (3 3 1);
    delta       0.001;
}
```

File fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.1                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
```

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         CrankNicolson 1;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
}

divSchemes
{
    default         none;

    div(phi,U)      Gauss linearUpwind grad(U);
    div(phi,T)      Gauss linearUpwind grad(T);
    div(phicp,T)    Gauss linearUpwind grad(T);
}

laplacianSchemes
{
    default         none;
    laplacian((kappa|rho),T)   Gauss linear limited 1;
    laplacian(nu,U)            Gauss linear limited 1;
    laplacian((1|A(U)),p_rgh)  Gauss linear limited 1;
    laplacian(rAUf,p_rgh)      Gauss linear limited 1;
```

```
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         limited 1;
}
fluxRequired
{
  default            none;
  p_rgh;


// ************************************************************************* //
```

File fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.1                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
solvers
{
   p_rgh
    {
        solver          GAMG;
        preconditioner  DILU;
        mergeLevels     1;
        smoother GaussSeidel;
        agglomerator faceAreaPair;
        nCellsInCoarsestLevel 100;
        tolerance       1e-8;
        relTol          0.001;
        nPreSweeps   0;
        nPostSweeps    2;
    }

    p_rghFinal
    {
        $p_rgh;
        tolerance       1e-8;
        relTol          0;
    }

   "(U|UFinal)"
    {
        solver          GAMG;
        preconditioner  DILU;
        mergeLevels     1;
        smoother GaussSeidel;
        agglomerator faceAreaPair;
        nCellsInCoarsestLevel 100;
        tolerance       1e-8;
        relTol          0.0;
        nPreSweeps   0;
        nPostSweeps    2;
    }
```

```
    "(T|TFinal)"
    {
        solver          GAMG;
        preconditioner  DILU;
        mergeLevels     1;
        smoother GaussSeidel;
        agglomerator faceAreaPair;
        nCellsInCoarsestLevel 100;
        tolerance       1e-8;
        relTol          0.001;
        minIter         1;
        nPreSweeps   0;
        nPostSweeps    2;
    }
}


PIMPLE
{
    momentumPredictor no;
    nOuterCorrectors 150;
    nCorrectors     3;
    nNonOrthogonalCorrectors 0;
//nNonOrthogonalCorrectors 6;
    pRefCell        0;
    pRefValue       0;
    maxLiquidFractionIteration 3000;
    maxLiquidFractionResidual  1e-06;
    darcyTermLiquidFractionRelaxFactor 0.9;
    writeUIntegral 0.0;
    writeLiquidFraction 1.0;
    secondsToWriteMyData 2.0;
    onlyConduction 0; // 1 for only conduction
maxwell 0.0;
corcione 0.0;
ponderateKappa 0;
```

```
    residualControl
    {

       "(p_rgh|T)"
        {
            tolerance 1e-3;
            relTol 0;
        }
    }
}

relaxationFactors
{
    fields
    {
p_rgh 0.3;
    }
    equations
    {

 "(T|TFinal)" 0.3;
        "(U|k|epsilon|R)" 0.3;
        "(U|k|epsilon|R)Final" 0.3;
    }
}

// ************************************************************************* //
```

References

[1] V.R.Voller. A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems. Int.J. Heat Mass Transfer, 30(8):1709–1719, 1987.

[2] V.R. Voller, M. Cross, and N.C. Markatos. An enthalpy method for convection/diffusion phase changes. Int. J. Num. Meth. Engng, 24:271–284, 1987.

[3] S. Jana, S. Ray, and F. Dust. A numerical method to compute solidification and melting processes. Applied mathematics modelling, 31:93–119, 2007.

[4] N. Hannoun, V. Alexiades, and Tsun Zee Mai. Resolving the controversy over tin and gallium melting in a rectangular cavity heated from the side. Angew. Chem. Int. Ed, 53(15):1521–3773, 2014.

[5] Amy S. Fleischer. Thermal energy storage using phase change materials. Fundamentals and applications. Springer, 2015.

[6] B.R. Munson, A.P. Rothmayer, T.H. Okiishi, and W.W. Huebsch. Fundmentls of fluid mechanics 7th edtion. Wiley, 1990.

[7] Vasilios Alexiades and Alan D. Solomon. Mathematical modelling of melting and freezing processes. Hemisphere publising corporation, 1980.

[8] John H. Lienhard IV and John H. Lienhard IV. A heat transfer textbook 3rd ed. Cambridge, MA: J.H. Lienhard V,, 2000.

[9] Michael Favre-Marinet and Sedat Tardu. Convective Heat Transfer. Hermes Science/Lavoisier, 2008.

[10] DICCA University of Genoa. `http://www.wolfdynamics.com`.

[11] Hrvoje Jasak. Error analysis and estimation for the finite volume method with applications to fluid flows. PhD thesis, Imperial college of science, technology and medicine, 1996.

[12] Christopher J. Greenshields. OpenFOAM User Guide v. 4.0. http://openfoam.org, 2016.

[13] B. Zivkovic and I. Fujii. An analysis of isothermal phase change of pcm within rectangular and cylindrical containers. Solar Energy, 70(51-61), 2001.

[14] C. Vuik. Some historical notes on the stefan problem. Faculty of Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 1994.

[15] J. Crank. Free and moving boundary problems. Clarendon Press, 1984.

[16] A. Gil, M. Medrano, I. Martorell, A. Lazaro, P. Dolado, B. Zalba, and L.F. Cabeza. State of the art in thermal storage for power generation. Renewable and sustanaible energy reviews, 14(1):31–35, 2010.

[17] Phase Energy. `http://phase-energy.com/how-pcms-work/`.

[18] M. Farid Mohammed, M. Khudhair Amar, Siddique Ali K. Razack, and Said Al-Hallaj. A review on phase change energy storage: materials and applications. Energy conversion and management, 45:1957–1615, 2004.

[19] Dr. Dong Choon Hyun, Nathanael S. Levinson, and Younan Xia Prof. Unyong Jeong. Emerging applications of phase-change materials (pcms): Teaching an old dog new tricks. Angew. Chem. Int. Ed, 53(15):1521–3773, 2014.

[20] T. Revankar, Purdue University Travis Croy, School of Nuclear Engineering, and NASA Glenn Research Center Patrick J. George, Space Flight Project Branch. `http://www.purdue.edu/uns/html4ever/010607.Revankar.solar.html`.

[21] M.M. Farid and X. D. Chen. Proc. Inst. Mech. Eng., 213:83 – 92, 1999.

[22] F. Tiarks, K. Landfester, and M. Antonietti. Chem. Phys. Lett., 504:180–184, 2011.

[23] Y. Hong, W. Wu, J. Hu, A.A. Voevodin, L. Chow, and M. Su. Langmuir, 17:908–918, 2001.

[24] M. Wutting and N. Yamada. Nat. Mater., 6:824–832, 2007.

[25] W. Su, L. Ma, Z. Sun, and C. Wang. Simultaneous detection of multiple biomarkers with over three orders of concentration difference using phase change nanoparticles. Anal. Chem., 83:2215–2219, 2011.

[26] M. Su, Y. Luo, M. Hossain, Zhang M., Hong Y., and C. Wang. Nanoscale, 4:3237–3241, 2012.

[27] J.P. Collete, P. Rochus, R. Peyrou-Lauga, Pin O., N. Nutal, M. Larnicol, and J. Crahay. Phase change material device for spacecraft thermal control. 62^{nd} International astronautical congres, 2011.

[28] A.D. Brent, V.R. Voller, and K.J. Reid. Enthapy-porosity technique for modeling convection-difussion phase change: application to the melting of a pure metal. Numerical Heat Transfer, 13:297–318, 1987.

[29] J. Martínez. Simulación de la dinámica de fusión en materiales con cambio de fase con nanopartículas dispersas en microgravedad. PhD thesis, Universidad Politécnica de Madrid, 2016.

[30] Swaminathan C.R. and Voller V.R. A general enthalpy method for modeling solidification processes. Metallurgical transactions B, 23B:651–664, 1992.

[31] F. Ziegler. The multiple meaning of the stefan-number(and relatives) in refrigeration. International journal of Refrigeration, 33(7):1343–1349, 2010.

[32] Bénard H. Les tourbillons cellulaires dans une nappe liquide. Rev.Gén.Sci Pure Appl., 11:1261–1271, 1900.

[33] M.J. Block. Surface tension as the cause of benard cells and surface deformation in a liquid film. Nature, 178(650), 1956.

[34] J.A. Maroto, V. Perez-Muñuzurri, and Romero-Cano M.S. Introductory analysis of bénard-marangoni convection. Europen journal of physics, 28:311–320, 2007.

[35] P. Lovass, Michal Branicki, Rita Tóth, Artur Braun, Kohta Suzuno, Daishin Ueyama, and István Lagzi. Maze solving using temperature-induced marangoni flow. RSC Advances, 5:48563–48568, 2015.

[36] D.A. Nield. titulofalso. J. Fluid Mech., 19:341, 1964.

[37] Diego M. Maza Ozcoidi. Transición al caos en convección de Bénard-Marangoni con pequeña relación de aspecto. PhD thesis, Universidad de Navarra, 1995.

[38] Charles Hirsch. Numerical Computation of Internal and External Flows (2ªEdition). Elsevier, 2007.

[39] O.C. Zienkiewicz and Y.K. Cheung. Finite elements in the solution of field problems. The Engineer, pages 507–510, 1965.

[40] S.V. Patankar. Numerical heat transfer and fluid flow. Hemisphere, 1990.

[41] B. Wiwatanapataphee, Y.H. Wu, Siew P.F. Archapitak, J., and B. Unyong. A numerical study of the turbulent flow of molten steel in a domain with a phase-change boundary. Journal of computational and applied mathematics, 116:297–318, 2004.

[42] C. Gau and R. Viskanta. Melting and solidification of a pure metal on a vertical wall. J. Heat Transfer, 108:174–181, 1986.

[43] J.A. Dantzig. Modelling liquid-solid phase changes with melt convection. Angew. Chem. Int. Ed, 53(15):1521–3773, 2014.

[44] M.M. Cerimele, D. Mansutti, and Pistella F. A front-fixing method for fluids in liquid/solid phase change with a benchmark test. Europena congres on computational methods in applied sciences and engineering, ECCOMAS 2000, Barcelona 11-14 September 2000.

[45] A.D. Brent, Voller V.R., and K.J. Reid. Enthalpy-porosity technique for modeling convection-diffusion phase change: application to the melting of a pure metal. Numerical heat transfer, 13:651–664, 1988.

[46] Chatterje D. Lattice boltzman modeling for melting/solidification processes. CSIR-Central Mechanical Engineering Research Institute, India, 2011.

[47] Yunus A. Çengel. Heat transfer: A practical approach 2nd edition. Higher Education, 1998.