

Developing a Liquid Fraction Incorporated Wet-Node Boundary Condition Scheme for Conjugate Heat Transfer in the Filter-Matrix Lattice Boltzmann Method

TN2983: Bachelor Thesis

Joran Wind

Developing a Liquid Fraction Incorporated Wet-Node Boundary Condition Scheme for Conjugate Heat Transfer in the Filter-Matrix Lattice Boltzmann Method

TN2983: Bachelor Thesis

by

Joran Wind

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on Friday February 20, 2026 at 14:00.

Student number: 5826063
Project duration: September 1, 2025 – February 20, 2026
Thesis committee: Dr. Ir. M. Rohde, TU Delft, supervisor
Dr. Ir. D. Lathouwers TU Delft

Abstract

The Molten Salt Fast Reactor could play a key role in future energy supply. A built-in safety feature of the MSFR is the freeze-plug. In case of emergency, this plug melts and leads away the molten salt. The melting process of the plug, in which conjugate heat transfer plays an important role, should therefore be thoroughly understood. In earlier research, conjugate heat transfer turned out to be incompatible with natural convection using the Filter-Matrix Lattice Boltzmann Method, as instabilities arose at the conjugate border. This could be the result of inaccurate information on the liquid fraction near the conjugate border. To get around this issue, a new liquid fraction incorporated boundary scheme was proposed and tested with respect to the numerical benchmark of Pater & Kaaks [1] and Borstlap [2] respectively. This was done using both the temperature and the total enthalpic approach of the Filter-Matrix Lattice Boltzmann Method.

The new scheme uses the total enthalpy of the nodes next to the boundary in the phase change material domain. This value is used to determine whether the phase front is near the conjugate boundary and to determine the temperature for the conjugate boundary condition.

In the first assessed case, a square cavity filled with molten salt adjacent to a cold pipe was considered. Grid convergence was tested for 4 different boundary condition extrapolation methods. Three of those methods showed grid convergence, while one yielded remained inconclusive. The first three methods showed second-order accuracy for small grid sizes, while the last again remained dubious. Lastly, energy conservation was checked for the different methods. The loss of energy from the system for the different methods remained between 0.04% and around 0.08%. One method actually showed an increasing energy, with a total increase of around 0.14%.

In the second case, first an intermediate configuration was studied. The model of Borstlap showed non-physical results near the conjugate boundary when freezing and subsequent melting took place. Three extrapolation methods were tested to see if they did better. One method acted similar to the model of Borstlap, one performed somewhat better and one returned a total fluid field. After this intermediate test, the method that acted similar and the one that returned the fluid field were used to model the results from benchmark cases by Pater & Kaaks [1] and Borstlap [2]. The first method showed good physical results and small errors at three chosen temperature probes ($<0.5\%$) and the overall phase front ($<1\%$). The method of the total fluid field showed grave non-physical results and seemed to not conduct any heat through the conjugate boundary, likely because it was not accurate enough to use in the newly developed algorithm. A future assessment of the method that handled the intermediate configuration somewhat better might prove valuable.

Contents

Abstract	i
1 Introduction	1
1.1 Molten Salt Fast Reactor	1
1.1.1 Design and advantages	1
1.1.2 The Freeze Plug	2
1.2 Lattice Boltzmann Method	3
1.3 Research Questions and Thesis Outline	3
1.3.1 Research Questions	3
1.3.2 Thesis Outline	3
2 Theory	4
2.1 Thermodynamics	4
2.1.1 Conservation of thermal energy	4
2.2 Physical processes in the freeze-plug	5
2.2.1 Heat conduction	5
2.2.2 conjugate heat transfer	6
2.3 Principles of the Lattice Boltzmann Method	6
2.3.1 Kinetic Theory	6
2.3.2 Discretizing the Boltzmann Equation	6
2.3.3 Linking the Boltzmann Equation and the Enthalpy Equations	7
2.3.4 The LBM Algorithm: the Basics	8
3 Numerical Method	9
3.1 Filter-Matrix method	9
3.1.1 D2Q9 scheme	9
3.1.2 Temperature FM-LBM	10
3.1.3 Total Enthalpy FM-LBM	11
3.1.4 Lattice Units	12
3.2 Boundary Conditions in the FM-LBM	12
3.2.1 Wet-Node Solution Vector Boundary Condition	12
3.2.2 Liquid-Fraction Incorporated WNSV BC	14
3.3 Algorithm	16
3.4 Numerical Assessment Methods	17
3.5 Assessed Cases	18
3.5.1 Case 1: Fluid Hot Salt Cavity with Adjacent Cold Pipe	18
3.5.2 Case 2: Benchmark Study	19
4 Results and Discussion	21
4.1 Case 1: Hot Salt Cavity	21
4.2 Case 2: Benchmark: Melting of the Freeze-Plug	26
4.2.1 Intermediate configuration	26
4.2.2 Full configuration	27
5 Conclusion and Recommendations	31
5.1 Conclusions	31
5.2 Recommendations	32
References	33
A Derivations of f_L Incorporated Boundary Temperatures	35

Introduction

In the Paris agreement of 2015, the world agreed to keep global warming 'well below 2 °C above pre-industrial levels' to avoid the catastrophic consequences of climate change[3]. To achieve this, greenhouse gas emissions must be drastically reduced. In a time where energy demand is increasing globally, a mix of renewable energy sources is required to decarbonize the energy sector. Although sources like solar, wind, and hydropower have a growing share of the total power supply, they come with intermittency issues as they are mainly dependent on weather conditions. In addition, supply and demand must be balanced at all times on the energy market. This means that if there is a higher demand, the supply should be adjusted immediately, which is not always possible with only solar and wind power. An instant adjustable energy source is therefore of high importance[4]. To create a stable backbone for our energy system, nuclear energy can be of considerable importance. Nuclear reactors have a limited effect on climate change in terms of emissions and can supply a consistent amount of energy during periods when other renewables do not. This has led to renewed interest in nuclear technology in recent years, with governments planning to invest billions in new reactors[5].

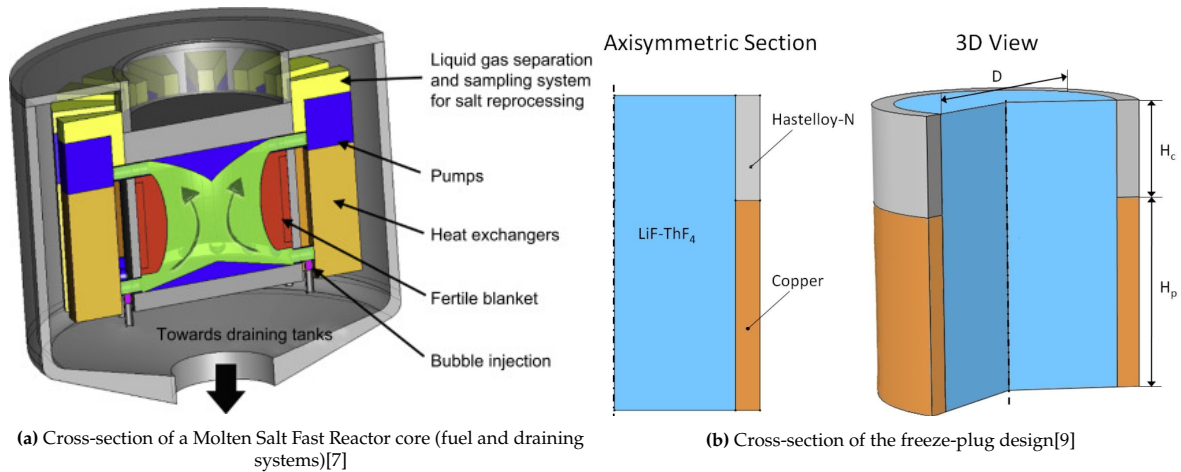
Nuclear energy comes with certain disadvantages, such as the risk of nuclear accidents and radioactive nuclear waste which has to be stored safely. To root out these risks, six generation IV reactors are being developed. One of those is the Molten Salt Fast Reactor (MSFR)[6].

1.1. Molten Salt Fast Reactor

1.1.1. Design and advantages

The first experimental studies on Molten Salt Reactors were performed in the 1950s and 1960s at Oak Ridge National Laboratory, but were eventually terminated because of the need for intensive chemical processing to remove fission products to avoid neutron captures leading to minor actinides. With the proposal for the Molten Salt Fast Reactor, using a fast neutron spectrum, the path to new research has opened again[7].

The current reference MSFR is a 3GW_{th} reactor with a total fuel volume of 18 m³, operated at a maximum fuel temperature of 750 °C under atmospheric pressure. The fuel salt is composed of lithium fluoride and thorium fluoride. The nuclear reactions take place in the core, with the salt serving both as fuel and as coolant. An illustration of the reactor core is shown in figure 1.1a. In the core, the salt flows from bottom to top, into one of 16 identical groups of pumps and heat exchangers that are located around the core. Here, the salt exchanges its heat to a secondary system for energy production. In the exchangers, helium bubbles are injected to separate gaseous and non-soluble fission products from the salt, to prevent corrosion inside the system[7, 8].



MSFRs offer some significant advantages over traditional pressurized water reactors. Firstly, they operate at atmospheric pressure, which is possible due to the high boiling temperature of the salt (around 1800 °C), reducing the risk of coolant leakage or system explosion. The risk of sudden rapid pressure increases is also eliminated by the liquid state stability of the fuel. Moreover, the MSFR produces less nuclear waste as a result of fuel recirculation. It could even use the waste of traditional reactors, solving the challenge of storing nuclear waste[4, 9]. The core of the reactor is connected to draining tanks by a pipe that is sealed with a so-called freeze-plug.

1.1.2. The Freeze Plug

The freeze-plug comes into play in the case of an emergency like a power outage, a failure inside the reactor core, a planned reactor shutdown or any other event that requires complete fuel drainage to prevent damage to the reactor. It is a valve made of frozen fuel salt, which is kept cold by cooling the surrounding pipe. In emergency cases, the cooling process shuts off and the melting process should occur passively, only depending on decay heat from the core and gravity. After melting, the fuel salt passes down through the pipe connecting the reactor core and drainage tanks, where it is cooled and stored[7, 9]. Figure 1.1b shows a schematic of the freeze-plug design.

The melting time of the freeze-plug is of great interest. To prevent any permanent damage to parts inside the reactor core, the salt should be drained before reactor components reach 1200 °C. It is not necessary for the plug to melt completely; only the circumference is enough for gravity to pull it down and open the drainage pipe. This adjacent, initially cooled pipe in which the valve is formed plays a vital role in the melting process. The pipe conducts decay heat from the reactor core, causing the melting process to be dominated mainly by heat conduction through the pipe[10]. It can be concluded that it is crucial to understand and simulate this process properly.

Several studies have been done on how the freeze-plug melts. Tiberge et al. [9] investigated the effects of various design parameters, such as recess depth and the form of the plug. They concluded that heat conduction through the surrounding pipe should be further looked into. Pater & Kaaks [1] did a numerical benchmark study for the freeze-plug with three different Computational Fluid Dynamics (CFD) codes; namely STAR-CCM+, OpenFOAM and DGFlows. This study emphasized carefully choosing the conjugate heat transfer modelling approach when simulating a problem where conjugate heat transfer and phase change are combined, as is the case when considering the freeze-plug. Borstlap [2] used the Filter-Matrix Lattice Boltzmann Method (FM-LBM) and a wet-node boundary approach to combine conjugate heat transfer, phase change and natural convection in the context of the freeze-plug, but this showed severe instabilities especially when phase change occurred right next to the conjugated wall. Further research must be done into combining conjugate heat transfer and phase change with the FM-LBM, before implementing natural convection into the model.

1.2. Lattice Boltzmann Method

The lattice Boltzmann method (LBM) is a CFD method with the Boltzmann equation as its physical basis. Its most standard use is as a second-order accurate solver for the compressible Navier-Stokes equation. It looks at the mesoscopic scale of fluids, meaning that the method considers particle distributions rather than individual particles. These distributions interact with each other through collisions and define the macroscopic properties of the system which is simulated. A big advantage is that interactions between nodes are linear, while non-linearity is treated locally. This makes the method ideal for parallel computing, for instance on GPUs, increasing the simulation speed[11].

This thesis makes use of multiple aspects of the method. Firstly, **conductive heat transfer resulting in phase change** is tracked. The LBM has proven to be successful in this area in earlier research[12]. Furthermore, **conjugate heat transfer** has been studied, including with a normal-based extrapolation method[13], with which second-order accuracy was achieved. As mentioned above, the **Filter-Matrix method** was used by Borstlap [2] and this will be extended to this thesis. It was originally proposed by Eggels et al. [14] and further improved by Zhuo & Zhong [15]. Bus & Rohde have also used it to model convective melting and solidification [16]. Its main advantage is its user-friendly control over macroscopic quantities, which is particularly beneficial in the case of boundary conditions. Furthermore, it is more stable than the more conventional BGK (Bhatnagar-Gross-Krook) operator, especially in turbulent and high Prandtl-number flows.

1.3. Research Questions and Thesis Outline

As discussed above, additional research should be done on combining conjugate heat transfer and phase change in the FM-LBM before natural convection can be added to the model and the melting time of the freeze-plug configuration can be accurately predicted. This thesis will therefore focus on further developing a conjugate heat transfer boundary condition using the FM-LBM in the context of the freeze-plug, with emphasis on phase change close to the conjugated wall. To reach this goal, first a new boundary condition scheme is developed that incorporates the latent heat of the phase change material in different extrapolation orders. This scheme is then analyzed by checking for grid convergence and determining its order of accuracy. Energy conservation will also be verified. Secondly, the model is benchmarked with the studies of Pater & Kaaks [1] and Borstlap [2]. The model used builds upon the one developed by Borstlap[2], in an effort to make it possible to later implement natural convection.

1.3.1. Research Questions

This leads to the following research questions:

1. How can the liquid fraction near a conjugate boundary be taken into account in the wet-node solution vector boundary condition scheme in the FM-LBM while imposing conjugate boundary conditions?
2. How does a liquid-fraction-incorporated conjugate boundary condition scheme perform with respect to known numerical benchmarks?

1.3.2. Thesis Outline

In this chapter, a short introduction to the MSFR was given, as well as to the Lattice Boltzmann Method. The existing literature on the research topic was briefly discussed and the research questions were formulated. In order to properly answer these questions, this thesis will have the following outline. Chapter 2 will discuss the necessary theoretical background of the physics used in conjugate heat transfer and phase change and will give an introduction to the theory behind the Lattice Boltzmann Method. Thereafter, chapter 3 describes the Filter-Matrix LBM and the developed boundary condition schemes. Chapter 4 shows and discusses the results of the simulations. Finally, chapter 5 concludes this thesis and provides recommendations for further research.

2

Theory

Before answering the research questions, a basic understanding of the ruling equations is necessary. This chapter will therefore first explore the field of thermodynamics in section 2.1. Subsequently, the physical phenomena in the freeze-plug considered in this thesis are discussed in section 2.2. The chapter concludes with the principles of the lattice Boltzmann theory in section 2.3.

2.1. Thermodynamics

2.1.1. Conservation of thermal energy

One of the main principles of physics is that the energy of a closed system is always conserved. This principle can be formulated in different ways, of which two will be presented in this section. This will first be done in terms of temperature, which is sufficient for treating heat transfer in materials without phase change. If phase change does occur, a modified version of the first equation is needed, expressed in terms of enthalpy. This second equation will be more suitable for phase change materials.

Energy conservation in terms of temperature

In an incompressible fluid with constant specific heat c_p , the equation for thermal energy conservation is given by:

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \mathbf{u} \cdot \nabla T = \rho c_p \nabla \cdot (\alpha \nabla T) + Q''' \quad (2.1)$$

Where T is the temperature, c_p the specific heat, α thermal diffusivity and Q''' a volumetric heat source (or sink) [W/m^3]. Viscous dissipation is neglected here. The left-hand side describes the rate of change of temperature in the volume and the energy moved with the fluid flow. The right-hand side includes heat conduction and a source term[17]. As mentioned above, this equation can be used in materials without phase change. In the context of the freeze-plug, this applies to the pipe surrounding the valve.

Energy conservation in terms of enthalpy

In phase change materials, latent heat has to be considered. In these situations, the **total** enthalpy H is a more effective quantity to use instead of temperature. To eventually arrive at an equation for conservation of total enthalpy, the total enthalpy is split into two parts. The first part is the sensible heat h , the energy that heats up a material. The second part is the latent heat ΔH , which is released or stored in a phase change and does not actually heat up a material[16].

$$H \equiv h + \Delta H = h + f_L L \quad (2.2)$$

Where f_L corresponds to the liquid fraction of the material, which defines the phase; 0 for solid, 1 for liquid and between 0 and 1 if the material is in the *mushy* zone, meaning it is in the process of changing phase. It is calculated by:

$$f_l = \begin{cases} 0 & \text{if } H < H_s \\ \frac{H-H_s}{H_l-H_s} & \text{if } H_s \leq H < H_l \\ 1 & \text{if } H \geq H_l \end{cases} \quad (2.3)$$

If the heat source in equation 2.1 is considered to be caused by absorption or discharge of latent heat, it can be described as[17]:

$$Q''' = -\left[\frac{\partial(\rho\Delta H)}{\partial t} + \nabla \cdot (\rho \mathbf{u} \Delta H)\right] \simeq -\frac{(\partial \rho f_L L)}{\partial t} \quad (2.4)$$

Thereby ignoring the second term for pure materials due to uniform latent enthalpy for the liquid[17]. Plugging equation 2.4 into equation 2.1 and substituting $c_p \nabla T$ for ∇h , one obtains the following:

$$\rho^\phi \frac{\partial h^\phi}{\partial t} + \rho^\phi \mathbf{u} \cdot \nabla h^\phi = \nabla \cdot (\rho^\phi \alpha^\phi \nabla h^\phi) - \rho^\phi \frac{\partial f_L^\phi L}{\partial t} \quad (2.5)$$

Where ϕ describes the phase, being liquid, solid, or mushy (respectively, l, s, m). By assuming constant c_p^ϕ and ρ^ϕ throughout unchanged phases and combining the two transient terms in equation 2.5, one arrives at the desired conservation equation for the total enthalpy H [2, 16, 17]:

$$\frac{\partial H^\phi}{\partial t} + \mathbf{u} \cdot \nabla h^\phi = \nabla \cdot (\alpha^\phi \nabla h^\phi) \quad (2.6)$$

This equation can be used in the so-called total enthalpy-based approach. Its main advantage is that temperature and enthalpy are tracked together, saving considerable computational resources[17].

The temperature can instinctively be recovered from the enthalpy:

$$T = \begin{cases} H/c_p^s & \text{if } H < H_s \\ T_s + f_L(T_l - T_s) & \text{if } H_s \leq H < H_l \\ T_l + \frac{H-H_l}{c_p^l} & \text{if } H \geq H_l \end{cases} \quad (2.7)$$

Where T_s and T_l are the solidification and liquidus temperatures and H_s and H_l the corresponding enthalpies given by $H_s = c_p^s T$ and $H_l = H_s + L$ [17]. In the mushy zone and in the case of different values in different phases, the specific heat and thermal conductivity are given by[2]:

$$c_p^m = \frac{c_p^s + c_p^l}{2}, \lambda^m = \frac{\lambda^s + \lambda^l}{2} \quad (2.8)$$

2.2. Physical processes in the freeze-plug

The goal of this study is to successfully combine two phenomena in the freeze-plug; phase change and conjugate heat transfer. The concept of liquid fraction and latent heat has already been introduced, so in this section heat conduction and conjugate heat transfer will briefly be discussed.

2.2.1. Heat conduction

If the fluid velocity is zero and the specific heat c_p^ϕ is not dependent on the location, heat transport by convection no longer occurs and equation 2.6 reduces to Fourier's law:

$$\frac{\partial H^\phi}{\partial t} = \nabla \cdot (\alpha^\phi c_p^\phi \nabla T^\phi) \quad (2.9)$$

2.2.2. conjugate heat transfer

Conjugate heat transfer takes place at a solid-fluid interface. At this interface, two boundary conditions arise:

$$T_{f,W} = T_{s,W} \quad (2.10)$$

$$-\lambda_f \frac{\partial T}{\partial n}|_{f,W} = -\lambda_s \frac{\partial T}{\partial n}|_{s,W} \quad (2.11)$$

With $T_{f,W}$ and $T_{s,W}$ denoting the temperature evaluated at the boundary, λ_f and λ_s the thermal conductivities of the fluid and solid and n the normal direction of the interface [2].

In the process of solving the conjugate boundary problem in this work, the heat equations of the fluid and solid domains are solved sequentially. The phase change material is handled first using the total enthalpy-based approach and the wall structure second using the temperature-based approach.

2.3. Principles of the Lattice Boltzmann Method

Modeling equation 2.1 and equation 2.6, which describe thermodynamics at the macroscopic level, turns out to be fairly hard since analytical solutions are not available for most situations. To avoid this challenge, the Lattice Boltzmann Method can be used. This method deals with thermodynamics on a mesoscopic scale. Where macroscopic approaches keep track of properties like temperature or density and microscopic approaches look at individual particles, the LBM keeps track of particle *distributions*.

In order to get an understanding of the LBM, first kinetic theory is discussed. Thereafter, the obtained Boltzmann equation is discretized and linked to equation 2.1 and equation 2.6. In the end, the basic steps of the method are presented.

2.3.1. Kinetic Theory

Kinetic theory describes fluids and gasses on a mesoscopic scale with its main quantity, the particle distribution function $f(\mathbf{x}, \boldsymbol{\xi}, t)$. This function not only describes the usual density $\rho(x, t)$, but also incorporates a three-dimensional velocity space[11]. Both density and velocity can be calculated from f with the following equations:

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \boldsymbol{\xi}, t) d^3 \boldsymbol{\xi} \quad (2.12)$$

$$\rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) = \int \boldsymbol{\xi} f(\mathbf{x}, \boldsymbol{\xi}, t) d^3 \boldsymbol{\xi} \quad (2.13)$$

When differentiated with respect to time, one finds how the particle distribution evolves over time. If this derivation is performed, the Boltzmann equation is obtained:

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} f + \frac{F}{\rho} \nabla_{\boldsymbol{\xi}} f \equiv \Omega(f) \quad (2.14)$$

On the left-hand side, the distribution function is advected with the velocity $\boldsymbol{\xi}$ of its particles and forces influencing the velocity. On the right is the *collision operator*. It acts as a source term and denotes the local redistribution of f due to collisions (hence the name)[11]. What this operator looks like is discussed in chapter 3.

2.3.2. Discretizing the Boltzmann Equation

Solving the obtained Boltzmann equation analytically is still quite hard. However, it can be solved fairly easily numerically. To do so, the discretized particle distribution function is introduced first: $f_i(\mathbf{x}, t)$, which is defined in a lattice of points (or nodes) $\mathbf{x}$ with spacing Δx at times t with time steps Δt . Δx is

given in lattice space $[ls]$ and Δt in lattice time $[lt]$. f_i is discretized in a set of velocities $\mathbf{c}_i$. The velocities c_i come with corresponding weighting coefficients w_i and form velocity sets $\{c_i, w_i\}$. The number of spatial dimensions d the sets cover and the number of discrete velocities q lead to a scheme called DdQq. Among the most commonly used schemes are D1Q3, D2Q9 and D3Q19 [11]. The density and velocity can still be calculated with weighted sums of all $f_i(\mathbf{x}, t)$

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t) \quad (2.15)$$

$$\rho \mathbf{u}(\mathbf{x}, t) = \sum_i \mathbf{c}_i f_i(\mathbf{x}, t) \quad (2.16)$$

By differentiating f_i with respect to time, the *lattice* Boltzmann equation is found:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Delta t \Omega_i(\mathbf{x}, t) \quad (2.17)$$

2.3.3. Linking the Boltzmann Equation and the Enthalpy Equations

Now that the numerical Boltzmann equation is derived, one might ask how it simplifies solving the Navier-Stokes equations or equation 2.1 and equation 2.6. The key is that they are connected via the *Chapman-Enskog* analysis, which shows that the Navier-Stokes equations can be derived from the Boltzmann equation. The Navier-Stokes equations are given by[18]:

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = -\nabla P + \rho \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (2.18)$$

Where ∇P is the pressure gradient, ν the kinematic viscosity and $\mathbf{f}$ external force. After some modifications are made

$$\rho \mathbf{u} \rightarrow \rho c_p T, \quad \rho \mathbf{u} \nabla \mathbf{u} + \nabla P \rightarrow \rho c_p \mathbf{u} \cdot \nabla T, \quad \nu \rightarrow \alpha, \quad \mathbf{f} \rightarrow Q''' \quad (2.19)$$

the thermal energy conservation equation 2.1 is retrieved. This means that the temperature and enthalpy conservation equations can be solved using an adjusted version of the LBM. This starts by defining a discretized temperature distribution function g_i and a discretized **total** enthalpy distribution m_i . They have the same velocity sets as f_i in the DdQq schemes. How they transform over time steps Δt is respectively given by[2]:

$$g_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = g_i(\mathbf{x}, t) + \Delta t \Omega_i^g(\mathbf{x}, t) \quad (2.20)$$

$$m_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = m_i(\mathbf{x}, t) + \Delta t \Omega_i^m(\mathbf{x}, t) \quad (2.21)$$

The temperature T and the total enthalpy H are determined in a similar way to how the density was gained from f_i , namely:

$$T(\mathbf{x}, t) = \sum_i g_i(\mathbf{x}, t) \quad (2.22)$$

$$H(\mathbf{x}, t) = \sum_i m_i(\mathbf{x}, t) \quad (2.23)$$

2.3.4. The LBM Algorithm: the Basics

The LBM algorithm works with time steps Δt . In each of those time steps, two things happen after each other. First, the collision step takes place. The post-collision distribution function f^* is determined by the collision operator:

$$f^*(\mathbf{x}, t) = f_i(\mathbf{x}, t) + \Omega_i(\mathbf{x}, t) \quad (2.24)$$

After the collision, propagation to neighbouring nodes in the direction of the velocities c_i takes place. One ends up with:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f^*(\mathbf{x}, t) \quad (2.25)$$

And one time step is completed[11].

3

Numerical Method

In the previous chapter, the principles of the lattice Boltzmann method were presented. The collision operator Ω_i was introduced, but has yet to be defined. This chapter starts by detailing how the LBM is put to use in this thesis. In section 3.1 the Filter-Matrix method is introduced as collision scheme. Section 3.2 describes how boundary conditions can be handled within this scheme. The new boundary condition scheme tested in this research is also introduced. Next, section 3.3 describes the algorithm with which the simulations are done. Section 3.4 will elaborate on the numerical assessment methods used to test the newly formulated boundary condition scheme. The concluding section 3.5 introduces the cases that are assessed.

3.1. Filter-Matrix method

This thesis makes use of the Filter-Matrix LBM (FM-LBM), which is more accurate and has greater stability than the more conventional lattice Bhatnagar-Gross-Krook model, which works with a collision operator that (in a really short description) brings the distribution towards its local equilibrium in a relaxation time τ [15]. Furthermore, the FM-LBM gives relatively easy control over macroscopic quantities such as enthalpy and temperature, which is especially convenient when dealing with boundary conditions.

3.1.1. D2Q9 scheme

Before entering the realm of the FM-LBM, first a choice must be made on which $DdQq$ is used. In this study, the D2Q9 scheme is used. This means that simulations are performed in 2 dimensions with 9 discrete velocities $\mathbf{c}_i$. A schematic view of these 9 velocities is given in figure 3.1.

As mentioned in section 2.3.2, the velocities come in sets with corresponding weighting coefficients. For D2Q9, these are[11]:

$$w_i = \begin{cases} \frac{4}{9} & \text{for } i = 0 \\ \frac{1}{9} & \text{for } i = 1, 2, 3, 4 \\ \frac{1}{36} & \text{for } i = 5, 6, 7, 8 \end{cases} \quad (3.1)$$

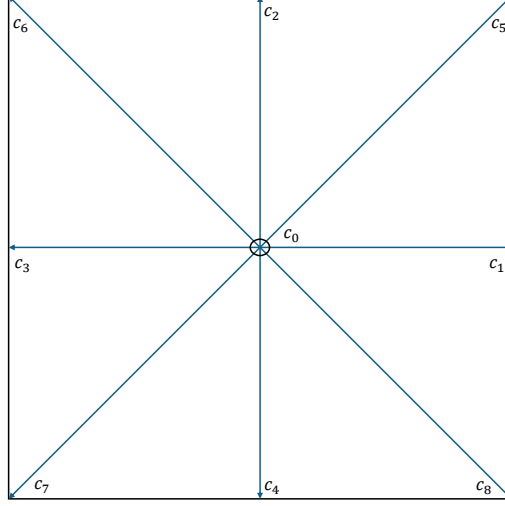


Figure 3.1: Schematic overview of the velocities in the D2Q9 scheme. There are 8 'moving' velocities c_{1-8} , and a static velocity c_0 which stays at the node in the middle

3.1.2. Temperature FM-LBM

To derive the FM-LBM, first equation 2.20 is shifted half a grid point and half a time step:

$$g_i(\mathbf{x} + \frac{\mathbf{c}_i \Delta t}{2}, t + \frac{\Delta t}{2}) - g_i(\mathbf{x} - \frac{\mathbf{c}_i \Delta t}{2}, t - \frac{\Delta t}{2}) = \Delta t \Omega_i(g) \quad (3.2)$$

After a first order Taylor expansion is performed, the function looks as follows:

$$g_i(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}) = g_i(\mathbf{x}, t) \pm \frac{\Delta t}{2} (\mathbf{c}_i \cdot \nabla \pm \partial_t) g_i(\mathbf{x}, t) + \mathcal{O}(\Delta t^2) \quad (3.3)$$

$$= g_i(\mathbf{x}, t) \pm \frac{\Delta t}{2} \Omega_i(g_i) + \mathcal{O}(\Delta t^2) \quad (3.4)$$

A definition of the collision operator is found in the form of a matrix multiplication[2, 15]:

$$g_i(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}) = \sum_k w_i E_{ik} \beta_k^\pm(\mathbf{x}, t) \quad (3.5)$$

Where E_{ik} is the filter-matrix (hence the name of the method) and $\beta^\pm$ the temperature solution vector. β^- is the pre-collision solution vector, β^+ the post-collision solution vector. The solution vector can then be found by a reversed calculation:

$$\beta^\pm(\mathbf{x}, t) = \sum_i E_{ki} g_i(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}) \quad (3.6)$$

With $(E_{ki})^{-1} = w_i E_{ik}$. E_{ki} and $\beta^\pm$ are given by, respectively:

$$E_{ki} = [1, c_{ix}, c_{iy}, 3c_{ix}^2 - 1, 3c_{ix}c_{iy}, 3c_{iy}^2 - 1, c_{ix}(3c_{iy}^2 - 1), c_{iy}(3c_{ix}^2 - 1), \frac{(3c_{ix}^2 - 1)(3c_{iy}^2 - 1)}{2}]^T \quad (3.7)$$

$$\beta^\pm(\mathbf{x}, t) = \begin{bmatrix} T \\ u_x T + \frac{-6\alpha \pm \Delta t}{6} \frac{\partial T}{\partial x} \\ u_y T + \frac{-6\alpha \pm \Delta t}{6} \frac{\partial T}{\partial y} \\ S_1^\pm \\ S_2^\pm \\ S_3^\pm \\ T_1^\pm \\ T_2^\pm \\ F^\pm \end{bmatrix} \quad (3.8)$$

With $S_{1,2,3}^\pm$, $T_{1,2}^\pm$ and $F^\pm$ higher-order terms that are set to zero in calculations[15].

3.1.3. Total Enthalpy FM-LBM

Now that the collision scheme for temperature is known, the step to the enthalpy scheme is straightforward, though one important aspect should be noted. In the enthalpy scheme, the **total** enthalpy distribution is first divided into two parts; the sensible part m_i^s and the latent part m_i^l :

$$m_i = m_i^s + m_i^l \quad (3.9)$$

$$m_i^l = [f_L \cdot L, 0, 0, 0, 0, 0, 0, 0]^T \quad (3.10)$$

One can see that the latent part of the enthalpy stays at the same node, as it is part of the static part of the distribution. The collision step is performed with the sensible part m_i^s [16].

The remaining part of the collision step is very similar to that of the temperature distribution. The filter-matrix E_{ki} is the same and the mathematical description looks as follows:

$$m_i^s(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}) = \sum_k w_i E_{ik} \gamma_k^\pm(\mathbf{x}, t) \quad (3.11)$$

Where, with respect to equation 3.5 $\beta_k^\pm(\mathbf{x}, t)$ is replaced by $\gamma^\pm(\mathbf{x}, t)$, the new enthalpy solution vector. $\gamma^\pm(\mathbf{x}, t)$ is calculated and given by, respectively:

$$\gamma^\pm(\mathbf{x}, t) = \sum_i E_{ki} m_i^s(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}) \quad (3.12)$$

$$\gamma^\pm(\mathbf{x}, t) = \begin{bmatrix} h \pm \frac{\Delta t}{2} \frac{Q}{\rho} \\ u_x h + \frac{-6\alpha \pm \Delta t}{6} \frac{\partial h}{\partial x} \\ u_y h + \frac{-6\alpha \pm \Delta t}{6} \frac{\partial h}{\partial y} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.13)$$

Where higher-order terms are, once more, set to zero. In the first element of the solution vector, a source term Q is introduced, which conceptually differs from the temperature solution vector. The final step in the collision is to add the subtracted latent heat distribution back to the sensible enthalpy distribution, to regain the total enthalpy distribution function.

3.1.4. Lattice Units

LBM simulations are done in so-called lattice units. To obtain these lattice units from (usually) the SI units of some physical quantity ϕ , a conversion factor is defined:

$$C_\phi = \frac{\phi}{\phi^*} \quad (3.14)$$

With ϕ being the physical quantity in SI units and ϕ^* the quantity in lattice units. The basis of conversion factors used in this research is:

$$C_L = \frac{\Delta x}{\Delta x^*}, C_v = \frac{v}{v^*}, C_\rho = \frac{\rho}{\rho^*}, C_T = 1 \quad (3.15)$$

where Δx^* and $\Delta \rho^*$ are set to unity. For a wet-node grid, C_l (L for length) is given by $\frac{L_x}{(N_x-1)}$. The wet-node grid will be discussed in the next section 3.2. Although velocity does not play a role in the simulations done in this thesis, lattice viscosity is still chosen as a free parameter to make it easier to eventually implement a momentum solution vector in future research. The only other free parameter is then Δx^* . Together, the conversion factors from equation 3.15 form a complete set, with which all other conversion factors can be calculated. Of them, the most notable is the time conversion factor, given by $C_t = \frac{C_l^2}{C_v}$. In the rest of this report, the (*)-notation for lattice units will be omitted. Note that the lattice unit of space is ls (lattice spacing) and for time lt (lattice time).

3.2. Boundary Conditions in the FM-LBM

The main focus of this thesis is to test a newly developed boundary condition scheme that combines conjugate heat transfer and phase change at the conjugate border. Therefore, a thorough understanding of how boundary conditions function in the FM-LBM is of significant importance. After taking a look at the solution vectors in equations 3.8 and equation 3.13, it can be concluded that the FM-LBM gives relatively good access and control over the most important macroscopic properties, temperature and (sensible) enthalpy. These properties are extracted from the solution vectors, simply by taking the first element of the vectors. This element can then be altered to some desired value and then be transformed back into the lattice Boltzmann *space*, after which the LBM algorithm can continue[2].

3.2.1. Wet-Node Solution Vector Boundary Condition

Wet-node scheme

The method used to impose boundary conditions that will be used is called the Wet-Node Solution Vector Boundary Condition (WNSV BC), originally proposed and derived by Rohde and later analyzed by Borstlap[2]. Before the procedure that is followed is explained, one must first know what a wet-node scheme is. In a wet-node scheme, the computational border lies infinitesimally close to the physical boundary, but still inside the fluid domain. This means that the standard LBM algorithm is also performed on the boundary nodes[2, 11]. A schematic view is given in figure 3.2.

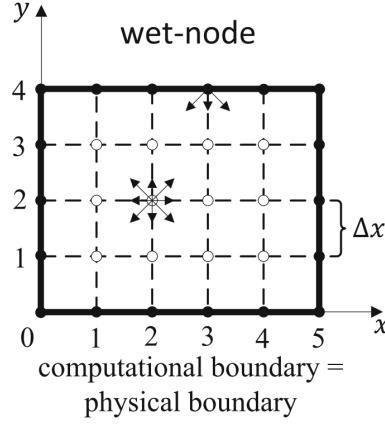


Figure 3.2: Schematic view of the wet-node boundary scheme. The open circles depict fluid nodes, solid circles boundary nodes[11]

Types of boundary conditions

In this paper, two types of boundary conditions are addressed. First off, adiabatic boundary conditions on the outer edges (Neumann BC) and secondly the conjugate boundary condition at the phase change material(PCM)-solid interface at the wall of the freeze-pipe (combination of Neumann and Dirichlet BC).

Imposing the boundary conditions

The process of imposing the desired boundary conditions will first be explained for the temperature solution vector, after which some adjustments are given in section 3.2.2 to apply it to the enthalpy solution vector.

The first step in the process is making an (educated) guess for the current solution vector at the boundary $(\beta_k^{i=0})_{extrap}$. This is done by performing an extrapolation with known neighbouring solution vectors, see figure 3.3. In this study, extrapolation is done with various schemes:

$$(\beta_k^{i=0})_{extrap} = \beta_k^{i=1,t} - \beta_k^{i=1,t-1} + \beta_k^{i=0,t-1} \quad (3.16a)$$

$$(\beta_k^{i=0})_{extrap} = \frac{5}{2}\beta_k^{i=1} - 2\beta_k^{i=2} + \frac{1}{2}\beta_k^{i=3} \quad (3.16b)$$

$$(\beta_k^{i=0})_{extrap} = \beta_k^{i=0,t-1} - 2\beta_k^{i=1,t-1} + \beta_k^{i=2,t-1} + 2\beta_k^{i=1,t} - \beta_k^{i=2,t} \quad (3.16c)$$

With equation 3.16a a 1st order spatial and temporal extrapolation, equation 3.16b a 2nd order spatial extrapolation and equation 3.16c a 2nd order spatial and 1st order temporal extrapolation.

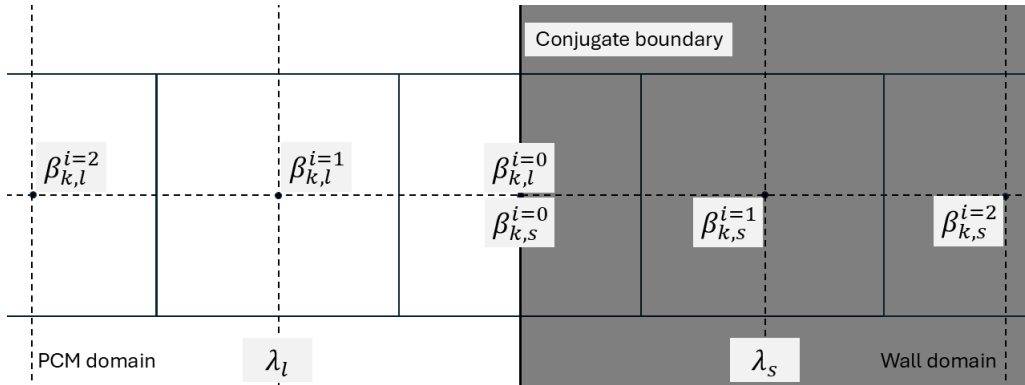


Figure 3.3: Nodes in the PCM and wall domains with temperature solution vectors β_k^i .

The second step is to impose the boundary condition. To this end, all vector elements in the extrapolated solution vector should be changed to the desired values. This is done in the following way:

$$\beta_k^{i=0} = (\beta_k^{i=0})_{extrap} - \begin{bmatrix} T_{extrap} \\ u_x T_{extrap} \\ u_y T_{extrap} \\ 0, 0, 0, 0, 0, 0 \end{bmatrix} + \begin{bmatrix} T_{BC} \\ u_x T_{BC} \\ u_y T_{BC} \\ 0, 0, 0, 0, 0, 0 \end{bmatrix} \quad (3.17)$$

Where T_{extrap} can be extracted from the extrapolated solution vector, as it is its first element $(\beta_0^{i=0})_{extrap}$. T_{BC} is logically the desired boundary value.

After writing out expansions for the temperature gradients on the walls, the value of T_{BC} for Neumann boundary conditions is given by:

$$T_{BC} = \beta_0^{i=1} \text{ (first order)} \quad (3.18a)$$

$$T_{BC} = \frac{4}{3}\beta_0^{i=1} - \frac{1}{3}\beta_0^{i=2} \text{ (second order)} \quad (3.18b)$$

For the conjugate boundary, equation 2.10 and equation 2.11 are combined to the following:

$$T_{BC} = \frac{1}{\lambda_s + \lambda_l} (\lambda_l \beta_{0,l}^{i=1} + \lambda_s \beta_{0,s}^{i=1}) \text{ (first order)} \quad (3.19a)$$

$$T_{BC} = \frac{1}{3(\lambda_s + \lambda_l)} [\lambda_l (\beta_{0,l}^{i=2} - 4\beta_{0,l}^{i=1}) + \lambda_s (-\beta_{0,s}^{i=2} + 4\beta_{0,s}^{i=1})] \text{ (second order)} \quad (3.19b)$$

See figure 3.3 for a clarification on which nodes are used.

The third and last step is to translate the constructed boundary solution vector back to LB space:

$$g_i(x=0, t) = \sum_k w_i E_{ik} \beta_k^{i=0} \quad (3.20)$$

Note that this equation differs from equation 3.5, as no additional (half) time-step is performed. The entire process of imposing a boundary condition is performed in one and the same time-step [2].

3.2.2. Liquid-Fraction Incorporated WNSV BC

In this section, the process of imposing boundary conditions in the phase change material (PCM) is described. The PCM makes use of the enthalpy solution vector. First, a distinction is made between Neumann BCs and conjugate BCs.

Neumann boundary conditions

Neumann BCs are implemented similarly to the temperature solution vector case. The same steps are performed, but temperature properties are replaced by enthalpy properties:

$$\beta \rightarrow \gamma, \quad T_{extrap} \rightarrow h_{extrap}, \quad g_i \rightarrow m_i^s \quad (3.21)$$

The desired boundary temperature must be converted to an enthalpy value. This can be done by a modified version of equation 2.7:

$$H_{BC} = \begin{cases} c_p^s T_{BC} & \text{if } f_L^{i=0} = 0 \\ c_p^s T_s + f_L^{i=0} L & \text{if } 0 < f_L^{i=0} < 1 \\ c_p^s T_s + c_{p,l}(T_{BC} - T_l) + L & \text{if } f_L^{i=0} = 1 \end{cases} \quad (3.22)$$

Conjugate boundary conditions

The collision step of the enthalpy solution vector differs slightly from the temperature solution vector, as can be recalled from section 3.1.3. Before collision, the latent heat at the node is subtracted from the static enthalpy population, so that m_i^s is retrieved (equation 3.9). Hence, one must be careful with which vectors to use in the first step of the BC-process to be sure that in every step the *sensible* enthalpy is considered. However, via this approach it may be difficult to retain knowledge of the local value of the liquid fraction, since the latent heat was subtracted earlier on. In the earlier approach by Borstlap, the expansions used in the temperature BC process (3.19a, 3.19b) were also applied to the conjugate boundaries. Instabilities arose when the phase front was at the conjugate boundary and natural convection was taken into account[2]. Natural convection causes velocities on fluid nodes. To suppress these velocities at the phase front and to ensure no-slip boundary conditions, an accurate value of the liquid fraction at the front is necessary. In this study, a liquid-fraction-incorporated approach is tested for conjugate boundary conditions with the aim of tracking the liquid fractions at the boundary nodes more precisely. Its algorithm is similar to that of the temperature BCs and of the Neumann enthalpy BCs, but has some modifications. Most importantly, the liquid fraction and thus the latent heat is taken into account while extrapolating to find an educated guess of the boundary node solution vector. The algorithm will be given in a stepwise description.

Calculating the sensible enthalpy and f_L at the boundary node

The first step is, again, to extrapolate the neighbouring nodes to gain an approximation of the boundary node solution vector. This is not done with the sensible enthalpy distribution or the enthalpy solution vector, but with the full enthalpy distribution m_i (see equation 2.21). β_k^i in equation 3.16 is replaced by m_i^x (see figure 3.4 for more clarity). After extrapolation, two things are calculated from the extrapolated $m_i^{x=0}$: the total enthalpy H_{extrap} with equation 2.23 and subsequently the liquid fraction f_L at the boundary node using equation 2.3. Now, having the total enthalpy H_{extrap} and the liquid fraction f_L , the sensible enthalpy h_{extrap} can be calculated with equation 2.2.

T_{BC} determined with f_L near boundary

The next step is to determine T_{BC} . In order to do this accurately, the liquid fraction at the conjugate border is taken into account. To understand how this is done, it must first be understood that the properties in the nodes can be seen as properties of a surrounding cell. See figure 3.4.

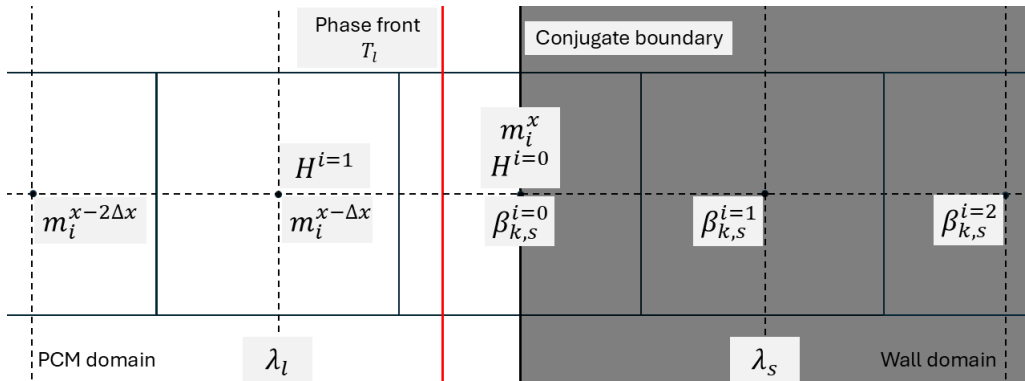


Figure 3.4: Nodes in the solid (black) and PCM (white) domain, with their according cells and labels. The red line depicts the phase front.

The liquid fraction at some node thus represents where the phase front is in the corresponding cell. In reality, the boundary cell is only half the size of other, non-boundary cells. The red line in figure 3.4 represents the phase front. Physically, it cannot exceed the boundary. f_L is calculated using the local total enthalpy, which is actually an enthalpy *density*. This means that a value between 0 and 1 is still reasonable, as the volume of the cell is not of interest in the calculation. An f_L value of 0 (no liquid) therefore means that the phase front is exactly at the boundary or not present in the boundary cell. A value of 1 implies that the phase front is at the other end of the cell or has gone out of the boundary cell. The temperature of the phase front is known to be the liquidus temperature T_l of the PCM (the fuel salt

of the reactor, in our case). This value can thus be used to perform an extrapolation to determine the boundary node temperature. In the Taylor expansion used to formulate a new T_{BC} , the liquid fraction is scaled to give the location of the phase front in the boundary cube:

$$f_{L,n} = \begin{cases} \frac{1}{2} \frac{H^{i=0} - H_s}{H_l - H_s} & \text{if } H^{i=1} \geq H_l \\ \frac{1}{2} - \left(\frac{1}{2} \frac{H^{i=0} - H_s}{H_l - H_s} \right) & \text{if } H^{i=1} \leq H_l \end{cases} \quad (3.23)$$

Naturally, this is done only when the phase front is inside the boundary cell. In other words, when $H_s < H^{i=0} < H_l$. Otherwise, the old extrapolation equation 3.19 can be used. This leads to the following formulation of a new liquid-fraction-incorporated T_{BC} :

$$T_{BC} = \frac{1}{\lambda_l - \lambda_s(f_{L,n} - \frac{1}{2})} [\lambda_l T_l - \lambda_s(f_{L,n} - \frac{1}{2})\beta_{0,s}^{i=1}] \quad (3.24a)$$

$$T_{BC} = \frac{\lambda_l(T_l - \beta_{0,l}^{i=1}(\frac{1}{4} - f_{L,n} + f_{L,n}^2)) - \lambda_s(f_{L,n}^2 - \frac{1}{4})(2\beta_{0,s}^{i=1} - \frac{1}{2}\beta_{0,s}^{i=2})}{\lambda_l(\frac{3}{4} + f_{L,n} - f_{L,n}^2) - \frac{3}{2}\lambda_s(f_{L,n}^2 - \frac{1}{4})} \quad (3.24b)$$

For a more elaborate explanation of how $f_{L,n}$ and both equations 3.24a and 3.24b were derived, see Appendix-A.

Calculating h_{BC} and the new f_L

The following step is to convert T_{BC} to a value of the sensible enthalpy. This is done through:

$$h_{BC} = \begin{cases} c_p^s T_{BC} & \text{if } T_{BC} \leq T_l \\ c_p^s T_s + c_{p,l}(T_{BC} - T_l) & \text{if } T_{BC} > T_l \end{cases} \quad (3.25)$$

while simultaneously calculating the new liquid fraction resulting from the boundary condition:

$$f_{LBC}^{i=0} = \begin{cases} 0 & \text{if } T_{BC} < T_s \\ \frac{T_{BC} - T_s}{T_l - T_s} & \text{if } T_s < T_{BC} < T_l \\ 1 & \text{if } T_{BC} > T_l \end{cases} \quad (3.26)$$

Now all desired values are known to fill in equation 3.17 (of course again by first swapping out $\beta_k^{i=0}$ for $\gamma_k^{i=0}$ and T_{extra}, T_{BC} for h_{extra}, h_{BC}). Now the boundary condition solution vector is constructed, it can be transformed back to LB-space with equation 3.20. The sensible enthalpy distribution is then retrieved. The last and final step is to add the latent heat back with the new liquid fraction from equation 3.26. At last, the total enthalpy distribution at the boundary is calculated, accompanied by its up to date liquid fraction.

3.3. Algorithm

In this section, an overview of the FM-LBM algorithm is given. The salt domain will be named the phase change material (PCM) and the adjacent solid will be referred to as the wall. In the PCM, the enthalpy distribution function is of interest, and in the wall the temperature distribution function. A third one, the momentum distribution function f_i from section 2.3.1, plays a minor role in the algorithm. It is commenced to make sure all used computational functions work properly, but has no place in collisions and is not propagated since the velocity stays zero. Simulations are run with Python in VSCode.

Initialization

1. PCM: The initial macroscopic properties of the PCM are chosen by the user. Then a liquid fraction field is created, chosen by the user. It can be all fluid, all liquid or somewhere in between. With these properties, the distribution functions f_i and m_i are constructed. The total enthalpy and temperature fields are calculated from the distribution functions.

2. Wall: The initial macroscopic properties of the wall are chosen by the user. With these properties, the distribution functions f_i and g_i are created.
3. Combined: after initializing both distribution functions, the conjugate boundary condition is applied, before the first collision even takes place.

LBM time iterations

1. Collision in PCM domain

- Obtain the sensible enthalpy from the total (3.9)
- Construct γ_k^- (3.12)
- Extract the sensible enthalpy ($h_s = \gamma_0^-$)
- Update γ^- to γ^+ (3.13)
- Transform γ_k^+ back to post-collision sensible enthalpy distribution m_i^s (3.11)
- Return the latent heat part to m_i^s to obtain the total enthalpy distribution function m_i

2. Propagation of m_i

3. Collision in the wall domain

- Construct β_k^- (3.6)
- Extract T ($T = \beta_0^-$)
- Update β^- to β^+ (3.8)
- Transform β_k^+ back to post-collision temperature distribution (3.5)

4. Propagation of g_i

5. **Update macroscopic properties** total enthalpy H (2.23), liquid fraction f_L (2.3), c_p and λ (2.8)
6. **Boundary conditions non-shared** performed on non-conjugate boundaries (Neumann BCs, 3.2.1).
7. **Boundary conditions conjugate** performed on the conjugate boundary between the PCM and wall domains (3.2.2).
8. **Update macroscopic properties** total enthalpy H (2.23), liquid fraction f_L (2.3), c_p and λ (2.8)
9. **Return to first step/stop.** Return to the first step to do another iteration. Stop if the maximum number of time steps is reached (given by the user) or steady-state is reached. Steady-state is reached if some quantity ϕ has met the convergence criterion $\delta_\phi[2]$:

$$\sqrt{\frac{\sum_i (\phi(\mathbf{x}_i, t) - \phi(\mathbf{x}_i, t - 1000\Delta t))^2}{\sum_i (\phi(\mathbf{x}_i, t - 1000\Delta t))^2}} < \delta_\phi \quad (3.27)$$

3.4. Numerical Assessment Methods

To properly answer the research questions, two cases are studied and used to verify the model. Both cases are in the context of the freeze-plug. The first case considers a square cavity filled with initially molten salt, bordering an initially cold wall that simulates the pipe adjacent to the freeze-plug. In the second case, the new boundary condition scheme is compared to a known computational benchmark from Pater & Kaaks [1]. This configuration adds more complexity to the first case, by also simulating decay heat from the reactor. The exact configurations and used macroscopic values are given in section 3.5. In this section, the methods used to verify the test cases are discussed.

Error Calculation When an analytical solution or a reliable computational benchmark is available, as in 3.5.2, the error between the simulation and the known solution can be calculated. In this study, the L_2 -error is used to properly compare the results with the research of Borstlap[2], who used the same error method. The L_2 -error norm is given by:

$$\epsilon_\phi = \sqrt{\frac{\sum_i (\phi_i - \phi_{i,ref})^2}{\sum_i (\phi_{i,ref})^2}} \quad (3.28)$$

Grid Convergence and Order of Accuracy By approximating partial differential equations with numerical methods, a discretization error is introduced. This is generally the biggest computational error that results from the LBM [2, 11]. By increasing the number of points in the grid and keeping the physical length of the model the same, the partial differential equations of chapter 2 should be solved more accurately and consequently the discretization error should decrease. Physical quantities of the computational solution of a system should thus converge asymptotically to their analytical values by increasing the grid size.

The LBM is a second order solver of the Navier-Stokes equations. This means that the error between physical reality and the model should decrease with $\sim \frac{1}{\Delta x^2}$ as the number of grid points increases. The error between a modelled quantity ϕ_i and some reference solution ϕ_0 is given by[11]:

$$\epsilon = \phi_i - \phi_0 \approx \alpha \Delta x_i^p + \mathcal{O}(\Delta x_i^{p+1}) \quad (3.29)$$

With $\Delta x = \frac{1}{N}$ the distance between nodes in the grid, p the order of accuracy and α a constant. One can observe that, as long as Δx is sufficiently small, the first term dominates the error. After omitting the higher order term and taking the logarithm on both sides, one ends up with:

$$\log(\epsilon) = p \log(\Delta x_i) + \log(K) \quad (3.30)$$

By plotting $\log(\epsilon)$ against $p \log(\Delta x)$, the order of accuracy can be obtained if a linear relation is found. A remark should be made on which reference quantity is chosen. If there is an analytical solution available, the choice is easily made. If there is no analytical solution available, a computational benchmark is a reasonable replacement. In the case of no analytical solution or computational benchmark, another option is to use a finest-grid solution. The reference solution is then approximated by the solution that is acquired from simulating with a suitable fine grid. To make this work, the grid refinement ratio $r = \Delta x_{fine-1} / \Delta x_{fine}$ should not be too close to zero; the fine grid needs to be sufficiently finer than the grids to which it is compared[2].

Energy Conservation A vital feature of computational boundary conditions, is that they should conserve the energy of the system (as is important in all parts of physics). In a closed system, the energies of the PCM domain and the wall domain should always sum to the same value. The error in the total energy can be quantified as:

$$\epsilon_{E_{total}} = \frac{(E_{PCM} + E_{wall}) - E_{total}(t = 0)}{E_{total}(t = 0)} \cdot 100\% \quad (3.31)$$

Where the reference quantity is chosen as $E_{total}(t = 0)$ in equation 3.28, since taking 0 would naturally prove difficult in calculating the error.

3.5. Assessed Cases

3.5.1. Case 1: Fluid Hot Salt Cavity with Adjacent Cold Pipe

As briefly mentioned above, the first tested case is a square cavity filled with molten salt at an initial temperature of $T_0 = 842K$. Next to it is an initially cold pipe at temperature $T_0 = 831K$. See figure 3.5 for a schematic view of the configuration. Both temperatures are chosen near the values of the benchmark tested in case 2, to best simulate the freeze-plug. All boundaries are adiabatic, except for the conjugated border between the PCM and wall domains. The PCM domain is handled with enthalpy distributions, the wall domain with temperature distributions. Adiabatic boundary conditions are handled with the WNSV boundary scheme. The conjugated boundary is imposed with the liquid fraction incorporated WNSV boundary scheme. The different extrapolation schemes tested are given in table 3.1. *BCold* should be highlighted. It does not use the liquid fraction incorporated T_{BC} (equation 3.24), but it does test the new algorithm for determining the liquid fraction at the boundary.

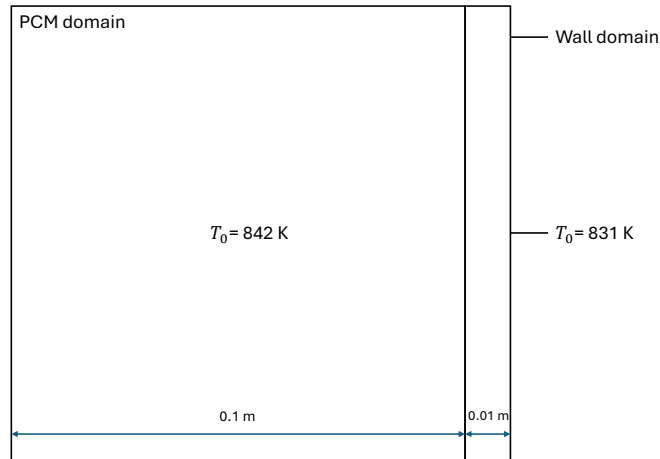


Figure 3.5: Case 1 configuration: a square cavity filled with hot salt next to a cold wall. T_0 indicates initial temperatures. All walls are adiabatic.

Table 3.1: Different WNSV BC schemes tested.

Name method	Extrapolation Method SV Conjugate Boundary	Neumann	Conjugate
BC1t	1 st order spatial & temporal (3.16a)	1 st order (3.18a)	3.24a
BC2	2 nd order spatial (3.16b)	2 nd order (3.18b)	3.24b
BC2t	2 nd order spatial & 1 st temporal (3.16c)	2 nd order (3.18b)	3.24b
BCold	2 nd order spatial & 1 st temporal (3.16c)	2 nd order (3.18b)	3.19b

The macroscopic properties used are given in table 3.2. Properties of the salt in solid form are given with subscript s , properties of the fluid form are given with subscript l . For the metal wall, Hastelloy-N is chosen. The salt is assumed to be a pure material, but a computational difference between T_s and T_l is introduced to properly incorporate the liquid fractions: $T_l = T_s + 0.002K$ [1]. There is no heat source present, so the source term Q in the first element of equation 3.13 is left out.

Domain	Property	Physical Value	Unit	Lattice Value	Lattice Unit
Salt	$\rho_s = \rho_l$	4390	kg/m^3	1.0	ls^{-3}
	$\lambda_s = \lambda_l$	1.5	$W/(m \cdot K)$	$1.5 / C_\lambda$	$ls/(lt^3 \cdot lK)$
	$c_{p,s}$	815	$J/(kgK)$	$815/C_{c_p}$	$ls^2/(lt^2 \cdot lK)$
	$c_{p,l}$	1000	$J/(kgK)$	$1000/C_{c_p}$	$ls^2/(lt^2 \cdot lK)$
	T_s	841	K	841	lK
	L	$159 \cdot 10^3$	J/kg	$159 \cdot 10^3/C_{lat}$	ls^2/lt^2
	ν	$7.5 \cdot 10^{-4}/\rho$	m^2/s	1/40	ls^2/lt
Metal	ρ	8860	kg/m^3	$8860/C_\rho$	ls^{-3}
	λ	23.6	$W/(m \cdot K)$	$23.6 / C_\lambda$	$ls/(lt^3 \cdot lK)$
	c_p	578	$J/(kgK)$	$578/C_{c_p}$	$ls^2/(lt^2 \cdot lK)$

Table 3.2: Used macroscopic properties in case 1. All are chosen the same as in [1].

3.5.2. Case 2: Benchmark Study

In the second case, the model is compared to a benchmark study by Pater & Kaaks[1]. In this study, the melting of the freeze-plug was simulated using 3 different codes; StarCCM+, OpenFoam and DGFlows. The configuration is given in figure 3.6. The same macroscopic properties as in case 1 are used and can

be found in table 3.2. The *BC2* and *BC2t* methods are used for extrapolation and boundary conditions, see table 3.1.

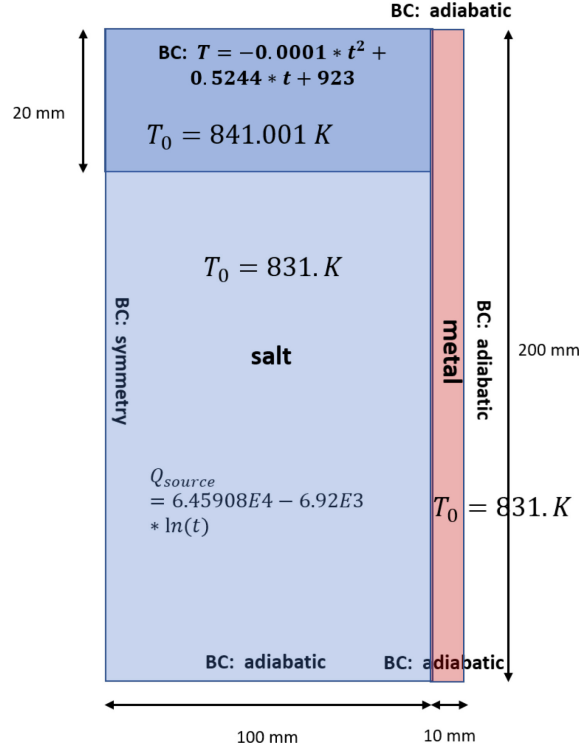


Figure 3.6: Benchmark case configuration. T_0 indicates initial temperatures. All BCs can be seen in the figure, BC: symmetry is also taken as adiabatic. The light blue color represents initial solid salt, dark blue represents initial molten salt [1].

Note that the configuration now has a rectangular shape. Initially, the top 10% of the salt is fluid, while the bottom is solid. Gravity is set to zero to prevent natural convection from occurring. Melting thus occurs only as a result of heat conduction, both in the PCM and at the conjugate border between the domains. The melting of the plug is driven by two sources. First, there is a time-dependent boundary condition at the upper boundary which simulates the heating of the molten salt in the reactor above the freeze-plug. This boundary condition is given by:

$$T(t) = -0.0001t^2 + 0.5244t + 923 \quad [K] \quad (3.32)$$

A second heat source is added by considering the radiated heat from the reactor. This heat source term, which is applied on the entire PCM domain (both solid and fluid), is approximated to be 1% of the decay heat of the reactor and is given by:

$$Q(t) = 6.45908 \cdot 10^4 - 6.92 \cdot 10^3 \cdot \ln(t) \quad [Wm^{-3}] \quad (3.33)$$

Results and Discussion

In this chapter, the results of the cases discussed in section 3.5 are presented. The first case is evaluated in section 4.1. In this section, grid convergence and energy conservation will be checked and the order of accuracy will be determined for the various extrapolation orders used. The second assessed case is presented in section 4.2. Here, the results will be compared to the benchmark from Pater & Kaaks [1] and Borstlap [2].

4.1. Case 1: Hot Salt Cavity

To verify grid convergence behaviour of the liquid-fraction-incorporated boundary condition scheme, both the temperature fields in the PCM and wall domain were analyzed. In the PCM temperature field, a square grid of 81 points was selected with x- and y-coordinates both ranging from 0.01 m to 0.09 m. The average temperature of these points was calculated at time 100 s and plotted in figure 4.1 for three different combinations of extrapolation methods from table 3.1.

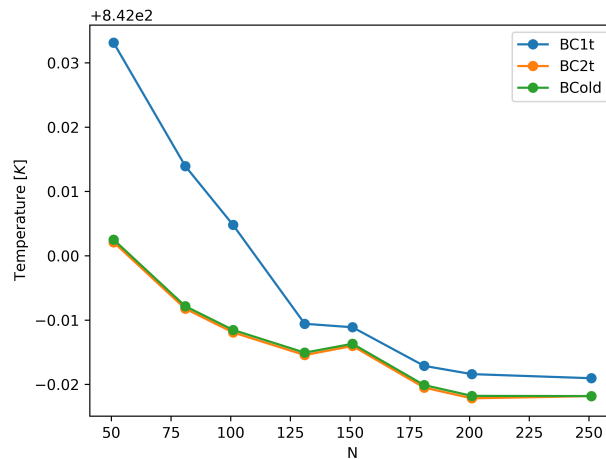


Figure 4.1: Grid convergence results for different extrapolation methods. Note that the temperature axis displays differences around 842 K.

It can be seen that the average temperatures of the *BC1t*, *BC2t* and *BCold* methods converge for an increasing grid size *N*. *BC2t* and *BCold* coincide almost completely. The temperature remains around the starting temperature of 842 K. The overshoot of ~ 0.03 K for *BC1t* and ~ 0.003 K for *BC2t* and *BCold* could result from the numerical heat diffusion related to phase change near the conjugate border, as was also found by Borstlap [2].

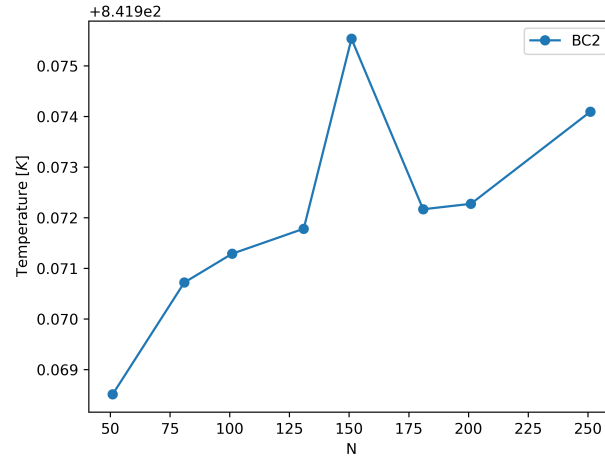


Figure 4.2: Grid convergence results for extrapolation method *BC2*. Note that the temperature axis displays differences around 841.9 K.

In figure 4.2 the results for the *BC2* method were plotted. It shows somewhat converging behaviour for increasing grid size, but the increases notably for grid sizes 151 and 251. The total change in temperature is only around $7 \cdot 10^{-4}\%$ when comparing the largest grid size to the smallest. This makes it hard to say whether the temperature field converges.

In the wall domain, a grid of points was chosen with x-coordinates ranging from 0.001 m to 0.009 m and y-coordinates ranging from 0.01 m to 0.09 m. Again, for three extrapolation methods, the average temperature found at these points was calculated at time 100 s and plotted in figure 4.3.

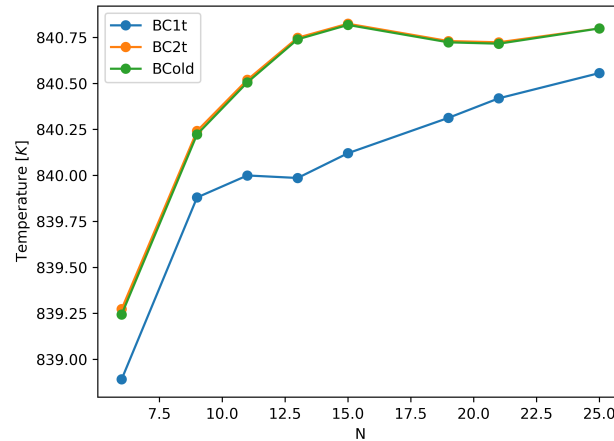


Figure 4.3: Grid convergence results for different extrapolation methods.

As in the PCM domain, converging behaviour can be seen for the *BC1t*, *BC2t* and *BCold* methods. However, it should be said that the grid size of the wall domain is relatively small compared to the PCM domain. This results from the fact that the width of the wall domain is only 10% of the width of the PCM domain.

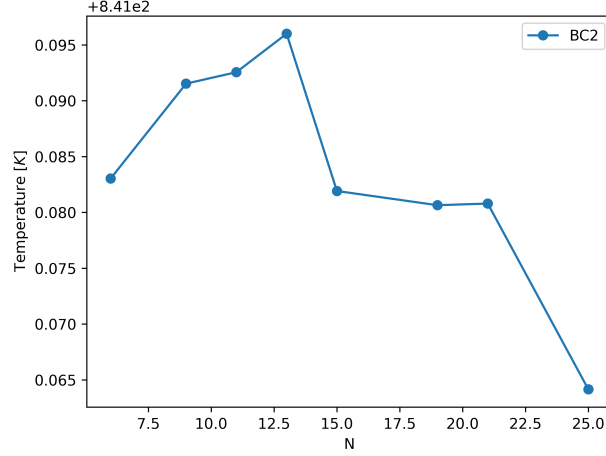


Figure 4.4: Grid convergence results for extrapolation method BC2.

In figure 4.4, the results for the BC2 method are shown. No converging behaviour can be seen, though again it should be said that the grid size in the wall is small and grid convergence behaviour at larger grid sizes can not be ruled out entirely. Considering all the above, the *BC1t*, *BC2t* and *BCold* methods show converging behaviour, while for the *BC2* this remains difficult to conclude.

The next step is to determine the order of accuracy of all methods. The procedure to do this is described in section 3.4. Since there is no analytical solution available for this case, a fine grid solution is taken as reference; $N_{fine} = 251$. The error is then defined as $\epsilon_T = |T_{N_{fine}} - T_N|$, see equation 3.29. Plotting the results of this error function leads to figures 4.5 and 4.6.

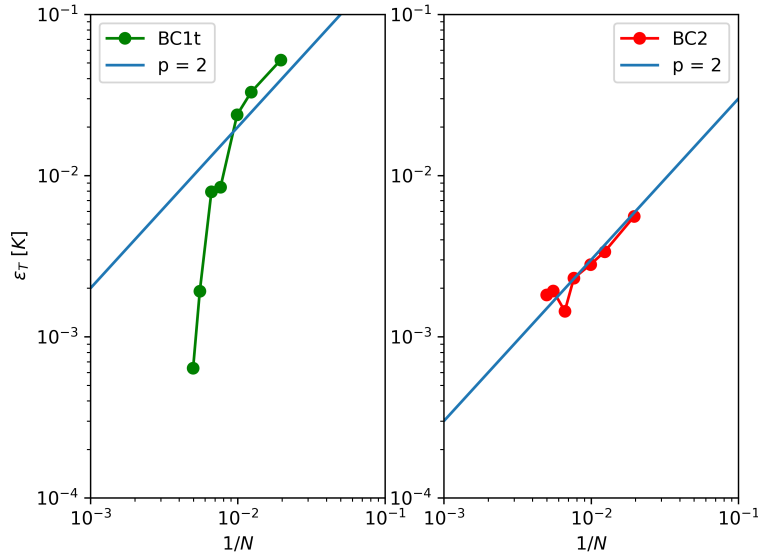


Figure 4.5: Order of accuracy analysis for the *BC1t* (left) and *BC2* (right) extrapolation methods, together with the theoretical $p = 2$ line. Finer grid sizes are on the left side of the figures, coarser grid sizes on the right side.

In figure 4.5, the results for the *BC1t* and *BC2* methods are shown. The errors are higher than those found for *BC2t* and *BCold*, which is likely the result of the fact that *BC1t* and *BC2* are lower order

extrapolation methods than *BC2t* and *BCold*. The theoretical $p = 2$ line from equation 3.30, the second-order accurate line, is shown as reference in both figures. *BC1t* follows this line for the first 3 grid sizes, $N = 51, 81$ and 101 . After that, the slope becomes rapidly steeper, especially for $N = 181$ and $N = 201$. This could be because the reference grid size of $N = 251$ is not 'fine' enough to compare to. *BC2* shows good correspondence with the reference slope line, though it must be noted that a relatively high temperature was found for $N = 251$ (figure 4.2) which could tamper with the order of accuracy test.

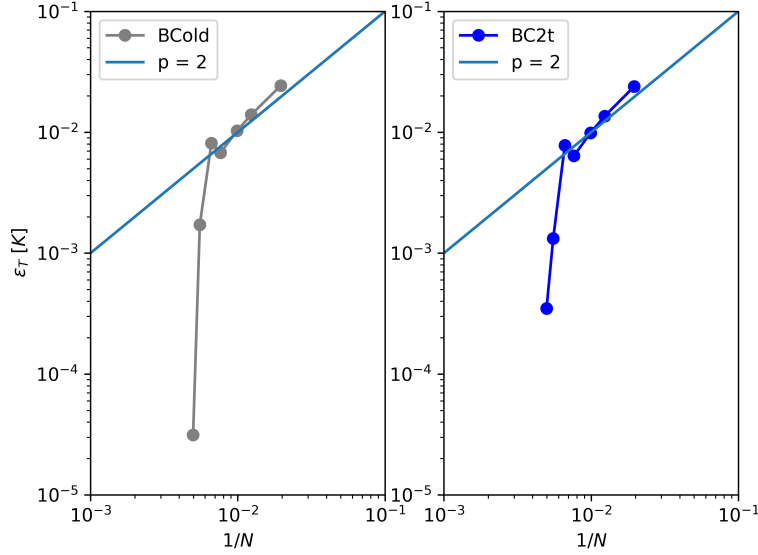


Figure 4.6: Order of accuracy analysis for the *BCold* (left) and *BC2t* (right) extrapolation methods, together with the theoretical $p = 2$ line. Finer grid sizes are on the left side of the figures, coarser grid sizes on the right side.

In figure 4.6, the results for the *BCold* and *BC2t* methods are shown. Similar to *BC1t*, these methods correspond well with the theoretical slope line even up to a grid size of $N = 151$, but clearly diverge for grid sizes 181 and 201. Again, this can probably be attributed to the 'fine' solution not being exact enough. All in all, it can be concluded that all the methods show the theoretical second-order solver features for smaller grid sizes, though the *BC2* remains uncertain in grid convergence and order of accuracy.

The last feature of the methods that will be checked is energy conservation. As mentioned in 3.4, an important part of a boundary condition is that it should conserve energy. To measure the loss of energy from the system, equation 3.31 was formulated. The results of this equation with grid size $N = 251$ are plotted in figures 4.7 and 4.8.

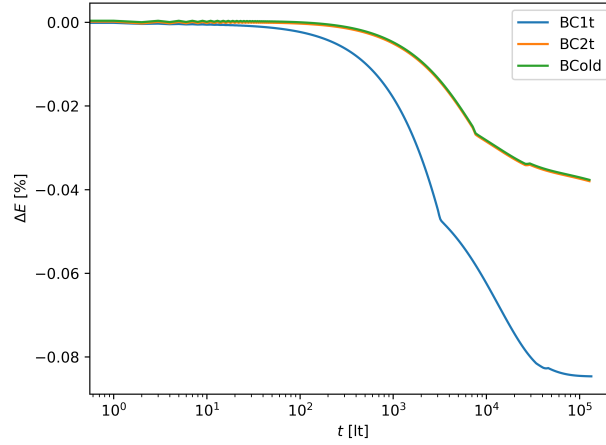


Figure 4.7: Measure of the energy loss in a percentage of the initial energy. Measured with boundary methods $BC1t$, $BC2t$ and $BCold$. Note that the time axis is logarithmically scaled.

In figure 4.7, the results for methods $BC1t$, $BC2t$ and $BCold$ are shown. Energy loss remains limited to 0.08 % for the $BC1t$ method and to 0.04 % for the $BC2t$ and $BCold$ methods. The energy loss is the largest between $lt = 10^3$ and $lt = 10^4$. It could be that in this time interval, phase change occurs and numerical heat loss takes place. Time convergence according to equation 3.27 was measured shortly after $lt = 10^5$.

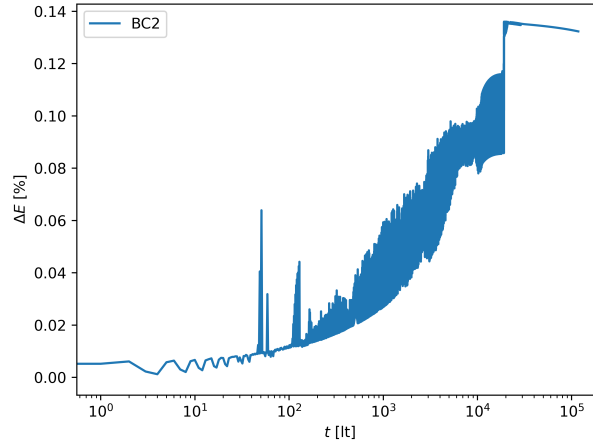


Figure 4.8: Measure of the energy loss in a percentage of the initial energy. Measured with boundary method $BC2$. Note that the time axis is logarithmically scaled.

The total energy error in the $BC2$ method (figure 4.8) is somewhat higher than the error of the other methods. The main energy loss occurs in the same lattice time regime, namely $10^3 - 10^4 lt$. It does not happen smoothly as in figure 4.7, but with oscillations. A remarkable feature of the energy conservation analysis of $BC2$ is the fact that energy is gained during the simulation, which can be concluded from the positive $\Delta E\%$. Whether this effect might also result from phase change, inaccurate extrapolation or something else should be studied further, for example by tracking the phase of the system in the first 10^4 time steps. In conclusion, no method shows full energy conservation, but methods $BC1t$, $BC2t$ and $BCold$ lose only a slight fraction of energy. The $BC2$ method actually gains energy during the simulation, which should be studied further in order to fully understand this effect.

4.2. Case 2: Benchmark: Melting of the Freeze-Plug

4.2.1. Intermediate configuration

Before turning to the results of the full configuration as described in section 3.5.2, an intermediate configuration is analyzed. This configuration is a mix of case 1 and case 2 from section 3.5. It has the same geometric size as the benchmark case (figure 3.6), but is initially filled completely with fluid salt. The time dependent temperature boundary at the top is also included. The decay heat source $Q(t)$ is omitted. The initial temperatures of both the PCM and wall domain are the same as in case 1 (842 K for the PCM domain, 831 K for the wall domain). The used grid size is $NX \times NY = 101 \times 201$. This configuration is considered, since it showed non-physical results near the phase front with the model used by Borstlap [2]. Results are shown for $It = 30000$, in figure 4.9.

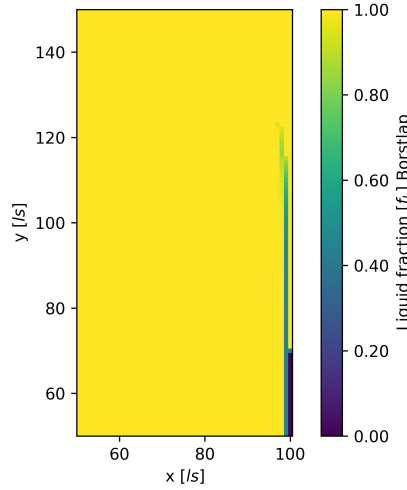
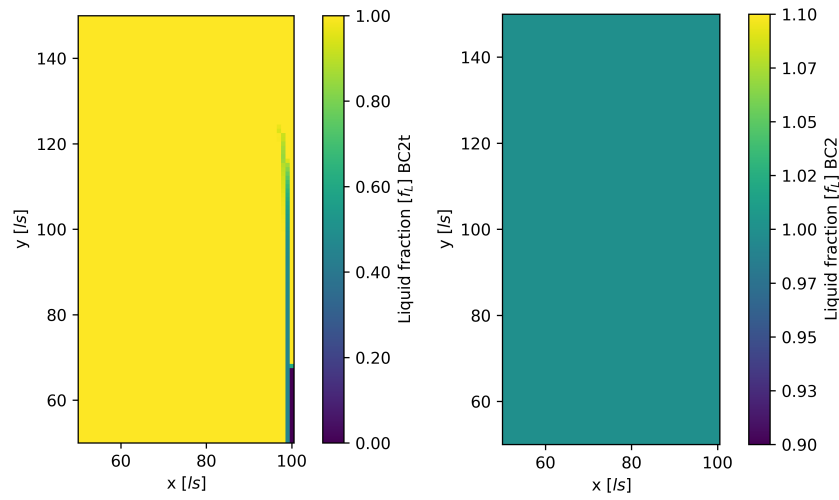
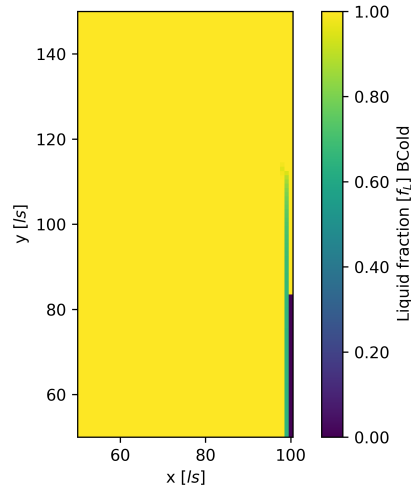


Figure 4.9: Zoomed in view of the liquid fraction field at $It = 30000$, using the model of Borstlap [2].

During the simulation of this configuration, the ‘wobble’ near the conjugate boundary emerged in figure 4.9 at lattice times between 10000 and 80000. It is non-physical; any solid salt formed at the boundary in the beginning of the simulation, should melt uniformly from top to bottom. In the simulation, this process was handled well at the boundary nodes, but not at the node directly to the left of the boundary node. When adding fluid flow to simulations, this could lead to instabilities. The liquid fraction is used in multiple methods to suppress velocities in the mushy and solid zones of phase change material [2]. If the values near the conjugate border are inaccurate, velocities can not be correctly adjusted, and non-physical results are more likely to be found. To see whether the liquid-fraction-incorporated boundary scheme can simulate this freezing from the wall domain more accurately, the same simulation is run with the *BC2t*, *BC2* and *BCold* methods. Results are shown in figure 4.10a, figure 4.10b and figure 4.10c. The *BC2t* field is the same as in figure 4.9. The *BC2* field is uniform, the freezing and melting process has already passed. This could be a better physical representation of reality, which will be tested in the following section. The *BCold* method still has the ‘wobble’ as in the model from Borstlap, but it is shorter and less wide at the top. As *BCold* is a combination of the new extrapolation algorithm for determining the liquid fraction and sensible enthalpy at the boundary and a T_{BC} for which this liquid fraction is not used, this could indicate that the new algorithm is more accurate than the one used by Borstlap.



(a) Zoomed in view of the liquid fraction field at $t = 30000$, using the $BC2t$ extrapolation method. (b) Zoomed in view of the liquid fraction field at $t = 30000$, using the $BC2$ extrapolation method.



(c) Zoomed in view of the liquid fraction field at $t = 30000$, using the $BCold$ extrapolation method.

4.2.2. Full configuration

Using the full configuration of case 2, the melting of the freeze-plug without natural convection was simulated using the $BC2t$ and $BC2$ boundary methods (see table 3.1). These two methods were chosen since $BC2t$ is a higher order method than $BC1t$ and most of its results were approximately the same as for the $BCold$ method. As this simulation is a transient case, an imposed (non-lattice) time of 2500 [s] was simulated with a grid size of $NX \times NY = 201 \times 401$. Results are shown in figures 4.11 and 4.12, with the isotherms on the left and the liquid fraction field on the right.

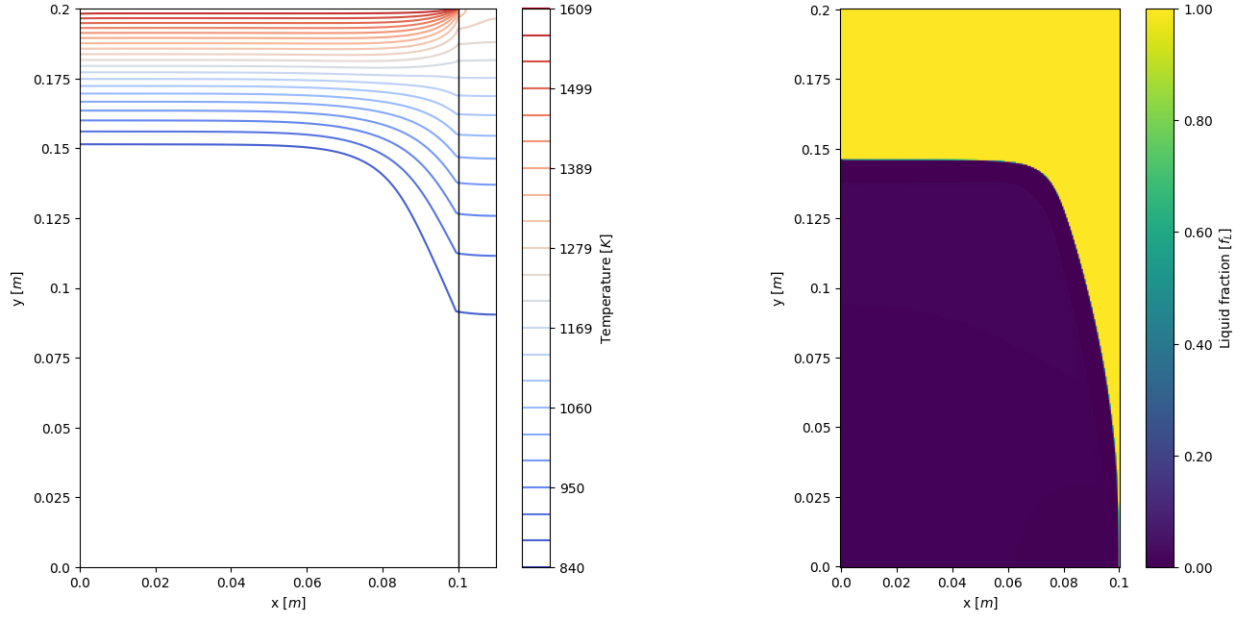


Figure 4.11: Results of the melting simulation of the freeze-plug at $t = 2500$ [s] with the $BC2t$ method. The used grid size is $N_x \times N_y = 201 \times 401$. On the left, the isotherms in the PCM and wall domain are depicted. On the right the liquid fraction profile in the PCM domain.

At first sight, the results of the $BC2t$ (figure 4.11) method are physically intuitive. The melting process at the conjugate boundary is enhanced by conjugate heat transfer, since the metal wall conducts heat faster than the salt does ($\lambda_{metal} > \lambda_{salt}$). Furthermore, the entire phase front is lowered as a result of the time-dependent boundary condition at the upper boundary. The results of the $BC2$ method do not show physically correct behaviour. The isotherms do not stretch into the wall domain, which could indicate that heat does not pass through the conjugate boundary. This could be the result of the extrapolation not being accurate enough to use in the liquid-fraction incorporated algorithm, as the method did show good results in the study by Borstlap. As the metal in the wall is not heating up, no melting takes place at the conjugate boundary.

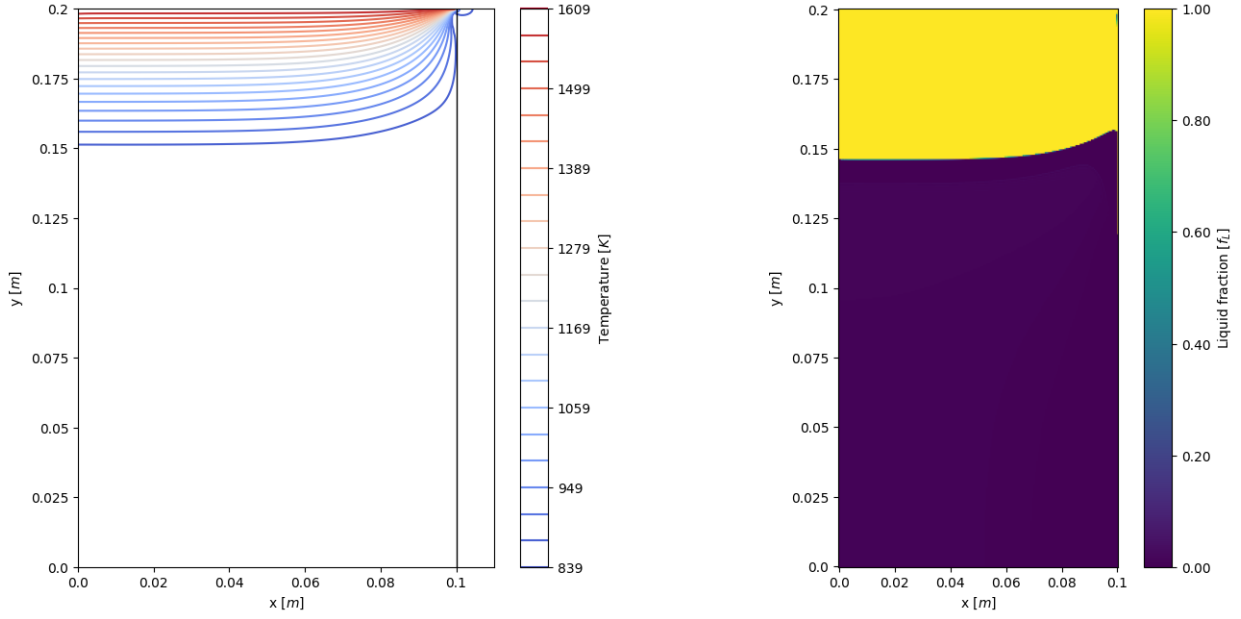


Figure 4.12: Results of the melting simulation of the freeze-plug at $t = 2500$ [s] with the $BC2$ method. The used grid size is $N_x \times N_y = 201 \times 401$. On the left, the isotherms in the PCM and wall domain are depicted. On the right the liquid fraction profile in the PCM domain.

In figure 4.13, the phase fronts of both methods are compared with one benchmark code of Pater & Kaaks [1] and the results of Borstlap [2]. The $BC2t$ model coincides remarkably well with both benchmark codes. As expected, the $BC2$ model does not show correspondence with the benchmark cases near the conjugate boundary.

The L_2 -error norms for the phase fronts are displayed in table 4.1.

	$T(x = 0mm)$	$T(x = 75mm)$	$T(x = 90mm)$	Phase Front
StarCCM+	0.13%	0.26%	0.36%	0.88%
Borstlap	$6.4 \cdot 10^{-6} \%$	0.006 %	0.017%	0.025 %

Table 4.1: L_2 -error norms between the $BC2t$ model and the benchmark models.

$BC2$ was left out of the error calculations, as it is qualitatively very far removed from the benchmark models. From the error results, it appears that the $BC2t$ performs well compared with both benchmark models, as errors are kept under 1%.

Three temperature probes are presented in figure 4.14; at the left boundary ($x = 0mm$), at ($x = 75mm$) and at ($x = 90mm$). At all three probes, the $BC2t$ model shows good correspondence. This can be checked numerically by calculating the L_2 -error norm (3.28). The values of these errors are also given in table 4.1. The model shows no errors larger than 0.88% to any of the benchmark cases. From this, it can be said that the $BC2t$ model can be used to further investigate how natural convection can be added to the model in future research. Furthermore, it could be very valuable to analyze the $BCold$ method, since it showed better results in section 4.2.1.

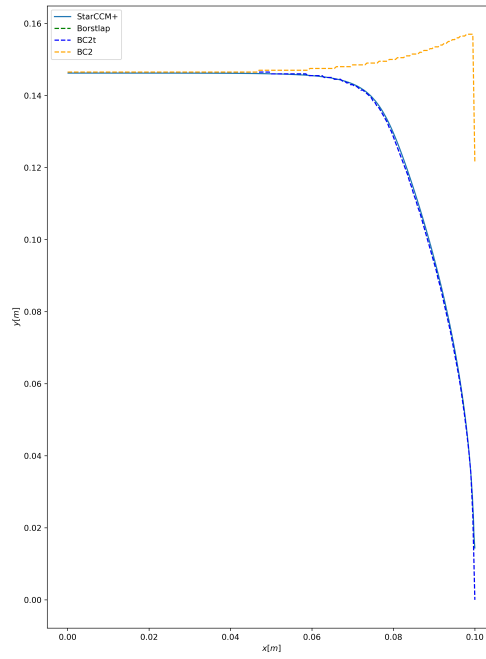


Figure 4.13: Modelled phase front position at $t = 2500$ [s]. The used grid size is 201×401 . Compared to the StarCCM+ results from [1] and the results from Borstlap [2].

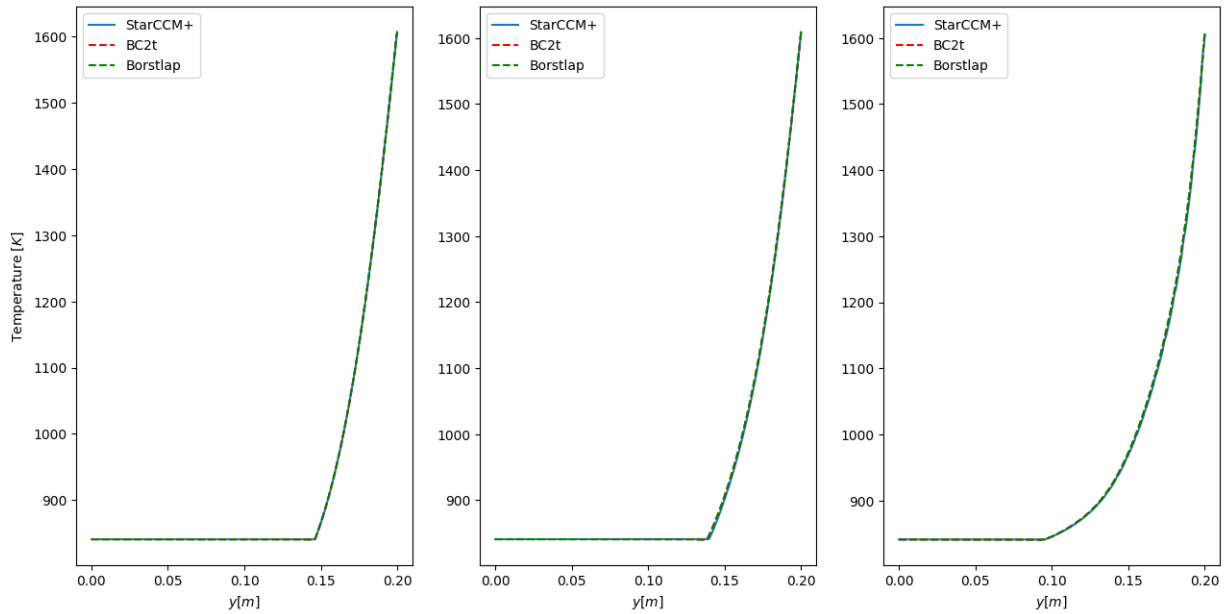


Figure 4.14: Temperature profiles at three different x -probes. The used grid size is 201×401 . Compared to the StarCCM+ results from [1] and results from Borstlap [2].

Conclusion and Recommendations

In the search for a stable and clean energy source for the future, nuclear reactors can play a key role. One of the new generation reactors is the MSFR, which eliminates major disadvantages of conventional nuclear energy like long term radioactive waste and explosion risks. An important built-in safety feature of the MSFR is the freeze-plug. In case of emergency, this plug should melt in the right time to prevent damage to the reactor. Hence, a thorough understanding of this process is vital. Conjugate heat transfer plays an important role in the melting process of the freeze-plug. In earlier research, conjugate heat transfer was simulated successfully using the Filter-Matrix Lattice Boltzmann Method (FM-LBM), but it turned out to be incompatible with the implementation of natural convection, another key element of the melting process. When fluid velocities are considered in the FM-LBM, the liquid fraction plays an important role in imposing no-slip boundary conditions near the conjugate border. Therefore, accurate information about the liquid fraction near conjugate boundaries is necessary. This lead to the formulation of two research questions. First of all, it was asked how the liquid fraction near a conjugate boundary can be taken into account in the algorithm that imposes a conjugate boundary condition. Subsequently, it was asked how this newly developed boundary condition scheme performed with respect to know numerical benchmarks. To answer these questions, a new liquid-fraction-incorporated conjugate boundary algorithm that keeps track of the liquid fraction was proposed and implemented into the wet-node solution vector boundary condition scheme. After that, several properties of the scheme were tested with different extrapolation methods. Finally, it was compared to two benchmark cases of Pater & Kaaks [1] and Borstlap [2] respectively. This was done using both the temperature and the total enthalpic approach of the Filter-Matrix Lattice Boltzmann Method.

5.1. Conclusions

The new algorithm for conjugate boundary conditions in the wet-node solution vector boundary condition scheme uses the total enthalpy of the nodes next to the boundary in the PCM domain. The total enthalpy distributions of these nodes are extrapolated to the boundary node, after which an extrapolated liquid fraction f_L is determined at the boundary. This value is used to determine whether the phase front is near the conjugate boundary and to determine the temperature for the conjugate boundary condition.

After formulating the new algorithm, two cases were tested to verify the new scheme. In the first one, a square cavity filled with molten salt adjacent to a cold pipe was tested. In the cavity, the enthalpic FM-LBM was used, while in the pipe a temperature solution vector kept track of the temperature. Grid convergence was tested for 4 different boundary condition extrapolation methods by taking a number of temperature points at a certain time. The first order spatial and temporal extrapolation method ($BC1t$), second order spatial and first order temporal extrapolation method ($BC2t$) and an extrapolation method combining the new algorithm and a conjugate temperature boundary condition in which the liquid fraction was not taken into account ($BCold$) showed grid convergence, while a second order spatial extrapolation method ($BC2$) yielded ambiguous results with respect to grid convergence. The first

three methods showed approximate second-order accuracy for small grid sizes as did the second order spatial method, though again dubious. More research to the order of accuracies would be desirable to get definite conclusions. Additionally, a larger adjacent wall grid could be implemented into the configuration to check more trustfully for grid convergence in both domains. Lastly, energy conservation was checked for the different schemes. Though not fully conserved, the loss of energy from the system for the different methods remains around 0.08% for the *BC1t* method and 0.04% for the *BC2t* and *BCold* methods. The *BC2* method actually showed an increasing energy, with a total increase of around 0.14% of the total initial energy.

In the second case, an intermediate configuration was studied first, which was less complex than the benchmark cases. The model of Borstlap showed non-physical results near the conjugate boundary when freezing and subsequent melting took place. The *BC2t*, *BCold* and *BC2* were tested to see if they showed a better handling of this case. The *BC2t* method acted similar to the model of Borstlap, while the *BCold* performed somewhat better and the *BC2* returned a total fluid field. After this intermediate test, the *BC2t* and *BC2* methods were used in an attempt to model the results from benchmark cases by Pater & Kaaks [1] and Borstlap [2]. The *BC2t* method showed good physical results and small errors at three chosen temperature probes (<0.5%) and the overall phase front (<1%). The *BC2* method showed grave non-physical results and seemed to not conduct any heat through the conjugate boundary. This is likely because it is not accurate enough to use in the newly developed algorithm. The performance of the *BCold* method was not compared to the benchmark cases. Doing this in the future could be valuable as the method did perform somewhat better in the intermediate case.

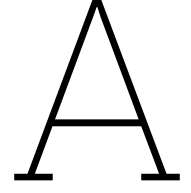
5.2. Recommendations

The research carried out in this thesis could open up some questions for future research. The new scheme itself could be more thoroughly tested, for instance by simulating a situation for which an analytical solution is available to better determine the order of accuracy. If this solution is not available, simulations with larger grid sizes could also suffice. Possibly the biggest question that remains is whether the new boundary scheme can help overcome the instabilities that arise when simulating natural convection near a conjugate boundary that combines both enthalpy and temperature solution vectors. If the implementation of natural convection into the model succeeds, simulations in which design parameters of the freeze-plug are altered could be done to check which configuration is the most efficient and melts the quickest.

References

- [1] Mateusz Pater et al. "A numerical benchmark for modelling phase change in molten salt reactors". In: *Annals of Nuclear Energy* 194 (2023), p. 110093. ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2023.110093>.
- [2] L. Borstlap. "Developing a Filter-Matrix Lattice Boltzmann Boundary Method for Conjugate Heat Transfer and Phase Change". MA thesis. Delft University of Technology, 2024.
- [3] United Nations. *Paris Agreement*. https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=en. Accessed: 14-1-2026.
- [4] A. Ho et al. "Exploring the benefits of molten salt reactors: An analysis of flexibility and safety features using dynamic simulation". In: *Digital Chemical Engineering* 7 (2023), p. 100091. ISSN: 2772-5081. DOI: <https://doi.org/10.1016/j.dche.2023.100091>.
- [5] Charles McCombie and Michael Jefferson. "Renewable and nuclear electricity: Comparison of environmental impacts". In: *Energy Policy* 96 (2016), pp. 758–769. ISSN: 0301-4215. DOI: <https://doi.org/10.1016/j.enpol.2016.03.022>.
- [6] Generation IV International Forum and U.S. DOE Nuclear Energy Research Advisory Committee. *A Technology Roadmap for Generation IV Nuclear Energy Systems*. 2002.
- [7] M. Allibert et al. "7 - Molten salt fast reactors". In: *Handbook of Generation IV Nuclear Reactors*. Ed. by Igor L. Pioro. Woodhead Publishing Series in Energy. Woodhead Publishing, 2016, pp. 157–188. ISBN: 978-0-08-100149-3. DOI: <https://doi.org/10.1016/B978-0-08-100149-3.00007-0>.
- [8] Jérôme Serp et al. "The molten salt reactor (MSR) in generation IV: Overview and perspectives". In: *Progress in Nuclear Energy* 77 (2014), pp. 308–319. ISSN: 0149-1970. DOI: <https://doi.org/10.1016/j.pnucene.2014.02.014>.
- [9] Marco Tiberga et al. "Preliminary investigation on the melting behavior of a freeze-valve for the Molten Salt Fast Reactor". In: *Annals of Nuclear Energy* 132 (2019), pp. 544–554. ISSN: 0306-4549. DOI: <https://doi.org/10.1016/j.anucene.2019.06.039>.
- [10] Indarta Kuncoro Aji. "Investigation of Basic Parameters in Developing High-Performance Freeze Valve for Molten Salt Reactor". PhD thesis., 2020. DOI: [10.18952/00009626](https://doi.org/10.18952/00009626).
- [11] T. Krüger et al. *The Lattice Boltzmann Method: Principles and Practice*. Springer International Publishing AG Switzerland, 2017.
- [12] W.-S. Jiaung, J.-R. Ho, and C.-P. Kuo. "Lattice Boltzmann method for the heat conduction problem with phase change". In: *Numerical Heat Transfer: Part B: Fundamentals* 39.2 (2001), pp. 167–187.
- [13] Guigao Le, Othmane Oulaid, and Junfeng Zhang. "Counter-Extrapolation Method for Conjugate Interfaces in Computational Heat and Mass Transfer". In: *Physical Review E* 91 (Mar. 2015), p. 033306. DOI: [10.1103/PhysRevE.91.033306](https://doi.org/10.1103/PhysRevE.91.033306).
- [14] J.G.M. Eggels and J.A. Somers. "Numerical simulation of free convective flow using the lattice-Boltzmann scheme". In: *International Journal of Heat and Fluid Flow* 16.5 (1995), pp. 357–364. ISSN: 0142-727X. DOI: [https://doi.org/10.1016/0142-727X\(95\)00052-R](https://doi.org/10.1016/0142-727X(95)00052-R).
- [15] Congshan Zhuo, Chengwen Zhong, and Jun Cao. "Filter-matrix lattice Boltzmann model for incompressible thermal flows". In: *Phys. Rev. E* 85 (4 Apr. 2012), p. 046703. DOI: [10.1103/PhysRevE.85.046703](https://doi.org/10.1103/PhysRevE.85.046703).
- [16] Celeke Bus, Thorben Besseling, and Martin Rohde. "A Filter-Matrix Lattice-Boltzmann Methodology for Convective Melting and Solidification". In: *International Journal for Numerical Methods in Fluids* 97.10 (2025), pp. 1379–1393. DOI: <https://doi.org/10.1002/flid.70001>.
- [17] R. Huang, H. Wu, and P. Cheng. "A new lattice Boltzmann model for solid-liquid phase change". In: *International Journal of Heat and Mass Transfer* 59 (2013), pp. 295–301. DOI: [10.1103/PhysRevE.85.046703](https://doi.org/10.1103/PhysRevE.85.046703).

-
- [18] *Fysische Transportverschijnselen: Denken in Balansen* | TU Delft OPEN Books — doi.org. <https://doi.org/10.5074/t.2023.002>. [Accessed 23-01-2026].



Derivations of f_L Incorporated Boundary Temperatures

In this appendix, the derivation for the liquid fraction incorporated boundary condition temperature is given. This is done for the first order and is then expanded to the second order.

First order

First, the boundary conditions for conjugate heat transfer from section 2.2.2 are restated:

$$T_{f,W} = T_{s,W} \quad (\text{A.1})$$

$$-\lambda_f \frac{\partial T}{\partial n}|_{f,W} = -\lambda_s \frac{\partial T}{\partial n}|_{s,W} \quad (\text{A.2})$$

Where the subscript f denotes the fluid (or phase change material), the subscript s the solid, W the wall, n the normal direction at the wall, T the temperature and λ the thermal conductivity. In order to implement these equations into the FM-LBM, they must be discretized. This can be done by making a Taylor expansion around the wall nodes and using neighbouring nodes to estimate the temperature derivatives at the wall. In this appendix, this is only done for the case where the phase front lies in the wall cell, hence when $H_s < H^{i=0} < H_l$. To arrive at an expansion for the location of the phase front, first the position coordinate of the phase front is determined, under the assumption that the phase front in a cell can be approximated by a vertical line. Its position can be defined as $x_W - \frac{1}{2}\Delta x + f_{L,n}\Delta x$, with $f_{L,n}$ ranging from 0 to $\frac{1}{2}$. See figure A.1.

In order to determine $f_{L,n}$, two possibilities of phase change at the conjugate border should be considered. The first case is the melting of the PCM due to a hot adjacent wall. This situation is displayed in figure A.1.

The first order Taylor-expansion of the temperature at the phase front position around the wall is given by:

$$T_f(x_W - \frac{1}{2}\Delta x + f_{L,n}\Delta x) = T(x_W) + T'_f(x_W)(-\frac{1}{2} + f_{L,n})\Delta x \quad (\text{A.5})$$

An expression for the point to the right of the wall (in the solid domain) can be found in a similar way:

$$T_s(x_W + \Delta x) = T_s(x_W) + T'_s(x_W)\Delta x \quad (\text{A.6})$$

Rewriting equations A.5 and A.6 into expressions for $\frac{\partial T}{\partial x}|_{f/s,W} (= T'(x_{f/s,W}))$ and filling this in equation A.2 leads to:

$$-\lambda_f \frac{T_f(x_W - \frac{1}{2}\Delta x + f_{L,n}\Delta x) - T_f(x_W)}{(-\frac{1}{2} + f_{L,n})\Delta x} = -\lambda_s \frac{T_s(x_W + \Delta x) - T_s(x_W)}{\Delta x} \quad (\text{A.7})$$

Now, using equation A.1 and realizing that the temperature at the phase front is actually T_l , one finally arrives at an expression for the first order temperature condition at the conjugated wall:

$$T_{BC} = \frac{1}{\lambda_l - \lambda_s(f_{L,n} - \frac{1}{2})} [\lambda_l T_l - \lambda_s(f_{L,n} - \frac{1}{2})T(x_W + \Delta x)] \quad (\text{A.8})$$

Second order

To derive the second order expression for T_{BC} , the Taylor-expansion for the phase front is extended to a second order:

$$T_f(x_W - \frac{1}{2}\Delta x + f_{L,n}\Delta x) = T_f(x_W) + T'_f(x_W)(-\frac{1}{2} + f_{L,n})\Delta x + \frac{(-\frac{1}{2}\Delta x + f_{L,n}\Delta x)^2}{2!} T''_f(x_W) \quad (\text{A.9})$$

Expressions for the temperature at $T_f(x - \Delta x)$, $T_s(x + \Delta x)$ and $T_s(x + 2\Delta x)$ can be found in a similar way:

$$T_f(x_W - \Delta x) = T_f(x_W) - T'_f(x_W)\Delta x + \frac{\Delta x^2}{2!} T''_f(x_W) \quad (\text{A.10a})$$

$$T_s(x_W + \Delta x) = T_s(x_W) + T'_s(x_W)\Delta x + \frac{\Delta x^2}{2!} T''_s(x_W) \quad (\text{A.10b})$$

$$T_s(x_W + 2\Delta x) = T_s(x_W) + T'_s(x_W)2\Delta x + \frac{\Delta 4x^2}{2!} T''_s(x_W) \quad (\text{A.10c})$$

An extra step with respect to the first order must be taken before filling in the boundary conditions. The expression for $T(x - \Delta x)$ must be multiplied by $(\frac{1}{4} - f_{L,n} + f_{L,n}^2)$ and subtracted from $T(x_W - \frac{1}{2}\Delta x + f_{L,n}\Delta x)$ to cancel out the second derivative in the Taylor-expansions. This leads to the following:

$$T'_f(x_W) = \frac{T_f(x_W - \frac{1}{2}\Delta x + f_{L,n}\Delta x) - (\frac{1}{4} - f_{L,n} + f_{L,n}^2)T_f(x_W - \Delta x) - (\frac{3}{4} + f_{L,n} - f_{L,n}^2)T_f(x_W)}{(-\frac{1}{4} + f_{L,n}^2)\Delta x} \quad (\text{A.11})$$

The derivative in the solid domain is given by:

$$T'_s(x_W) = \frac{-\frac{1}{2}T_s(x_W + 2\Delta x) + 2T_s(x_W + \Delta x) - \frac{3}{2}T_s(x_W)}{\Delta x} \quad (\text{A.12})$$

This can be obtained by a strategy to that in equation A.11, by combining a linear combination of $T(x_W + \Delta x)$ and $T(x_W + 2\Delta x)$ to cancel out the second derivative term in the Taylor-expansions. Now the boundary condition equations A.1 and A.2 can be used together with the fact that the temperature at the phase front is T_l and (after doing some algebra) the second order temperature boundary condition is found:

$$T_{BC} = \frac{\lambda_l(T_l - T_f(x_W - \Delta x)(\frac{1}{4} - f_{L,n} + f_{L,n}^2)) - \lambda_s(f_{L,n}^2 - \frac{1}{4})(2T_s(x_W + \Delta x) - \frac{1}{2}T_s(x_W + 2\Delta x))}{\lambda_l(\frac{3}{4} + f_{L,n} - f_{L,n}^2) - \frac{3}{2}\lambda_s(f_{L,n}^2 - \frac{1}{4})} \quad (\text{A.13})$$