Developing a Filter-Matrix Lattice Boltzmann Boundary Method for Conjugate Heat Transfer and Phase Change

Master Thesis

Lotte Borstlap

Delft University of Technology

TU Delft

Developing a Filter-Matrix Lattice Boltzmann Boundary Method for Conjugate Heat Transfer and Phase Change

Master Thesis

by

Lotte Borstlap

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on February 29 2024 at 10:00h

Student Number:	4573757	
Project Duration:	April 2023 - Februari 2024	
Thesis committee:	Dr. Ir. M. Rohde	TU Delft, supervisor
	Dr. Ir. D. Lathouwers	TU Delft
	Dr. Ir. B. Bera	TU Delft

Faculty of Applied Sciences · Delft University of Technology

Highlights

The most notable contributions of the current research to existing literature are:

- *The development of a novel Lattice Boltzmann by adding a wet-node boundary condition to a filter-matrix double distribution function. With the ability of modelling conjugate heat transfer & phase change and showing second-order accuracy for Dirichlet momentum boundary conditions*
- *The ability to model a MSFR freeze-plug geometry with conductive melting with a double-distribution-filter-matrix lattice Boltzmann method*

Abstract

Nuclear reactors can offer a stable and reliant source of renewable energy, which is highly important to fill the gaps left by intermittent renewable energy sources like solar and wind. The Molten Salt Fast Reactor can be promising because of its reduced issues with waste management and high safety standards. The freeze-plug safety system is an integral part of the MSFR, melting in the case of an emergency and thereby allowing drainage of the reactor core. This research aims to develop a Filter-Matrix Lattice Boltzmann model to simulate two crucial factors in the freeze-plug melting: convective phase change and conjugate heat transfer. To that end, novel filter-matrix-wet-node boundary conditions are developed.

The Filter Matrix Lattice Boltzmann Method (FM-LBM) with double-distribution functions (DFF) was combined with the total-enthalpy based enthalpy method and a novel wet-node-solution-vector boundary condition (WNSV BC). The model was benchmarked with three different problems. First, it showed good correspondence to benchmark studies for a steady-state square cavity natural convection problem. Second, modelling of steady-state conjugate heat transfer in combination with natural convection also showed a small error with numerical benchmark studies. Third, phase change was implemented, which showed excellent results for conduction melting with respect to two numerical benchmark studies. However, a larger error was found for convective melting, expectedly caused by unphysical numerical diffusion of latent heat across the phase front. Future research could examine if grid refinement near the phase front can reduce this error.

The individual model components were combined in modelling a freeze-plug geometry, in two subsequent stages. First, a freeze-plug geometry combining conjugate heat transfer and conductive melting was modelled, in very good correspondence to a benchmark study. The second stage also included natural convection and thereby convective melting, but showed severe instabilities. Future research should look into the WNSV BC's ability to combine conjugate BCs with the no-slip phase front BC.

Different extrapolation schemes from the wet-node-solution-vector boundary condition were examined for their numerical performance. Two extrapolation schemes showed second-order-accuracy in modelling a Poiseuille flow. The found error with the analytical solution was somewhat higher than for a conventional half-way-bounce-back scheme. For modelling a conjugate natural convection case, an order of accuracy between $1.6 - 2.0$ was found. As no analytical solution for this case was available, the order of convergence was calculated with a finest-grid approach and an approach suggested by Eça & Hoekstra. Future research can aim to verify the second-order-accuracy in thermal flows by modelling a case with an available analytical solution.

Contents

Nomenclature

Symbol	Definition	Unit
A_{mush}	mushy constant	$kg/m^3/s$
B_f	weight factor	
C_ϕ	conversion factor for ϕ	[ϕ]
$\mathbf{c}_i$	discretized lattice speed	ls/lt
c_p	specific heat capacity	$J/kg/K$
c_s	lattice speed of sound	ls/lt
d	wall thickness	m
E	energy	$lJ = ls^2/lt^2$
E_{ik}, E_{ki}	filter matrix	
f_i	density distribution function	
f_L	liquid fraction	
$\mathbf{f}$	body force	N/m^3
g	gravitational acceleration	m/s^2
g_i	temperature distribution function	
H	total enthalpy	J/kg
H_l	liquid enthalpy	J/kg
H_s	solid enthalpy	J/kg
h	sensible enthalpy	J/kg
m_i	enthalpy distribution function	
m_i^l	latent enthalpy distribution function	
m_i^s	sensible enthalpy distribution function	
K	constant	
L	latent heat	J/kg
N	amount of grid points	
p	order of accuracy	
Q'''	volumetric heat source	W/m^3
r	grid refinement ratio	
s_{av}	average phase front position	
T	temperature	K
T_0	reference temperature	K
T_C	cold temperature	K
T_H	hot temperature	K
T_l	liquidifus temperature	K
T_s	solifidus temperature	K
t	time	s
$\mathbf{u}$	velocity	m/s
v_0	characteristic lattice velocity	ls/lt

Symbol	Definition	Unit
α	thermal expansion coeff.	$1/K$
α_k	momentum solution vector	
β_k	temperature solution vector	
γ_k	enthalpy solution vector	
Δt	lattice time step	lt
Δx	lattice spacing	ls
δ_ϕ	time convergence criterion	
ϵ	error	$\%$
θ	non-dimensional time	
κ	thermal diffusivity	m^2/s
λ	conductivity	$W/m/K$
ν	kinematic viscosity	m^2/s
ρ	density	kg/m^3
ρ_0	reference density	ls^{-3}
$\sigma_{1,2}$	small factor	
ϕ	any quantity/solution	$[\phi]$
ϕ_0	reference quantity/solution	$[\phi]$
$\phi_{in,out}$	surface in-, out- flux	$lW = ls^2/lt^3$
φ	phase tag	
Ω_i	collision operator	
Ω_i^s	special collision operator	

Non-dimensional numbers	
Co	Courant number
Fo	Fourier number
Ma	Mach number
$Nu_{(0,H)}$	Nusselt number (at $x = 0, H$)
Pe	Péclet number
Pr	Prandtl number
Ra	Rayleigh number
Ste	Stefan number

Abbreviation	
BC	Boundary Condition
BGK	Bhatnagar-Gross-Krook
CFD	Computational Fluid Dynamics
EHM	Eça-Hoekstra Method
FM	Filter-Matrix
GIF	Generation International Forum
HBB	Half-Way-Bounce-Back
LB(M)	Lattice Boltzmann (Method)
LBE	Lattice Boltzmann Equation
MRT	Multi-Relaxation-Time
MS(F)R	Molten Salt (F)ast Reactor
NSE	Navier-Stokes Equations
WNSV	Wet-Node-Solution-Vector

Chapter 1

Introduction

A mix of different renewable energy sources is required to limit the global temperature increase to 1.5 $^{\circ}\mathrm{C}$ by 2030 [1], as internationally agreed in the Paris Agreement. While renewable energy sources like solar, wind and hydropower have been growing rapidly, their inherent intermittency limits the potential for full-scale expansion. Therefore, nuclear energy can offer a reliable addition to the global renewable energy mix. Nuclear energy can supply consistent, base-load power, which is vital in a world where both industry and households are increasingly electrifying their energy demand. Although historic accidents sometimes still shed a negative light on nuclear energy, it has gradually received renewed attention as a part of the energy transition in recent years.

The development of nuclear technologies is coordinated by the Generation International Forum (GIF) [2]. Previous generations of nuclear reactors have led the way from proof-of-concepts to economically reliable reactors with advanced safety systems. The current Generation-IV reactors aim to include reliable passive safety systems, sustainability, efficiency and cost-effectiveness. Six different promising nuclear reactors have been selected within Generation IV, amongst which is the Molten Salt Fast Reactor (MSFR).

The MSFR research efforts are currently coordinated by SAMOSAFER [3], a consortium funded by the EU as part of the European Energy Roadmap 2050 and of which TU Delft is one of the leaders. SAMOSAFER's overarching objective is to ensure the MSFR's compliance with all expected regulations by 2050. At the moment, the design of the MSFR is conceptual and relies on numerical modelling to assess potential safety risks before starting construction. Central to the MSFR's safety studying heat transfer within the reactor core and its surrounding system. The current research is focused on numerically modelling of heat transfer in the context of one of the safety features: the so-called freeze-plug.

This chapter starts with an introduction to the design of the MSFR and freeze-plug in section 1.1. Relevant literature studies on the freeze-plug and the numerical simulation method are discussed in sections 1.2 and 1.3, respectively. Finally, the research question and thesis outline are discussed in section 1.4.

1.1 Molten Salt Fast Reactor

The MSFR design stems from the earlier developed thermal molten salt reactor (MSR); a graphite-moderated, liquid-fueled fission reactor first developed in the 1950s by the Oak Ridge National Lab Despite its promising results in a test reactor [4], the program was shut down in the 1970s, partly because its thermal neutron spectrum required intensive chemical processing to avoid neutron captures leading to minor actinides. In the last two decades, the MSR technology regained interest, also giving rise to the proposal of the Molten Salt **Fast** Reactor.

1.1.1 Design

The current MSFR design concept [5][6] is a fast-spectrum, 3 GW_{th}, breeder reactor operating in the thorium fuel cycle, under ambient pressure and at 750 $^{\circ}$C. The MSFR is filled with a molten fuel salt

which itself also acts as a primary coolant. The preliminary design, as shown in figure 1.1, is a single compact cylinder (2.25m x 2.25m) filled with fuel salt and surrounded by a ring of thorium-salt in the walls. The ring acts as a breeding blanket, in which non-fissile Th-232 is transmuted to fissile U-233 upon capturing a neutron and via two beta-decays. This happens without any solid moderator and thus in the fast neutron spectrum. The pumps circulate the fission-heated fuel salt in the core from bottom to top, where it flows out towards a number of identical sections. Here, the salt releases its thermal energy to the heat exchangers for power production. In each section, a helium bubble separation system is present, which separates the gaseous fission products and (semi-)noble metal particles from the fuel salt, preventing corrosion in the primary circuit [5]. Lastly, a drainage towards safety tanks is present at the bottom of the core, sealed with a freeze-plug. Section 1.1.2 will elaborate further on this freeze-plug mechanism.

The MSFR offers several advantages with respect to conventional solid-fueled nuclear reactors [7]. First, while one of the main arguments against nuclear energy is the need for long-term storage, the MSFR produces waste that is mostly free of long-lived actinides. This is a result of the MSFR's fast-spectrum operation and its ability to perform real-time waste processing. In addition, the MSFR could offer a solution to the waste storage problem of conventional nuclear reactors, as it is able to burn long-lived waste as a fuel. Another main advantage of the MSFR is its inherent safety, owing to its liquid core. The liquid is free to expand upon heating, causing a strong negative temperature reactivity feedback loop. The liquidity of the salt also makes it possible to passively remove decay heat during accidents, via the drainage of fuel through a freeze-plug-sealed pipe.

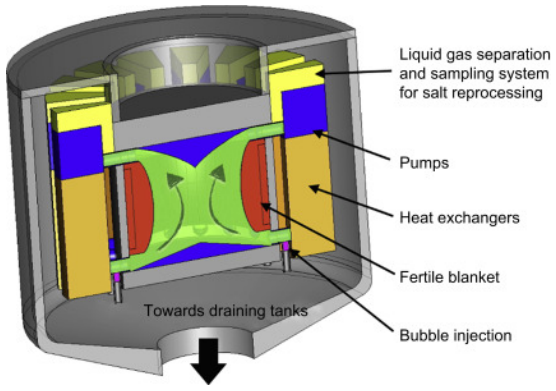


Figure 1.1: Schematic view of cross-section of MSFR fuel circuit and draining system [8].

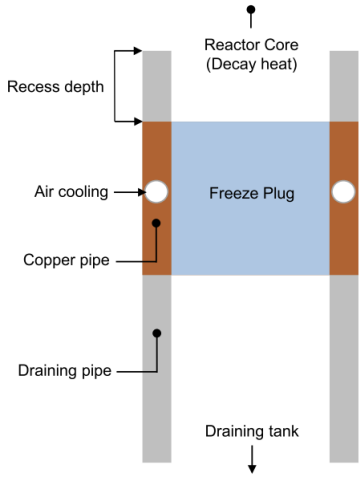


Figure 1.2: Vertical cross section of the freeze-plug design [9]

1.1.2 Freeze-Plug

The context of this thesis is the freeze-plug, a key safety component of the MSFR. The most far-developed design of the freeze-plug was built by Giraud [10]. A simplified, schematic overview is given in figure 1.2. The freeze-plug consists of a barrier of solidified salt that sits in an actively air-cooled part of the drainage pipe and stays frozen during normal operations. In case of an accident, the freeze-plug offers a heat sink solution: its active cooling shuts off and the decay heat from the reactor will cause the freeze-plug to melt. The opened drainage pipe allows the fuel salt to flow into emergency dump tanks, where it is stored under sub-critical conditions. The melting time of the freeze-plug is an important feature: it must be short enough such that the decay heat in the core does not reach critical temperatures (> 2100 °C[9]) in case of an emergency, as this can structurally damage the core. The adjacent, metal drainage pipe wall can play a crucial role in enhancing the melting, as it conducts and stores decay heat coming from the reactor core. Hence, the freeze-plug does not have to melt all the way through: melting its outer circumference near the hot wall is enough for the freeze-plug to fall down and thereby open the drainage pipe. In conclusion, it is of significant importance to understand the melting of the freeze-plug and which parameters influence it to assess the MSFR's safety.

1.2 Review of Literature on the Freeze-Plug

Insights on the suitable design parameters of the freeze-plug can be offered through (a combination of) experimental and numerical research. However, experimental investigations on the freeze-plug that fully include the complexity of the MSFR and fuel salt are scarce [11]. Aji [12] conducted basic investigations on the freeze-plug parameters and found that heat conduction within the surrounding pipe structures substantially impacts the melting time. Aji also studied the effect of natural convection [13] and found that it prolonged the melting time in a vertical position. However, when inclining the freeze-plug with an angle, the effect of natural convection contributed to shorter melting times. the effect of natural convection could contribute to shorter melting times when inclining the freeze plug with an angle to the horizontal axis. Although Aji's results provided interesting insights into the general importance of the surrounding wall and natural convection on the melting time, they could only be generalised for non-realistic, small-diameter plugs. Tiberga et al. [9] performed a preliminary investigation on the influence of various design parameters (e.g. sub-cooling, recess depth) on its melting time with a numerical COMSOL method. The study concluded that melting only based on decay heat was likely to be unfeasible, and recommended to further study heat transferred from the surrounding wall structures. Tartaglia [14] studied the combination of convection, phase change and wall effects in the freeze-plug by using a 3D OpenFOAM model and also found the significant importance of the wall effect on the melting time. However, the study simplified the wall effect by modelling it with a time-dependent temperature boundary condition instead of considering coupled heat conduction through the wall. Therefore, Tartaglia advised to develop a multi-region solver that is capable of dealing with the different structural components of the freeze plug system. Pater & Kaaks [11] recently developed a numerical benchmark of the freeze-plug and combined the modelling of phase change, conjugate heat transfer, natural convection and radiation. Results were modelled with three different codes (STAR-CCM+, OpenFoam, DGFlows) and were mostly consistent between the three of them. However, the addition of conjugate heat transfer showed discrepancies. In conclusion, literature indicates that the effects of natural convection and conjugate heat transfer with surrounding wall structures are of significant importance to the freeze plug's melting time, but numerically benchmarking these effects accurately has proved to be complicated. Therefore, the current thesis focuses on coupling convective phase change (melting) and conjugate heat transfer at a fluid-solid interface.

1.3 Review of Literature on the Lattice Boltzmann Method

The Lattice Boltzmann Method (LBM) is a Computational Fluid Dynamics (CFD) method that has received increasing interest over the past decades [15]. Where conventional CFD methods (e.g. finite difference, finite volume, finite elements) take a *macroscopic* view on a fluid by solving the Navier-Stokes equations, the LBM treats a fluid as an ensemble of particles described by a distribution function in a *mesoscopic* view. These ensembles move through space and interact with other ensembles through *collisions*. A major advantage of the LBM is that it treats non-linearity locally and is therefore highly suitable for parallel computing.

This research uses the LBM for numerically modelling the freeze-plug and thereby spans across different aspects of the method: (convective) phase change, conjugate heat transfer, wet-node boundary conditions and the filter-matrix collision scheme. The most relevant literature on those different aspects is summarised in this section.

- Multiple studies have proven the LBM to be successful for **convective phase change** problems (for example Huber et al. [16], Betrand et al. [17]). Most of these studies use a so-called enthalpy-updating scheme [18], treating the inclusion of phase change's latent heat into the temperature equation in an iterative manner. As this process greatly increases the computational cost, Huang et al. [19] proposed an alternative method to avoid iterations: the total-enthalpy-based approach. This method has attracted much attention due to its computational efficiency and accuracy [20].

- **Conjugate heat transfer** deals with the interaction of heat at a solid-fluid interface and its application within LBM is a relatively new practise [21]. It requires defining a suitable boundary condition at the fluid and solid interface. Wang et al [22] made a first attempt by adopting a steady-state half-lattice division treatment for the fluid-solid interface. Other schemes using a *wet-node* approach, where the conjugate interface is *on* the lattice nodes, have also been developed [23][24]. However, they mostly required the specific capacities of the fluid and solid to be identical. Li et al.

[25][26] recognised the importance of the interface location and proposed a method that could deal with transient problems. However, the mathematical formulation is relatively complicated even in a small-dimensional case and was not very computationally efficient [27]. Le et al. [28] proposed performing extrapolation along the normal direction, transforming the conjugate boundary condition to a Dirichlet boundary condition. Second-order accuracy was achieved using this method.

- The **wet-node boundary** approach could offer an instinctive and elegant approach to solving conjugate heat transfer problems. There are three well-known wet-node boundary schemes. The equilibrium scheme (ES) [29] uses the so-called equilibrium distribution to enforce macroscopic quantifies on the boundary. Despite its simplicity and excellent stability, it reduces the LBM to first-order accuracy. The non-equilibrium extrapolation scheme [30] and the non-equilibrium bounce-back method [31] propose an improvement on the ES, achieving second-order accuracy. However, they show moderate to low stability and complex extension to more complex geometries.

- All previously mentioned work employ conventional collision schemes: Bhatnagar–Gross–Krook (BGK) or Multiple-Relaxation-Time (MRT) collision. Eggels et al. [32] proposed an alternative collision scheme, the **Filter-Matrix** (FM) method. Zhuo & Zhong [33][34] further improved the FM-LBM and showed its superior stability with respect to BGK in incompressible thermal flows. Another great advantage of the FM-LBM is that it gives the user great control over the macroscopic quantities in question. Recently, the FM-LBM has proved its effectiveness in multiple MSc theses, including transient freezing of eutectic materials [35], turbulent thermal flows [36] and flow batteries [37][38].

1.4 Thesis Goals

This thesis' goal is to combine the separate knowledge on (convective) phase change, conjugate heat transfer, wet-node boundary conditions and the Filter-Matrix Lattice Boltzmann method. This will contribute to the extension of LBM applications, as the FM-LBM has not been applied in combination with wet-node boundary conditions and with conjugate heat transfer yet. In addition, it is aimed to contribute to a better understanding of the melting of the MSFR freeze-plug through numerical modelling. This leads to the following research question:

How can the Filter-Matrix lattice Boltzmann method be implemented to model the melting behaviour of the freeze-plug and heat transfer in the freeze-plug's surrounding structures?

- How can conjugated heat transfer best be modelled?
- What are suitable wet-node boundary conditions?
- What is the effect of the design parameters of the freeze-plug on its melting rate?

1.4.1 Outline

This thesis will answer the research question and test the developed algorithm. To that end, the report has the following structure:

- **Theory** (chapter 2) discusses the fundamental physics of the governing equations of conjugate heat transfer, phase change and natural convection. Also, an introduction to the theory behind the Lattice Boltzmann Method is given.
- **Numerical method** (chapter 3) presents the specific implementation of the Lattice Boltzmann method. This includes the Filter-Matrix collision operator, different boundary conditions, lattice units and a complete overview of the used algorithm. Additionally, a remark on numerical verification methods is made.
- **Benchmark results** (chapter 4) aims to verify the three individual aspects of the model: natural convection, conjugate heat transfer and convective melting. Benchmarking is performed by comparing results to other numerical studies available in literature.
- **Freeze-plug results** (chapter 5) combines the individual model elements into one case, modelling the freeze-plug. Results are compared to a numerical benchmark study.
- **Wet-node boundary condition results** (chapter 6) performs an analysis of the wet-node boundary conditions in terms of its grid convergence, order of accuracy, time convergence and energy conservation. Different variations in the wet-node boundary scheme are tested.
- Finally, **conclusions** & **recommendations** (chapter 7) are presented.

Chapter 2

Theory

The research question exploring conjugate heat transfer, phase change, natural convection and lattice Boltzmann boundary conditions spans across both fundamental physics and computational methods. This chapter is dedicated to laying out relevant theories within these topics as a fundamental basis for the remainder of the report. Section 2.1 covers macroscopic physical governing equations for thermal flows and phase change. Building further onto this, section 2.2 describes the theory behind three physical phenomena which arise in the freeze-plug. Lastly, section 2.3 introduces the lattice Boltzmann method (LBM) and its general workings, without going into depth about its numerical application yet.

2.1 Physical Governing Equations

The central equations describing the fundamental physics of convective, conjugate heat transfer and phase change can be derived from fluid dynamics and thermodynamics. A conventional approach is to make use of the continuum approximation: the *microscopic* behaviour of individual particles are considered negligible, such that the fluid can be characterised by its continuous, *macroscopic* properties [39]. The resulting conservation laws for mass and momentum are given in section 2.1.1, and for those energy in section 2.1.2.

2.1.1 Fluid Dynamics

Conservation of mass, in absence of sources/ sinks, is defined by the rate of mass change in a volume being equal to the mass flow rate over it's boundaries. This is described by the continuity equation [40]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.1}$$

with ρ the fluid density and $\mathbf{u}$ the fluid velocity. By assuming the fluid to be incompressible, the equation reduces to:

$$\nabla \cdot \mathbf{u} = 0 \tag{2.2}$$

The conservation of momentum for incompressible, Newtonian fluids is described by the equation [40]:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) = -\nabla P + \rho \nu \nabla^2 \mathbf{u} + \mathbf{f} \tag{2.3}$$

with ∇P the pressure gradient, ν the kinematic viscosity and $\mathbf{f}$ the external force. The left side of the equation describes the rate of change momentum flow and convection. The right side represents the total force due to the pressure gradient, viscous drag and external forces. The continuity- and momentum equation together are known as the incompressible Navier-Stokes equations (NSE).

2.1.2 Thermodynamics

Similar to momentum, the energy is also conserved. This section first describes the energy equation in terms of temperature. Thereafter, the temperature equation is modified to allow expression in terms of enthalpy, which is more suitable when dealing with the description of phase change.

The conservation of energy in an incompressible fluid with constant specific heat c_p and neglecting viscous dissipation is given by the heat equation:

$$\rho c_p \frac{\partial T}{\partial t} + \rho c_p \mathbf{u} \cdot \nabla T = \rho c_p \nabla \kappa \nabla T + Q''' \tag{2.4}$$

with T temperature, c_p specific heat, κ thermal diffusivity, Q''' a volumetric heat source/sink $[W/m^3]$. The left-hand side represents the rate of change of temperature in the volume and the advection of energy. The right-hand side describes the diffusion of heat and a source term [41]. In this thesis, the energy equation in terms of temperature T is used when dealing with domains without phase changes.

Phase Change: Enthalpy Equation

When dealing with phase change it is more convenient to work with the energy equation in terms of total enthalpy H, instead of temperature. This section aims to find the energy equation in terms of H, based on a derivation by Bus & Rohde [42]. Phase change is defined by a change in total enthalpy H, consisting of a change in sensible enthalpy h (due to a change in temperature T) and in latent enthalpy ΔH (due to the internal energy released or required to enable the phase change):

$$H \equiv h + \Delta H = c_p T + f_L L \tag{2.5}$$

with f_L the local liquid fraction and L the latent heat. The liquid fraction defines the phase of the current: 0 for solid, 1 for liquid and between 0 and 1 the so-called *mushy zone*. The absorption/release of latent heat L forms a heat sink/source Q''' in equation 2.4, which can be described by [43]:

$$Q''' = -\frac{\partial(\rho \Delta H)}{\partial t} = -\frac{\partial(\rho f_L L)}{\partial t} \tag{2.6}$$

The equation for sensible enthalpy h can be found by filling in Q''' from equation 2.6 into the heat equation 2.4 and using the definition of h stated in 2.5. This gives the following expression:

$$\rho^\varphi \frac{\partial h^\varphi}{\partial t} = \rho^\varphi \mathbf{u} \cdot \nabla h^\varphi = \nabla(\rho^\varphi \kappa^\varphi \nabla h^\varphi) - \rho^\varphi \frac{\partial f_L^\varphi L}{\partial t} \tag{2.7}$$

in which the definition $\varphi = [\mathbf{l}iquid, \mathbf{s}olid, \mathbf{m}ushy]$ is introduced, to describe h for the whole domain in one equation. To arrive at the conservation equation for total enthalpy H, the two transient terms in equation 2.7 are combined, and the following assumptions are made:

- $c_p^\varphi \approx$ constant, $\rho^\varphi \approx$ constant, i.e. the density and specific of the material are approximately constant within the same phase.
- $\nabla f_L L \approx 0$ [43], i.e. the advection of latent heat can be ignored hence changes in f_L are only incorporated in the time derivative. This can be assumed because velocities near the mushy zone are approximately zero.

Finally, this leads to the energy equation in terms of total enthalpy H:

$$\frac{\partial H^\varphi}{\partial t} + \mathbf{u} \cdot \nabla h^\varphi = \nabla \cdot \left(\kappa^\varphi \nabla h^\varphi \right) \tag{2.8}$$

With this equation, the explicit temperature dependency in solving for energy is eliminated - called the *total enthalpy-based* approach [43]. In other words, by solving equation 2.8 the temperature and enthalpy are tracked simultaneously, instead of needing an implicit scheme to iteratively include the latent source term into the temperature equation (the *enthalpy-updating* approach [18]).

The temperature can be retrieved from the temperature-enthalpy relationship, which in liquid-solid phase change is characterized by a jump in enthalpy at the melting point. This is visualised in figure 2.1, in which an instantaneous jump is seen for *pure* materials ($T_m = T_s$) and a transition regime for so-called *eutectic* materials ($T_m = T_s + \delta$). In a formula, this is given by:

$$T = \begin{cases} H/c_p^s & \text{if } H < H_s \\ T_s + f_L(T_l - T_s) & \text{if } H_s \leq H < H_l \\ T_l + \frac{H - H_L}{c_p^l} & \text{if } H > H_l \end{cases} \tag{2.9}$$

with T_s, T_l the solidifus- and liquidifus temperature and H_s, H_l are the solid and liquid enthalpy. The latter are given by $H_s = c_p^s T$, $H_l = H_s + L$. The calculation of the liquid fraction f_L is given by:

$$f_L = \begin{cases} 0 & \text{if } H < H_s \\ \dfrac{H - H_s}{H_l - H_s} & \text{if } H_s \leq H < H_l \\ 1 & \text{if } H \geq H_l \end{cases} \tag{2.10}$$

When the thermophysical properties specific heat and thermal conductivity have different values for different phases, their value in the mushy zone can be set by their respective fractions [44]:

$$c_p^m = \frac{c_p^s + c_p^l}{2}, \quad \lambda^m = \frac{\lambda_p^s + \lambda_p^l}{2} \tag{2.11}$$

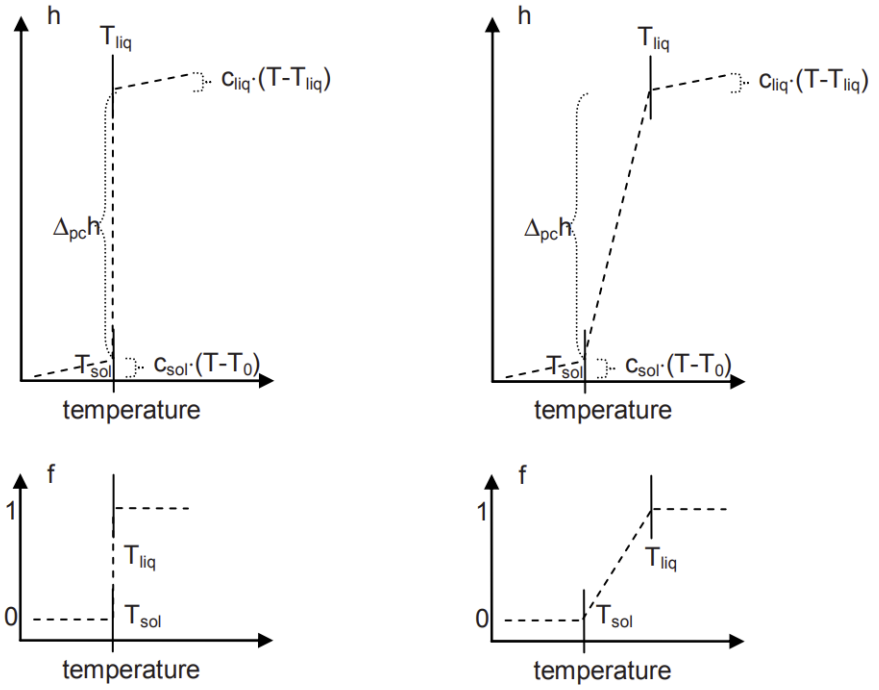


Figure 2.1: Left: $H(T)$ function for a pure substance, with a step at the melting temperature. Right: $H(T)$ function for eutectic substance, with a linear rise in melting range. [44]

2.2 Phenomena in the Freeze-plug

In the context of the freeze-plug, this research studies (the combination of) three physical phenomena: natural convection, conjugate heat transfer and convective melting. This section introduces the theory behind these phenomena and, if applicable, their governing equations and related non-dimensional numbers.

2.2.1 Natural Convection

Natural convection is the flow in a fluid driven by temperature differences. Its effect on a fluid can be described by using the *Bousinessq approximation*. It assumes that local temperature differences are small and can therefore be linearised around a reference temperature T_0. Also, it assumes that density changes are induced only by these temperature changes (i.e. pressure variations are negligible), such that it can also be linearised around a reference density ρ_0: $\rho = \rho_0 + \Delta\rho$, with $\Delta\rho << \rho_0$. Thereby, a Taylor approximation leads to [41]:

$$\rho(T) \approx \rho_0 - \rho_0\,\alpha\big(T - T_0\big) \tag{2.12}$$

with α the thermal expansion coefficient. In the momentum equation (2.3) it is assumed that density is constant ($\rho = \rho_0$), except for in the gravitational force term. The latter is called the buoyancy force and is given by the following definition:

$$\mathbf{f} = \big(\rho(T) - \rho_0\big)\mathbf{g} = \alpha\rho_0(T - T_0)\mathbf{g} \tag{2.13}$$

Natural convection problems can be characterised with the non-dimensional numbers Rayleigh (Ra) and Prandtl (Pr). The Rayleigh number captures the intensity of natural convection and is defined as the ratio of the timescale of diffusive thermal transport to the timescale of convective thermal transport:

$$Ra = \frac{g\alpha(T - T_0)L^3}{\nu\kappa} \tag{2.14}$$

with L the characteristic length of the geometry. The Prandtl number Pr indicates the relative thickness of momentum and temperature boundary layers in a natural convection domain, and thereby the rate of heat transfer. The formulation of Pr is given by the ratio between momentum diffusivity and thermal diffusivity:

$$Pr = \frac{\nu}{\kappa} \tag{2.15}$$

2.2.2 Convective Melting

Melting of the freeze-plug is expected to be influenced by heat transfer through natural convection [12]. A case of convective melting can be illustrated by the geometry shown in figure 2.2. A two-dimensional box is filled with ice and the left boundary is kept at hot temperature $T = T_1$, making the ice to melt over time. It is commonly accepted that this transient convective melting problem consist of four transition regimes [16][45]. The first regime is for short melting times, when the heat transfer is entirely defined by conduction. The melting front is still parallel to the vertical wall and the width of the molten area is too small for natural convection to develop. In the second regime the liquid boundary layer has grown and convection starts taking place mostly in the top part, making it melt faster than in the bottom part. This is called the *mixed* regime because both conduction and convection play a role. In the next regime, the convection zone extends to the entire height of the cavity, and conduction is negligible. Finally, the melting front has reached the right boundary the sold starts shrinking.

In addition to non-dimensional numbers Ra, Pr, convective melting is also described by the Stefan number (Ste). The latter is defined as the ratio between sensible heat and latent heat L and thereby expresses the rate of phase change:

$$Ste = \frac{c_p\Delta T}{L} \tag{2.16}$$

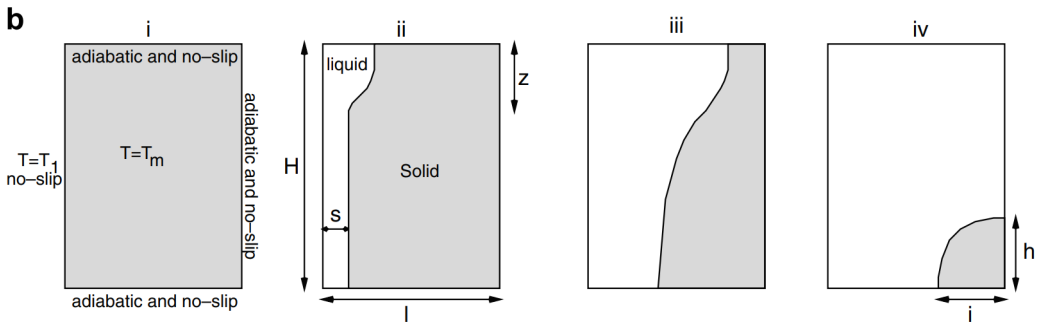


Figure 2.2: i) Conduction regime ii) Mixed regime iii) Convection regime iv) Shrinking solid [16].

2.2.3 Conjugate Heat Transfer

Conjugate heat transfer refers to the interactive heat transfer between fluid and an adjacent solid. It is mostly characterised by the combination of conduction (in the solid) & convection (in the fluid) and its complexity lies in the boundary interface that couples the two domains. On this interface, the following boundary conditions for temperature hold:

$$T_{f,W} = T_{s,W} \tag{2.17}$$

$$-\lambda_f \left.\frac{\partial T}{\partial n}\right|_{f,W} = -\lambda_s \left.\frac{\partial T}{\partial n}\right|_{s,W} \tag{2.18}$$

in which subscripts f, s note the respective fluid and solid domain, subscript W the spatial position of the shared interface, λ the thermal conductivity ($\lambda = \kappa/(\rho c_p)$) and n the normal direction of the interface.

When solving conjugate problems numerically, there are mainly two different methods of conjugating the solid and fluid domain solutions. The first method is a direct approach in which the complete set of equations is solved in both the domains simultaneously. The second method is the iterative approach where the fluid and solid domains are solved separately and only their solutions are conjugate at the interface.

2.3 Fundamentals of the lattice Boltzmann method

In previous sections, fluid dynamics has been described by viewing it at the macroscopic scale. However, in fact three different physical scales can be used to describe fluids: microscopic, mesoscopic and macroscopic. Where the microscopic scale tracks individual particles and the macroscopic scale treats properties like density, velocity and temperature, the mesoscopic scale lies in between the two. It describes the fluid flow by the *distribution* of particles. In this research the fluid- and thermodynamics will be modelled in a mesoscopic approach, by using the Lattice Boltzmann Method (LBM).

Section 2.3.1 starts by introducing kinetic theory on which the LBM is based, and the resulting Boltzmann equation is discretized in section 2.3.2. Thereafter, section 2.3.3 introduces the basic concept of a LBM's basic algorithm. Chapter 3 will further elaborate on the specific algorithm implementation.

2.3.1 Kinetic Theory

Kinetic theory is a gas and fluid representation on the mesoscopic scale, in which the central quantity is the particle distribution function $f(\boldsymbol{x}, \boldsymbol{\xi}, t)$. The particle distribution function describes the behaviour of an ensemble of atoms or molecules, at each point in space $\boldsymbol{x}$, moving with microscopic velocity $\boldsymbol{\xi}$ at each moment of time t. The distribution function is connected to macroscopic variables density ρ and velocity $\mathbf{u}$, by integrating over all possible velocities ξ at $(\mathbf{x}, t)$ [15]:

$$\rho(\boldsymbol{x}, t) = \int f(\boldsymbol{x}, \boldsymbol{\xi}, t) \boldsymbol{d}^3\xi \tag{2.19}$$

$$\rho(\boldsymbol{x}, t)\boldsymbol{u}(\boldsymbol{x}, t) = \int \boldsymbol{\xi} f(\boldsymbol{x}, \boldsymbol{\xi}, t) \boldsymbol{d}^3\xi \tag{2.20}$$

The time evolution of distribution function f is described by the so-called Boltzmann equation:

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla_x f + \frac{F}{\rho} \nabla_\xi f = \Omega(f) \tag{2.21}$$

This can be interpreted as a kind of advection equation. The left hand side shows the distribution function f being advected with particle velocity ξ and the forces F affecting this velocity. On the right hand side the source term is represented by a so-called *collision operator* $\Omega(f)$. The collision operator imitates inter-molecular behaviour as distributions move forwards and collide with each other. As in all physical collisions, the quantities mass, momentum and energy must be conserved and the collision operator takes care of that. It also ensures that distribution function f will locally evolve to an equilibrium f^{eq}. Chapter 3 will elaborate further on how to define the collision operator within the LBM.

2.3.2 Discretization of the Boltzmann Equation

In order to solve the Boltzmann equation 2.21 numerically, the distribution function f needs to be discretized. Discrete distribution function $f_i(\mathbf{x}, t)$ is defined in a square lattice of points $\mathbf{x}$ with lattice spacing Δx $[ls]$ and at times t with lattice time steps Δt $[lt]$. The units ls and lt refer to lattice space and lattice time. Subscript i in f_i refers to its discretization in a set of velocities $\mathbf{c}_i$. The choice of spatial discretization dimension (d) and number of discrete velocities (q) together result in a discretization scheme called DdQ9. In this research, the 2D space is discretized via the D2Q9 scheme. A visualisation for the resulting f_i with $i = 0, 1, .., 8$, in one lattice point is shown in figure 2.3. The discretized velocity set $\mathbf{c}_i$ is completed by attributing corresponding weights w_i to each direction, for D2Q9 given by:

$$w_i = \begin{cases} \frac{4}{9} & \text{for } i = 0 \\ \frac{1}{9} & \text{for } i = 1, 2, 3, 4 \\ \frac{1}{36} & \text{for } i = 5, 6, 7, 8 \end{cases} \tag{2.22}$$

Based on the discretizations, the lattice Boltzmann equation (LBE) now takes the form:

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i \Delta t, t + \Delta t) - f_i(\boldsymbol{x}, t) = \Delta t \Omega_i(\boldsymbol{x}, t) \tag{2.23}$$

with $\Omega_i(\mathbf{x}, t)$ the discretized collision operator. Furthermore, the macroscopic momentum variables can be found from the distribution function via a similar method as in the continuous case (equations 2.19, 2.20):

$$\rho(\boldsymbol{x}, t) = \sum_i f_i(\boldsymbol{x}, t) \tag{2.24}$$

$$\rho(\mathbf{x}, t)\boldsymbol{u}(\boldsymbol{x}, t) = \sum_i \boldsymbol{c}_i f_i(\boldsymbol{x}, t) \tag{2.25}$$

Lastly, the equilibrium distribution function is given by

$$f_i^{eq} = w_i \rho \Big(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} + \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \Big) \tag{2.26}$$

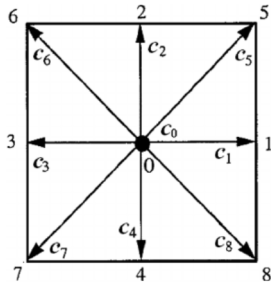


Figure 2.3: Schematic of the directions $i = 0, 1, ..., 8$ in the velocity set D2Q9 of $\{\mathbf{c}_i, w_i\}$ [46].

Chapsman-Enskog Analysis

Having found a suitable LBE to use for solving fluid dynamics, the question might remain how the LBE exactly relates to a problem's macroscopic governing equations (section 2.1). The answers lies in the so-called *Chapsman-Enskog analysis*, which shows that the Navier-Stokes equation can be retrieved from the LBE. The full analysis lies beyond the scope of this report, but the interested reader can employ [15] for reference. An important point that arises from the Chapman-Enskog analysis, is that second-order accuracy in LBM can be achieved by assuming certain higher-order-terms to be negligible. This assumption is valid as long as the *Mach* number remains small. The resulting stability requirements are discussed in section 3.3.

LBM for Thermodynamics

Until this point, only the density distribution function f_i has been discussed, which can solve for macroscopic properties density and velocity. In the context of thermal- & phase change flows, the energy is to be solved as well. Several methods exist to include energy into a LBM, for example the *hybrid* [47] and *multi-speed* [48] approach. The method that is employed in this thesis, is the *double-distribution function*, which treats heat in a very similar manner as momentum. The idea is to introduce a second distribution function to describe the flow of heat, next to the distribution function f_i for momentum. The two distributions are then explicitly coupled in the collision step. In this way, all calculations are performed within the LBM method (unlike for example the hybrid method), which proves to be very successful in terms of stability, accuracy and parallezability [49].

The Navier-Stokes momentum equation 2.3 and the heat equation 2.4 have strong similarities, which are visible when making the following changes to the NSE:

$$\rho\mathbf{u} \rightarrow \rho c_p T, \qquad \rho\mathbf{u}\nabla\mathbf{u} + \nabla P \rightarrow \rho c_p \mathbf{u}\cdot\nabla T, \qquad \nu \rightarrow \kappa, \qquad \mathbf{f} \rightarrow Q''' \tag{2.27}$$

Thereby, the heat equation could in fact be understood as a NSE for heat [15], making it easy to adapt the LBM to temperature and enthalpy problems. The discretization of temperature distribution function g and enthalpy distribution function h in the D2Q9 follows the same set $\mathbf{c}_i, w_i$ and their evolution is described by:

$$g_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) = g_i(\mathbf{x},t) + \Delta t\,\Omega_i^{(g)}(\mathbf{x},t) \tag{2.28}$$

$$m_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) = m_i(\mathbf{x},t) + \Delta t\,\Omega_i^{(m)}(\mathbf{x},t) \tag{2.29}$$

Similarly to equations 2.24 and 2.25, the macroscopic temperature T and total enthalpy H can be recovered from distribution functions g_i and m_i:

$$T(\mathbf{x},t) = \sum_i g_i(\mathbf{x},t) \tag{2.30}$$

$$H(\mathbf{x},t) = \sum_i m_i(\mathbf{x},t) \tag{2.31}$$

2.3.3 Collision and Streaming

When solving the discretized distribution function equations (2.23, 2.28, 2.29) in a numerical algorithm, two subsequent processes take place in each time step. These are called *collision* and *propagation* and are illustrated in figure 2.4, taking f_i as an example. In the collision step, distribution function f_i is redistributed at each lattice node by a collision operator ($\Omega(f_i)$, resulting in a post-collision function f^*:

$$f_i^*(\mathbf{x},t) = f_i(\mathbf{x},t) + \Delta t\,\Omega_i(\mathbf{x},t) \tag{2.32}$$

Subsequently, the post-collision populations f^* are propagated from the lattice node $\mathbf{x}$ to neighbouring node $\mathbf{x}+\mathbf{c}_i\Delta t$, following their individual lattice directions $\mathbf{c}_i$. Thereby one time step Δt [lt] is completed:

$$f_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) = f_i^*(\mathbf{x},t) \tag{2.33}$$

This general structure in the LBM's algorithm also provides intuition for its (earlier introduced) suitability in parallel computing. Namely, the computational *expensive* process (collision) happens locally, while the computationally *in-expensive* process (streaming) happens non-locally. Or in Succi's [50] words: 'Non-linearity is local, while non-locality is linear'.

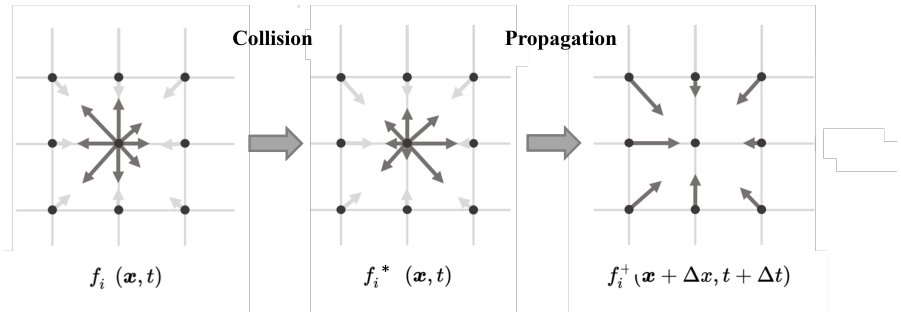


Figure 2.4: Schematic of general LBM algorithm [38]. Distributions are redistributed during collision. Propagation streams the distributions to their adjacent nodes

Chapter 3

Numerical method: Lattice Boltzmann

This chapter describes the implementation of the lattice Boltzmann method for a thermal fluid dynamics model including conjugate heat transfer and phase change. The Filter-Matrix is used as a collision scheme, described in section 3.1. Various options for the treatment of boundary conditions within LBM are described in section 3.2. Thereafter, section 3.3 comments on how conversion parameters and lattice units are used within the LBM. This concludes all "mathematical ingredients" of the used LBM. How these are formed into a LBM algorithm is described in section 3.4. Finally, section 3.5 moves away from the LBM, and elaborates on methods for verification of numerical simulations.

3.1 Collision Schemes

In the previous chapter, collision operator Ω_i was introduced. Mulitple schemes to model Ω_i exist, the simplest and most commonly used one being the *Bhatnagar-Gross-Krook* (BGK) operator. In short, the BGK-operator brings the distribution function towards its local equilibrium f_i^{eq} on a timescale defined by relaxation time τ. The relaxation time is bound to a numerical range in which stability and accuracy are ensured, which is directly related to the kinematic viscosity ν. Therefore, the main disadvantage of BGK is that it is viscosity dependent: high viscosities result in reduced accuracy while low viscosities lead to instabilities [15]. Another collision scheme named the Filter-Matrix (FM) has shown to be more stable than the conventional BGK scheme. It also has the advantage of giving the user great control over the macroscopic quantities in question, as will be experienced in section 3.2.2.

This thesis utilises the FM-LBM for all three different distributions functions. The current section introduces the formulas associated with the FM-LBM. First this is done by example of the density distribution function f_i (section 3.1.1). Application on the temperature- and enthalpy distribution is very similar and given in sections 3.1.2 and 3.1.3, respectively. Note that later on, section 3.4 will show how to implement the FM-LBM model in an iterative algorithm.

3.1.1 Filter-Matrix for Momentum

The FM-LBM makes use of a staggered coordinate-time definition of the particle distribution f_i, i.e. equation 2.23 is shifted half a grid- and time step:

$$f_i\Big(\mathbf{x}+\frac{\mathbf{c}_i\Delta t}{2},t+\frac{\Delta t}{2}\Big)-f_i\Big(\mathbf{x}-\frac{c_i\Delta t}{2},t-\frac{\Delta t}{2}\Big)=\Delta t\,\Omega_i(f) \tag{3.1}$$

By combining a second-order Taylor expansion around $f_i(x,t)$ on equation 3.1 and a first-order expansion on equation 3.14, the staggered formulation can be rewritten into:

$$f_i\Big(\mathbf{x}\pm\frac{\mathbf{c}_i\Delta t}{2},t\pm\frac{\Delta t}{2}\Big)=f_i(\mathbf{x},t)\pm\frac{\Delta t}{2}\Big(c_i\cdot\nabla\pm\frac{\Delta t}{2}\partial_t\Big)f_i(\mathbf{x},t)+\mathcal{O}(\Delta t^2) \tag{3.2}$$

$$=f_i(\mathbf{x},t)\pm\frac{\Delta t}{2}\;\Omega_i(f_i)+\mathcal{O}(\Delta t^2) \tag{3.3}$$

By using the Chapsman-Enksnog analysis [51], a definition of the collision operator $\Omega_i(f)$ in terms of macroscopic properties can be found. This is written in the following form of a matrix multiplication:

$$f_i\Big(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}\Big) = \sum_k \omega_i E_{ik} \alpha_k^{\pm}(\mathbf{x}, t) \tag{3.4}$$

in which E_{ik} and $\alpha_k^{\pm}$ are the so-called *filter-matrix* and *solution vector*. The superscript $\pm$ indicates the *pre-* and *post*-collision solution vectors. The filter matrix is reversible (with $\omega_i E_{ik} = (E_{ki})^{-1}$), such that the solution vector can be found with a reverse operation:

$$\alpha_k^{\pm}(\mathbf{x}, t) = \sum_i E_{ki} f_i\Big(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}\Big) \tag{3.5}$$

Filter-matrix E_{ik} is independent of time and position and in the D2Q9 scheme, it is defined by:

$$\mathbf{E}_{ik} = \Big[1,\ 3c_{ix},\ 3c_{iy},\ \tfrac{3c_{ix}^2-1}{2},\ 3c_{ix}c_{iy},\ \tfrac{3c_{iy}^2-1}{2},\ \tfrac{3c_{ix}(3c_{iy}^2-1)}{2},\ \tfrac{3c_{iy}(3c_{ix}^2-1)}{2},\ \tfrac{(3c_{ix}^2-1)(3c_{iy}^2-1)}{2}\Big] \tag{3.6}$$

The solution vector is time and position dependent and changes every time step due to collision. It is defined by a set of macroscopic properties, for the D2Q9 scheme:

$$\alpha_k^{\pm}(\mathbf{x}, t) = \begin{bmatrix} \rho \\ \rho u_x \pm \dfrac{\Delta t}{2} f_x \\ \rho u_y \pm \dfrac{\Delta t}{2} f_y \\ 3\rho u_x u_x + \rho(-6\nu \pm \Delta t)\dfrac{\partial u_x}{\partial x} \\ 3\rho u_x u_y + \rho\dfrac{-6\nu \pm \Delta t}{2}\left(\dfrac{\partial u_x}{\partial y} + \dfrac{\partial u_y}{\partial x}\right) \\ 3\rho u_x u_y + \rho(-6\nu \pm \Delta t)\dfrac{\partial u_y}{\partial y} \\ T_1^{\pm} \\ T_2^{\pm} \\ F^{\pm} \end{bmatrix} \tag{3.7}$$

The last three terms $T_1^{\pm}$, $T_2^{\pm}$, $F^{\pm}$ are higher-order, unphysical terms that originate from the $\mathcal{O}(\Delta t^2)$ term in equation 3.2. Therefore they are set to zero in calculations.

3.1.2 Filter Matrix for Temperature

The FM-LBM for temperature is set up in a manner very similar to momentum. The staggered formulation for temperature density function g_i is given by:

$$g_i\Big(\mathbf{x} + \frac{\mathbf{c}_i \Delta t}{2}, t + \frac{\Delta t}{2}\Big) - g_i\Big(\mathbf{x} - \frac{\mathbf{c}_i \Delta t}{2}, t - \frac{\Delta t}{2}\Big) = \Delta t\, \Omega_i(g) \tag{3.8}$$

Collision is also performed by a matrix multiplication with filter matrix E_{ki} and now with *temperature* solution vector $\beta_k^{\pm}$:

$$g_i\Big(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}\Big) = \sum_k \omega_i E_{ik} \beta_k^{\pm}(\mathbf{x}, t) \tag{3.9}$$

$$\beta_k^{\pm}(\mathbf{x}, t) = \sum_i E_{ki} f_i\Big(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2}\Big) \tag{3.10}$$

The filter matrix E_{ik} is unchanged with respect to momentum, thus given by equation 3.6. As the solution vector deals with macroscopic quantities, and $\beta_k^{\pm}$ now describes temperature, it ofcourse defined differently than the momentum solution vector $\alpha_k^{\pm}$. For D2Q9, the temperature solution vector $\beta_k^{\pm}$ is given by [33]:

$$\beta_k^{\pm}(\mathbf{x}, t) = \left[T, u_x T + \frac{-6\kappa \pm \Delta t}{6}\frac{\partial T}{\partial x}, u_y T + \frac{-6\kappa \pm \Delta t}{6}\frac{\partial T}{\partial y}, S_1^{\pm}, S_2^{\pm}, S_3^{\pm}, T_2^{\pm}, T_2^{\pm}, F^{\pm}\right]^T \tag{3.11}$$

where $S_{1,2,3}^{\pm}$, $T_{1,2}^{\pm}$ and $F^{\pm}$ are the higher-order terms and set to zero in calculations.

3.1.3 Filter Matrix for Enthalpy

Although the FM-LBM is also very similar for enthalpy, a slight alteration is required before presenting its collision scheme. Namely, the *total* enthalpy distribution function m_i is split into a *sensible* part m_i^s and a *latent* part m_i^l, similarly to equation 2.5:

$$m_i = m_i^s + m_i^l \tag{3.12}$$

$$m_i^l = [f_L \cdot L, 0, 0, 0, 0, 0, 0, 0]^T \tag{3.13}$$

Thereby, the latent heat is only part of the static population ($i = 0$) and the collision can be performed with the sensible enthalpy distribution function m_i^s. This approach was first introduced in the MSc thesis by Besseling [52] and originally inspired by Huang et al. [43].

Apart from this alteration, the formulations for enthalpy are similar to those of momentum and temperature in respective sections 3.1.1 and 3.1.2. The staggered formulation defined for the sensible enthalpy distribution function is given by:

$$m_i^s\Big(\mathbf{x} + \frac{\mathbf{c}_i\Delta t}{2}, t + \frac{\Delta t}{2}\Big) - m_i^s\Big(\mathbf{x} - \frac{\mathbf{c}_i\Delta t}{2}, t - \frac{\Delta t}{2}\Big) = \Delta t\Omega_i(m_i^s) \tag{3.14}$$

The collision is again performed with the same filter matrix E_{ki} and with the enthapy solution vector $\gamma_k^{\pm}$:

$$m_i^s\Big(\mathbf{x} \pm \frac{\mathbf{c}_i\Delta t}{2}, t \pm \frac{\Delta t}{2}\Big) = \sum_k \omega_i E_{ik}\gamma_k^{\pm}(\mathbf{x}, t) \tag{3.15}$$

$$\gamma_k^{\pm}(\mathbf{x}, t) = \sum_i E_{ki} m_i^s\Big(\mathbf{x} \pm \frac{\mathbf{c}_i\Delta t}{2}, t \pm \frac{\Delta t}{2}\Big) \tag{3.16}$$

$$\gamma_k^{\pm}(\mathbf{x}, t) = \left[h,\ u_x h + \frac{-6\kappa \pm \Delta t}{6}\frac{\partial h}{\partial x},\ u_y h + \frac{-6\kappa \pm \Delta t}{6}\frac{\partial h}{\partial y},\ 0, 0, 0, 0, 0\right]^T \tag{3.17}$$

in which the higher-order terms are set to zero.

3.2 Boundary Conditions

The formulation of boundary conditions (BCs) in the LBM is typically a non-trivial task, as rather than specifying the macroscopic variables of interest (such as velocity and temperature), LB boundary conditions apply to the *distribution functions*. This gives rise to more degrees of freedom in the distribution functions than the available set of macroscopic variables. A wide range of different LBM BC methods is available, which can be divided into two general families: *link-wise*, and *wet-node*. Their main characterics are described in the following way:

- The **link-wise** boundary scheme is shown in figure 3.1(a): the computational boundary is located on the lattice *links*, i.e. the computational- and physical boundary do not coincide. BCs are mostly modelled with so-called *bounce-back methods*.
- The **wet-node** boundary scheme is shown in figure 3.1(b): the computational boundary is located on the lattice *nodes*, i.e. the computational- and physical boundary are the same. Or to be precise: the boundary node is assumed to lie *infinitesimally close* to the actual boundary, but still inside the fluid domain. Therefore the standard LBM steps that apply in the bulk (i.e. collision and streaming), also apply in the same way on the wet-node.

The different geometries that will be modelled in this thesis require different macroscopic boundary conditions types: (1) no-slip and isothermal (i.e. Dirichlet) BCs on an outer edge, (2) adiabatic (i.e. Neumann) BCs on an outer edge, (3) conjugate BCs at a solid-fluid interface and (4) no-slip BCs on the phase front. The following sections will describe the rationale and mathematics of the used methods for those four macroscopic BC types.

Note that the sections are categorized by LB boundary *method* rather than by macroscopic boundary condition *type*. Section 3.2.1 describes the conventional, link-wise half-way-bounce-back (HBB) method for a no-slip BC. Section 3.2.2 describes the newly introduced wet-node-solution-vector BC that can deal with Dirichlet, Neumann and conjugate BCs. Lastly, section 3.2.3 describes two different methods for describing the no-slip at a phase front.

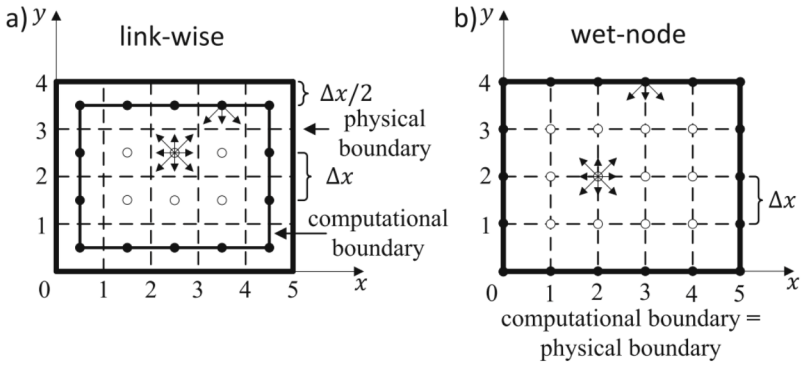


Figure 3.1: Two discretizations of the same domain with (a) link-wise and (b) wet-node boundary conditions. Fluid nodes are illustrated as open circles, boundary nodes as solid circles [15]

3.2.1 Half-Way-Bounce-Back: No-Slip

Mostly, solid boundaries have a no-slip boundary condition, which states that the fluid velocity at the boundary must be zero. In the LBM it is very common to apply this with a bounce-back method, thanks to the scheme's simplicity and inherent mass conservation [15]. A disadvantage is that bounce-back methods only guarantee higher than first order accuracy if the surface is located exactly half-way and aligned with the lattice. It is also viscosity dependent when used together with the BGK scheme [15][53].

The principle of a bounce-back scheme is that populations at the wall are reflected back to where they came from. The half-way-bounce-back-method (HBB) is pictured in figure 3.2: distributions leaving the boundary node meet the surface halfway through the streaming step (at $(t + \frac{\Delta t}{2})$, are reflected back with velocity $\mathbf{c}_j = \mathbf{c}_i$ and arrive at the boundary at the end of the streaming step (at $t + \Delta t$). Therefore, the the HBB boundary scheme is mathematically described by [15]:

$$f_j\left(\mathbf{x}_b, t - \frac{\Delta t}{2}\right) = f_i\left(\mathbf{x}_b, t + \frac{\Delta t}{2}\right) \tag{3.18}$$

in which $\mathbf{x}_b$ is the position of the boundary and subscript j denotes the opposite direction of i. In this thesis, the HBB scheme is only used in chapter 6 for comparison to a wet-node boundary technique.

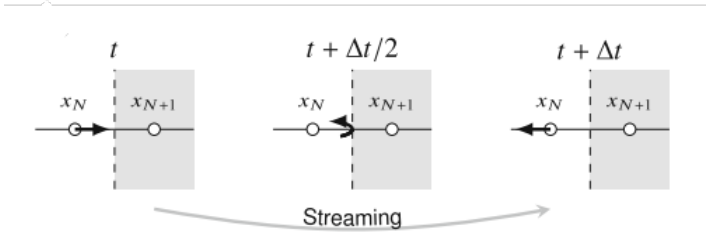


Figure 3.2: Half-way-bounce-back boundary method: distribution is bounced-back over the course of one lattice time step Δt [15].

3.2.2 Wet-Node-Solution-Vector: Dirichlet, Neumann, Conjugate

This research applies a new boundary condition, originally proposed and derived by M. Rohde, that combines a wet-node boundary approach with the Filter-Matrix collision scheme: the wet-node-solution-vector (WNSV) boundary condition. Its main idea is to take advantage of a shared ability of both wet-nodes and the FM-LBM: easy accessibility and control of macroscopic values. For the FM-LBM, this 'easy control' can be easily understood by recalling the form of the solution vectors α_k (3.7), β_k (3.11) and γ_k (3.17) for respectively momentum, temperature and enthalpy. Namely, one can see that its formulation enables to intuitively "switch" (or "filter", hence the name) between the macroscopic- and LB (mesoscopic) space at every point $\mathbf{x}$. Thereby, on the one hand: when the solution vector is known, the local macroscopic quantity in question can be retrieved from the solution vector's elements by $\rho = \alpha_0$, $u_{x,y} = \alpha_{1,2}/\alpha_0$, $T = \beta_0$, $h = \gamma_0$. And the other way around: when a local macroscopic boundary condition is implied, the values can be substituted into its solution vector and filtered back to LB space. The WNSV BC will use this property to easily imply a macroscopic boundary value for a distribution function.

The WNSV BC consists of three steps and works both for different distribution functions (f_i, g_i, m_i) as for different types of boundary conditions (Dirichlet, Neumann, conjugate). In the remainder of this section, the three-step-method of the WNSV BC will first be explained by the example of temperature, for all three BC types. Thereafter, the small changes in the method are presented for it to also apply to enthalpy and momentum.

Temperature boundary conditions

Step1. Initial guess to find $(\beta_k^{i=0})_{extrp}$

First, the known domain-node solution vectors $\beta_k^{(i>0)}$ are extrapolated to find an initial guess of the solution vector at the boundary $(\beta_k^{i=0})_{extrp}$. See figure 3.3 for a visualisation. The extrapolation can be executed with a variation of different schemes, for instance by a 0th order spatial, 1st order spatial, 1st order spatial & temporal, 2nd order spatial or 2nd order spatial & 1st order temporal extrapolation, respectively given by:

$$(\beta_k^{i=0,t})_{extrp} = \begin{cases} \beta_k^{i=1} & \text{(3.19a)} \\ 2\beta_k^{i=1} - \beta_k^{i=2} & \text{(3.19b)} \\ \beta_k^{i=1,t} - \beta_k^{i=1,t-1} + \beta_k^{i=0,t-1} & \text{(3.19c)} \\ \frac{5}{2}\beta_k^{i=1} - 2\beta_k^{i=2} + \frac{1}{2}\beta_k^{i=3} & \text{(3.19d)} \\ \beta_k^{i=0,t-1} - 2\beta_k^{i=1,t-1} + \beta_k^{i=2,t-1} + 2\beta_k^{i=1,t} - \beta_k^{i=2,t} & \text{(3.19e)} \end{cases}$$

in which ubscript i indicates the spatial position aa t the time instance.

Step 2. Imply desired boundary value

Second, the vector elements in $(\beta_k^{i=0})_{extrp}$ are adjusted with the desired macroscopic boundary value to find $\beta_k^{i=0}$. Based on the standard formulation of β_k (3.11) this is done in the following manner:

$$\beta_k^{i=0} = (\beta_k^{i=0})_{extrp} - \begin{bmatrix} T_{extrp}, \\ u_x T_{extrp} \\ u_y T_{extrp} \\ 0,0,0,0,0,0 \end{bmatrix} + \begin{bmatrix} T_{BC}, \\ u_x T_{BC} \\ u_y T_{BC} \\ 0,0,0,0,0,0 \end{bmatrix} \tag{3.20}$$

In which it is defined that $T_{extrp} \equiv (\beta_0^{i=0})_{extrp}$ and T_{BC} is the imposed boundary value, which value depends on the type of imposed boundary condition:

- *Dirichlet* ($T_{i=0} = T_W$) - No additional operation is required, as the temperature value is simply already known:

$$T_{BC} = T_W \tag{3.21}$$

- *Neumann* ($dT/dn|_{i=0} = 0$) - The known flux is transformed into a Dirichlet BC by expanding it in the first- or second order, respectively given by:

$$T_{BC} = \begin{cases} \beta_0^{i=1} & \text{(3.22a)} \\ \frac{4}{3}\beta_0^{i=1} - \frac{1}{3}\beta_0^{i=2} & \text{(3.22b)} \end{cases}$$

  in which it was used that $T^i = \beta_0^i$ (see equation 3.11).

- *Conjugate boundary* (see 2.17, 2.18) - A similar strategy as for the Neumann case can be employed. By writing out expansions of equations 2.17 and 2.18 and combining for T_{BC}, one achieves for respectively first- and second order:

$$T_{BC} = \begin{cases} \frac{1}{\lambda_s + \lambda_l}\left(\lambda_l \beta_{0,l}^{i=1} + \lambda_s \beta_{0,s}^{i=1}\right) & \text{(3.23a)} \\ \frac{1}{3(\lambda_s + \lambda_l)}\left(\lambda_l(-\beta_{0,l}^{i=2} + 4\beta_{0,l}^{i=1}) + \lambda_s(\beta_{0,s}^{i=2} - 4\beta_{0,s}^{i=1})\right) & \text{(3.23b)} \end{cases}$$

  See figure 3.4 for a visualisation of adjacent fluid and solid.

Step 3. Transform back to LB space
The constructed boundary solution vector $\beta_k^{i=0}$ can now be filtered back to the LB space with the the FM matrix multiplication:

$$g_i(x=0,t) = \sum_k \omega_i E_{ik} \beta_k^{i=0} \tag{3.24}$$

Important to note is that this is *not* the same as in the collision matrix multiplication 3.9, as at the boundary no additional (staggered) time-stepping is performed. In fact all three steps have taken place in the same time step.

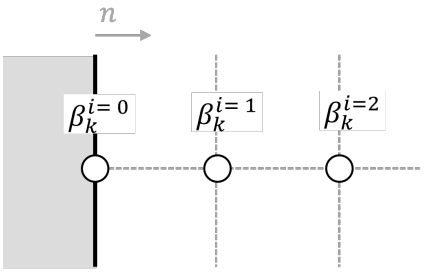


Figure 3.3: Nodes in fluid domain with local temperature solution vector β_k^i

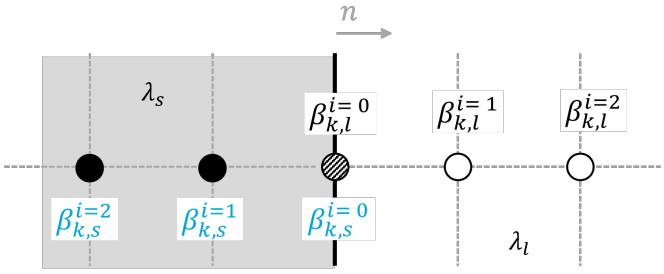


Figure 3.4: Nodes in the solid (grey) and fluid (white) domain, with corresponding temperature solution vectors $\beta_{k,s/l}$ and thermal conductivity $\lambda_{s/l}$

Enthalpy-temperature boundary conditions
The same three steps are performed, but now replacing the temperature properties for enthalpy properties. i.e. solution vector $\beta \longrightarrow \gamma$ in equations 3.19, 3.20, 3.23, 3.24, replacing $T_{extrp} \longrightarrow h_{extrp}$ in equation 3.20 and replacing $g_i \longrightarrow m_i^s$ in equation 3.24.

However, a small alteration is needed in step 2, as the found T_{BC} for an imposed temperature boundary condition (via either equation 3.19, 3.22 or 3.23) needs to be transformed to an enthalpy boundary condition H_{BC} This relation is given by a rewritten form of 2.9:

$$H_{BC} = \begin{cases} c_{p,s} T_{BC} & \text{if } f_L^{i=0} = 0 \\ c_{p,s} T_s + f_L^{i=0} L & \text{if } 0 < f_L^{i=0} < 1 \\ c_{p,s} T_s + c_{p,l}(T_{BC} - T_l) + L & \text{if } f_L^{i=0} = 1 \end{cases} \tag{3.25}$$

Momentum boundary conditions
For momentum, only a no-slip boundary condition (i.e. Dirichlet) BC is employed in this research. The same three-step method is employed for momentum with the obvious replacements: $\beta \longrightarrow \alpha$ in equations 3.19, 3.24 and replacing $g_i \longrightarrow f_i$ in equation 3.24.

hOWEVER step 2 (equation 3.20 is defined a little differently due to the different formulation for the momentum solution vector, while the idea remains exactly the same:

$$(\alpha_k^{i=0})_{BC} = (\alpha_k^{i=0})_{extrp} - \begin{bmatrix} 0 \\ \rho(u_x)_{extrp} \\ \rho(u_y)_{extrp} \\ 3\rho(u_x^2)_{extrp} \\ 3\rho(u_x u_y)_{extrp} \\ \rho(u_y^2)_{extrp} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ (\rho u_x)_{BC} \\ \rho(u_y)_{BC} \\ 3\rho(u_x^2)_{BC} \\ 3\rho(u_x u_y)_{BC} \\ \rho(u_y^2)_{BC} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.26}$$

In which $\mathbf{u}_{BC}$ is only ever given by no-slip boundary conditions in this research.

3.2.3 Enthalpy-Porosity and Immersed-Boundary: Phase Front

When modelling phase change, a boundary condition should be imposed that sets the velocity in the bulk of the ice and on the phase front to zero (i.e. no-slip). As the phase front moves and is typically of a complex shape, implementation of the regular HBB no-slip boundary method (section 3.2.1) would become non accurate and cumbersome. Two methods that have proven to be successful in previous studies for the no-slip BC at the phase front are the enthalpy-porosity method and the immersed-boundary method.

Enthalpy-Porosity

The enthalpy-porosity method introduces a Darcy momentum sink $S_{x,y}$ in the Navier-Stokes equation [18]:

$$\rho\Big(\frac{\partial u_{x,y}}{\partial t} + u_{x,y}\nabla\mathbf{u}\Big) = -\frac{\partial P}{\partial x,y} + \rho\nabla(\nu\nabla u_{x,y}) - S_{x,y}u_{x,y} \tag{3.27}$$

$S_{x,y}$ represents an frictional resistance force per unit mass, given by:

$$S_{x,y} = A_{mush}\frac{(1-f_L)^2}{f_L^3+\sigma_1}. \tag{3.28}$$

A_{mush} is a constant which is commonly set to an universal value of 10^4-10^8 $[kg/m^3/s]$, and σ_1 is a small value which prohibits division by zero (when $f_L = 0$). The formulation of $S_{x,y}$ effectively ensures that in phase-changing cells, the porous medium resistance term dominates over the transient, convective and diffusive effects originating out of molecular interaction mechanisms. On the other hand, in totally solid elements ($f_L = 0$), the term $S_{x,y}$ has a large value, forcing any velocity to zero. In a fully liquid element ($f_L = 1$), the term is zero, and the usual Navier-Stokes equations are retrieved.

Chakraborty & Chatterjee [18] implemented the enthalpy method in a hybrid BGK-LBM. In the MSc thesis by Bus [35], it was successfully implemented in FM-LBM, by proposing a modification to the definition of the momentum solution vector: $\alpha_k^{\pm}$:

$$\alpha_{1,2}^{\pm} = \Big(\rho \mp \frac{\Delta t}{2}S_{x,y}\Big)u_{x,y} \pm \frac{\Delta t}{2}f_{x,y} \tag{3.29}$$

Giving also that velocities can be found by:

$$u_{x,y} = \frac{\alpha_{1,2}^{-} + \frac{\Delta t}{2}f_{x,y}}{\rho + \frac{\Delta t}{2}S_{x,y}} \tag{3.30}$$

Immersed-Boundary

The immersed-boundary method [54] makes uses of an altered collision operator Ω_i^S that is a function of the liquid fraction. In this way, a special post-collision distribution function f_i^S is introduced, which remains unchanged in the liquid phase ($f_L = 1$) but will impose zero velocity in the solid region ($f_L = 0$) and a no-slip on the liquid-solid interface:

$$f_i^S\Big(\mathbf{x}+\frac{\mathbf{c}_i\Delta t}{2}, t+\frac{\Delta t}{2}\Big) - f_i\Big(\mathbf{x}-\frac{\mathbf{c}_i\Delta t}{2}, t-\frac{\Delta t}{2}\Big) = \Big((1-B_f)\Omega_i + B_f\Omega_i^S\Big)\Delta t \tag{3.31}$$

B_f is a weighting factor that depends on the local solid fraction:

$$B_f = \frac{(1-f_L)\sigma_2}{f_L+\sigma_2} \tag{3.32}$$

in which σ_2 is a term that is orignially defined by the dimensionless relaxation time [54], and is taken as a small term to avoid division by zero. The collision term Ω_i is given by its usual formulation (3.1) and the special collision term Ω_i^s is given by:

$$\Omega_i^s = f_j\Big(\mathbf{x}-\frac{\mathbf{c}_i\Delta t}{2}, t-\frac{\Delta t}{2}\Big) - f_i\Big(\mathbf{x}-\frac{\mathbf{c}_i\Delta t}{2}, t-\frac{\Delta t}{2}\Big) - f_j^{eq}(\rho,\mathbf{u}) + f_i^{eq}(\rho,\mathbf{u}_s) \tag{3.33}$$

in which subscript j denotes the opposite direction of i, f^{eq} is given by equation 2.26 and solid velocity $\mathbf{u}_s = \mathbf{0}$.

3.3 Lattice Units

3.3.1 Conversion Parameters

Lattice Boltzmann simulations are mostly performed in dimensionless *lattice units*. The non-dimensional value ϕ^* is obtained by dividing a dimensional physical quantity ϕ by a reference quantity called the *conversion factor* C_ϕ: $\phi^* = \phi/C_\phi$. In this report, the following basic non-dimensionalisations are used:

$$C_L = \frac{\Delta x}{\Delta x^*} = \Delta x, \qquad C_\nu = \frac{\nu}{\nu^*}, \qquad C_\rho = \frac{\rho}{\rho^*} = \rho, \qquad C_T = 1 \tag{3.34}$$

The lattice spacing, lattice time step and reference density are set to unity: $\Delta x^* = \Delta t^* = \rho^* = 1$, which is a common choice within LBM [15]. Note that the temperature is not set to lattice units. Also note that for a link-wise grid C_L is given by $C_L = L_x/N_x$, while for a wet-node grid this is $C_L = L_x/(N_x - 1)$. The four conversion factors as defined in 3.34 from a complete set: all other conversion factors can be derived from them by either using the law of similarity (dimensionless numbers must be identical in all unit systems) or by matching units. This leads to:

$$C_t = \frac{C_L^2}{C_\nu}, \qquad C_\kappa = \frac{C_L^2}{C_t}, \qquad C_g = \frac{C_L}{C_t^2}, \qquad C_{c_p} = \frac{C_L^2}{C_t^2 C_T}, \qquad C_\lambda = \frac{C_L}{C_t^3 C_T}, \qquad C_{Lat} = \frac{C_t^2}{C_L^2} \tag{3.35}$$

Note that in a given system (i.e. all physical parameters are known), only two free lattice parameters of choice are left: the grid size Δx^* and lattice viscosity ν^*. It is crucial to balance these simulation parameters in such a way that a suitable compromise of accuracy, stability and efficiency is achieved. Two stability considerations, based on non-dimensional numbers, are handled in this research and explained in the following two sections.

In the remainder of this thesis, the asteriks (*) notation will be dropped.

3.3.2 Mach number

The lattice velocity has to stay below certain limits within LBM. This originates from a point introduced in section 2.3.2; the LBE does no exactly solve the NSE, but differs from it with an unphysical term of order $\mathcal{O}(u^3)$. This term remains small if the Mach number remains small, i.e. $Ma \equiv v_0/c_s << 1$. When $c_s = \sqrt{1/3}$, this leads to the lattice velocity to have the limit $v_0 < 0.1$ $[ls/lt]$.

When dealing with natural convection, the characteristic velocity v_0 is equal to [34][55].:

$$v_0 = \sqrt{g\alpha\Delta T(N_X - 1)}, \ (ls/lt) \tag{3.36}$$

When initialising a LB simulation, v_0 can be altered to meet the Ma-criterium by choosing a larger N or smaller lattice viscosity ν.

3.3.3 Courant and Péclet number

The Courant number is a common way to analyse stability in CFD aplications. It compares the speed $\Delta x/\Delta t$ at which the information propagates in the model with the physical speed $\mathbf{u}$ at which the fluid field is advected:

$$Co = \frac{\mathbf{u}\Delta t}{\Delta x} \tag{3.37}$$

If $Co > 1$, the simulation cannot propagate the physical solution quickly enough, which tends to make the simulation unstable [15]. By taking $v_0 < 0.1$ into account, this mostly met for flow problems. However, when dealing with *thermal* flow, an equivalent non-dimensional number can be used to control stability: the Péclet number. It is defined as the ratio of the rate of advection of a physical quantity by the flow, to the rate of the (thermal) diffusion of the same quantity driven by a gradient. The Péclet number is defined as [1][15][56]:

$$Pe = \frac{\mathbf{u}\Delta x}{\kappa} \tag{3.38}$$

It indicates the speed u of the advected fluid field compared to the rate at which thermal diffusion $\kappa/\Delta x$ information propagates. Therefore if Pe is too large, the advection dominates and non-physical results can occur.

[1] Note the similarities between the Courant and Péclet number: by substituting penetration theorem for heat $\Delta x = \sqrt{\pi\kappa\Delta t}$ into 3.37, it becomes clear that Péclet can be seen as a 'thermal Courant number'.

Suga [56] performed a stability analysis for advection-diffusion problems for different Pe and Co, in the BGK LBM. Very stable systems were found for $Pe \leq 10$ and $Co \leq 0.4$. When Pe was larger, $10 \leq Pe \leq 100$, stability was recovered most of the times, if also $Co \leq 0.1$. However in these cases larger errors occurred and sometimes oscillating solutions were found. Although these results cannot be directly applied to the current research because BGK-LB was used instead of FM-LB, during this research it was made sure to keep Pe in the order of 10^1.

3.4 Numerical Application: The Algorithm

This section gives a step-by-step overview of FM-LBM wet-node algorithm, thereby combining the equations given in the previous sections of this chapter. Note that the *total* algorithm is given, for the most complex case modelled in this thesis: the combination of convective melting in a fluid and conjugate heat transfer in an adjacent solid. Throughout this research, also less complex cases were modelled. The algorithm for these cases can simply be obtained by leaving out the non-relevant steps.

The algorithm presented here consist of initialization and the iterative scheme. The fluid domain will be referred to as the *PCM* (phase change material) and to the solid domain as the *wall*. The relevant distribution functions in the PCM are momentum distribution function f_i & enthalpy distribution function m_i, for the wall this is temperature distribution function g_i.

Initialization

- PCM: Choose initial macroscopic fluid properties (according to section 3.3) and use them to build α_k^- (equation 3.7), γ_k^- (equation 3.17). f_i^-, m_i^- follow from equation 3.4, 3.15
- Wall: Choose initial macroscopic solid properties (according to section 3.3) and use them to build β_k^- (equation 3.11). g_i^- follows from equation 3.9

Iterative Scheme

1. **Collision in PCM** to calculate post-collision enthalpy function m_i:
   (a) Substract m_i^l from m_i to obtain m_i^s (3.12, 3.13).
   (b) Construct pre-collision solution vector γ_k^- (3.16)
   (c) Extract macroscopic sensible enthalpy h^s from solution vector ($h_s = \gamma_0^-$)
   (d) Update solution vector γ_k^- to γ_k^+ (3.17)
   (e) Calculate post-collision enthalpy distribution function m_i^s (3.15)
   (f) Add m_i^l to g_i^s to retrieve total enthalpy distribution function m_i (3.12, 3.13).
2. **Propagate** m_i on PCM domain
3. **Boundary conditions** on non-shared (i.e. non-conjugate) boundaries in PCM with wet-node-solution-vector method, for Dirichlet and Neumann BCs (section 3.2.2)
4. **Collision in wall** to calculate post-collision temperature function g_i:
   (a) Construct pre-collision vectors β_k^- (3.10)
   (b) Extract macroscopic temperature T from solution vector ($T = \beta_0^-$)
   (c) Update solution vector β_k^- to β_k^+ (3.11)
   (d) Calculate post-collision temperature distribution function g_i (3.9)
5. **Propagate** g_i on wall domain
6. **Boundary conditions** on non-shared (i.e. non-conjugate) boundaries in wall with wet-node-solution-vector method, for Dirichlet and Neumann BCs (section 3.2.2)
7. **Boundary conditions** on shared (i.e. conjugate) boundary by PCM and wall domain (section 3.2.2)
8. **Update macroscopic quantities** H (2.31), f_L (2.10), c_p, λ (2.11) from m_i.
9. **Collision in PCM** to calculate post-collision density distribution function f_i:

(a) Construct pre-collision vector α_k^- (3.5)
(b) Update buoyancy force f_y (2.13)
(c) (*Option 1 phase front*) Update momentum sink term $S_{x,y}$ (3.28) to apply the enthalpy-porosity method.
(d) Extract macrsoscopic density ρ and $\mathbf{u}$ from solution vector ($\rho = \alpha_0^-$, $u_{x,y}$ with 3.30)
(e) Update solution vector α_k^- to α_k^+ (3.7)
(f) Calculate post-collision density distribution function f_i (3.4)
(g) (*Option 2 phase front*) Update B (3.32) and Ω_i^s(3.33) and calculate the post-IBM density distribution function (3.31)

10. **Propagate** f_i on PCM domain

11. **Boundary conditions** on all boundaries with wet-node-solution-vector method for no-slip (Dirichlet, section 3.2.2

12. **Update quantities** ρ (2.24), $\mathbf{u}$ (2.25) from f_i.

13. **Return** to first step for new time iteration. Repeat until at desired time-step (transient case) or until a quantity $\phi = T\mathbf{u}$ has met **convergence** criterion δ_ϕ (steady-state):

$$\sqrt{\frac{\sum_i(\phi(\mathbf{x_i},t) - \phi(\mathbf{x_i},t-1000\Delta t))^2}{\sum_i(\phi(\mathbf{x_i},t-1000\Delta t)^2}} < \delta_\phi \tag{3.39}$$

3.5 Numerical Verification Methods

This section moves away from the Lattice Boltzmann method, to the more general topic of numerical verification methods. When computational simulations are employed to model a physical situation, it is essential to build confidence in its correctness via verification and validation. Where validation deals with physics and addresses the appropriateness of the model in reproducing experimental data, verification deals with mathematics and addresses the correctness of the numerical solution to a given model [57]. Or stated in the words of Blottner [58]: validation is about ”solving the right equations” while verification is about ”solving the equations right”.

Within verification, the presence of numerical errors will always arise in CFD models and they can consist of three different types: (1) round-off errors, stemming from the finite precision of computers, (2) the iterative error, due to the non-linearity of mathematical equation, (3) the discretization (or truncation) error, a consequence of the approximations made to transform the partial differential equations into a discrete form. Generally, the discretization error is the major error source in a Lattice Boltzmann simulation [15]. Luckily, the relative importance of the discretization error decreases with grid refinement [59].

Three different methods for verifying the codes in this thesis are used. Given by, in order of increasing rigor [15][60]:

- **Error quantification**. The computational solution of a variable ϕ is compared to a known reference solution ϕ_{ref}, which can consist of an analytical solution or an other computational benchmark. In this research, the so-called L_2-error norm is used:

$$\epsilon_\phi = \sqrt{\frac{\sum_i(\phi_i - \phi_{i,ref})^2}{\sum_i(\phi_{i,ref})^2}} \tag{3.40}$$

- **Grid convergence**. As the lattice grid size is reduced, the discrete form of a numerical method should be able to better approach the continuous PDEs. This can be assessed by computing the obtained error for a series of different grid sizes. Upon decreasing the grid size, the outcome of the numerical model should convergence towards an asymptotic value.

- **Order of accuracy**. The rate at which the discretization error tends to zero shows the related order of accuracy. For example, a second-order-accurate model would show the error to scale with Δx^2.

While the methods for assessing error quantification and convergence are quite straight-forward, the order of accuracy sometimes needs a more delicate approach. The following section elaborates a bit more on the order of accuracy.

3.5.1 Order of Accuracy

The order of accuracy is linked to the discretizaton error via an equation that finds its origin in Richardson extrapolation. It is quantified by simply taking the difference between the numerical estimate ϕ_i (with i referring to the grid refinement level) and a reference solution ϕ_0 [61][62]:

$$\epsilon_{\phi_i} \approx \phi_i - \phi_0 = K\Delta x_i^p + \mathcal{O}(\Delta x_i^{p+1}) \tag{3.41}$$

in which Δx_i the typical cell size for the chosen refinement level,K a constant value, p the observed order of accuracy and $\mathcal{O}(\Delta x_i^{p+1})$ the higher-order error terms.

When Δx_i is sufficiently small, the higher order terms are negligible compared to the Δx^p-term and the solutions are in the so-called *asymptotic range of convergence* [63]. Ignoring the $\mathcal{O}(\Delta x_i^{p+1})$-term in equation 3.41 and subsequently taking the logarithm of both sides, the following equation is obtained:

$$\log(\epsilon_{\phi_i}) = p \cdot \log(\Delta x_i) + \log(K) \tag{3.42}$$

This formulation leads to a method to determine the observed order of accuracy p: by plotting a range $\log(\epsilon_{\phi_i})$ versus $\log(\Delta x_i)$ a linear relation should be found and the corresponding slope represents p. It should be noted that sensible results can only be obtained when a suitable choice for reference solution ϕ_0 is made. This research uses three different methods to choose ϕ_0 and thereby obtain the order of accuracy p:

- **Exact solution**. When a system is modelled of which the exact solution is known, the reference solution can be simply taken equal to this value: $\phi_0 = \phi_{exact}$. When employing a constant grid refinement ratio r ($r = \Delta x_2/\Delta x_1 = \Delta x_3/\Delta x_2 =$ .. etc), a minimum of two grid solutions ϕ_i is needed to calculate p. Unfortunately, for many problems an exact solution is not available.

- **Finest-grid solution**. Another commonly used method is to approximate the exact solution with a finest-grid solution: $\phi_0 = \phi_{fine}$. Note that caution should be taken when choosing a suitable finest-grid-solution, as the grid refinement ratio $r = \Delta_{fine-1}/\Delta_{fine}$ should not be too small. In other words, the finest-grid i_{fine} should not be too close to the second-finest value. However, employing a high r may come at an impractically high computational cost.

- **Eça-Hoekstra Method (EHM)**. Sometimes, an analytical solution is unavailable and a finest-grid solution lies beyond a practical application. Eça & Hoekstra [59] proposed a linear-least-squares method to work around this limitation. The idea is to define a function S that determines the error norm of the observed value ϕ and its predicted value ϕ_0, following the definition in 3.41. S is given by:

$$S(K,\, \phi_0,\, p) = \sqrt{\sum_{i=1}^{n} \left(\phi_i - (\phi_0 + K\Delta x_i^p)\right)^2}, \tag{3.43}$$

  By minimizing the function of S via $\frac{\partial S}{\partial K} = \frac{\partial S}{\partial \phi_0} = \frac{\partial S}{\partial p} = 0$, the following system of non-linear equations is found:

$$\begin{cases} K = \dfrac{\sum_{i=1}^{n} \phi_i \Delta x_i^p - \left(\sum_{i=1}^{n} \phi_i\right)\left(\sum_{i=1}^{n} \Delta x_i^p\right)}{\sum_{i=1}^{n} \phi_i \Delta x_i^{2p} - \left(\sum_{i=1}^{n} \Delta x_i^p\right)\left(\sum_{i=1}^{n} \Delta x_i^p\right)} \\ \phi_0 = \sum_{i=1}^{n} \phi_i - K \sum_{i=1}^{n} \Delta x_i^p \\ \sum_{i=1}^{n} \phi_i \Delta x_i^p \log(\Delta x_i) - \phi_0 \sum_{i=1}^{n} \Delta x_i^p \log(\Delta x_i) - K \sum_{i=1}^{n} h_i^{2p} \log(\Delta x_i) = 0 \end{cases} \tag{3.44}$$

  When these equations are solved with an iterative scheme, estimations for ϕ_0, K, p can be found, with the minimum amount of inputs being $n = 4$.

Chapter 4

Benchmark Results

The goal of this research is to correctly apply the Filter-Matrix lattice Boltzmann method with wet-node boundary conditions in simulating the combination between three phenomena in the freeze-plug: conjugate heat transfer, phase change and natural convection. This goal is subdivided into three parts. The first part aims to justify the implementation of the three phenomena separately. In the second part, these three phenomena are combined into one model simulating the melting freeze-plug. The third part focuses specifically on the wet-node-solution-vector boundary condition method, and its performance in terms of order of accuracy, time convergence and energy conservation.

This chapter focuses on the first part of the results: the three phenomena are justified by comparing their results to a number of literature benchmark studies. In section 4.1 a natural convection flow is benchmarked by a standard case of a side-heated cavity. The addition of conjugate heat transfer is simulated and benchmarked in section 4.2 by adding a conducting wall to the cavity. Lastly, section 4.3 shows the results of the combination of phase change (melting) and natural convection, by modelling a side-heated cavity filled with a phase-change material.

4.1 Natural Convection

By using the FM-LBM for the simulation of a natural convection flow, the implementation of the coupling between the velocity- and temperature field is checked, as well as the ability of the wet-node-solution-vector BC to simulate Dirichlet and Neumann boundary conditions. The modelling of natural convection in a square cavity is a common benchmarking case within the field of computational fluid dynamics and two different numerical studies are used in this work for benchmarking: (1) de Vahl Davis [64] who used a traditional finite difference method, and Zhuo & Zhong [33] who employed a FM-LBM.

4.1.1 Geometry and Input Parameters

The modelled geometry is shown in figure 4.1. It consist of a two-dimensional square cavity of size H x H filled with air and at initial temperature T_0. At $t > 0$, the temperature vertical walls is set to a cold temperature $T_C < T_0$ on the left side and a hot temperature $T_H > T_0$ on the right side. The top and bottom walls are adiabatic and a no-slip boundary condition for velocity is implied on all four boundaries. The simulation is initiated by a zero velocity and T_0 in the fluid domain. Driven by the buoyancy force, the velocity- and temperature profile will transition to a steady-state.

The physical and numerical (lattice) parameters used in the model are given in table 4.1. The behaviour of a natural convection flow is uniquely characterized by the non-dimensional numbers Pr and Ra, as mentioned before in section 2.2.1. Pr is chosen at its characteristic value for air. Three different values for Ra were simulated to test the model's sensitivity to different velocities. The latter is done in the laminar regime, i.e. $Ra \leq 10^6$ [65]. Ra is implied through varying the temperature difference ΔT between the hot and cold wall via the definition of Ra (equation 2.14). The lattice viscosity ν is chosen for each of the three Rayleigh numbers, such that characteristic velocity of the flow (equation 3.36) ensures the model to run in the incompressible regime. All other input variables can be chosen based on the definitions of Ra and Pr and the conversion factors as formulated in equations 3.34, 3.35. All thermophysical

properties are assumed to be independent of temperature.

The wet-node-solution-vector boundary condition (section 3.2.2) was used for all boundary conditions, with the use of first order extrapolation schemes (equation 3.19b, 3.22a). Lastly, whether the problem has reached the steady-state situation is checked by the residual value of the temperature and velocity as defined in equation 3.39, with $\delta_{T,\mathbf{u}} = 10^{-6}$ [33].

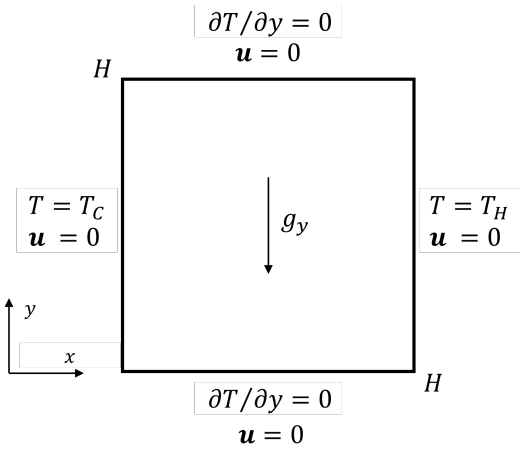


Figure 4.1: The geometry of the H x H cavity natural convection case, with isothermal vertical walls, adiabatic horizontal walls and no-slip on all boundaries.

Description	Variable	Physical value	Unit	Lattice value	Unit
Height & width cavity	H x H	0.001 x 0.001	$[m]$	$(N-1)$ x $(N-1)$	$[ls]$
Prandtl number	Pr	0.7	-	0.7	-
Rayleigh number	Ra	$10^4, 10^5, 10^6$	-	$10^4, 10^5, 10^6$	-
Kinematic viscosity	ν	$8.56 \cdot 10^{-7}$	$[m^2/s]$	$\frac{1}{100}, \frac{1}{100}, \frac{1}{150}$	$[ls^2/lt]$
Initial temperature	T_0	1	$[K]$	1	$[K]$
Wall temperature	$T_{H,C}$	$T_0 \pm \Delta T/2$	$[K]$	$T_0 \pm \Delta T/2$	$[K]$
Density	ρ	1.293	$[kg/m^3]$	1.0	$[ls^{-3}]$
Gravitational acceleration	g	9.81	$[m/s^2]$	$9.81/C_g$	$[ls/lt^2]$
Thermal expansion coefficient	α	$2.07 \cdot 10^{-4}$	$[K^{-1}]$	$2.07 \cdot 10^{-4}$	$[K^{-1}]$
Thermal diffusivity	κ	$1.23 \cdot 10^{-6}$	$[m^2/s]$	$1.23 \cdot 10^{-6}/C_\kappa$	$[ls^2/lt]$

Table 4.1: Definition of physical- and lattice parameters for simulation of natural convection of air. Three different lattice viscosities correspond to the respective three Rayleigh numbers. Note that the problem is uniquely characterised by the parameters above the dashed line, the ones below are provided for completeness. All conversion parameters C are specified in 3.34, 3.35.

4.1.2 Heat Transfer and Velocity Results

The modelled temperature- and velocity fields in the square cavity for $Ra = 10^4, 10^5, 10^6$ are shown in figure 4.3. The results are physically sensible: the fluid close to the right wall is heated, flows upward, and travels along the adiabatic top wall to the cold wall, where it is cooled and thereby streams back downwards. As Ra increases, the temperature gradients near the vertical walls become larger, while in the middle of the domain the flow is quasi-motionless. Overall, this result serves as a qualitative benchmark of the applied method for natural convection. Ofcourse, the next step is to perform a quantitative benchmark

Grid Dependence

Before comparing the modelled results to the two benchmark studies, a grid dependence study is performed: the grid is refined (i.e. the amount of lattice points N x N increases) until the results stay approximately the same upon further refinement. The numerical output value that is used, is the average Nusselt Nu_0 along the cool wall (i.e. at $x = 0$), serving as a characterisation of the heat transfer.

Following [32][33], Nu_0 is given by[1]:

$$Nu_0 = -\frac{1}{\Delta T}\int_0^H \left(\frac{\partial T}{\partial x}\right)_{x=0} dy \tag{4.1}$$

The integral in this equation is computed numerically using Simpson's rule. Furthermore, the gradient $\partial T/\partial x$ can be directly calculated from the local value of the solution vector $\beta_1(x=0,y)$ via its formulation 3.11. This approach is preferred over calculating the gradient with a finite-difference method, as the latter could introduce a new discretization error into the result.

The results from the grid dependence study for Nu_0 for the three different Rayleigh numbers are shown in figure 4.2. For all three Rayleigh numbers, the figures show converging behaviour towards an asymptotic value for Nu_0 as the amount of grid points N x N increases. The threshold to indicate grid convergence was set at $(Nu(N+20) - Nu(N))/Nu(N) < 0.1\%$. For $Ra = 10^4, 10^5, 10^6$ this was achieved at respectively $N = 181, 181, 221$. It should be noted that it is somewhat surprising that grid convergence for $Ra = 10^4, 10^5$ is achieved at the same grid size, as it is expected that a larger Ra needs a larger amount of grid points because of increased (small-scale) complexities in the temperature- and velocity field. Overall, the grid convergence is a first indication of the correct implementation of the numerical model, but this will be further substantiated by comparing the results of the finest-grid solution to numerical benchmark studies.

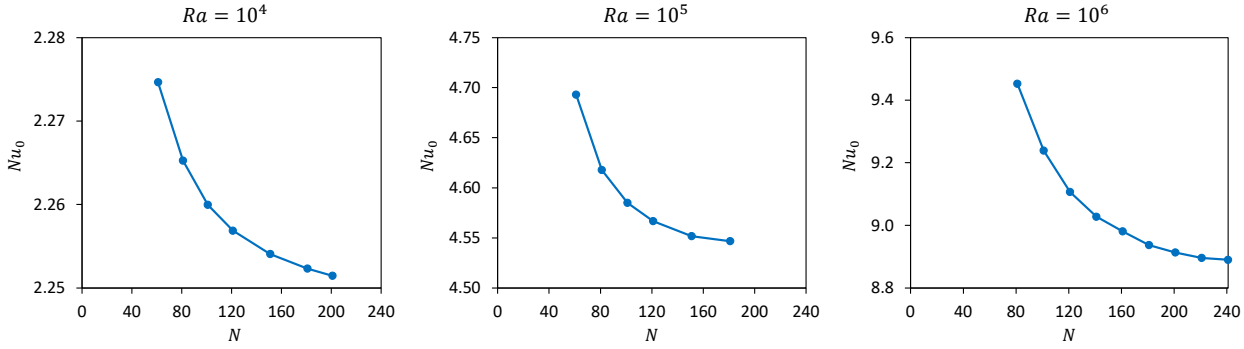


Figure 4.2: Grid dependence study for three different Rayleigh numbers Ra: the output average Nusselt number Nu_0 for increasing amount of grid points (i.e. decreasing grid size $1/\Delta x = H/(N-1)$

Error Quantification

In addition to the characteristic Nu_0, four other numerical values were used to benchmark the results: the velocity component along the vertical central line $(u_x)_{max}$ and its location y_{max} & the maximum vertical velocity component along the horizontal central line $(u_y)_{max}$ and its location x_{max}. Following [33][64], the velocity components and locations are scaled in the following way, enabling a generalised comparison with the benchmark values:

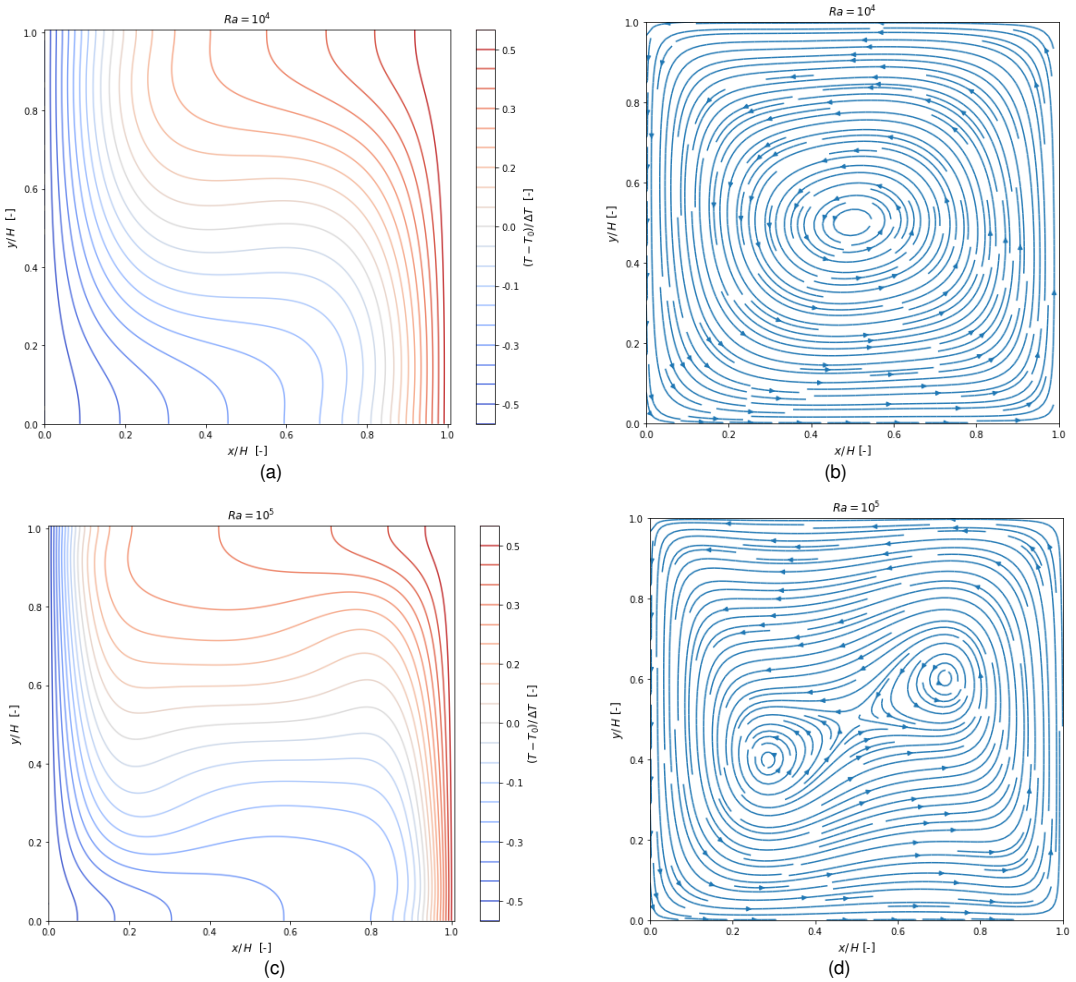
$$\bar{\mathbf{u}} = \frac{\mathbf{u}(N-1)}{\kappa}, \qquad \bar{\mathbf{x}} = \frac{\mathbf{x}}{H} \tag{4.2}$$

The results on the finest grid for the five characteristic numerical values Nu, $(u_x)_{max}$, y_{max}, $(u_y)_{max}$, y_{max} and their errors to the benchmark results from Zhuo & Zhong [33] and by De Vahl Davis [64] are given in table 4.2, for three values of Ra. It can be observed that all characteristic values show a small error under 0.9% with the benchmarks, except for y_{max}, which shows an error $> 1.0\%$. The latter is explained by the discrete nature of the coordinate: it has an error margin of $\pm 1/(N-1) \approx \pm 0.005$. This equals an $\pm 2.8\%$ error to the benchmark value, which is in the range of the observed errors. Another observation is that for larger Ra, the overall the errors increase somewhat (max $0.5\,\%pt$). However this is still within an acceptable margin, and the errors are expected to decline even further for an even finer grid. To conclude, the grid convergence and small errors to the two benchmarks indicates that the developed FM-LBM with WNSV BCs performs well in modelling coupled velocity- and thermal flows with Dirichlet- and Neumann boundary conditions, in the laminar regime.

[1] See appendix A for nondimensionalisation

	Study	Nu_0	$\bar{u}_{x,max}$	$\bar{y}_{max}$	$\bar{u}_{y,max}$	$\bar{x}_{max}$
$Ra = 10^4$	[64]	2.238	16.178	0.177	19.617	0.881
	[33]	2.245	16.183	0.178	19.627	0.882
	Current	2.232	16.189	0.175	19.631	0.880
	Error [%]	0.27 - 0.58	0.07 - 0.04	1.13 - 1.17	0.07 - 0.02	0.01 - 0.02
$Ra = 10^5$	[64]	4.509	34.73	0.145	68.59	0.934
	[33]	4.521	34.74	0.144	68.62	0.935
	Current	4.543	34.74	0.147	68.55	0.933
	Error [%]	0.75- 0.49	0.00 - 0.03	1.38 - 2.08	0.06 - 0.10	0.11 - 0.21
$Ra = 10^6$	[64]	8.817	64.63	0.150	219.36	0.962
	[33]	8.819	64.91	0.148	220.20	0.960
	Current	8.890	65.05	0.150	220.54	0.963
	Error [%]	0.83 - 0.80	0.65 - 0.22	0.00 - 1.33	0.54 - 0.15	0.10 - 0.31

Table 4.2: Results of current study for numerical characteristic values of natural convection flow, for three different input Rayleigh numbers, compared with benchmark values from De Vahl Davis [64] and Zhuo & Zhong [33]. Note that the two benchmark studies used a geometry that is mirrored in the y-axis with respect to the current geometry; i.e. the hot & cold wall are reversed. Therefore their values for $\bar{\mathbf{x}}_{max}$ were mirrored $(1 - \bar{\mathbf{x}}_{max})$ for comparison. Also note that [33] used two different methods in their study; when the results differed, the average was taken.



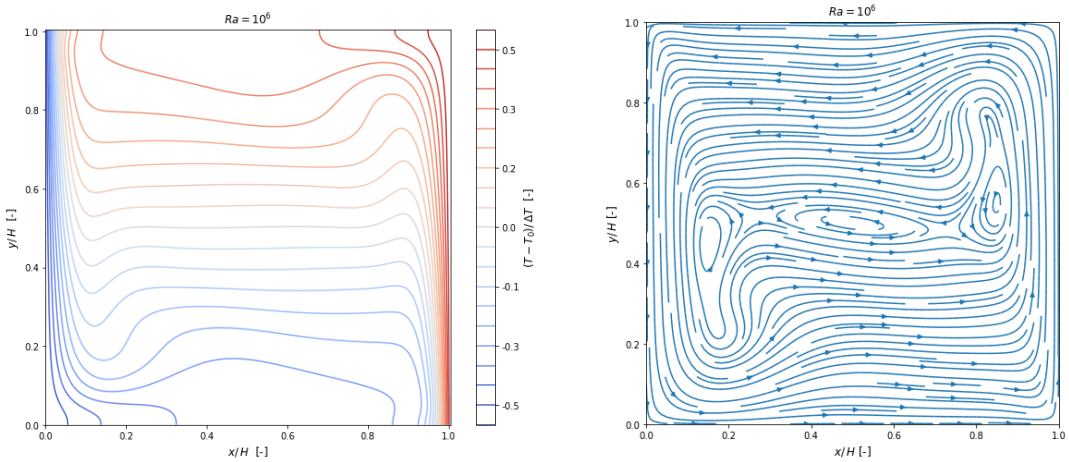


Figure 4.3: Results for natural convection in side-heated cavity, for $Ra = 10^4, 10^5, 10^6$. Left figures show temperature isotherms, right figures velocity streamlines.

4.2 Conjugate Heat Transfer

The second benchmarking case has the goal to correctly use the proposed FM-LBM with WNSV BCs to model *conjugate* thermal boundary conditions. To this end, the geometry of the previous benchmark in section 4.1 is extended by adding a solid, conducting wall. In the context of the freeze-plug, the conducting wall imitates the pipe wall structure in which the freeze-plug is positioned. Two numerical studies are used to benchmark the results: (1) a finite-difference model by Kaminksky & Prakash [66] and (2) a finite-element model by Misra & Sarkar [67].

4.2.1 Geometry and Input Parameters

The modelled geometry is shown in figure 4.4: it is an extension of the previous geometry (figure 4.1), but now one of the vertical walls of the square cavity is of non-zero thickness d and consists of a solid, conducting material. The right edge of the solid wall is kept at hot temperature T_H, the left wall of the cavity is at cold temperature T_C, and all other outer boundaries are adiabatic. Every boundary of the fluid domain has a no-slip condition for velocity. The algorithm treats the fluid- and solid domains as two different numerical domains, interacting with each other at the shared boundary via the conjugate thermal boundary conditions (equations 2.17, 2.18). All boundary conditions are implied with the WNSV BC with first order extrapolation schemes (equations 3.19b, 3.22a, 3.23a). The solid wall has an higher thermal diffusivity than the fluid, i.e. ratio $k_\lambda \equiv \lambda_s/\lambda_f > 1$. The thermal diffusivity ratio is assumed to have the same ratio $k_\lambda = k_\kappa \equiv \kappa_s/\kappa_f$, which is equivalent to stating that the heat capacitance $c_p\rho$ is identical in both domains, as the thermal diffusivity is defined by $\kappa = \lambda/(\rho c_p)$. This assumption is sensible since the problem is steady-state, for which only the thermal conductivity plays a role in the temperature distribution, and the heat capacitance of the fluid & solid is not relevant.

Most input parameters were taken identical to the previous case (table 4.1), additional parameters are given in table table 4.3. Three different Rayleigh numbers are tested and ν is again set accordingly to ensure an incompressible regime. Ra is implied through setting tge temperature difference ΔT via the definition of Ra (equation 2.14). Next to the Rayleigh- and Prandtl number, the problem is uniquely defined by thermal conductivity ratio k_λ and wall thickness ratio d/H. For the latter two, a value was chosen that is in the order of magnitude of the freeze-plug set-up and that was available in the two used benchmarks [66] & [67]. All corresponding lattice values can be calculated via the usual conversion parameters in equation 3.34 and 3.35. Again, all thermo-physical quantities were assumed to be independent of temperature and the time-convergence criteria 3.39 with $\delta_{\mathbf{u},T} = 10^{-6}$ were employed.

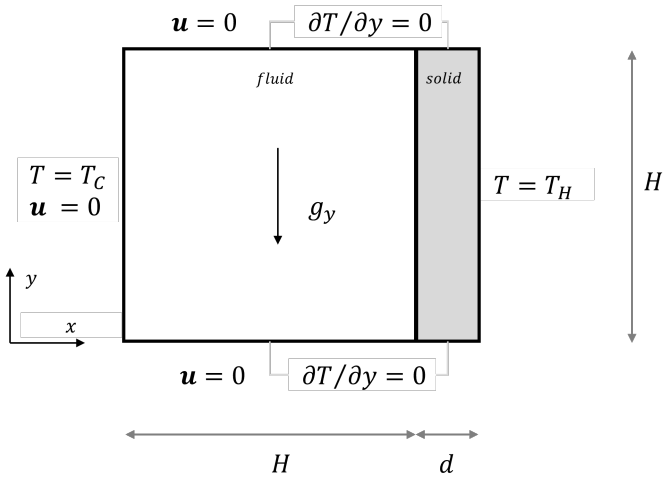


Figure 4.4: Geometry conjugate heat transfer: cavity filled with fluid and adjacent solid wall.

Description	Variable	Physical Value	Unit	Lattice Value	Unit
Wall thickness ratio	d/H	0.2	-	0.2	-
Rayleigh number	Ra	$10^4, 7 \cdot 10^4, 7 \cdot 10^5$	-	$10^4, 7 \cdot 10^4, 7 \cdot 10^5$	-
Kinematic viscosity	ν			$1/100,\ 1/100,\ 1/150$	$[ls^2/lt]$
Ratio solid/liquid λ, κ	$\lambda_s/\lambda_f = \kappa_s/\kappa_f$	5	-	5	-

Table 4.3: Definition of parameters and non-dimensional numbers used in the simulation of conjugate heat transfer and natural convection of air, additional to used values given in table 4.1. Three different lattice viscosity's correspond to the respective three Rayleigh numbers.

4.2.2 Heat Transfer and Velocity results

The modelled temperature- and velocity results for $Ra = 10^4, 7 \cdot 10^5, 7 \cdot 10^6$ are shown in figure 4.6 and show a similar counter-clockwise flow pattern as the previous case with the thin-walled cavity (figure 4.3). It can be observed that for a lower Rayleigh number the isotherms in the fluid near the wall are more parallel to each other, which is due to the conduction-dominated thermal flow. As the Rayleigh number increases, convection plays a larger role, the flow field becomes more asymmetric and a non-uniform heat flux in the y-direction along the solid-fluid interface is observed. All in all, the results indicate good physical behaviour, but must ofcourse be further validated with the benchmark studies.

Grid Dependence

A grid dependence study is carried out: the characteristic Nusselt number Nu_H along the hot wall is calculated for different amount of used gridpoints N. Nu_H is calculated via 4.1, but replacing $x = H$. The results for the three different Rayleigh numbers are shown in figure 4.5. All three cases show converging behaviour towards a characteristic Nusselt number as the grid is refined.

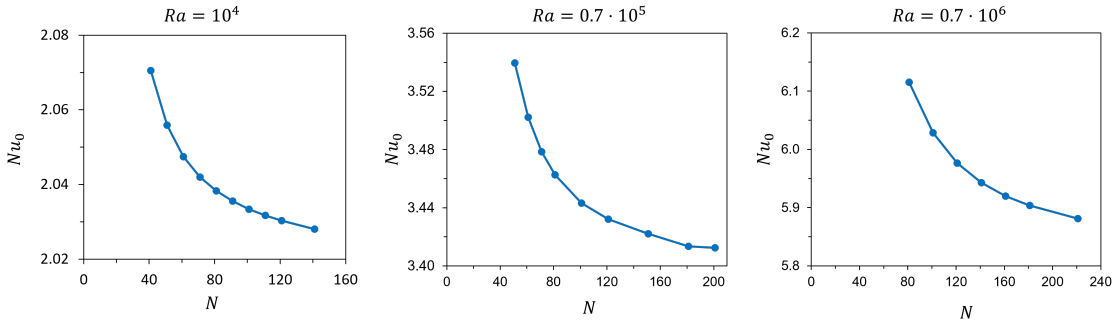


Figure 4.5: Grid convergence for Nusselt number Nu while grid size N increases, for three different Rayleigh numbers.

Error Quantification

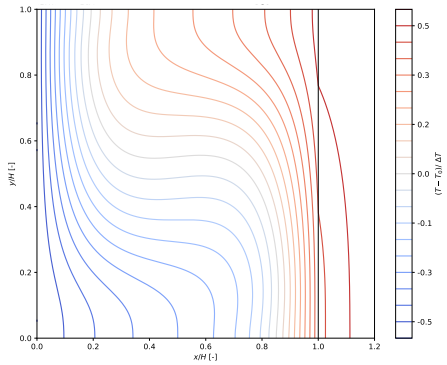
The results are benchmarked by the output Nusselt number Nu_H, given in table 4.4. The obtained Nusselt numbers all show an acceptable error of $< 1.0\%$ with respect to the benchmark studies by Kaminsky & Prakash [66] and Misra et al. [67]. Another interesting point is that the two benchmarks both model the solid- and liquid domains as one computational domain, assigning a very high viscosity in the solid region, thereby solving the thermal flow in the fluid cavity and conduction wall simultaneously, and automatically satisfying the conjugate boundary conditions. Whereas the current model treats the fluid and wall on separate domains and links them through the WNSV BC, as also previously introduced in the algorithm lay-out (section 3.4). Finding the same results for these different numerical treatments actually strengthens the verification.

Unfortunately, characteristic values on the velocity field ($\mathbf{u}_{max}$ and $\mathbf{x}_{max}$) were not provided in the two benchmark studies, hence they were not used in the comparison. Nevertheless, this does not undermine the verfication of the code, as compared to the previous benchmark (section 4.1) the only addition is complexity in heat transfer. The Nusselt number captures this complexity. All characteristics and boundary conditions of the velocity field have remained the same.

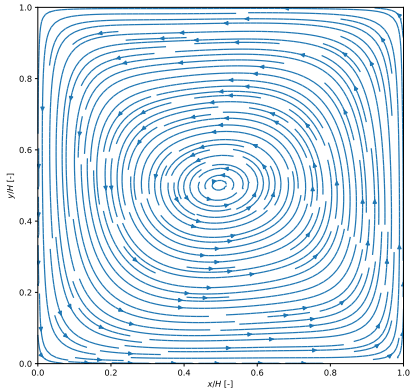
In conclusion, the WNSV BC successfully modelled conjugate heat transfer, in combination with natural convection. Different numerical aspects of the WNSV BC for this case will be studied in more detail in chapter 6.

	Nu_H			
	[66]	[67]	Current	Error
$Ra = 10^4$	-	2.0213	2.025	0.18 %
$Ra = 0.7 \cdot 10^5$	3.42	3.436	3.423	0.09 - 0.38 %
$Ra = 0.7 \cdot 10^6$	5.89	5.910	5.857	0.56 - 0.90 %

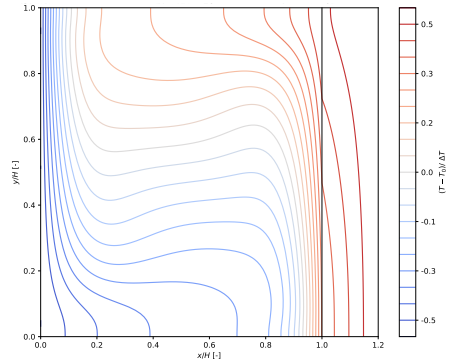
Table 4.4: Results of current study for average Nusselt number along the right, hot cavity wall (Nu_H), for three different input Rayleigh numbers, compared with benchmark values from Kaminsky & Prakash [66] and Misra et. al [67].



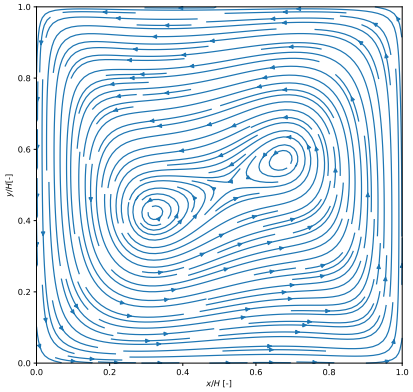
(a) $Ra = 10^4$



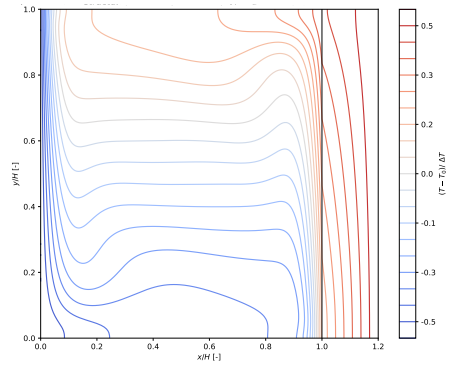
(b) $Ra = 10^4$



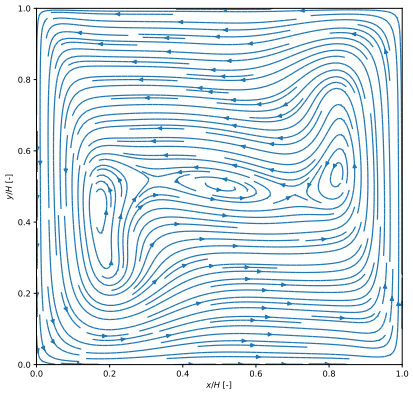
(c) $Ra = 0.7 \cdot 10^5$



(d) $Ra = 0.7 \cdot 10^5$



(e) $Ra = 0.7 \cdot 10^6$



(f) $Ra = 0.7 \cdot 10^6$

Figure 4.6: Results for natural convection in side-heated square cavity with conjugate heat transfer for 3 Rayleigh numbers. Left figures show temperature field with isotherms, black vertical line is the liquid-solid interface. Right figures show velocity field with streamlines in the liquid domain.

4.3 Convective Melting

This section describes the third benchmark case: the combination of phase change (melting) and natural convection. The results are compared with numerical benchmark studies by Huber [16] and Jourabian [68]. These two studies also used an LBM, but with slightly different numerical implementations than in the current model. Namely, they solved collision with the BGK-scheme, treated phase change as a source term in the thermal LB equation (i.e. the *source-based enthalpy method*) and imposed the no-slip condition on the phase-front was with a bounce-back on the solid nodes. In theory, a different numerical model should ofcourse lead to the same physical results.

4.3.1 Geometry and Input Parameters

The geometry consist of a 2D square cavity filled with an initially frozen fluid ($T = T_0 = T_m$). At $t > 0$ the left boundary is set at a hot temperature $T_H > T_0$ and the other three boundaries are adiabatic in their normal directions. All boundaries are no-slip for velocity. The WNSV BC is used to imply all outer boundary conditions, with first order extrapolation schemes (equations 3.19b, 3.22a). The phase front no-slip BC was handled with the immersed boundary method. The used input parameters are given in table 4.5. Two different Rayleigh numbers are tested, and the lattice viscosity ν will be modified accordingly to ensure to be operating in the in compressible regime (equation 3.36). Ra is implied through setting tge temperature difference ΔT via the definition of Ra (equation 2.14). It is assumed that the properties of the solid and liquid phases are constant for temperature, i.e. $\kappa_s = \kappa_l$, $\lambda_s = \lambda_l$, $c_{p,s} = c_{p,l}$.

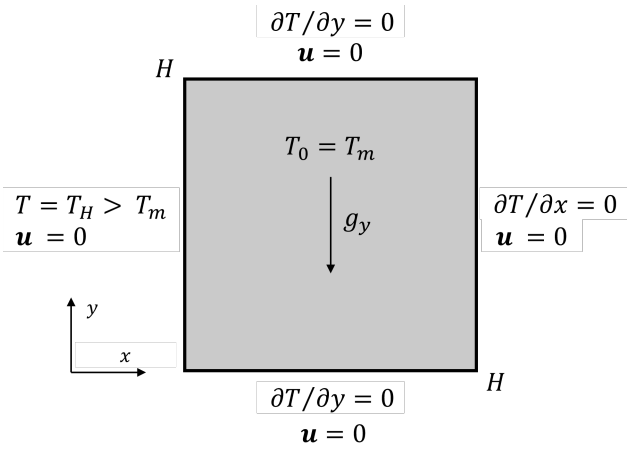


Figure 4.7: Geometry of square cavity filled with phase change material, initially solid.

Description	Variable	Physical value	Unit	Lattice Value	Unit
Height x Width cavity	H x H	0.6 x 0.6	$[m]$	$(N-1)$ x $(N-1)$	$[ls]$
Prandtl number	Pr	1.0	-	1.0	-
Stefan number	Ste	10	-	10	-
Rayleigh number	Ra	$5 \cdot 10^4$, $1.7 \cdot 10^5$	-	$5 \cdot 10^4$, $1.7 \cdot 10^5$	-
Kinematic viscosity	ν	0.0075	$[m^2/s]$	$1/45$, $1/80$	$[ls^2/lt]$
Wall temperature	T_H	$T_0 + \Delta T$	$[K]$	$T_0 + \Delta T$	$[K]$
Initial temperature	T_0	301.14	$[K]$	301.14	$[K]$
Solidifying temperature	T_s	301.14	$[K]$	301.14	$[K]$
Melting temperature	T_l	$T_s + 0.002$	$[K]$	$T_s + 0.002$	$[K]$
Thermal expansion coeff.	α	$2.5 \cdot 10^{-4}$	$[K^{-1}]$	$2.5 \cdot 10^{-4}$	$[K^{-1}]$
Thermal conductivity	λ	1.5	$[W/m/K]$	1.5 / C_λ	$[ls/lt^3/K]$
Thermal diffusivity	κ	$7.5 \cdot 10^{-3}$	$[m^2/s]$	$7.5 \cdot 10^{-3}/C_\kappa$	$[ls^2/lt]$
Specific heat	c_p	0.0456	$[Ws/kg/K]$	$0.0456/C_{c_p}$	$[ls^2/lt^2/K]$
Latent heat	L	$159 \cdot 10^3$	$[Ws/kg]$	$159 \cdot 10^3/C_{Lat}$	$[ls^2/lt^2]$
Gravitational acceleration	g	9.81	$[m/s^2]$	$9.81/C_g$	$[ls/lt^2]$

Table 4.5: Input parameters for convective melting case. Two different lattice viscosities ν correspond to two different Rayleigh numbers. Note that the problem is uniquely characterised by the parameters above the dashed line, the ones below are provided for completeness. All conversion parameters C are specified in 3.34, 3.35.

The problem is now transient instead of steady-state. Generalised comparison in time with the benchmark studies therefore requires the introduction of non-dimensional time θ, which is a product of the Stefan (equation 2.16) and Fourier number:

$$\theta = Fo \cdot Ste, \qquad Fo = \frac{\kappa t}{(N-1)^2} \tag{4.3}$$

4.3.2 Heat Transfer and Phase Front Results

The results for convective melting are visualised in the Appendix for $Ra = 5 \cdot 10^4$ (figure B.1) and $Ra = 1.7 \cdot 10^5$ (figure B.2): the left figures show the isotherms for the temperature distribution, the middle figures the liquid fraction ($f_L = 0$ for solid and $f_L = 1$ for liquid) and the right figures the streamlines of the velocity field. Three different non-dimensional time instances θ are displayed, showing the different melting regimes. Like introduced in section 2.2.2, the problem starts with a conduction regime in which the melting front is almost parallel to the wall, then transitions into a mixed regime of both conduction & convection, and lastly into a convection dominated regime. Also, comparison between the results for the two Rayleigh numbers shows the expected result: natural convection and thus melting is enhanced as the Rayleigh number is increased.

A quantitative benchmark is performed by looking at the time evolution of two characteristic values: the average phase front position $s_{av}(\theta)$ and Nusselt number and $Nu_0(\theta)$. The first is defined by equation 4.4, the latter by the usual definition given in equation 4.1.

$$s_{av}(\theta) = \frac{1}{H} \int_0^H x_{front}(y, \theta)\, dy \tag{4.4}$$

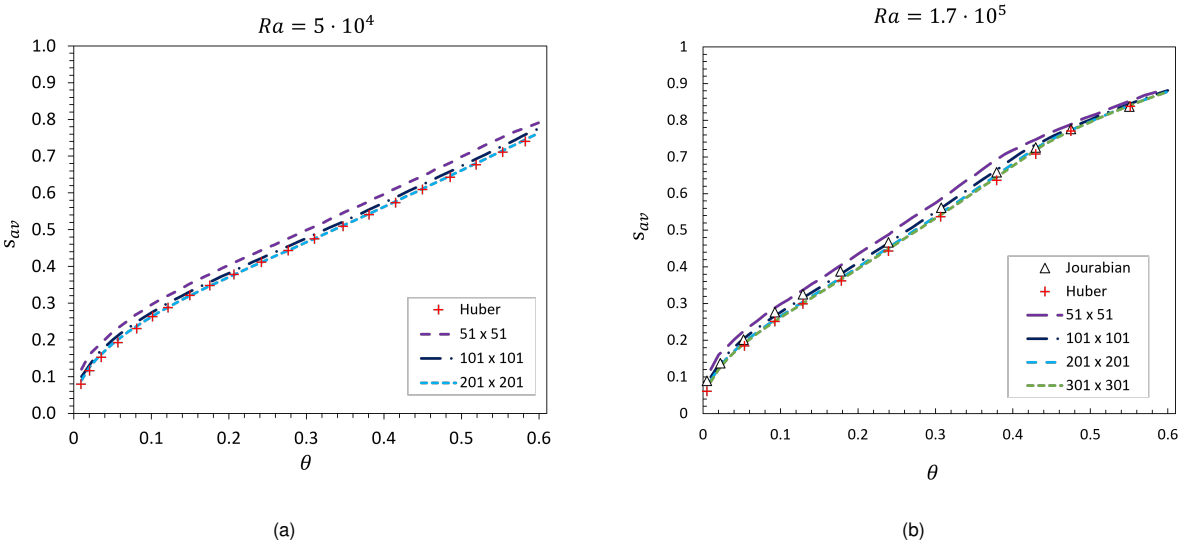
Average Phase Front

The results for the evolution of the average phase front position $s_{av}(\theta)$ are given in figure 4.8a for $Ra = 5 \cdot 10^4$ and figure 4.8b for $Ra = 1.7 \cdot 10^5$. Both Rayleigh numbers were compared with the results from Huber [16], but Jourabian [68] only provided results for $Ra = 1.7 \cdot 10^5$. Four different grid refinements (N x N) were employed until the results showed to be approximately constant. This was the case at $N = 201$for $Ra = 5 \cdot 10^4$, and at $N = 301$ for $Ra = 1.7 \cdot 10^5$. The simulations for both Rayleigh numbers show a good correspondance to the benchmark case(s). The maximum local error with the benchmark cases is 0.3% (w.r.t Huber) for $Ra = 5 \cdot 10^4$ and $1.7\% - 4.0\%$ (w.r.t Huber, Jourabian resp.). The latter is acceptable, because the two benchmark studies show an reciprocal 6.1% difference between themselves. The small error is a first indication of the good performance of the current model. However, it should be noted that s_{av} is not a definite benchmark parameter, as the average can balance out effects on the top and bottom of the phasefront. The exact position of the phase front (i.e. $s(x, \theta)$) was not given by Jourabian and Huber. Therefore, the Nusselt number is also important.

Nusselt Number

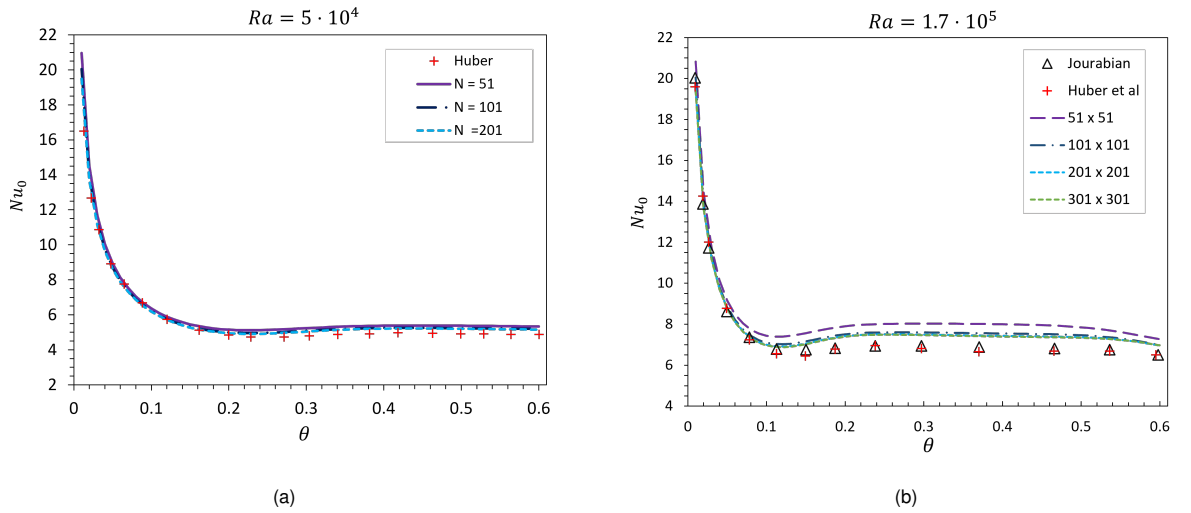
The results for the Nusselt number are given in figure 4.9a for $Ra = 5 \cdot 10^4$ and in figure 4.9b for $Ra = 1.7 \cdot 10^5$. The results were obtained from the same grid refinements (N x N) as for the melting front, and also showed grid convergence at the same amount of grid points. Qualitatively, the Nusselt number shows similar behaviour over time as the benchmark studies: an approximately linear decrease in Nusselt number is observed for shorter times (conduction melting), followed by a plateau value (convection regime). In the first regime (for small $\theta < 0.1$), the Nusselt number shows good quantitative agreement with the benchmark(s), as $\epsilon < 0.1\%$.

However, as θ increases and the effect of convection starts showing, the modelled Nusselt number begins to deviate from the benchmark results. The deviation is larger for $Ra = 1.7 \cdot 10^5$ ($\epsilon \approx 8 - 11.8\%$) than for $Ra = 5 \cdot 10^4$ ($\epsilon \approx 5\%$). Three different test were performed to see if they would influence the result: (1) decreasing the lattice viscosity ν (and thereby the timestep), (2) setting the weighting factor $B_f(x, y)$ in the immersed boundary condition (see section 3.2.3) to 1.0 in the mushy and solid zone, i.e. imposing a zero velocity, (3) using the enthalpy porosity method for the treatment of the no-slip at the boundary condition instead of the immersed-boundary method. The results did not improve for the first two points and the third point showed unphysical results.



(a) (b)

Figure 4.8: Results for phase front s_{av} evolution over time θ, for different grid sizes N x N. Compared with results by Huber et al. [16] and Jourabian [68].



(a) (b)

Figure 4.9: Results for Nusselt number Nu over time θ, for different grid sizes N x N. Compared to results by Huber et al [16] and Jourabian [68].

Two possible explanations for the deviation with the benchmarks are identified:

- *Unphysical numerical diffusion across the phase front.* It is found that the current model shows a slightly lower temperature in the solid phase than the initial $T_0 = T_m$, in contradiction with theoretical pure substance melting. This is illustrated in figure 4.10, which plots the temperature profile for $Ra = 1.7 \cdot 10^5$, at time $\theta = 0.2$ and at position-probe $y = 0.5H$. It showcases a 'temperature dip' to values beneath initial T_0 around $x = 0.35H$, the position of the phase front (see figure B.2), and these lower temperatures diffuse into the solid as well. Thereby the temperature difference is enlarged, increasing the flow, heat transfer and Nusselt number. Huber & Jourabian also commented on this local temperature decrease, and avoided its result by modifying the collision step to ensure no heat transfer in parts of the domain where the enthlapy has remained equal to the initial enthalpy. In other words, they made sure the sink term at the phase front was not spread to the rest of the solid, thereby not enlarging the temperature difference. The effect of numerical diffusion for this geometry with LBM is also known in other literature, for example by Huang & Wu [19]. They introduced a "magic parameter" to keep this effect under control. However, this approach was not possible to implement in the current model, as it was specifically for the LB MRT-collision scheme. It is recommended for future research to implement grid refinement near the phase front to overcome the diffusion, motivated by the smaller temperature dip for finer grids in figure 4.10.

- *Difference in phase change treatment*. While the current model uses an enthalpy-based method to solve for phase change, Huber & Jourabian both use an source-based method. The latter treats latent heat as a source term in the thermal LB equation, requiring an extra iteration for solving the problem and making the method implicit. This slightly underestimates the rate of heat transfer and thereby Nu_0 with respect to the total-enthalpy-based method that was used in this study.

It can be concluded that the proposed method shows good behaviour for modelling *conduction* melting. However, for *convective* melting it is not precluded if the deviation with the benchmark stems from a fundamental difference in methodology or from numerical errors. Convective melting will be further investigated in the next chapter.

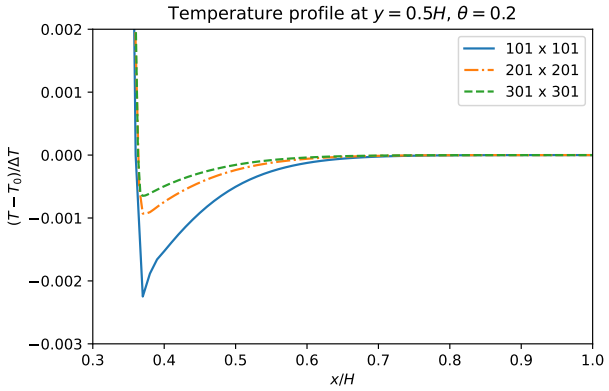


Figure 4.10: Illustration of nonphysical numerical diffusion: temperature profile at position $y = 0.5H$ and time $\theta = 0.2$. For $Ra = 1.7 \cdot 10^5$ and three different grid sizes N x N.

Chapter 5

Freeze-Plug Modelling Results

In this chapter the three individually modelled components from the previous chapter are combined into one model: simulating the melting MSFR freeze-plug. The results of the modelled freeze-plug are compared to a numerical study by Pater & Kaaks [11]. Their study was recently performed as a part of the SAMOSAFER [3] collaboration, and used three different codes to model the freeze-plug: Star-CCM+, OpenFoam and DGFlows. First, the description of the freeze-plug geometry is introduced in section 5.1. The simulation results are presented and compared to the benchmark study in two stages: without natural convection in section 5.2 and with natural convection in section 5.3.

5.1 Problem Description

The modelled set-up shown in figure 5.1, based on the MSFR freeze valve design described by Giraud et al. [10] and used in the numerical benchmark by Pater & Kaaks [11]. It consists of a rectangular cavity filled with a frozen fuel salt, with initially the top 10 % molten. Adjacent to the cavity is a vertical metal wall, which models the surrounding pipe structures of the freeze-plug. To save computational resources, the total geometry is modelled in 2D and cut in half at the vertical symmetry axis. The boundary conditions on the outer vertical edges, the total bottom boundary and the wall top boundary are adiabatic in their normal directions. A no-slip velocity boundary condition is applied on the metal-salt interface, the top boundary and at the melting front.

The melting of of the freeze-plug is driven by two heat sources: (1) decay heat from the reactor and (2) neutronic & photonic heat deposition. The decay heat is modelled by a time-dependent boundary condition at the top horizontal boundary of the cavity. Its definition was derived by Tiberga et al. [9], based on the average temperature in the reactor after shut-down, due to decay heat:

$$T_{BC}(t) = -0.001t^2 + 0.5244t + 923 \quad [K] \tag{5.1}$$

The heat deposition stems from energetic neutrons and photons travelling from the reactor vessel and heating up the salt in the freeze-plug. This effect is mimicked by adding a time-dependent, volumetric heat source to the entire (i.e. both molten- and frozen) salt volume. It is approximated to being 1% of the decay heat of the reactor, characterized by [69]:

$$Q_{source}(t) = 6.45908 \cdot 10^4 - 6.92 \cdot 10^3 \cdot \ln(t) \quad [Wm^{-3}] \tag{5.2}$$

Two physical phenomena further enhance the melting: conjugate heat transfer and natural convection. The first stems from the metal wall adjacent to the salt: as the metal is assumed to have a higher thermal conductivity than the salt, it acts as heat storage. This also gives rise to the second phenomenon: natural convection occurs as the salt closest to the wall will have lower density.

It should be noted that several phenomena that could be expected in the real-life freeze-plug are not taken into account in the simulation. These limitations include temperature-dependency of density and thermal conductivity (and accompanying volume changes), isothermal phase change, cracking phenomena in the solid due to strong temperature gradients at the interface, and forced convection due to re-circulation in the reactor [11].

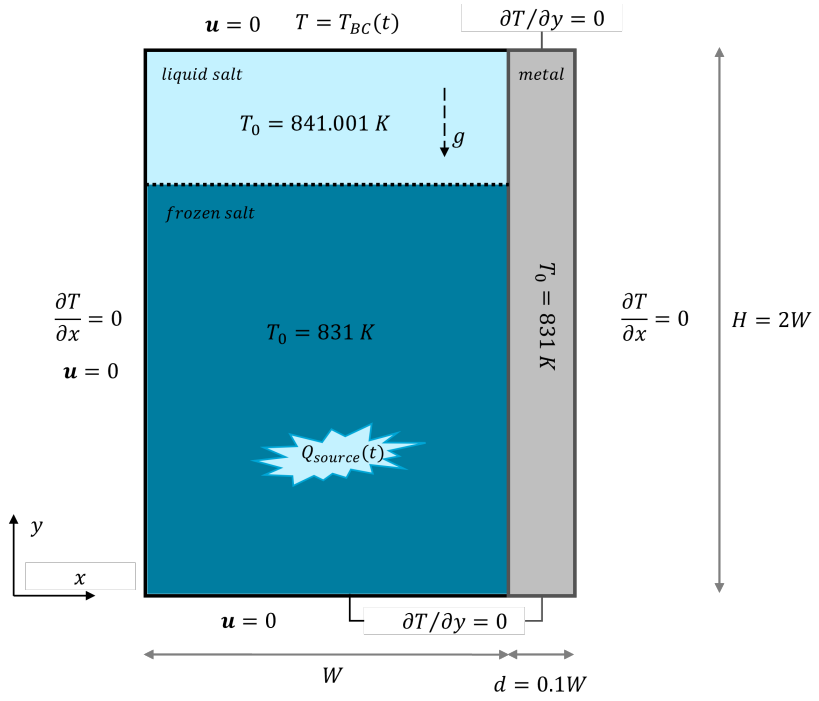


Figure 5.1: Freeze-plug set-up. Note Q_{source} is present in both frozen- and liquid salt.

The physical input parameters for the salt and the Hastelloy-N metal wall are given in table 5.1. The model assumes to deal a with the salt as a pure material. However, in terms of computational input parameters, a small difference of $\delta T = 0.002$ is taken between the solidification- and melting temperature: $T_l = T_s + \delta T$. The conversion to lattice values can be performed with a chosen lattice value ν and N_x x N_y per equations 3.34, 3.35. Their values were chosen differently for the two stages, and will be specified in their following, respective sections.

	Property	Physical value	Unit	Lattice Value	Unit
Salt	$\rho_s = \rho_f$	4390	kg/m^3	1.0	ls^{-3}
	$\lambda_s = \lambda_f$	1.5	$W/m/K$	$1.5/C_\lambda$	$ls/lt^3/K$
	$c_{p,s}$	815	$J/kg/K$	$815/C_{c_p}$	$ls^2/lt^2/K$
	$c_{p,l}$	1000	$J/kg/K$	$1000/C_{c_p}$	$ls^2/lt^2/K$
	T_m	841	K	841	K
	L	$159 \cdot 10^3$	J/kg	$159 \cdot 10^3/C_{Lat}$	ls^2/lt^2
	α	$2.5 \cdot 10^{-4}$	K^{-1}	$2.5 \cdot 10^{-4}$	K^{-1}
	ν	$7.5 \cdot 10^{-4}/\rho$	m^2/s	specified in 5.2, 4.3	ls^2/lt
	Pr	5	-	5	-
	Ste	0.063	-	0.063	-
Hastelloy-N	ρ	8860	kg/m^3	$8860/C_\rho$	ls^{-3}
	λ	23.6	$W/m/K$	$23.6/C_\lambda$	$ls/lt^2/K$
	c_p	578	$J/kg/K$	$578/C_{c_p}$	$ls^2/lt^2/K$

Table 5.1: Used thermophysical properties for fuel salt and Hastelloy N [9].

Lastly, it should be noted that previous models in chapter 4 have not dealt with the presence of a volumetric heat source $Q_{source}(t)$ yet. This requires a slight alteration in the first term of the enthalpy solution vector γ_k (equation 3.16), given by:

$$\gamma_0^{\pm}(\mathbf{x}, t) = h \pm \frac{\Delta t}{2} \frac{Q(t)}{\rho} \tag{5.3}$$

in which $Q(t)$ is the heat source in lattice units.The term $\frac{\Delta t}{2}$ originates from staggered time-stepping in the FM-LBM, note the analogy with this Q-source term for enthalpy and the f-source term for momentum in solution vector α_k (equation 3.5).

5.2 Results Freeze-Plug: Conductive Conjugate Melting

In this section the first stage of the freeze-plug modelling is described. The effect of natural convection is excluded by setting the gravitational acceleration g to zero, thereby isolating the coupling between transient, *conductive* phase change & conjugate heat transfer. In addition to the parameters from table 5.1, the lattice parameter $\nu = 1/40\ [ls^2/lt]$ is used. Visualisations of the temperature isotherms and liquid fraction after $t = 2500$ are given in figure 5.2. As expected, the time-dependent temperature equation on top has lowered the melting front and the presence of the solid wall made the ice close to the wall melt the fastest.

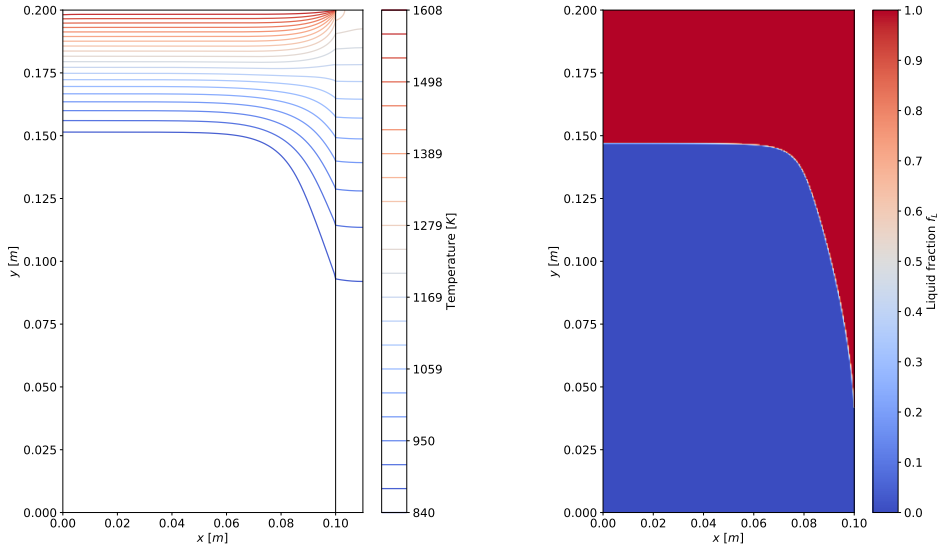


Figure 5.2: Results of modelling the freeze-plug at $t = 2500$ [s] and with fine grid size N_x x N_y = 201 x 401. (a) Temperature isotherms in the salt & wall. (b) Liquid fraction f_L in the salt: fluid at $f_L = 1$, frozen at $f_L = 0$

The results are benchmarked quantitatively with the Pater & Kaaks study. The phase front (x, y)-position and three different temperature probes $T(x = 0, y)$, $T(x = 0.075, y)$, $T(x = 0.09, y)$ were compared at time $t = 2500$ [s]. A grid study for different sizes N_x x N_y was performed and showed clear convergence. All phasefront- and temperature results of the grid convergence study (N_x x N_y = [31x61, 51x101,101x201, 151x301, 201x401]) are included in Appendix C. The results for the finest grid (N_x x N_y= 201 x 401) are plotted in figures 5.3 and 5.4. The corresponding $L2$-errors with respect to the three models in Pater & Kaaks benchmark are given in table 5.2. Note that the StarCCM+ & OpenFoam models taken together as they overlapped almost perfectly.

The total L_2 errors indicate very good ($\epsilon < 1\%$) correspondence to the StarCCM+ and OpenFoam results by Bouke & Kaaks, whereas there is a larger ($\epsilon < 3.5\%$) error with the DGFlows model. The latter could possibly be explained by the model's method of treating the fluid- and wall domain. Namely: while the current LBM, StarCCM+ and OpenFoam all solve on two separate domains, DGFlows solves the heat equation for the full domain in one time. Bouke & Kaaks hypothesized that the separate treatment could cause the solution to lag behind in time and therefore shows a slower melting rate. However, this effect would be expected to become smaller for smaller time steps, which does not seem to be the case in both the current study and also was not in Bouke & Kaaks. As the current LBM model retrieves the results of $2/3$ benchmark studies with an $< 1\%$ error, it can be concluded that it shows good results for the coupling between conjugate heat transfer and conductive melting.

	L_2-error with current model			
Benchmark Model [11]	$T(x=0,y)$	$T(x=0.075,y)$	$T(x=0.09,y)$	front(x,y)
StarCCM+ & OpenFoam	0.05 %	0.12 %	0.30 %	0.90 %
DGFlows	0.12 %	1.06 %	2.31 %	3.33 %

Table 5.2: L_2-errors for obtained results, with respect to three benchmark models from [11]. StarCCM+ & OpenFoam showed nearly identical results.

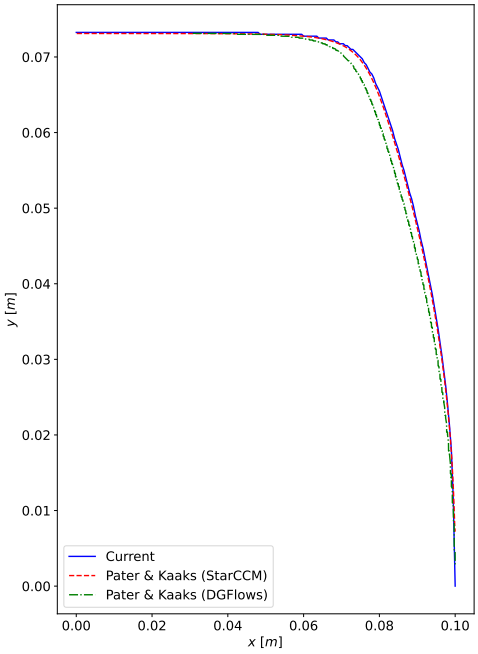


Figure 5.3: Results obtained for phase front position with used grid size 201 x 401. Compared with StarCCM+ & DGFlows benchmark models from [11]

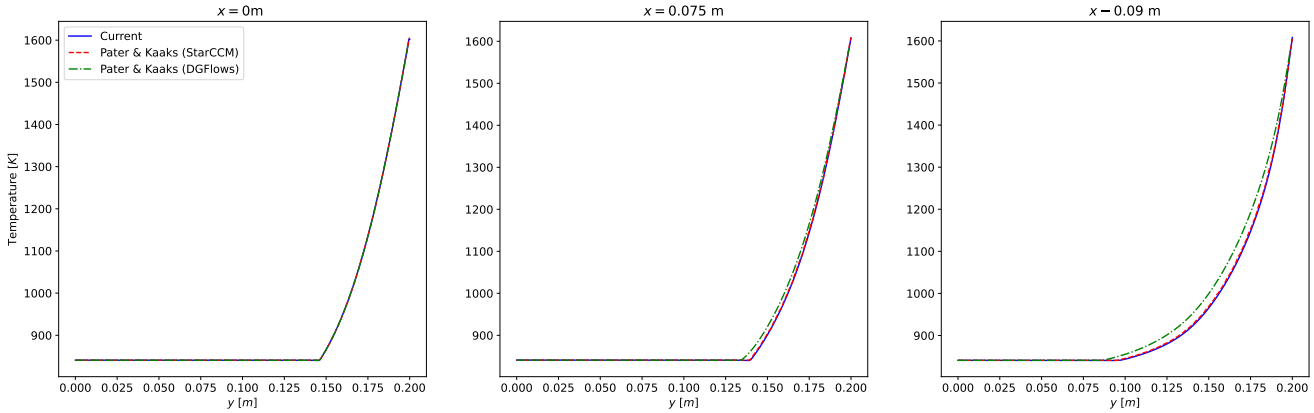


Figure 5.4: Results obtained for temperature vs y position at three different x-probes, with used grid size 201 x 401. Compared with StarCCM+ & DGFlows benchmark models from [11].

5.2.1 Wall Thickness

Several parameters are of interest for the ultimate design of the freeze-plug, one of which is the wall thickness of the surrounding pipe structure. This subsection looks at the influence of the wall-thickness d on the melting time t_{melt} of the freeze-plug. Note that t_{melt} can be defined as the time at which the uttermost lower-right corner of the freeze-plug is liquid ($f_L(0.1,0) = 1.0$): at that point the freeze-plug loses contact with the wall and is able to fall down, opening the drainage pipe. Simulations are run for different values of d, and with the same parameters as defined in table 5.1. The results for t_{melt} at

different d/W are plotted in figure 5.5.

It can be observed for $d < 0.3W$ that the melting time decreases with increasing wall thickness: more metal material can store more decay heat, thereby increasing the melting rate of the salt adjacent to the wall. However, after $d > 0.3W$ the melting time increases again: the larger material now takes longer to heat all the way through. It is interesting to see that an optimal value of the wall thickness exists, at which t_{melt} is at its minimum. For this case it thus lies around $d = 0.3W$.

Of course this is a very preliminary study on the wall thickness: other choices for example fluid & wall properties, initial conditions and boundary conditions would influence the melting times. However, the result does stress the importance of the careful choice of wall thickness in the freeze-plug design.

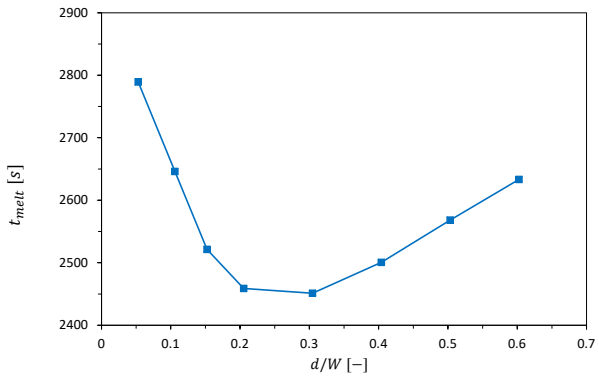


Figure 5.5: Freeze-plug melting time t_{melt} for different wall thickness ratios d/W.

5.3 Results Freeze-Plug: Convective Conjugate Melting

In this section, the second stage of the freeze-plug modelling is described: now the effect of natural convection is taken into account. The presence fluid velocities now restricts the stability requirements (see section 3.3), and therefore the lattice values N_x x N_y = 251 x 501 and $\nu = 1/150$ [ls^2/lt] are taken. This results in a big increase of computational resources opposed to the previous stage without natural convection.

Due to memory- and time constraints, it was not possible to model the resulting temperature and phase front profiles at large time $t = 2500$. Therefore, the solutions could not be benchmarked with the study by Pater & Kaaks. Instead, this section comments on the qualitative performance of simulations, at earlier t. It was found that different simulation results were obtained for different phase front boundary conditions: the immersed- and enthalpy-porosity boundary method (see section 3.2.3). Therefore, the results are discussed seperately.

5.3.1 Immersed-boundary method

The results at $t = 24$s with the immersed-boundary method are shown in figure 5.6. Next to the temperature isotherms and the liquid fraction, now also the velocity streamlines are shown. It should be noted that velocities below $u_{max}/100$ were cut off to zero for visualisation. Several observations from the plot do not seem physical. All three plots show instabilities near the phase front: the temperature fluctuates in both x- and y- direction and the no-slip boundary condition is not achieved. Furthermore, temperatures below the initial solid temperature $T_0 = 831\ K$ are found, likely due to unphysical numerical diffusion of latent heat at the melting front. The fluid salt near the wall is also colder than in the bulk. This effect is most pronounced at the point where the phase front touches the metal wall where the salt is re-freezing instead of melting.

All in all, it can be concluded that the model with immersed-boundary scheme shows severe instabilities. In future research, the effect of grid refinement near the phase front should be examined. Furthermore, the combination of the WNSV conjugate BC and immersed-boundary BC should be further looked into.

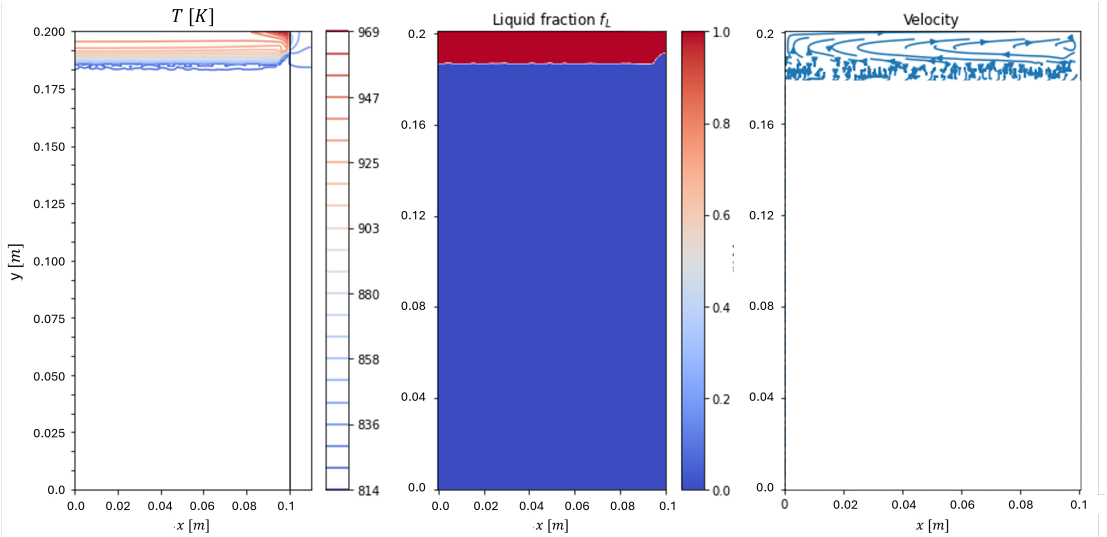


Figure 5.6: Simulation results with immersed-boundary method. For temperature isotherms (left), liquid fraction (middle) and velocity profile (right) in freeze-plug, after $t = 24$s

5.3.2 Enthalpy-porosity method

The simulation results at $t = 41$s with the enthalpy-porosity-method are shown in figure 5.7. Again, velocities below $u_{max}/100$ were cut off to zero for visualisation. The results are somewhat better than for the immersed-boundary method: so stability issues arise for temperature at the phase front. Also, the point of contact of the phase front and the metal wall is resolved better: the starts starts to melt in the vicinity of the wall. However, an unphysical heat sink near the phase front is still present and no perfect no-slip is achieved. Also, the temperature of the liquid salt near the metal wall is again colder than in the bulk, which can also be observed from the local downward streaming velocity lines.

All in all, it is hypothesized that the enthalpy-porosity method is more compatible with the conjugate WNSV BC than the immersed-boundary method. This effect can be due to the fact that the immersed-boundary-method bounces back distribution functions at the phase front, thereby complicating the interaction with WNSV BCs at the metal wall. In that sense, the enthalpy-porosity-method is more simple; it merely attributes a local momentum sink $S_{x,y}$ to the momentum equation. For the enthalpy-porosity method it is recommended to implement grid refinement and GPU simulation to study if physical solutions can be obtained.

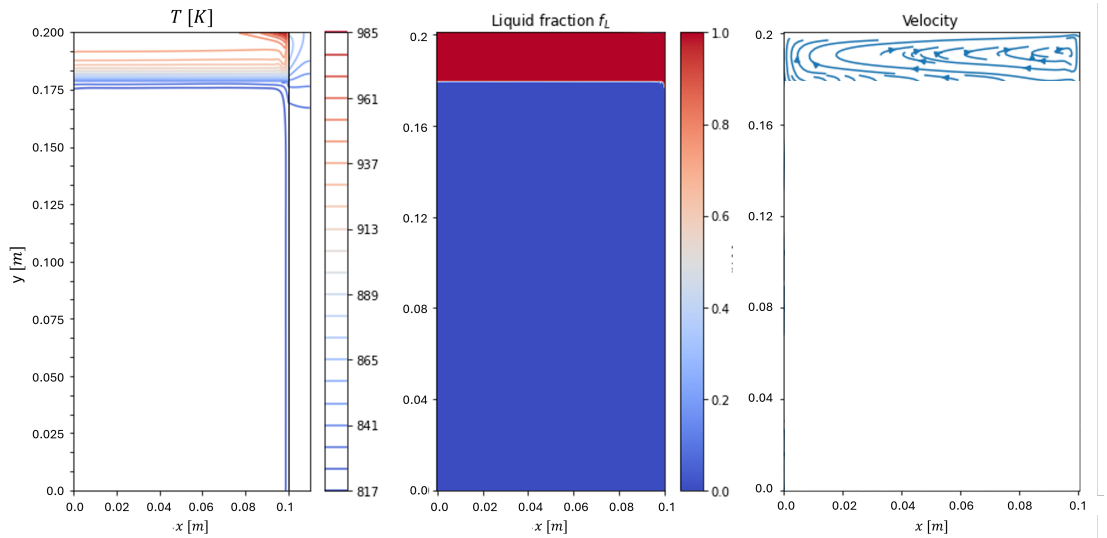


Figure 5.7: Simulation results with enthalpy-porosity method. For temperature isotherms (left), liquid fraction (middle) and velocity profile (right) in freeze-plug, after $t = 41$s

Chapter 6

Wet-Node Boundary Conditions Results

In the previous two chapters, the wet-node-solution-vector boundary condition (WNSV BC) has shown to be successful in modelling conjugate heat transfer in combination with either convective thermal flows or with conductive phase change. The current chapter focuses on the WNSV BC itself, by studying its performance in terms of order of accuracy, energy conservation and required time-iterations. A number of different variations within the WNSV BC are applied and compared. This is done for three different types of boundary conditions: Dirichlet, Neumann and conjugate,

First, the Dirichlet boundary condition is studied by modelling a simple Poiseuille flow in section 6.1. Thereafter, section 6.2 models the combination of Dirichlet, Neumann and conjugate boundary conditions in the more complex conjugate heat transfer case.

6.1 Dirichlet: Poiseuille Case

In this section five different *variations* of the WNSV BC are tested for a Dirichlet boundary condition. A *variation* is defined by its used extrapolation scheme in the WNSV BC to predict the boundary solution vector from known solution vectors in the domain. This methodology was described in detail in section 3.2.2. In this case, the *momentum* solution vector α_k is of interest. For clarity, the five wet-node variations are noted and labelled in table 6.1 with a reference to their corresponding extrapolation scheme.

The different variations are applied by modelling a Poiseuille flow. This is a suitable starting point because (1) only a no-slip (i.e. Dirichlet) boundary condition has to be handled and (2) the analytical solution for a Poiseuille flow is known.

Description	Abbreviation	Extrapolation method α_k	Group
0th order spatial	BC0	3.19a	A
1st order spatial & temporal	BC1t	3.19c	
1st order spatial	BC1	3.19b	B
2nd order spatial	BC2	3.19d	
2nd order spatial & 1st order temporal	BC2t	3.19e	

Table 6.1: Different variations of wet-node-solution-vector boundary condition. The division in group A & B will be explained in section 3.5.1

6.1.1 Geometry & Parameters

The set-up for a Poiseuille flow is shown in figure 6.1, and consist of a two-dimensional parallel plate geometry, with place distance H. No-slip boundary conditions are imposed on the horizontal walls and periodic boundary conditions on the vertical walls. The body force $f_x = \rho g_x$ drives the laminar flow. Its effect is captured in the macroscopic momentum equation 2.3 and in the numerical method via momentum solution vector $\alpha_k^{\pm}$ (3.7). It is assumed that the viscosity is uniform throughout the domain

and that no pressure gradient is present. This makes that the analytical solution for the velocity profile is given by [40]:

$$u_{x,analytical}(y) = \frac{g_x y}{2\nu}(H - y) \tag{6.1}$$

The used input parameters are given in table 6.2. Based on the maximum flow velocity in lattice units (u_{max}) an appropriate input body force g_x is calculated to achieve stable results: $g_x = 8u_{max}\nu/H^2$ [1].

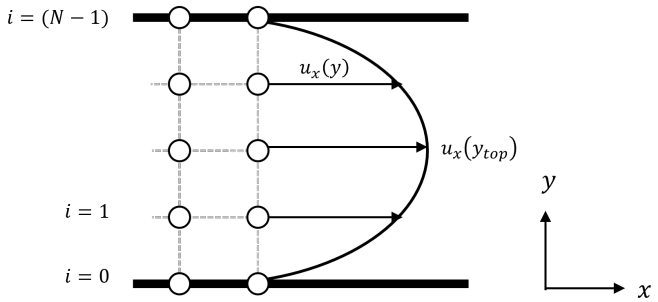


Figure 6.1: Poiseuille flow with wet-node lattice

Description	Variable	Physical value	Unit	Lattice Value	Unit
Height & length geometry	H x W	0.005 x 0.0025	$[m]$	$(N-1)$ x $\frac{1}{2}(N-1)$	$[ls]$
Density	ρ	1000	$[kg/m^3]$	1.0	$[ls^{-3}]$
Kinematic viscosity	ν	$1.4 \cdot 10^{-6}$	$[m^2/s]$	$1/3$	$[ls^2/lt]$
Maximum velocity	$u_{x,max}$	–	–	0.09	$[ls/lt]$

Table 6.2: Parameter definition of Poiseuille flow case. Physical parameters from water.

6.1.2 Time Convergence

All simulations were run until time-convergence was reached, ensuring the flow profile to be fully developed (i.e. steady-state is reached). This was handled by looking at the residual value of the velocity profile, for the following convergence criterium [70]:

$$\sqrt{\frac{\sum_i \left(u_x(y_i,t) - u_x(y_i,t-100\Delta t)\right)^2}{\sum_i u_x^2(y_i,t-100\Delta t)}} < 10^{-6} \tag{6.2}$$

The Poiseuille flow was modelled for different amounts of lattice points $N = [5, 9, 21, 41, 81, 161]$. The amount of time iterations N_t required to reach convergence at a lattice size N was noted for the five different WNSV BC variations (BC0, BC1, BC1t, BC2, BC2t). Also, simulations were run with the link-wise HBB method for comparison. No difference in N_t was found within the order of $100\Delta t$, hence mutual results for the required N_t are shown in figure 6.2.

The obtained velocity profile $u_x(y)$ for Poiseuille flow modelled by the five different variations, for a very fine grid ($N = 161$) is shown in figure 6.3, together with the analytical solution. It is clear that all variations reproduce the analytical solution up to a very small error.

6.1.3 Order of Accuracy

This section has the goal to determine if all WNSV BC variations show grid convergence, and with which order of accuracy. The error in the velocity profile was calculated with respect to the analytical solution via the L_2-error norm (equation 3.40), for different grid sizes N. To avoid overseeing potential local differences, the local error at the top of the velocity profile ($y = \frac{1}{2}$) was also studied next to the global error, but no notable results were found. The grid dependence results for the L_2-error for the five wet-node BCs are shown in figure 6.4, together with the results for the link-wise HBB BC for comparison. Note that some of the y-axes differ in scale per sub-figure. Figure 6.5 shows the same data, but now plotted on double-logarithmic axes. As explained in section 3.5.1, order of accuracy p can be determined from this double-logarithmic plot from its slope. A quick observation of both figure 6.4 and 6.5 shows that the BC variations can be classified into two groups, based on the general shapes of the figures: (A) HBB, BC0, BC1t and (B) BC1, BC2, BC2t. Therefore the groups' results are interpreted separately in the following two sections.

[1] This comes from u_x having its maximum value at $y = H/2$

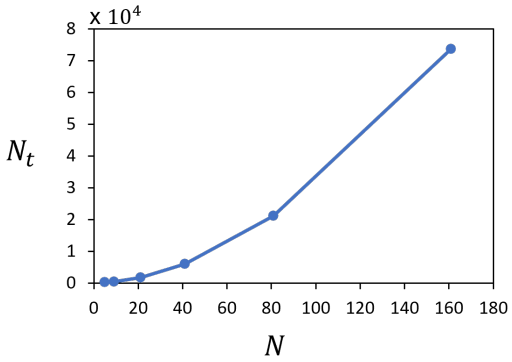


Figure 6.2: Required time iterations N_t to reach convergence citerium 6.2 for amount of grid points N in Poiseuille flow model. All five wet-node- and the link-wise HBB boundary method show the same results.

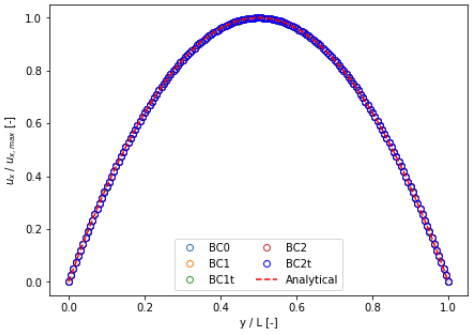


Figure 6.3: Velocity profile, non-dimensionalised via $u_x(y)/u_{x,max}$, for different variations in the WNSV BCs. $N_y = 161$

Results group (A): HBB, BC0, BC1t

The grid convergence results of HBB, BC0 and BC1t are plotted in figure 6.4a, 6.4b, 6.4c respectively. With increasing amount of grid points N, all three show a clear convergence towards approximately zero error. It should be noted that the absolute error in HBB is smaller than for BC0 and BC1t: while HBB already shows $\epsilon < 0.1\%$ for $N = 11$, the two wet-node BCs achieve $\epsilon < 0.1\%$ only at $N = 21$. This could be attributed to the fact that bounce-back boundary schemes (like HBB) are inherently mass conservative, while extrapolation schemes (like the WNSV) are not [15]. Therefore, the error in recovering the analytical solution for WNSV can be higher than for HBB.

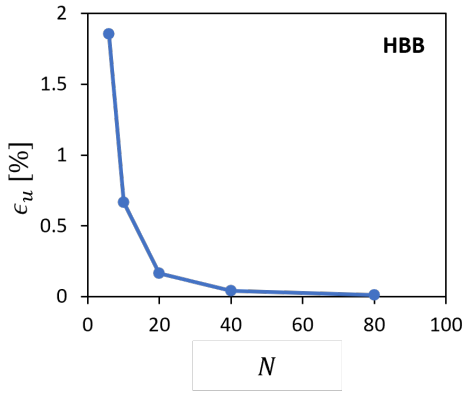
The double-logarithmic plots of the error ϵ_u with respect to $1/N$ for HBB, BC0 and BC1t are shown in subfigures 6.5a, 6.5b, 6.5c. The dotted line is a reference for slope $p = 2.0$. As expected [15], HBB shows second-order accuracy for the Poiseuille flow. Also BC0 and BC1t show a near-perfect second order accuracy, i.e. their lower-order extrapolation schemes have not degraded the inherent second-order accuracy of the Lattice Boltzmann method.

Results group (B): BC1, BC2, BC2t

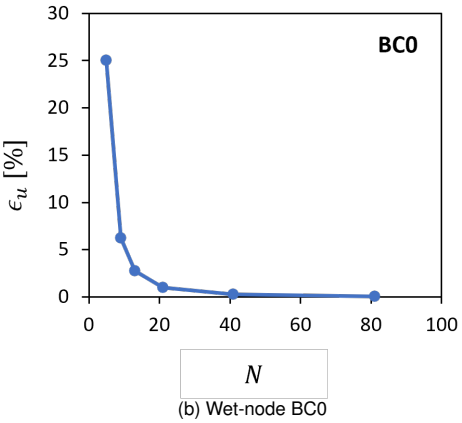
The grid convergence results for BC1, BC2 and BC2t are plotted in subfigures 6.4d, 6.4e, 6.4f. Note that the y-axis (error) is an order of magnitude $\sim 10^3$ smaller than in group (A). At a coarse grid $N = 9$, group (B) already shows a minimal error of $\epsilon \approx 10^{-12}\%$, which is in the order of machine precision. As the grid is refined the error grows, but is still very small, e.g. $\epsilon \approx 0.05\%$ at $N = 160$. Hence, it can be concluded that the extrapolation schemes of group (B) show close to no truncation error for the Poiseuille flow. This absence reveals other, smaller error sources: the round-off error and mass-conservation error. As the grid refines, the amount of iterations increases and the effect of the round-off error and mass leakage become more pronounced [70][71], explaining the upward trend of the error in the figures.

The logarithmic subfigures 6.5d, 6.5e and 6.5f for BC1, BC2 and BC2t are not really informative for the determination of the order of accuracy, because of the absent trunctation error. However, it can be seen that the slopes tend towards $p = -2$. This is in line with the second order scaling of the required amount of timesteps to the grid size ($N \propto N_t^2$) and thereby *increasing* ($p < 0$) the round-off & mass conservation error with the second power as the grid is refined.

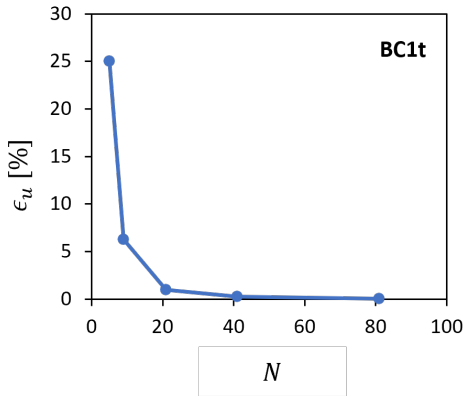
In conclusion, boundary schemes BC1, BC2, BC2t show a promising, very low truncation error in modelling the Poiseuille flow. Therefore, no conclusion about their order of grid convergence can be drawn at this point.



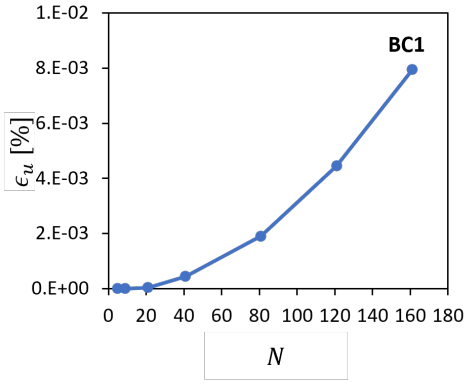
(a) Link-wise HBB



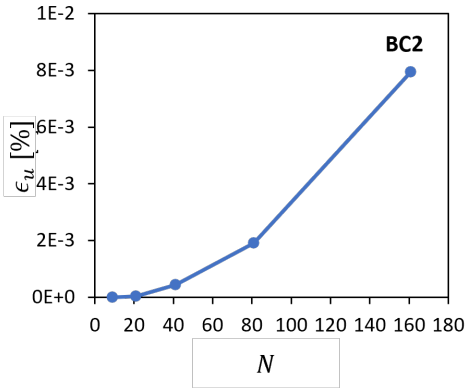
(b) Wet-node BC0



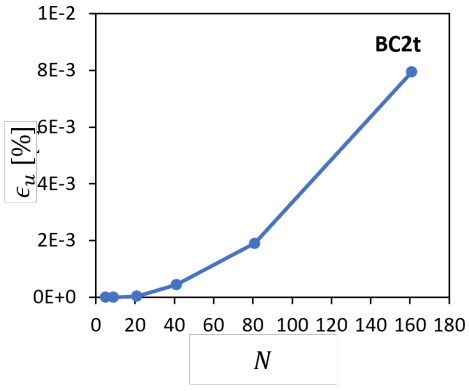
(c) Wet-node BC1t



(d) Wet-node BC1



(e) Wet-node BC2



(f) Wet-node BC2t

Figure 6.4: Grid convergence results: velocity error ϵ_u w.r.t the analytical solution for a Poiseuille flow, plotted against grid size N. Sub-figures for link-wise HBB and five different wet-node BC variations. Note that the y-axes of sub-figures (a)(b)(c) are a factor $10^2 - 10^3$ bigger than those of (d)(e)(f).

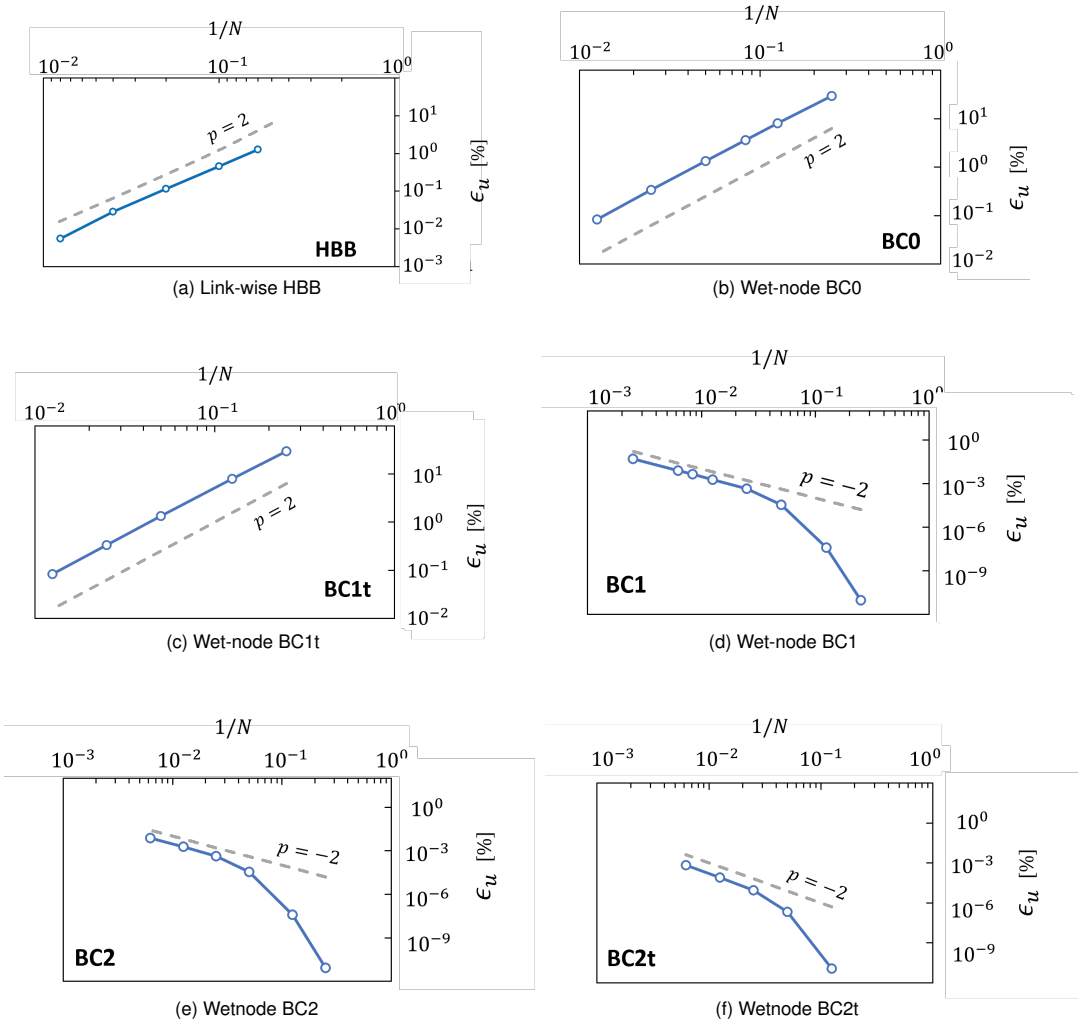


Figure 6.5: Order of accuracy results: velocity ϵ_u w.r.t. analytical solution for Poiseuille profile, against $1/N$, on double logarithmic axes. For link-wise HBB BC and five wet-node BCs. Dotted line shows a reference slope $p = \pm 2$.

6.1.4 Validation Eça-Hoekstra-method

The Eça-Hoekstra method (EHM) was introduced in section 3.5.1 for determining the order of accuracy when an analytical solution is not available ánd a fine grid solution is computationally hard to obtain. Despite the fact that neither is the case in the Poiseuille flow, it does provide a good opportunity to test the implementation and results of the EHM.

A situation in which the analytical- and finest grid solution were not available was mimicked by excluding the modelled velocity solutions for the most fine grids. In other words, the EHM study was carried out with the velocity solutions on only the four most coarse grids ($N = 5, 9, 11, 21$). This was only done for group (A), as the other group did not show grid convergence. By using the EHM minimization equation 3.44, predictions of the order of accuracy p and velocity reference value $\phi_0 = u_x(y_{top})$ were found. The results are given in table 6.3. The found reference value is presented as a scaled value to the analytical value $u_{x,analytical}(y_{top})$. It can be seen that the EHM predicts both the expected order of accuracy p (≈ 2.0) and the expected reference solution (up til 0.3% with the analytical value) for the Poiseuille flow.

Boundary condition	Order of accuracy p	Reference solution $\phi_0/\phi_{analytical}$
Expected	2.0	1.0
HBB	2.005961	0.99998301
BC0	1.999527	0.99998475
BC1t	1.999511	0.99998472

Table 6.3: Results for p and ϕ_0 of Eça-Hoekstra minimization method applied at Poiseuille velocity results, with inputs at $N = 5, 9, 11, 21$, for three different BC methods.

6.2 Neumann: Conjugate Heat Transfer Case

In this section the different wet-node boundary variations are tested for a more complex case, that involves both Dirichlet- snd (conjugate) Neumann boundary conditions: the conjugate heat transfer case. The geometry and used input parameters are identical to the used benchmark case for conjugate heat transfer in section 4.2: a side-heated cavity with adjacent solid wall. The simulations are carried out for one Rayleigh value: $Ra = 0.7 \cdot 10^4$ and with time convergence criterion 3.39.

Recall from section 3.2.2 that while a Dirichlet boundary condition only required an extrapolation in the solution vector (equation 3.19), (conjugate) Neumann BCs require additional extrapolation schemes: given in equations 3.22 and 3.23. For clarity, table 6.4 shows the abbreviations of the used WNSV BC variations and their corresponding combination of used extrapolation schemes. A new, sixth boundary scheme 'BCmix' is introduced, which combines a first order extrapolation of the solution vector with a second order extrapolation for the Neumann and conjugate BCs.

	Extrapolation method α_k, β_k	Adiabatic / Neumann	Conjugate	Group
BC0	3.19a	3.22a	3.23a	A
BC1t	3.19c	3.22a	3.23a	
BC1	3.19b	3.22a	3.23a	B
BCmix	3.19c	3.22b	3.23b	
BC2	3.19d	3.22b	3.23b	
BC2t	3.19e	3.22b	3.23b	

Table 6.4: Overview of different wet-node-solution-vector boundary condition variations.

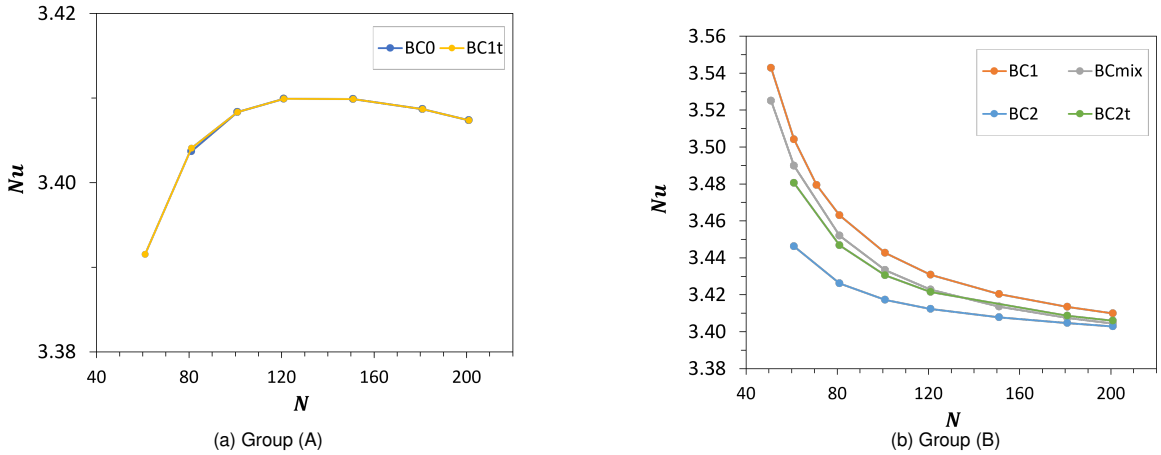
This section employs a similar approach as for the Poiseuille case: studying time convergence and order of accuracy for the six different wet-node boundary schemes. Additionally, energy conservation is studied as a third numerical performance indicator.

6.2.1 Order of Accuracy

The grid convergence results for the modelled Nusselt number Nu_H at the hot wall of the cavity are presented in figure 4.5. Every simulation was performed with $\nu = 1/100$ $[ls^2/lt]$ (see table 4.3), except for BC2 & BC2t, as instabilities arose. BC2 & BC2t therefore used $\nu = 1/50$.

The results for group (B) in figure 6.6b all show convergence approximately towards the expected value ($Nu = 3.42 - 3.436$ [66][67]). Note that while earlier for the Poiseuille flow, group B did not show a truncation error, now for the more complex case of conjugate thermal flows, it does. On the other hand, group (A) in figure 6.6a shows converging behaviour up til $N = 150$, but not for higher grid refinement. However, do note that the relative divergence is small $\sim 3 \cdot 10^{-2}\%$. It could be the case that after $N = 150$ the round-off & mass leakage error become in the same order of magnitude of the truncation error for group (A).

The order of accuracy is calculated for the schemes that show grid convergence: BC1, BCmix, BC2, BC2t (group B). Note that because no analytical solution for Nu is present, the error ϵ_{Nu} [%] is calculated with respect to the finest-grid solution. The latter was taken at $N_{fine} = 201$. The resulting errors at different grid spacings are plotted on double-logarithmic axes in figure 6.7. Note that the grid gets finer from right to left on the x-axis. The dotted line has slope $p = 2$ and is drawn for reference. It can be observed from the figure that for all four BCs in group B, the more coarse grids (up to $N = 121$) are on a straight line with approximately $p = 2$. However, the last points ($N = 151, 181$) show a rapidly increasing slope. This effect is probably attributed to the grid refinement ratios on these points being



(a) Group (A) (b) Group (B)

Figure 6.6: Grid convergence results for different boundary variations, conjugated heat transfer.

too high: $r_{151} = 201/151 = 1.3$, $r_{181} = 201/181 = 1.1$. In other words, $N_{fine} = 201$ is not "fine enough" to be compared to these points. A formal derivation of what is "fine enough" is lacking in literature. Freitas [72] suggests a minimum of $r = 1.3$, Roache [61] $r = 1.5$, while Ober et al. [73] point out that inaccuracies associated with the choice of a too-coarse-fine-grid solution can still occur when $r = 2$. As it was computationally challenging to achieve "fine-enough" reference solutions, also the Eça-Hoekstra method was employed to find the order of accuracy.

By performing the EHM for group (B), with Nusselt values for $N = 61, 81, 101, 121, 151, 181, 201$ as input values, the order of accuracy p and reference value $\phi_0 = Nu$ as given in table 6.5 were found. It can be sen that the EHM-found Nusselt numbers correspond to the trend in figure 6.6b and to the expected benchmark value (~ 0.6 % error). The found orders of accuracy are between $p = 1.6 - 1.74$. It is important to note that the outcome for p was influenced by the amount of used grid points, within a range of $p \pm 0.2$.

Combining the insides on the order of accuracy from both the finest-grid method and the Eça-Hoekstra Method, there is reason to believe that wet-node variations BC1, BCmix, BC2 and BC2t show an order of accuracy between $1.6 - 2.0$. It would be helpful to fully confirm this statement by studying the grid convergence for a conjugate heat transfer problem that has an analytical solution available.

Wet-node BC	p	$\phi_0 = Nu_H$
BC1	1.60	3.396
BCmix	1.66	3.392
BC2	1.71	3.396
BC2t	1.74	3.396

Table 6.5: Results order of accuracy p and reference solution ϕ_0 from Eça-Hoekstra method (EHM).

6.2.2 Time Convergence

The amount of iterations N_t needed at certain number of grids points N is plotted in figure 6.8, for all diferrent wet-node BCs. Note that BC2 & BC2t require ~ 2 times less time iterations, as $\nu = 1/50$ $[ls^2/lt]$ was used instead of $\nu = 1/100$ $[ls^s/lt]$ for the other four BCS. Therefore, the BCs showed no notable time convergence differences amongst eachother.

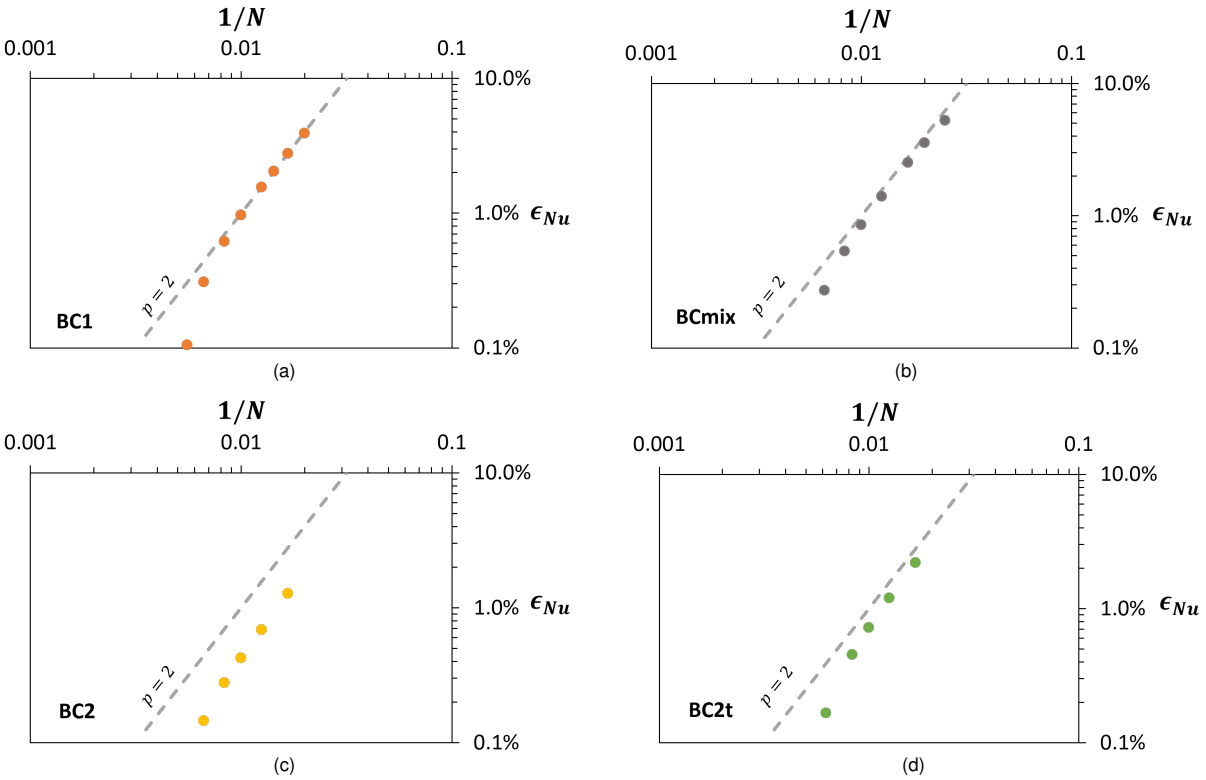


Figure 6.7: Conjugate: order of accuracy for different wetnode BCs

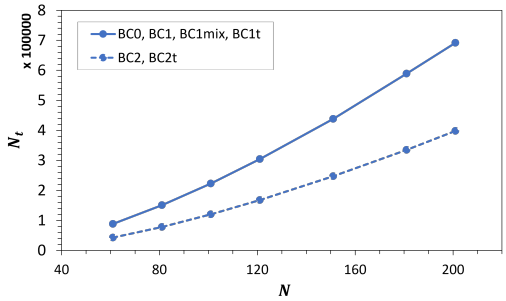


Figure 6.8: Time iterations N_t needed for grid size N to reach steady-state

6.2.3 Energy Conservation

Another important aspect of a numerical model and its boundary conditions, is that it should conserve energy. This can be checked by evaluating the thermal (transient) energy balance over the total geometry. The change in volumetric total energy ($\partial E/\partial t$) should be equal to the sum of in and outgoing surface fluxes $\phi_{in,out}$, given by [40]:

$$\left.\frac{\partial E}{\partial t}\right|_{volume} = \left(\phi_{in} - \phi_{out}\right)_{surface}$$
$$\int_0^H \int_0^H (\rho c_p)_l \frac{\partial T_l}{\partial t} dxdy + \int_0^H \int_H^{H+d} (\rho c_p)_s \frac{\partial T_s}{\partial t} dxdy = \int_0^H -\lambda_s \left.\frac{\partial T_s}{\partial x}\right|_{H+d} dy - \int_0^H -\lambda_l \left.\frac{\partial T_l}{\partial x}\right|_0 dy \tag{6.3}$$

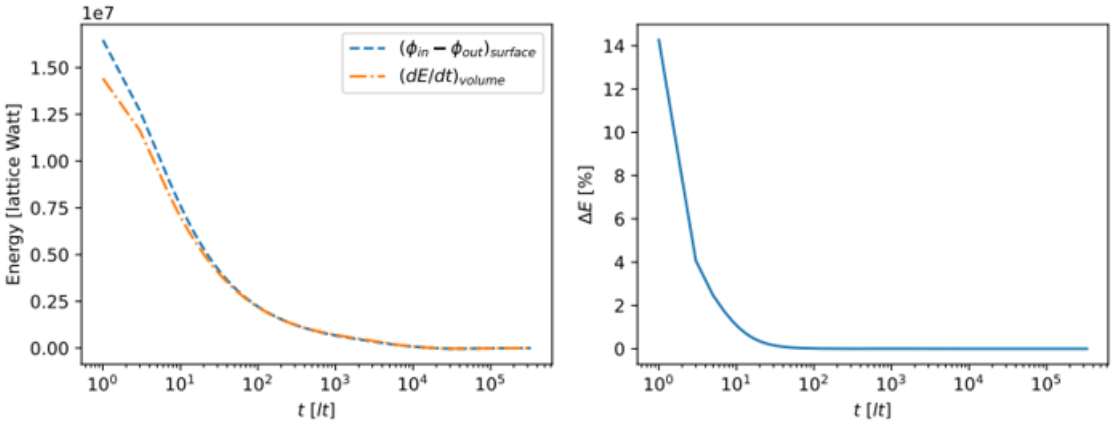
in which subscripts l, s denote the properties on the respective liquid and solid domain.

The results are plotted in figure 6.9, for all different WNSV BC varations. The left sub-figures plot $\frac{\partial E}{\partial t}$ and $(\phi_{in} - \phi_{out})$ of equation 6.3 together over the lattice time t. A fine lattice size $N = 181$ was taken for all plots. The x-axis is plotted logarithmically for better visibility. The right sub-figures show the difference ΔE between the left- and right-hand-side of equation 6.3 in terms of a fraction of the total volumetric

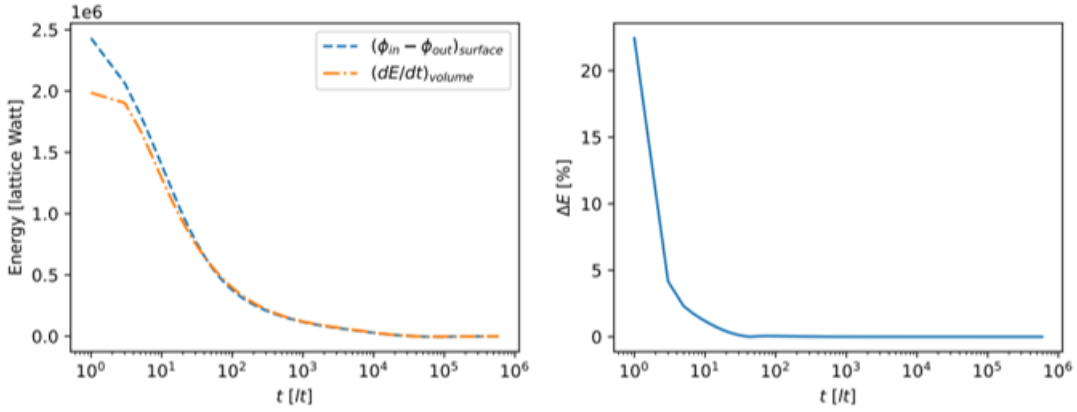
energy, calculated by:

$$\Delta E(t_i)[\%] = \frac{\left|(\phi_{in} - \phi_{out})_i - \frac{\partial E}{\partial t}_i\right|}{\sum_0^i \frac{\partial E}{\partial t}} \tag{6.4}$$

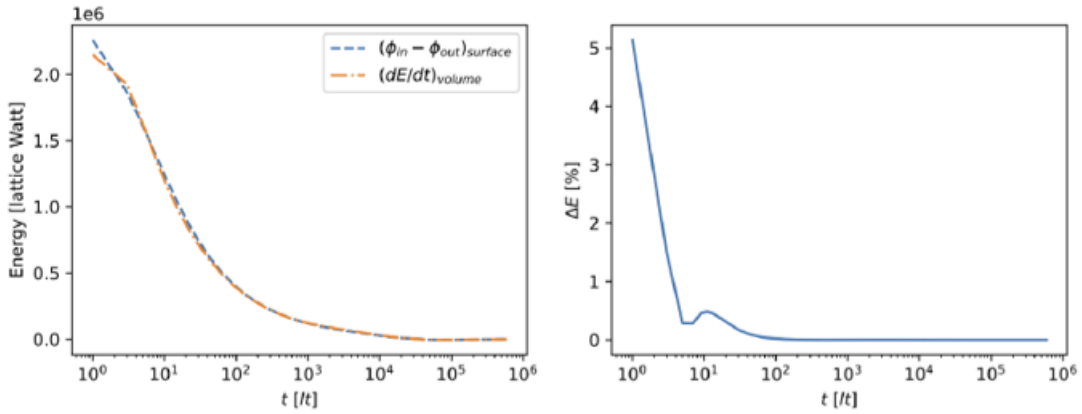
From the figures it can be observed that the $\frac{\partial E}{\partial t}$ and $(\phi_{in} - \phi_{out}$ show the same trend, but do not overlap for small t. In other words, energy is only conserved for larger t and not for small t. The error is bigger for lower-order- than for higher-order extrapolation schemes, as expected. The deviation at smaller t can be explained by the effect of *penetration depth/time*: the characteristic depth/time at which the temperature in the material as noticeably changed at a certain time/depth. The penetration time t_{pen} at point x is typically of scale $t_{pen,x} = x^2/(\pi\kappa)$ [40]. The figures show that the error ΔE reaches $\sim 0\%$ at $t = 20 - 30$ [lt], which is in line with the order of magnitude for the penetration depth at the first point in the domain ($x = \Delta x = 1$ [ls]).



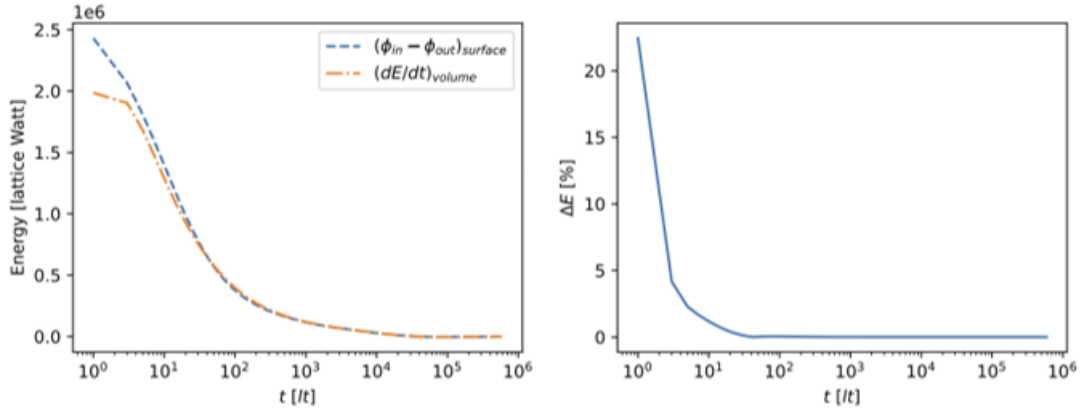
(a) BC0



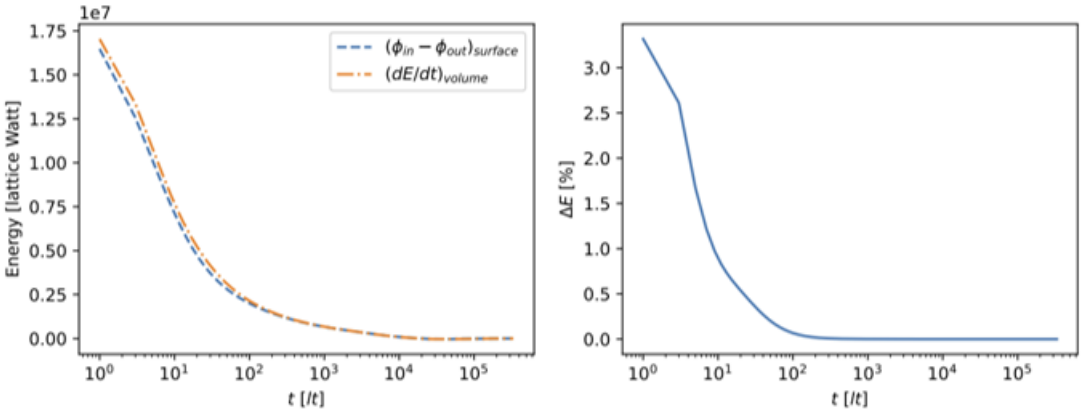
(b) BC1



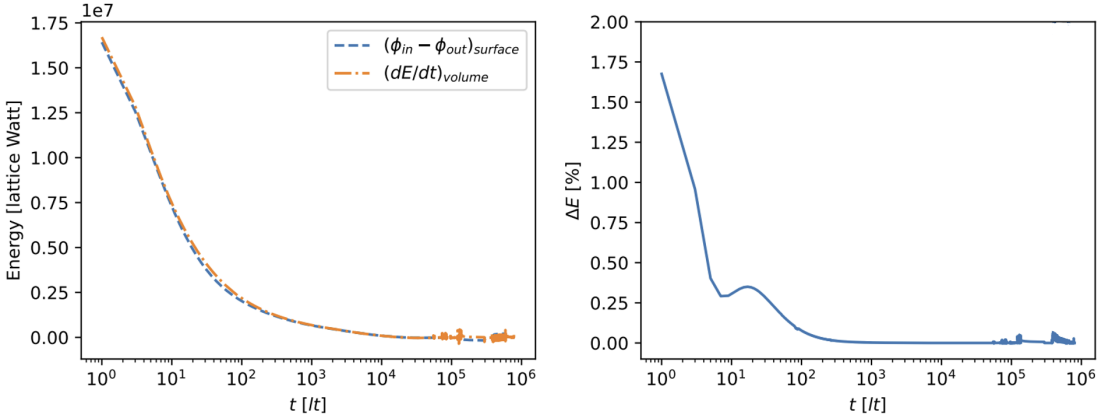
(c) BC1t



(d) BCmix



(e) BC2



(f) BC2t

Figure 6.9: Energy conservation for six different wet-node boundary conditions. Left: LHS and RHS of equation 6.3 [$lW = ls^2/lt^3$] over time [lt]. Right: energy difference ΔE [%] over time [lt]. x-axes are logarithmic. Lattice size $N = 181$

Chapter 7

Conclusions and Recommendations

Nuclear reactors can offer a stable and reliant source of renewable energy, which is highly important to fill the gaps left by intermittent renewable energy sources like solar and wind. The Molten Salt Fast Reactor is one of the reactors proposed by Generation IV and shows to be promising because of its reduced issues with waste management and high safety standards. The freeze-plug safety system is an integral part of the MSFR, melting in the case of an emergency and thereby allowing drainage of the reactor core. It is therefore crucial to study the freeze-plug's melting behaviour. It is known that both conjugate heat transfer with adjacent structures and natural convection in the liquid salt play an important role in the melting process. To this end, this thesis aimed develop a numerical model to simulate convective phase change and conjugate heat transfer. This was done with the Lattice Boltzmann Method with a Filter-Matrix collision scheme. In addition, it was aimed to contribute to the current state of knowledge on the FM-LBM by introducing a new wet-node boundary condition. This chapter discusses the conclusions of this work and some recommendations for further research.

7.1 Conclusion

The Filter Matrix Lattice Boltzmann Method (FM-LBM) with double-distribution functions (DFF) was implemented to simulate natural convection, conjugate heat transfer and phase change. The inherent advantages of the FM-LBM and wet-node boundary scheme were combined to develop a wet-node-solution-vector boundary condition (WNSV BC).

Benchmarks

The model was benchmarked with three different problems. First, it was applied to the well-defined square cavity natural convection problem, for three different Rayleigh numbers. By comparing the characteristic velocities and Nusselt number to benchmark studies by De Vahl Davis [64] and Zhuo & Zhong [33], an $< 0.9\%$ error was found, and the model was verified for convective thermal flows. Second, modelling of conjugate heat transfer was tested for a square cavity with one conducting wall and steady-state natural convection. The results for the Nusselt number showed an acceptable error of $< 1.0\%$ with respect to numerical benchmark cases by Kaminksy & Prakash [66] and Misra et al. [67], for three different Rayleigh numbers. Third, phase change was included by using a total-enthalpy scheme and an immersed-boundary scheme was handled to set the no-slip condition on the phase front. The results for the development of the phase front and of the Nusselt number were compared with two benchmark studies by Huber[16] an Jourabian [68], for two Rayleigh numbers. It was found that the results showed good correspondence for conduction melting, but a large error ($\sim 12\%$) in the Nusselt number was found as convection increased. This could probably be attributed to a fundamental difference in phase treatment between the current- and benchmark studies, or to unphysical numerical diffusion on the phase front in the current model.

The individual model components were aimed to be combined in modelling a freeze-plug geometry. This was done in two stages. First, the combination of conductive melting and conjugate heat transfer was modelled for the freeze-plug geometry, by discarding the effect of natural ocnvection. Comparison with three-fold results by Kaaks & Pater [11] showed good correspondenc. Especially with respect to the two

models that also used two separate computational domains for the solid and fluid, for which $\epsilon < 1.0\%$ was found. Second, the final freeze-plug case with the addition of natural convection was attempted to be modelled, but the results showed severe instabilities. Most likely, the numerical diffusion as seen in the previous convective phase change case was enlarged as the Rayleigh number in the freeze-plug case was high. Also, the combination of the no-slip boundary condition at the phase front and the conjugated boundary condition proved difficult to implement.

All in all, the wet-node-solution-vector boundary condition, with a first order extrapolation scheme, showed good performance of modelling Dirichlet and (conjugate) Neumann boundary conditions, for conjugate heat transfer combined with either convective heat transfer or conductive melting. The application to convective melting was not verified.

Wet-node Boundary Conditions
The wet-node-solution-vector boundary condition itself was examined in more detail. Different extrapolation schemes for the WNSV BC were adopted to test which one performed best. Two test cases were modelled, a Poiseuille flow and a conjugate heat transfer case.

The Poiseuille case tested a Dirichlet, no-slip boundary condition. The 0th order spatial (BC0) and first order spatial & temporal (BC1t) schemes showed convergence to an $< 1\%$ error with the analytical solution. Second-order-accuracy was obtained. However, the overall error was somewhat larger than for the link-wise HBB BC. The extrapolation schemes first order spatial (BC1), second order spatial (BC2) and second order & first order temporal (BC2t) showed machine precision and no discretization error. However, this revealed smaller-scale round-off and mass-conservation errors, which grew upon grid refinement.

The conjugate heat transfer case tested both Dirichlet and (conjugate) Neumann boundary conditions. The WNSV BC extrapolation schemes BC0 and BC1t did not show fully converging results for the Nusselt number, but BC1, BCmix, BC2 and BC2t did. The order of accuracy for the latter four BCs was studied and showed approximate second-order-accuracy. However, this was not conclusive, to the unavailability of an analytical- and finest-grid solution. Another approach called the Eça-Hoekstra Method (EHM) found an ~ 1.7 order of accuracy for BC1, BCmix, BC2 and BC2t. In addition, energy conservation over time was examined for all WNSV extrapolation schemes. Better energy conservation was found for higher order spatial schemes, as expected. However, none of the schemes was perfectly energy conservative.

Combining the results from the Poiseuille and conjugated case, the four boundary schemes BC1, BCmix, BC2 and BC2t showed to perform the best. Definite conclusion on their second-order accuracy needs further research.

7.2 Recommendations

Recommendations for further research can be subdivided into three categories: improvements on the wet-node-solution-vector boundary condition, improvement of the conjugate convective melting model and study on the freeze-plug.

7.2.1 Improve Wet-Node-Solution-Vector Boundary Condition

- **Confirmation second-order-accuracy** The determination of order of accuracy for proposed WNSV BC relied partly on comparison with a non-analytical solution and proved to be difficult. The second-order accuracy in thermal (conjugate) flows could be further confirmed by modelling a problem with an available analytical solution. For example a Poiseuille flow in a channel with conducing upper and lower walls, for which the analytical solution can be found in [74]. Some other examples for more complex, 2D conjugate problems are given in [75].

- **Mass conservative extrapolation** While bounce-back schemes are inherently mass conservative, extrapolation schemes are not. Further research could look into adjustments on the extrapolation schemes to ensure mass conservation. For reference, future researches can employ the mass conservative method suggested by Bao et al. [76] or the overview of LBM boundary conditions by Jananshaloo [77], for example.

- **Complex boundary geometries** Currently, the wet-node-solution-vector boundary condition is only employed for straight boundaries. Effectiveness for curved boundaries and 3D cases should be examined to generalise its applicability.

7.2.2 Improve Conjugate Convective Melting

- **Grid refinement** It is expected that grid refinement can (partly) overcome the errors and instabilities found in the two convective melting cases in this study. Especially near the phase front, where grid refinement could overcome unphysical numerical diffusion of the latent heat. Additionally, grid refinement reduces memory requirements.

- **Conjugate & phase front BCs** The combination of conjugate boundary conditions and no-slip velocity conditions a phase front showed to be difficult. Future study should assess how to combine them properly.

7.2.3 Study Freeze-Plug

- **Melting complexities** The molten salt and freeze-plug system in reality has many complexities that were not attempted to capture in this study, but nevertheless are important for assessing the MSFR safety. For example the effect of eutectic melting, time-dependent salt properties and radiative heat transfer on the melting behaviour should be examined.

- **Design parameters** The greater goal is to make suitable choices in the design of the freeze-plug system to ensure effective melting in case of emergencies. Two design choices that are not studied yet in detail are the optimal wall thickness of adjacent structures and the effect of the inclination angle. It is recommended to study this in the future.

Bibliography

[1] Intergovernmental Panel on Climate Change. *AR6 Synthesis Report*. IPCC, 2023.

[2] Generation IV International Forum DOE Nuclear Energy Research Advisory Committee. A technology roadmap for generation iv nuclear energy systems. Technical report, 2002.

[3] Samosafer. https://samosafer.eu/.

[4] HG MacPherson. The molten salt reactor adventure. *Nuclear Science and engineering*, 90(4):374–380, 1985.

[5] O. Benes S. Delpech O. Feynberg V. Ghetta D. Heuer D. Holcomb V. Ignatief J.L. Kloosterman J. Serp, M. Allibert. The molten salt reactor (msr) in generation iv: overview and perspectives. *Progress in Nuclear Energy*, 77:308–319, 2014.

[6] D.Heuer A.Laureau E. Merle-Lucotte C. Seuvre D. Gérardin, M.Allibert. Design evolutions of the molten salt fast reactor. 2017.

[7] M Allibert, M Aufiero, M Brovchenko, S Delpech, V Ghetta, D Heuer, A Laureau, and E Merle-Lucotte. Molten salt fast reactors. In *Handbook of Generation IV Nuclear Reactors*, pages 157–188. Elsevier, 2016.

[8] Z. Perkó S. Dulla D. Lathouwers M. Santanoceto, M. Tiberga. Preliminary uncertainty and sensitivity analysis of the molten salt fast reactor steady-state using a polynomial chaos expansion method. *Annals of Nuclear Energy*, 159:108311, 2021.

[9] D. Lathouwers M. Rohde J.L. Kloosterman M. Tiberga, D. Shafer. Preliminary investigation on the melting behavior of a freeze-valve for the molten salt fast reactor. *Annals of Nuclear Energy*, 132:544–554, 2019.

[10] P. Rubiolo M.T. Retamales J. Giraud, V. Ghetta. Development of a cold plug valve with fluoride salt. *Eur. Phys. JN*, 5:9, 2019.

[11] B. Lauritzen D. Lathouwers M. Pater, B. Kaaks. A numerical benchmark for modelling phase change in molten salt reactors. *Annals of Nuclear Energy*, 194:110093, 2023.

[12] Indarta Kuncoro Aji. *Investigation of Basic Parameters in Developing High-Performance Freeze Valve for Molten Salt Reactor*. PhD thesis, The University of Electro-Communications, 2020.

[13] T. Tokushima K. Enoki M. Kinoshita T. Okawa I. K., Aji. Convection effect of the freeze valve melting process on molten salt reactor. In *The Proceedings of the International Conference on Nuclear Engineering (ICONE) 2019.27*, page 1346. The Japan Society of Mechanical Engineers, 2019.

[14] C. Introini D. Tartaglia, A. Cammi and S. Lorenzi. Numerical modeling and simulation of melting phenomena for freeze valve analysis in molten salt reactors. *Nuclear Science and Engineering*, 197(12):3058–3081, 2023.

[15] A. Kuzmin O. Shardt G. Silva E. M. Viggen T. Krüger, H. Kusumaatmaja. The lattice boltzmann method. *Springer International Publishing*, 10(978-3):4–15, 2017.

[16] M. Manga O. Bachmann B. Chopard C. Huber, A. Parmigiani. Lattice boltzmann model for melting with natural convection. *International Journal of Heat and Fluid Flow*, 29(5):1469–1480, 2008.

[17] Betrand et al. Melting driven by natural convection. a comparison exercise: first results. 1999.

[18] D. Chatterjee S. Chakraborty. An enthalpy-based hybrid lattice-boltzmann method for modelling solid–liquid phase transition in the presence of convective transport. *Journal of Fluid Mechanics*, 592:155, 2007.

[19] Rongzong Huang and Huiying Wu. Phase interface effects in the total enthalpy-based lattice boltzmann model for solid–liquid phase change. *Journal of Computational Physics*, 294:346–362, 2015.

[20] Ya-Ling He, Qing Liu, Qing Li, and Wen-Quan Tao. Lattice boltzmann methods for single-phase and solid-liquid phase-change heat transfer in porous media: A review. *International Journal of Heat and Mass Transfer*, 129:160–197, 2019.

[21] C. Huber H. Karani. Lattice boltzmann formulation for conjugate heat transfer in heterogeneous media. *Physical Review E*, 91(2):023304, 2015.

[22] Jinku Wang, Moran Wang, and Zhixin Li. A lattice boltzmann algorithm for fluid–solid conjugate heat transfer. *International journal of thermal sciences*, 46(3):228–234, 2007.

[23] Fankong Meng, Moran Wang, and Zhixin Li. Lattice boltzmann simulations of conjugate heat transfer in high-frequency oscillating flows. *International journal of heat and fluid flow*, 29(4):1203–1210, 2008.

[24] M. Mirzaei M. Seddiq, M. Maerefat. Modeling of heat transfer at the fluid–solid interface by lattice boltzmann method. *International Journal of Thermal Sciences*, 75:28–35, 2014.

[25] Like Li, Chen Chen, Renwei Mei, and James F Klausner. Conjugate heat and mass transfer in the lattice boltzmann equation method. *Physical Review E*, 89(4):043308, 2014.

[26] Kaikai Guo, Like Li, Gang Xiao, Nick AuYeung, and Renwei Mei. Lattice boltzmann method for conjugate heat and mass transfer with interfacial jump conditions. *International Journal of Heat and Mass Transfer*, 88:306–322, 2015.

[27] Liqing Yue, Zhenhua Chai, Lei Wang, and Baochang Shi. A lattice boltzmann model for the conjugate heat transfer. *International Journal of Heat and Mass Transfer*, 165:120682, 2021.

[28] Guigao Le, Othmane Oulaid, and Junfeng Zhang. Counter-extrapolation method for conjugate interfaces in computational heat and mass transfer. *Physical Review E*, 91(3):033306, 2015.

[29] AA Mohamad and S Succi. A note on equilibrium boundary conditions in lattice boltzmann fluid dynamic simulations. *The European Physical Journal Special Topics*, 171(1):213–221, 2009.

[30] Zhaoli Guo, Chuguang Zheng, and Baochang Shi. An extrapolation method for boundary conditions in lattice boltzmann method. *Physics of fluids*, 14(6):2007–2010, 2002.

[31] Qisu Zou and Xiaoyi He. On pressure and velocity boundary conditions for the lattice boltzmann bgk model. *Physics of fluids*, 9(6):1591–1598, 1997.

[32] JGM Eggels and JA Somers. Numerical simulation of free convective flow using the lattice-boltzmann scheme. *International Journal of Heat and Fluid Flow*, 16(5):357–364, 1995.

[33] Congshan Zhuo, Chengwen Zhong, and Jun Cao. Filter-matrix lattice boltzmann model for incompressible thermal flows. *Physical Review E*, 85(4):046703, 2012.

[34] Congshan Zhuo and Chengwen Zhong. Les-based filter-matrix lattice boltzmann model for simulating turbulent natural convection in a square cavity. *International Journal of Heat and Fluid Flow*, 42:10–22, 2013.

[35] Celeke Bus. Simulating the transient freezing in cooled non-eutectic molten salt channel flow. Master's thesis, TU Delft, Reactor Institute Delft, 2022.

[36] Mees Wortelboer. Investigating gpu-accelerated double distribution function lattice boltzmann schemes for heat transfer and phase change in turbulent flows. Msc thesis, Delft University of Technology, Reactor Institute Delft, 2023.

[37] Willemijn Peters. Modelling the rheology and internal resistance of alginate-based semi-solid flow batteries. Msc thesis, Delft University of Technology, Reactor Institute Delft, 2022.

[38] Daniel van Bemmelen. Influence of turbulence on the internal conductivity and total electrical resistance of a carbon black suspension inside a semi-solid flow battery. Msc thesis, Delft University of Technology, Reactor Institute Delft, 2023.

[39] JN Reddy. Principles of continuum mechanics, 2009.

[40] R. Mudde H. v.d. Akker. *Fysische Transportverschijnselen-Denken in Balansen*. TU Delft Open, 2023.

[41] M.J. Tummers H.J.Jonker K. Hanjaliç, S. Kenjeres. *Analysis and Modelling of Physical Transport Phenomena*. VSSD, 2009.

[42] M. Rohde C. Bus. A filter matrix lattice-boltzmann method for melting and solidification in convective flows. *[To be submitted]*, 2024.

[43] Rongzong Huang, Huiying Wu, and Ping Cheng. A new lattice boltzmann model for solid–liquid phase change. *International Journal of Heat and Mass Transfer*, 59:295–301, 2013.

[44] F. Luisa H. Mehling, L. Cabeza. Heat transfer basics. *Heat and cold storage with PCM: An up to date introduction into basics and applications*, pages 105–136, 2008.

[45] A. Bejan P. Jany. Scaling theory of melting with natural convection in an enclosure. *International Journal of Heat and Mass Transfer*, 31(6):1221–1235, 1988.

[46] HB Luan, H Xu, L Chen, DL Sun, and WQ Tao. Numerical illustrations of the coupling between the lattice boltzmann method and finite-type macro-numerical methods. *Numerical Heat Transfer, Part B: Fundamentals*, 57(2):147–171, 2010.

[47] S. Chakraborty D. Chatterjee. A hybrid lattice boltzmann model for solid–liquid phase transition in presence of fluid flow. *Physics Letters A*, 351(4-5):359–367, 2006.

[48] Wen-Shu Jiaung, Jeng-Rong Ho, and Chun-Pao Kuo. Lattice boltzmann method for the heat conduction problem with phase change. *Numerical Heat Transfer: Part B: Fundamentals*, 39(2):167–187, 2001.

[49] F. W. Tavares K. V. Sharma, R. Straka. Current status of lattice boltzmann methods applied to aerodynamic, aeroacoustic, and thermal flows. *Progress in Aerospace Sciences*, 115:100616, 2020.

[50] S. Succi. *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford university press, 2001.

[51] M.Rohde. The filter-matrix lattice-boltzmann method. *[To be submitted]*, 2024.

[52] Thorben Besseling. Investigating adaptive mesh refinement criteria for a double-distribution fmlb scheme in melting and solidification. Msc thesis, TU Delft, Reactor Institute Delft, 2022.

[53] M. Rohde, D. Kandhai, and H.E.A. Derksen, J.J. van den Akker. Improved bounce-back methods for no-slip walls in lattice-boltzmann schemes: Theory and simulations. *Physical Review E*, 67(6):066703, 2003.

[54] H. Williams B. Cook, D. R. Noble. A direct simulation method for particle-fluid systems. *Engineering Computations*, 21(2/3/4):151–168, 2004.

[55] F Kuznik, J Vareilles, G Rusaouen, and G Krauss. A double-population lattice boltzmann method with non-uniform mesh for the simulation of natural convection in a square cavity. *International Journal of Heat and Fluid Flow*, 28(5):862–870, 2007.

[56] S Suga. Numerical schemes obtained from lattice boltzmann equations for advection diffusion equations. *Int. J. Mod. Phys.*, 2006.

[57] Christopher J Roy. Review of code and solution verification procedures for computational simulation. *Journal of Computational Physics*, 205(1):131–156, 2005.

[58] F. Blotther W. Oberkampf. Issues in computational fluid dynamics code verification and validation. *AIAA journal*, 36(5):687–695, 1998.

[59] M. Hoekstra L. Eça. A procedure for the estimation of the numerical uncertainty of cfd calculations based on grid refinement studies. *Journal of computational physics*, 262:104–130, 2014.

[60] K. Salari P. Knupp. *Verification of computer codes in computational science and engineering*. CRC Press, 2002.

[61] P. Roache. Perspective: a method for uniform reporting of grid refinement studies. 1994.

[62] P. Roache. Quantification of uncertainty in computational fluid dynamics. *Annual review of fluid Mechanics*, 29(1):123–160, 1997.

[63] M. D. Salas. Some observations on grid convergence. *Computers & fluids*, 35(7):688–692, 2006.

[64] G de Vahl Davis. Natural convection of air in a square cavity: a bench mark numerical solution. *International Journal for numerical methods in fluids*, 3(3):249–264, 1983.

[65] M. Krafczyk J. Tölke B Nachtwey C. v. Treeck, E. Rank. Extension of a hybrid thermal lbe scheme for large-eddy simulations of turbulent convective flows. *Computers & Fluids*, 35(8-9):863–871, 2006.

[66] DA Kaminski and C Prakash. Conjugate natural convection in a square enclosure: effect of conduction in one of the vertical walls. *International Journal of Heat and Mass Transfer*, 29(12):1979–1988, 1986.

[67] D Misra and A Sarkar. Finite element analysis of conjugate natural convection in a square enclosure with a conducting vertical wall. *Computer methods in applied mechanics and engineering*, 141(3-4):205–219, 1997.

[68] A. Darzi A.A.R. Darzi M. Jourabian, M. Farhadi. Convection-dominated melting of phase change material in partially heated cavity: lattice boltzmann study. *Heat and Mass Transfer*, 49:555–565, 2013.

[69] M Tano, P Rubiolom, J Giraus, and V Ghetta. Multiphysics study of the draining transient in the molten salt fast reactor. In *Proceedings of the 2018 International Congress on Advances in Nuclear Power Plants (ICAPP 18), Charlotte, NC, USA*, pages 8–11, 2018.

[70] P. A. Skordos. Initial and boundary conditions for the lattice boltzmann method. *Physical review E*, 48(6):4823, 1993.

[71] J. C.G. Vershaeve E. Le Coupanec. A mass conserving boundary condition for the lattice boltzmann method for tangentially moving walls. *Mathematics and Computers in Simulation*, 81(12):2632–2645, 2011.

[72] C. Freitas. Standards and methods for verification, validation, and uncertainty assessments in modeling and simulation. *Journal of Verification, Validation and Uncertainty Quantification*, 5(2):021001,

2020.

[73] J. N. Shadid C. Ober. Studies on the accuracy of time-integration methods for the radiation–diffusion equations. *Journal of Computational Physics*, 195(2):743–772, 2004.

[74] Pranab Kumar Mondal and Shibdas Dholey. Effect of conjugate heat transfer on the irreversibility generation rate in a combined couette–poiseuille flow between asymmetrically heated parallel plates: The entropy minimization analysis. *Energy*, 83:55–64, 2015.

[75] Wei Wu, Su-Ling Zhang, Shuang-Feng Wang, and Kong Ling. A lattice boltzmann model for interphase conjugate heat transfer. *Numerical Heat Transfer, Part B: Fundamentals*, 72(2):130–151, 2017.

[76] Jie Bao, Peng Yuan, and Laura Schaefer. A mass conserving boundary condition for the lattice boltzmann equation method. *Journal of Computational Physics*, 227(18):8472–8487, 2008.

[77] A. Fazeli H.A.M. Pesaran L. Jahanshaloo, N. A. C. Sidik. An overview of boundary implementation in lattice boltzmann method for computational heat and mass transfer. *International Communications in Heat and Mass Transfer*, 78:1–12, 2016.

Appendix A

Nondimensional Nusselt number

Nusselt is defined as [40]

$$Nu = \frac{hH}{\lambda} \tag{A.1}$$

with h the convective heat transfer coefficient, H the characteristic length, λ the thermal conductivity. This can be rewritten by using two different describtions of the heat rate. The first being given by Newton's law of cooling:

$$q_{Newton} = hH(T_s - T_\infty) = hH\Delta T \tag{A.2}$$

The second given by the conductive heat transfer:

$$q_{conductive} = -\lambda H \frac{\partial T}{\partial x} \tag{A.3}$$

The two heat transfer rates q_{newton} and $q_{conductive}$ must be equal, which leads to:

$$\frac{h}{\lambda} = \frac{\frac{\partial T}{\partial y}}{\Delta T} \tag{A.4}$$

Filling this into equation A.1, gives for the local Nusselt number:

$$Nu = \frac{H}{\Delta T}\frac{\partial T}{\partial x} \tag{A.5}$$

By integrating over y and dividing by its characteristic length H, the nondimensional, average Nusselt number as given and used in equation 4.1 is found.

Appendix B

Figures of convective melting model

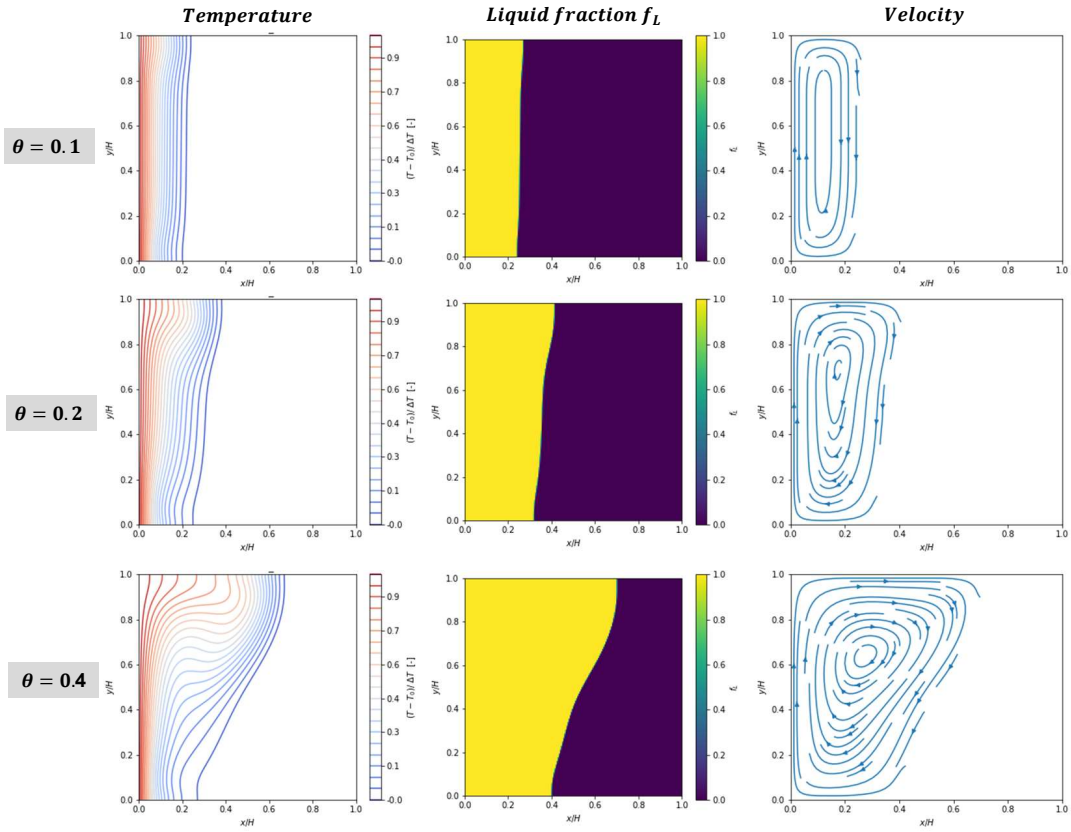


Figure B.1: $Ra = 5.0 \cdot 10^4$. At three different nondimensional times θ

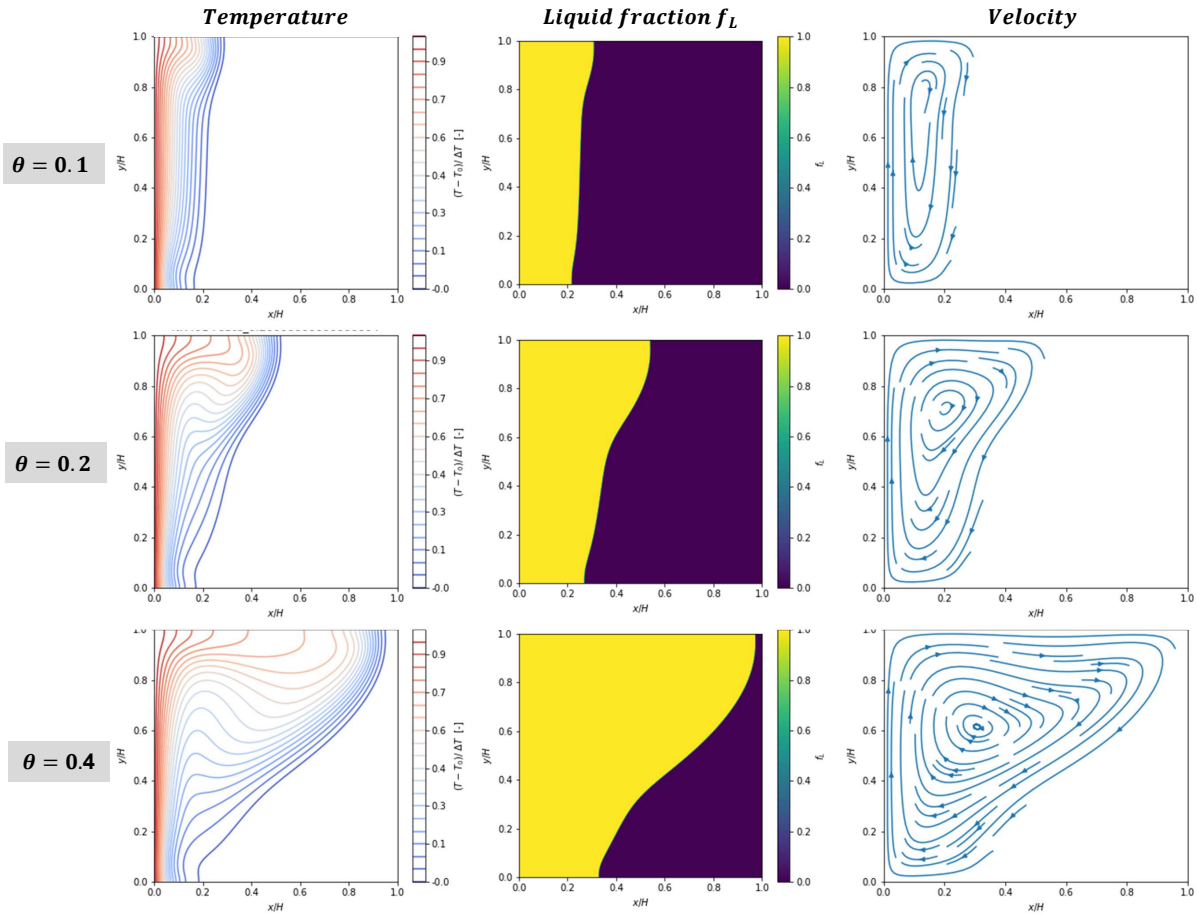


Figure B.2: Results for $Ra = 1.7e5$. At three different nondimensional times θ

Appendix C

Figures freeze-plug modelling without natural convection

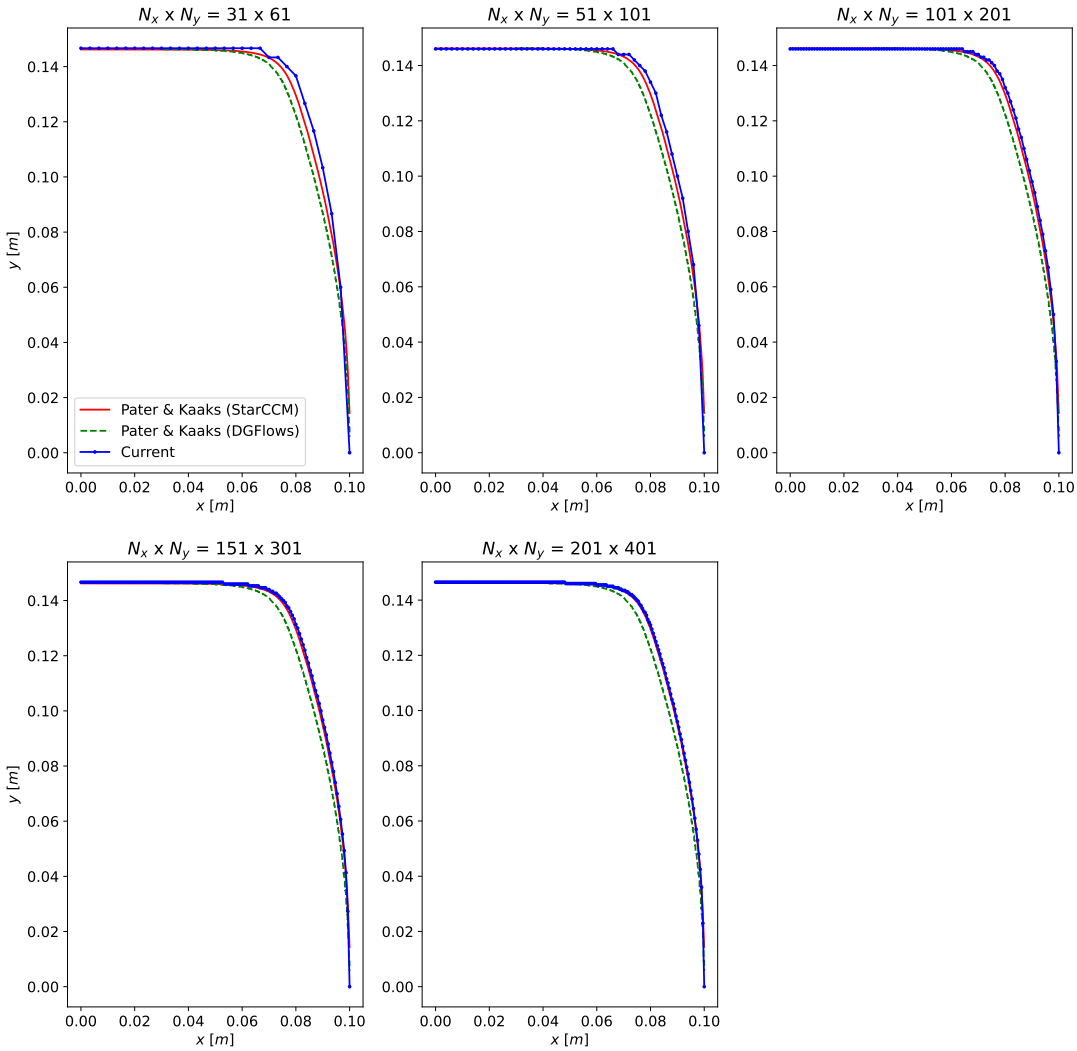


Figure C.1: Modelled phase front position for five different grid sizes N_x x N_y. Compared with benchmark study by Pater & Kaaks [11] for two used models

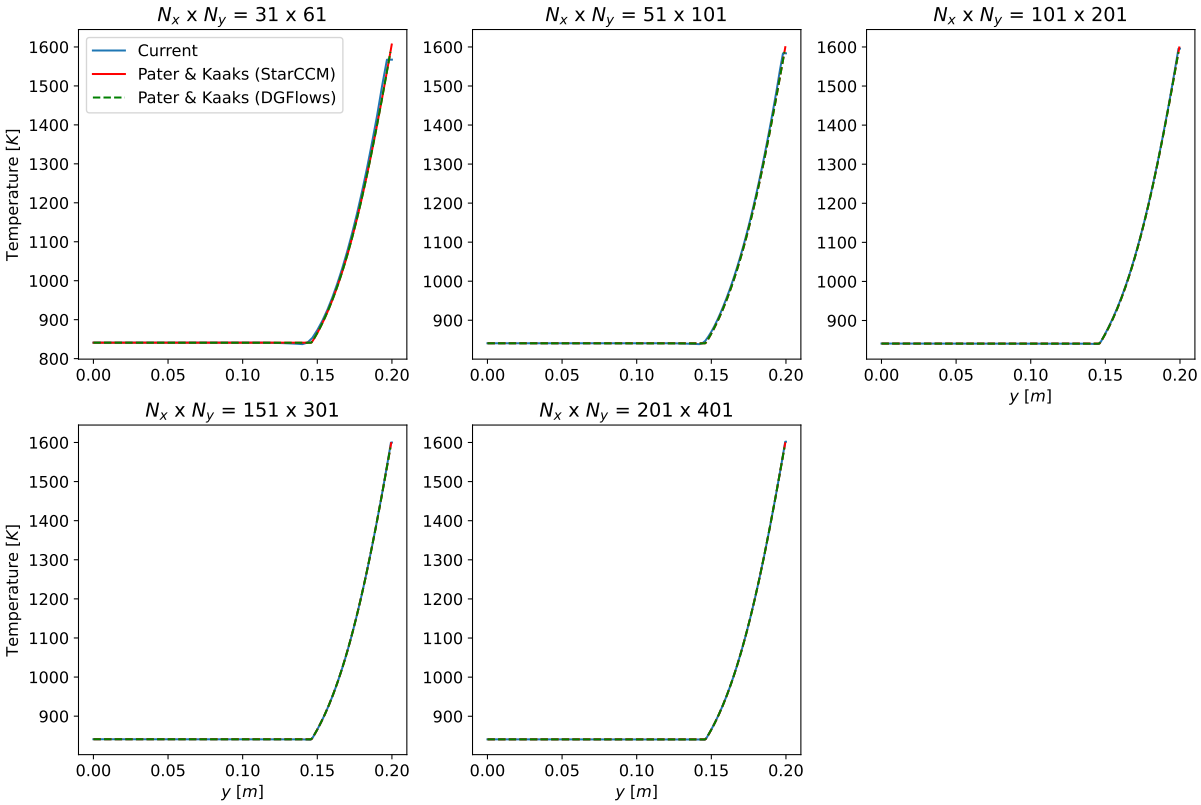


Figure C.2: Modelled temperature profile at probe $x = 0.0\,m$ for five different grid sizes N_x x N_y. Compared with benchmark study by Pater & Kaaks [11] for two used models

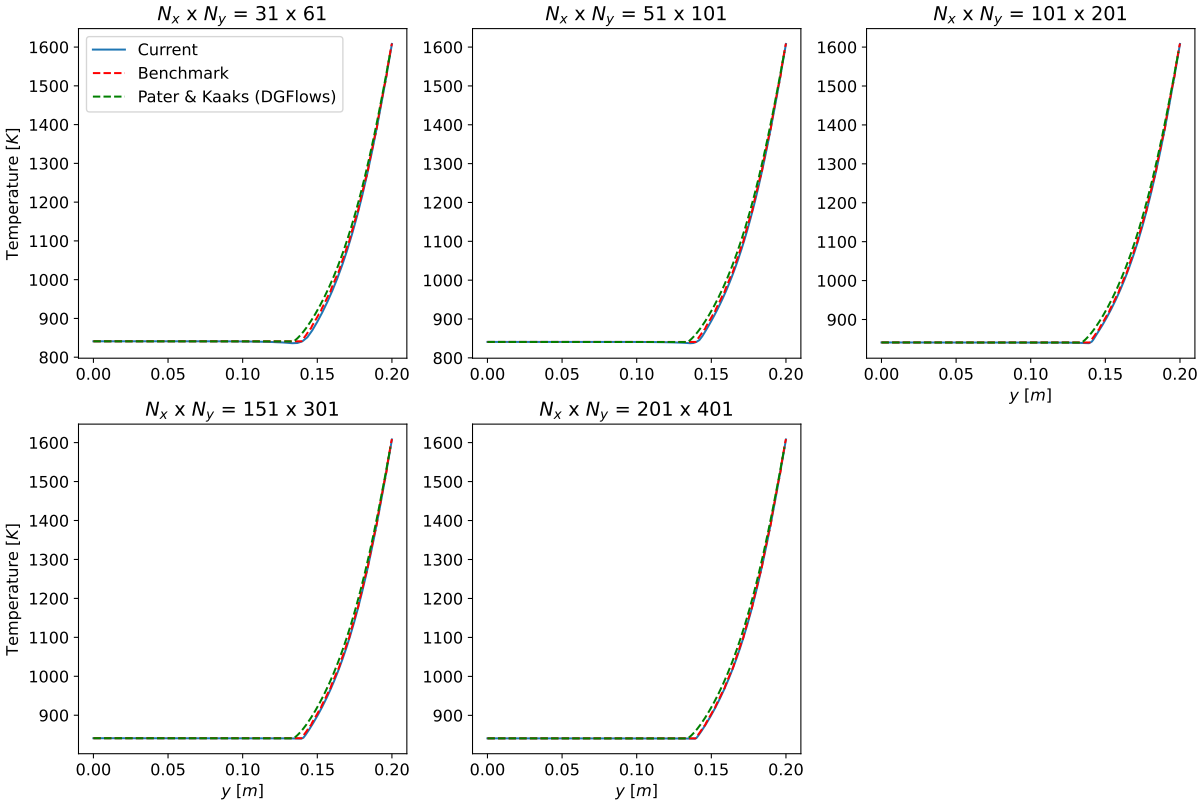


Figure C.3: Modelled temperature profile at probe $x = 0.075\,m$ for five different grid sizes N_x x N_y. Compared with benchmark study by Pater & Kaaks [11] for two used models

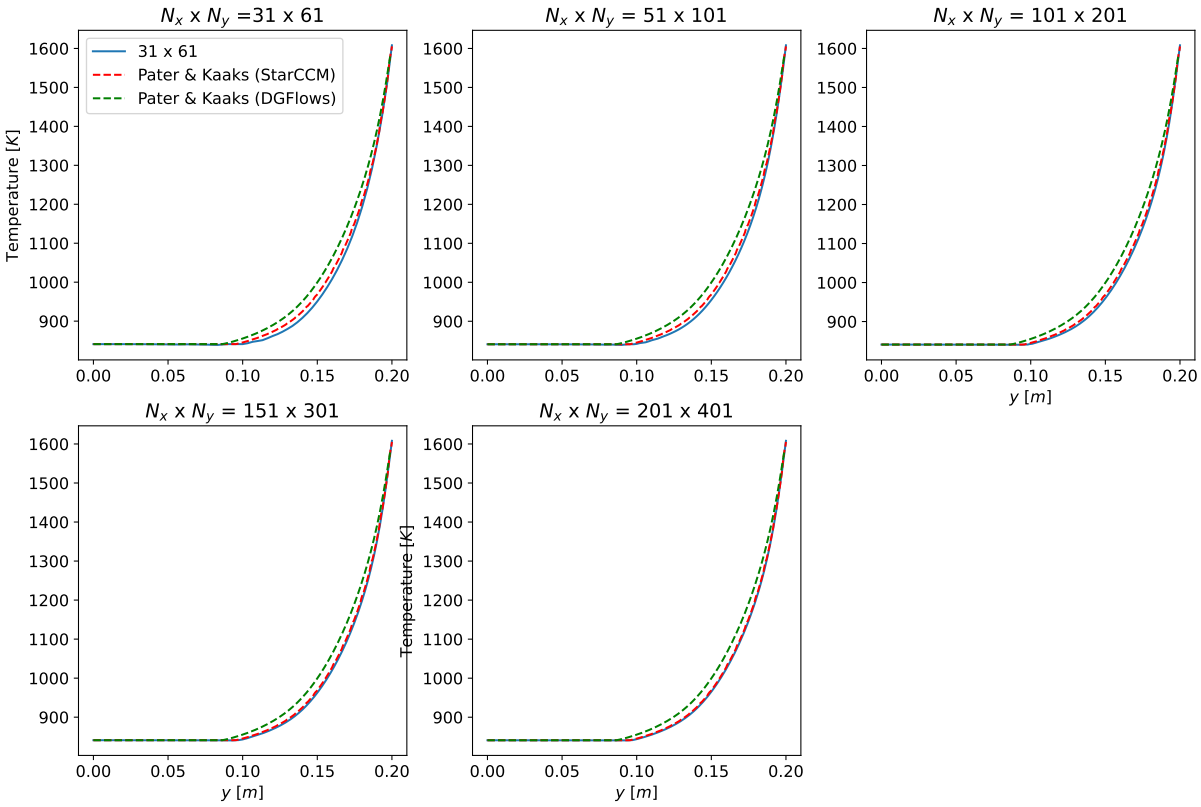


Figure C.4: Modelled temperature profile at probe $x = 0.090\,m$ for five different grid sizes N_x x N_y. Compared with benchmark study by Pater & Kaaks [11] for two used models