

Investigating GPU-accelerated Double Distribution Function Lattice Boltzmann Schemes for Heat Transfer and Phase Change in Turbulent Flows

Master Thesis

MSc. Applied Physics - Physics for Energy

Mees Wortelboer

Investigating GPU-accelerated Double Distribution Function Lattice Boltzmann Schemes for Heat Transfer and Phase Change in Turbulent Flows

Master Thesis

by

Mees Wortelboer

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday July 20, 2023 at 11:00 AM.

Student number: 5612241
Project duration: September 5, 2022 – July 12, 2023
Thesis committee: Dr. Ir. M. Rohde, TU Delft, supervisor
Dr. Ir. D. Lathouwers, TU Delft
Dr. Ir. J. Peeters, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



Abstract

Generation IV nuclear reactors are a vital tool for meeting the future demand of renewable energy. Heat exchangers are a key component for the functioning of these systems and heat transfer correlations are a widely employed tool in safety analysis and reactor design. Experimental and numerical research to find these correlations for different geometries and working fluids remains an active area of research. Due to the properties of the coolants and possible conditions within the heat exchangers, turbulent flows and freezing of coolant are feasible in multiple generation IV systems in transient conditions. Thus accurate heat transfer correlations covering both turbulence and freezing are highly relevant for the realisation of this new generation of nuclear power plants. This thesis aims to contribute to the available methods for determining heat transfer correlations for application in generation IV nuclear reactors.

To this end, a GPU-accelerated double distribution function Lattice Boltzmann (GPU-DDF-LB) model was implemented. A filter matrix (FM) collision operator was implemented for momentum flow. For the evolution of the total enthalpy distribution function, an optimal two-relaxation-time (OTRT) collision operator was compared to a thermal FM operator. The implementation of a thermal FM operator with a seven speed velocity scheme has not yet been presented in literature. Tests were performed for heating and freezing simulations in laminar flow and heating simulations in turbulent flow. A preliminary calculation of the heat transfer correlation for turbulent flow of lead-bismuth eutectic in an asymmetrically heated channel was performed.

In the laminar flow regime the results of the current model closely matched analytical solutions in the case of single sided heating. The operators for thermal flow showed similar results. In the case of transient freezing, the FM operator performed better than the OTRT operator. The FM operator produced a stable ice layer that relatively closely matched the analytical solution. The OTRT operator produced a more ragged ice layer with a large error compared to the analytical solution. Stability tests showed increasing the Reynolds number leads to large fluctuations and instabilities for both operators. These effects are found to decrease with decreasing Prandtl number and decreasing maximum numerical flow velocity.

The GPU implementation allowed direct numerical simulations of turbulent channel flows to be conducted within a time frame of several hours up to several days. The statistics of the turbulence produced by the FM collision operator for momentum flow shows good agreement to benchmark results. In the turbulent flow regime, the FM operator for thermal flow also performed better than the OTRT operator. It was found that both operators can reproduce average temperature profiles for asymmetrical heating with reasonable accuracy. However, both collision operators produced unphysical temperature fluctuations. The fluctuations produced by the OTRT operator were roughly five times larger than those produced by the FM operator. These fluctuations have been characterised and their influence in the case of single sided heating has been analysed. The unphysical fluctuations influence the fluctuating temperature field, making the results for both operators inaccurate. The results of the FM operator could be improved by applying a correction, whereas the results of the OTRT operator remained very inaccurate.

Finally a preliminary calculation of the heat transfer correlation for lead-bismuth eutectic flowing turbulently in a channel, asymmetrically heated with a constant wall temperature was performed. The results give lower Nusselt number values compared to correlations for pipe flow, which coincides with results from literature for constant wall heat flux. A stronger dependence of the Nusselt number on the Peclet number is observed compared to pipe flow, which is unexpected. More research is needed to assess the validity of the obtained correlation.

Preface

This master thesis represents the completion of my two year MSc. Applied Physics at the Technical University of Delft. I would like to thank my supervisor Martin Rohde, for accepting me into the research group of Reactor Physics and Nuclear Materials and for supervising my work over course of the last year.

Only during the first year of my masters, did I acquire the interest in nuclear physics and transport phenomena. It was triggered and stimulated by teachers such as Martin Rohde and his colleagues at the Reactor Institute, who in this way have been instrumental in my academic and personal development. I feel very fortunate to have been given the opportunity to combine these new found interests with my interest in numerical simulations, and to dedicate the final research of my academic career to a very small part of the greatest global problem of our time. The Msc. Applied Physics has provided me the opportunity to learn what I wanted to learn and allowed me to discover which direction I want to take.

This year has by no means been without challenge, and I would not have been able to complete it without the help of those around me. I want to extend my gratitude to my fellow students and researches of the RPNM research group. Although I spend much of this year working from Amsterdam, coming to the Reactor Institute and meeting people in the same ordeal as myself was always good for morale. I want to thank Anand for his help with the LB boundary conditions and Celeke and Thorben, who did research upon which I could build my own.

I want to thank all my friends and family who have helped me keep my spirits up during the past year. Lisa showed me that although her research into the representation of sensuality in Sex and the City could not lie further from my own topic, the grind of writing a master thesis is universally shared among disciplines. Daniel, both friend and fellow physics student, with whom I worked closely on parts of the numerical work and discussed all the highs and lows of the last two years. My girlfriend Stephanie, who has supported me and was always ready to listen to me when I was up or down and without whom I could not have finished this work. And all others who have in some way, shape or form helped me to keep smiling.

*Mees Wortelboer
Amsterdam, July 2023*

Contents

Abstract	i
Preface	ii
Nomenclature	v
1 Introduction	1
1.1 Generation IV Liquid Metal and Molten Salt Systems	2
1.1.1 Lead-cooled Fast Reactor	2
1.1.2 Molten Salt Reactor	2
1.1.3 Sodium-cooled Fast Reactor	3
1.2 Previous Research on Heat Transfer Correlations	3
1.3 Lattice Boltzmann Method	4
1.4 Thesis Goals	5
1.4.1 Research Questions	5
1.4.2 Outline	5
2 Theoretical Background	6
2.1 Governing Equations	6
2.1.1 Mass and Momentum Conservation	6
2.1.2 Energy Conservation	7
2.2 Non-dimensional Numbers	7
2.3 Turbulence	8
2.3.1 Characteristics of Turbulence	9
2.3.2 Turbulence Statistics	9
2.4 Kinetic Theory	11
2.4.1 Boltzmann Equation	11
2.5 Parallel Programming on Graphical Processing Units	11
2.5.1 Parallel Programming using CUDA	12
2.5.2 Three Hierarchies of NVIDIA GPU's	12
3 Numerical Methods	15
3.1 Lattice Boltzmann Basics	15
3.1.1 Discretizations & Velocity Sets	15
3.1.2 Lattice Boltzmann Equation - Collision & Streaming	16
3.1.3 Thermal Lattice Boltzmann Methods	17
3.2 Collision Operators	18
3.2.1 Filter-Matrix Operator for Momentum Flow	19
3.2.2 Collision Operators for Thermal Flow	20
3.3 Boundary Conditions	22
3.3.1 Solid Walls	22
3.3.2 Hydrodynamic Inlet	22
3.3.3 Thermal Inlet	22
3.3.4 Outlet	23
3.3.5 Phase Interface	23
3.4 GPU Accelerated Lattice Boltzmann Method	23
3.5 The Algorithm	25
3.5.1 Initialisation and Main Loop	25
3.5.2 Collision	27
3.5.3 Propagation and Boundary Conditions	27

4	Laminar Flow	28
4.1	Boundary Conditions	28
4.2	Poiseuille Flow	29
4.2.1	Analytical Solution and Initialisation	30
4.2.2	Results	30
4.3	Local Nusselt Number for Heating in Forced Convection	31
4.3.1	Initialisation	31
4.3.2	Results	32
4.4	Transient Freezing	34
4.5	A Note on the Stability of the Thermal Flow Models	36
4.6	Laminar Benchmark Conclusion	38
5	Turbulent Flow	39
5.1	Parallel Performance	39
5.2	Turbulent Channel Flow	40
5.2.1	Initialisation	41
5.2.2	Low <i>Re</i> Turbulence	42
5.2.3	High <i>Re</i> Turbulence	43
5.2.4	Turbulence Benchmark Conclusion	45
5.3	Thermal Benchmark in Turbulent Channel Flow	45
5.3.1	Constant Temperature	45
5.3.2	Single Sided Heating	47
5.3.3	Thermal Benchmark in Turbulent Flow Conclusion	48
5.4	Heat Transfer Correlation	49
5.4.1	Initialisation	49
5.4.2	Results	49
6	Conclusion	52
6.1	Conclusion	52
6.2	Recommendations	53
	References	55
A	Thermodynamic Properties	62
A.1	Water	62
A.2	Lead-bismuth Eutectic	62
B	Ice Layer Development	63

Nomenclature

Symbols

Symbol	Definition	Unit
c_i	discrete lattice velocity	ls/lt
c_s	lattice speed of sound	ls/lt
C_p	specific heat capacity	J/kgK
D_h	hydraulic diameter	m
E_{ik}, E_{ki}	filter matrices	
\mathcal{E}	error	
\mathbf{f}	body force	N/m^3
f_i	density distribution function	
f_i^{eq}	discrete equilibrium density distribution function	
f_l	liquid fraction	
\mathbf{g}	body force acceleration	m/s^2
g_i	enthalpy distribution function	
g_i^{eq}	discrete equilibrium enthalpy distribution function	
H	total enthalpy	J/kg
h	sensible enthalpy	J/kg
h_c	heat transfer coefficient	W/m^2K
\mathcal{H}	channel half-height	m
\mathcal{L}	macroscopic length scale	m
N	number of [<i>subscript</i>]	
P	pressure	Pa
T	temperature	K
t	time	s
\mathcal{T}	relaxation time	lt
\mathbf{u}	velocity	m/s
u, v, w	x, y, z velocity components	m/s
u_τ	wall shear velocity	m/s
\mathcal{U}	macroscopic velocity scale	m/s
\mathbf{q}''	conductive heat flux	J/m^2
α	thermal diffusivity	m^2/s
α_k	moment space solution vector for f_i	
β_k	moment space solution vector for g_i	
Δt	lattice time step	
Δx	lattice spacing	
λ	thermal conductivity	W/mK
ν	kinematic viscosity	m^2/s
ρ	density	kg/m^3
τ	stress	Pa
Ω_i	discrete collision operator	$1/s$
ω	vorticity	$1/s$
ω_i	weight of velocity vector	

Abbreviations

Abbreviation	Definition
D3Q7	three dimensional seven speed velocity set
D3Q19	three dimensional nineteen speed velocity set
DDF	Double Distribution Function
DNS	Direct Numerical Simulations
FM	Filter Matrix
GPU	Graphical Processing Unit
LBE	Lead-Bismuth Eutectic
LBM	Lattice Boltzmann Method
MRT	Multiple-Relaxation-Time
OTRT	Optimal-Two-Relaxation-Time
RMS	Root-Mean-Squared
SRT	Single-Relaxation-Time
TRT	Two-Relaxation-Time

Non-Dimensional Numbers

Symbol	Definition	Formula
B	ratio bulk viscosity to kinematic viscosity	ζ/ν
$Gz(x)$	local Graetz number	$RePrDh/x$
Nu	Nusselt number	$h_c\mathcal{L}/\lambda$
Pe	Peclet number	$U\mathcal{L}/\alpha$
Pr	Prandtl number	ν/α
Re	Reynolds number	$U\mathcal{L}/\nu$
Re_τ	wall shear Reynolds number	$u_\tau\mathcal{H}/\nu$
t^+	time	tu_τ/\mathcal{H}
u^+	velocity	u/u_τ
z^+	channel height	zu_τ/ν

1

Introduction

Global energy consumption has been rapidly increasing over the course of the last century and is expected to keep growing [1]. The need for carbon free energy production can not be understated. While carbon free sources of energy, such as wind, solar and hydro power are an active area of research and development, these resources generally do not produce a stable power output and are not yet able to cover the entire world energy demand. In the recent IPCC reports on the possible scenario's regarding climate change, increase in energy production by nuclear power plants is mentioned as an important part of all scenario's that keep the temperature increase below $2.0\text{ }^{\circ}\text{C}$ in 2060 [2].

Nuclear power production is an area of constant development. Since the world's first nuclear power plant became operational in 1951 [3], researches have continued to devise ways of making nuclear reactors safer and more efficient. The most promising nuclear energy systems that are currently being designed are referred to as 'generation IV nuclear reactors'. These reactors are designed to excel in safety and reliability and reduce long-lived waste production, making nuclear power cleaner, safer and economically more attractive. The Generation IV International Forum (GIF) is a collaboration between thirteen countries and the European Union to coordinate global efforts to realise this next generation of nuclear energy systems [4]. Six systems were selected: the Gas-cooled fast reactor (GFR), the Lead-cooled fast reactor (LFR), the Molten salt reactor (MSR), the Supercritical-water-cooled reactor (SCWR), the Sodium-cooled fast reactor (SFR) and the Very-high-temperature reactor (VHTR).

Heat transfer processes in fluid flows play a central role in the functioning of all of these reactor designs. A core component of any nuclear reactor is the heat exchanger. These components transfer the heat generated in the core to secondary and tertiary circuits, where heat is converted to electrical power or potentially used directly in industrial processes. Elaborate analysis of heat exchangers and the hydro- and thermodynamical processes therein, through both numerical and experimental work is a crucial part of the safety analysis and design process for generation IV nuclear reactors [5]. Heat transfer correlations are a widely employed tool to approximate heat flux between a fluid and a surface. They generally depend on the geometry of the flow, the properties of the working fluid and the operating conditions. While laminar flow characteristics can be calculated using analytical methods with few assumptions, turbulent flows require experimental data or numerical simulations [6]. Accurate heat transfer correlations allow the complex physical processes to be captured in much simpler mathematical expressions, greatly aiding design processes.

This thesis aims to contribute to the available methods for determining heat transfer correlations for application in generation IV LFR, SFR and MSR design and safety analysis. To this end research has been done on the development of a numerical model capable of simulating turbulent thermal flows through channels, incorporating freezing effects. The selected numerical scheme is the Lattice Boltzmann Method (LBM). The remainder of this chapter provides a brief overview on history and previous research on the relevant topics. Section 1.1 gives a brief introduction to a selection of generation IV systems, after which section 1.2 provides an overview on previous research on heat transfer correlations. Section 1.3 then gives an introduction to LBM. Finally section 1.4 formulates the goals and the research questions for the present work.

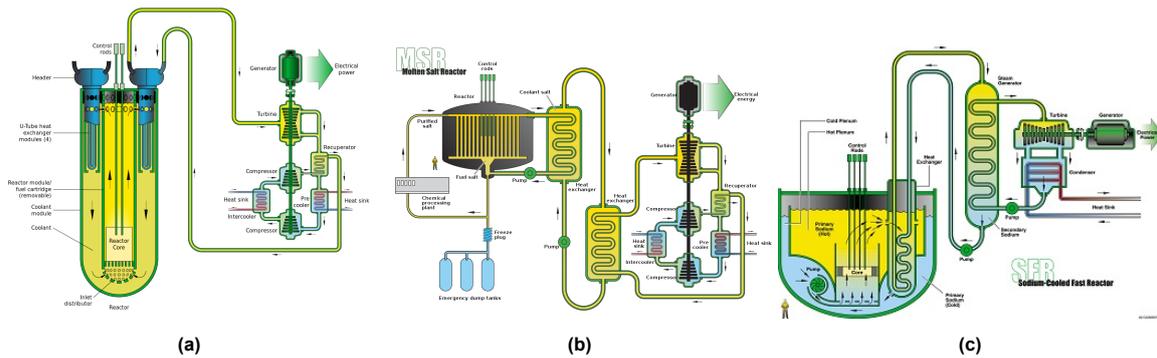


Figure 1.1: Schematics of (a) the Lead-cooled Fast Reactor [7], (b) the Molten Salt Reactor [8] and (c) the Sodium-cooled Fast Reactor [9]

1.1. Generation IV Liquid Metal and Molten Salt Systems

The generation IV nuclear energy systems are selected for their potential to meet standards set in the areas of sustainability, economics, safety/reliability and proliferation resistance [4]. The LFR, MSR and SFR systems make use of a primary coolant which is solid under ambient conditions. This section serves as a short introduction into these systems.

1.1.1. Lead-cooled Fast Reactor

The LFR (figure 1.1a) utilizes a fast neutron spectrum and is cooled by molten lead or lead-bismuth eutectic (LBE). Lead and lead-bismuth have excellent cooling properties, while exhibiting low neutron absorption or moderation tendencies. Both substances are chemically inert, liquid at normal operating temperatures, thus eliminating the need for high pressurisation, have a very high boiling point and are abundant in nature. These attributes allow better resource utilisation, longer core life, effective burning of minor actinides and open fuel pin spacing, which are all important factors in achieving sustainability, proliferation resistance, fuel cycle economics and enhanced passive safety mechanisms [7] [10].

Lead-cooled reactor technologies have been under research for more than 50 years. During the period from mid 1960s until 1990s the Soviet Union and later the Russian Federation gained significant industrial and operational experience with lead or lead-bismuth eutectic (LBE) cooled reactors. 12 reactors and 15 reactor cores were deployed in this period, with a focus on reactors for submarine propulsion. The Russian programme represents roughly 80 reactor-years of operating experience [11]. Modern research efforts include the BREST-OD-300 by the Russian Federation [12], ELFR by the European Union [13] and the SSTAR by the United States [14].

1.1.2. Molten Salt Reactor

In the MSR (figure 1.1b) the fuel is dissolved in a liquid lithium fluoride salt. Some advantages of liquid fuel are homogeneity of the fuel, heat production directly in the heat transfer fluid, possibility to easily reconfigure the geometry of the fuel to switch between optimized power production and storage with a passive cooling system based on natural circulation and the possibility to reprocess the fuel without stopping the reactor [15]. The candidate fuel mixtures of the MSR are liquid under normal operating temperatures, removing the need of pressurisation and increasing safety similarly to the LFR.

The first experimental research on MSR technologies go back to the 1960's in the United states. In the Oakridge National Laboratory the Molten Salt Reactor Experiment ran from 1965 up to 1969, logging more than 13 thousand hours at full power in this short period of time and acting as a prove of concept [16]. In the United Kingdom a large lead-cooled fast MSR using plutonium fuel dissolved in chloride salt was designed and experiments where undertaken from 1968 to 1973, after which funding was terminated [17]. In the 2014 Roadmap the GIF stated that much work is needed on molten salt technology before demonstration reactors are operational. The SAMOSAFER consortium [18] presents the efforts of the European Union in this regard. It aims to develop and demonstrate safety barriers for the MSR. Their objective is to ensure the MSR complies with all safety regulations near 2050. Several commercial initiatives exist, such as the Dutch start-up Thorizon [19] or the Danish Copenhagen

Atomics [20]. Construction of the Chinese experimental TMSR-LF1 reactor [21] was started in 2018 and was reported to have been concluded only 3 years later. Recently an operating licence for this experimental reactor was granted by the Chinese government [22].

1.1.3. Sodium-cooled Fast Reactor

The SFR (figure 1.1c) makes use of liquid sodium as the primary and secondary coolant. This allows a high power density with a low coolant volume fraction. Sodium reacts chemically with water and air, thus the SFR requires a sealed coolant system and oxygen-free environment. The oxygen-free environment prevents corrosion of the components. Important safety features include a reasonable margin to coolant boiling, a long thermal response time and a primary system that operates near atmospheric pressure [9].

Much of the SFR technology stems from former fast reactor programmes. The Dounreay Fast Reactor [23], located in Dounreay United Kingdom, started operation in 1960 and became the first fast spectrum nuclear reactor to supply electricity to a national power grid. It closed down in 1977, shortly after the Prototype Fast Reactor [24], also located in Dounreay, was connected to the national grid and become the worlds second connected fast spectrum reactor. The PFR was shut down in 1994. Both the DFR and the PFR systems were sodium cooled. Other examples of prototype SFR's that contributed to the current SFR knowledge are the Phénix and Super Phénix (France), the EBR-II (United States), the Monju (Japan) and the BN-600 (Russia) [25]. Of these systems, only the BN-600 is still in operation. Modern research efforts for SFR's include the Prototype Generation IV SFR (PSFR) by the Republic of Korea [26], the Japan SFR (JSFR) [27], the European SFR (ESFR) [28] and programmes in the United States [29].

These three generation IV systems utilize a coolant which is liquid under normal operating temperatures, but solid under ambient conditions. While this eliminates the need for core pressurization and allows for passive safety measures such as a freeze plug emergency draining system [30], it can lead to problems in transient conditions, such as the start-up, shut-down or in case of accidents. As temperatures drop below the freezing temperatures of the coolants, these fluids may start to freeze in valves or the heat exchangers, causing blockages and potentially damage the systems [31] [11]. Turbulent flows are feasible in heat exchangers for both molten salt and liquid metal coolants [5] [32], thus analysis of turbulent flows through heat exchanger geometries incorporating transient freezing is an important area of research for safety and efficiency analyses of generation IV nuclear reactors. Accurate heat transfer correlations for the relevant working fluids under specific operating conditions will be an important tool for the design of the nuclear reactors of the future. The next section will give an introduction of previous research on these topics.

1.2. Previous Research on Heat Transfer Correlations

Because of their practical importance heat transfer correlations have been actively researched of the course of the last century. This section provides an overview of some important works. In 1933 the first method for correlating forced convective heat transfer and pressure drop was proposed by Colburn [33] where he linked the Nusselt number (Nu) to the Reynolds (Re) and Prandtl (Pr) number for a wide range of both parameters. Since then much work has been done analytically, experimentally and numerically for various fluids and flow conditions to provide more detailed correlations. In 1978 Shah and London [34] published an extensive work on analytically obtained expressions for heat transfer correlations for laminar flow through pipes and square ducts geometries. Their expressions match found empirical relations [35] [36] [37] with good accuracy. A systematic summary of these relations is presented in [38].

For turbulent flows more work has been done for circular pipe geometries compared to channel geometries. Dittus and Boetler [39] proposed a simple correlation for high Re and medium to high Pr for pipes. Barnes and Jackson [40] performed heat transfer experiments in turbulent pipe flows with several gasses for a wide range of Re . Their expression for airflow was numerically tested with satisfying results [41]. Gnielinski made several contributions to the field proposing correlations and interpolation methods for the transitional turbulent regime [42] [43] [44]. More recently Taler [45] proposed correlations for Pr ranging from 0.1 to 1000 and Re ranging from 10^3 to 10^6 , achieving good results compared to literature and simple expressions. For symmetric heating in channel and square duct flow several correlations are presented [46] [47] [48] [49]. An extensive overview of these works is presented by David et al.

in [6]. In the same paper, David et al. present a correlation for asymmetrical heating. The proposed correlation in the above mentioned researches all consider fluids with Pr ranging $\mathcal{O}(0.1)$ to $\mathcal{O}(100)$. Liquid metals however are known to have lower Pr , leading to different heat transfer behaviour. Thus different correlations are needed.

The number of published works regarding heat transfer in liquid metals is relatively low, but recently more interest is seen in the topic in light of the research into liquid metal cooled reactors. In [50], [51] and [52] heat transfer to liquid metals in tube bundles were investigated. Cheng and Tak [53] investigated turbulent heat transfer to LBE flows in circular tubes, specifically for nuclear applications. Taler [54] proposes several new correlations for Pr ranging from 10^{-4} to 10^{-1} and Re ranging from 10^3 to 10^6 , extending his earlier work. His correlations correspond well to experimental data. An overview of empirical work with twelve different metals in a variety of geometries is presented in [55].

The presented studies do not consider the freezing of liquid metals in turbulent flows. Little work can be found on the the combination of freezing and turbulence. Recently phase-change emulsions have attracted attention and work has been done in this context [56] [57] [58]. However, the consistency of these emulsions and its applications are not applicable to target fluids for the generation IV systems. Gilpin et al. [59] present work on the instability of an ice-water interface in the presence of turbulent flow. Shibani and Özisik [60] present work on the steady-state freezing of high Pr liquids in pipes. However, generally accepted heat transfer correlations for the target fluids of the generation IV systems in turbulent flow under phase change conditions have yet to be proposed. The development of a versatile numerical model for the calculation of heat transfer correlations in forced convective turbulent channel flows incorporating freezing will therefore be a contribution to the current state of knowledge and aid in the design of the next generation of nuclear reactors.

1.3. Lattice Boltzmann Method

The Lattice Boltzmann Method is a numerical method for simulating fluid flows that has gained attention in the recent years as an alternative to conventional Navies-Stokes based CFD methods. Fluids are described as populations of particles moving through space and colliding with other populations. Due to a relatively simple basic algorithm, the LBM is a very versatile method for simulating a variety of flow situations [61]. Recent applications of the LBM include flows through porous media [62] [63], dendritic solidification [64] [65], and turbulent flows with Direct Numerical Simulations (DNS) [66] [67], Large-Eddy Simulations with various subgrid models [68] [69] and RANS methods [70] [71].

Another major advantage of LBM over conventional CFD solvers is its inherent parallelizability. Due to the structure of the algorithm, local and non-local processes are naturally uncoupled, allowing relatively straight forward implementation on massively parallel computational architectures like Graphical Processing Units (GPU) [72] [73]. These factors are the main driving force behind the increased interest taken in LBM in recent years.

All the before mentioned papers use either single-relaxation-time (SRT) or the multiple-relaxation-time (MRT) methods for modelling the inter particle collisions. SRT models are simple to implement but suffer from instabilities for low viscosity [74]. MRT models are more stable but require the selection of free parameters that significantly influence the result. Since there is no straightforward method to determine these parameters this is a significant downside of MRT models [61]. Two methods that more stable than SRT methods but more straightforward to implement are the two-relaxation-time (TRT) [75] method and the filter matrix (FM) method [76] [77] [78]. The present research uses the FM method for momentum flow and compares the performance of an Optimal TRT (OTRT) method to that of a FM method for thermal flow, combining these in two distinct GPU accelerated double distribution function Lattice Boltzmann (GPU-DDF-LB) models

Both the tested methods for thermal flow have not yet been tested on freezing problems in three-dimensional (3D) forced convective flows. The TRT method employed was adapted from [75], who tested it for solid-liquid phase change in laminar natural convective flows in a square cavity. The FM method was developed very recently by Besseling [79], who tested it for solid-liquid phase change in two-dimensional (2D) laminar natural convection. It was later used by Bus [80] who simulated forced convective freezing of eutectic and non-eutectic fluids in 2D laminar forced convection. Both methods have not yet been implemented on forced convection in 3D, laminar or turbulent. Neither the TRT or the FM method has been implemented to run on a GPU. To contribute to the current standing of the LBM knowledge, the present work tests the performance of both methods by simulating 3D forced

convection driven laminar and turbulent flows in a GPU based algorithm.

1.4. Thesis Goals

The goal of this thesis is to contribute to the determination of heat transfer correlations for nuclear applications. Fluids of interest are liquid metals (low Prandtl number) and molten salts (high Prandtl numbers). To this end, the aim is to develop a versatile and computationally efficient double distribution function Lattice Boltzmann (DDF-LB) model for simulating turbulent thermal flows under the influence of solid-liquid phase change. The following research questions were formulated.

1.4.1. Research Questions

1. Methodology

- How can a DDF-LB model in combination with a source-based enthalpy scheme for solid-liquid phase change be implemented on a GPU to perform direct numerical simulations of turbulent thermal flows within a reasonable time frame?
 - What are necessary changes to the algorithm for implementation on a GPU?
 - What are suitable input parameters and boundary conditions?
 - How much does the GPU implementation decrease computational cost compared to a CPU implementation?
- How do the OTRT and FM collision operators perform in combination with the source based enthalpy method to simulate freezing in turbulent forced convection?

2. Heat transfer correlations

- How does the Nusselt number depend on the Reynolds and Prandtl number in a turbulent channel flow in the presence of single sided freezing for low and high Pr fluids?

1.4.2. Outline

This concludes the introductory chapter. To test the developed algorithm and answer the research questions, benchmark cases are simulated and the results evaluated. The rest of this report will have the following structure:

- **Theoretical Background** discusses the fundamental physics governing the problems under consideration. The basic theory of parallel programming on GPU's is also discussed.
- **Numerical Methods** covers the specifics of the numerical methods used in the algorithm. The basics of LBM will be covered, followed by the specific implementation of the used collision operators, boundary conditions and the adjustments to the algorithm necessary for the GPU algorithm. This chapter concludes with a step-by-step overview of the algorithm.
- **Laminar Flow** presents the results of the performed benchmark cases in the laminar flow regime. The development to Poiseuille flow is discussed, followed by the addition of the thermal models in calculating local heat transfer in single sided heating. Finally the performance of the models for transient freezing is shown. The results are compared to analytical solutions found in literature. Findings on the influence of boundary conditions and input parameters are also discussed.
- **Turbulent Flow** presents the results of the benchmark cases performed in the turbulent flow regime. First the performance of the momentum flow model is shown and compared to established benchmark results from literature. After which the performance of the thermal flow models is evaluated using another established benchmark case from literature. Finally a preliminary calculation of the heat transfer correlation of LBE flowing turbulently through an asymmetrically heated channel is performed.

2

Theoretical Background

This chapter is dedicated to the fundamental theory that is important for simulating thermal flows and phase change processes in forced convection. Section 2.1 covers the continuum description of fluid flow and thermal flow, arriving at the Navier-Stokes equations for fluid flow and the total enthalpy equation for thermal flow. After this, section 2.2 gives an overview of the most important non-dimensional numbers for this research. In section 2.3 the turbulent flow regime is discussed. Section 2.4 then deals with the description of fluids at a different scale by introducing the basics of kinetic theory, which is where the LBM has its origins. Finally section 2.5 moves away from physics and goes into the important theoretical and physical concepts of parallel programming on graphical processing units.

2.1. Governing Equations

The central topics of this research are part of the fields of fluid mechanics and thermodynamics. These fields are concerned with describing fluids on a macroscopic scale. It would in principle be possible to describe the behaviour of a fluid by calculating the individual motion of its molecules. However, the average manifestation of these motions is more important for the macroscopic behaviour of the fluid and knowing the exact state of each particle is an unnecessary level of detail for most purposes. When the molecular density and region of interest is sufficiently large, the fluid can be described as a *continuum* and macroscopic properties such as density, velocity, pressure and temperature are defined everywhere and continuous. This continuum approximation is valid when the Knudsen number, defined as

$$Kn = l/\mathcal{L}, \quad (2.1)$$

where l is the molecular mean free path and \mathcal{L} is the length scale of the region of interest, is much smaller than unity. For water at room temperature $l = \mathcal{O}(-12)$, thus for most situations this is a good approximation to make [81]. Using the continuum approximation, conservation laws can be formulated for mass, momentum and energy. This section gives an overview of these equations

2.1.1. Mass and Momentum Conservation

Conservation of mass is described by the continuity equation. The rate of change of the mass in a control volume is equal to the mass flow rate over its boundaries. Neglecting the presence of sources or sinks, such as fission processes, this is given

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (2.2)$$

where ρ is the fluid density and \mathbf{u} is the fluid velocity [82].

Assuming conservation of momentum, the rate of change of momentum in a control volume is equal to the flow of momentum over its boundaries, body forces working on the element and stresses working on the boundaries of the element. Assuming an incompressible medium, this is described by the Cauchy momentum equation [82]:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}, \quad (2.3)$$

where σ is the stress tensor and f is the body force per unit volume. For incompressible flow of Newtonian fluids the stress tensor simplifies greatly and the Cauchy momentum equation reduces to the Navier-Stokes momentum equation:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u} + \mathbf{g}, \quad (2.4)$$

where P is the pressure, ν is the fluids kinematic viscosity and \mathbf{g} is the sum of the body force accelerations working on the fluid element [81].

Together with the continuity equation, this set of 4 equations is known as the incompressible Navier-Stokes equations. In combination with an appropriate set of boundary conditions and an initial condition these equations are in principle enough to completely determine the evolution of a flow field in a defined volume as a function of time [83]. In practice however, the nonlinear nature of the momentum equations makes it extremely difficult to find analytical solutions for flow problems in all but the simplest cases [84]. This will be discussed further in section 2.3.

2.1.2. Energy Conservation

Similarly to mass and momentum, a conservation equation can be formulated for the internal energy of a system. For the purposes of describing phase change, total enthalpy is a convenient description of the energy of a system. The total enthalpy H is a combination of the sensible enthalpy h and the latent heat L . The sensible enthalpy h is a form of internal energy often used in phase change problems, defined as

$$dh = C_p dT, \quad (2.5)$$

where C_p is the specific heat capacity of the fluid and T is the temperature. Generally C_p depends on the phase of the material and T . The rate of change of the sensible enthalpy of a system, without considering viscous dissipation or shock waves, is described by [85]

$$\rho \frac{\partial h}{\partial t} + \rho \mathbf{u} \cdot \nabla h = -\nabla \cdot \mathbf{q}'' + q''', \quad (2.6)$$

where \mathbf{q}'' is the conductive heat flux over the boundary and q''' is a volumetric heat source.

During phase transitions intermolecular bonds are formed or broken. The energy associated with solid-liquid phase change is called the latent heat L . The total enthalpy is defined as

$$H \equiv h + f_l L, \quad (2.7)$$

where f_l is the liquid fraction at each point in the domain. The conservation equation for total enthalpy can be derived from equation (2.6). In this derivation it is assumed that the source term accounts for the change in L , the density of the material is approximately constant for each phase, the specific heat is approximately constant for each phase and the advection of latent heat is negligible. This last assumption means changes in f_l can be incorporated in the time derivative only. These assumptions lead to the following equation for total enthalpy [86],

$$\frac{\partial H^k}{\partial t} + \mathbf{u} \cdot \nabla h^k = \nabla \cdot (\alpha^k \nabla h^k), \quad (2.8)$$

where α is the thermal diffusivity of the material and the superscript k denotes the phase of the material. By solving this equation, the temperature and liquid fraction are tracked in tandem, meaning the phase interface is naturally tracked.

Now that the governing equations have been introduced, the next section will cover the most important non-dimensional numbers describing fluid and thermal flows. These numbers will play an important role in this thesis.

2.2. Non-dimensional Numbers

In the fields of fluid mechanics and thermodynamics non-dimensional numbers play an essential role. They allow the description of processes independent of physical scales. In the field of computational physics they are a tool for describing processes when certain parameters are subject to numerical

constraints. This will become apparent in the chapter about the LBM. In the present work the three most important non-dimensional numbers are the Reynolds number Re , the Prandtl number Pr and the Nusselt number Nu . This section gives an introduction of these quantities.

The Reynolds number Re is a dimensionless quantity that helps to predict certain patterns of the flow. It gives a measure for the ratio of the inertial forces versus the viscous forces and is defined as

$$Re = \frac{U\mathcal{L}}{\nu}, \quad (2.9)$$

where U and \mathcal{L} are a characteristic velocity and length scale of the flow. A small value of Re means the flow is dominated by viscous forces. This regime is called the laminar flow regime which is characterised by parallel and symmetric motions. A large value of Re means the flow is dominated by inertial forces. This regime is called the turbulent flow regime, characterised by chaotic and swirling motions [83]. This will be further discussed in the next section.

Re can be formulated using different characteristic velocities and length scales. A formulation that is often used in turbulence research is the so-called wall shear Reynolds number Re_τ . It is defined using the wall shear velocity. For channel flow the wall shear velocity is defined as

$$u_\tau = \sqrt{\frac{\tau_s}{\rho}} = \sqrt{g\mathcal{H}}, \quad (2.10)$$

where τ_s is the averaged wall shear stress, g is the acceleration in the flow direction and \mathcal{H} is the channel half-height [83]. The second expression comes from the stationary macroscopic force-balance in a channel. The wall shear Reynolds number is now defined as.

$$Re_\tau = \frac{u_\tau\mathcal{H}}{\nu}. \quad (2.11)$$

The Prandtl number Pr is a material property giving the ratio between momentum transport and heat transport capacities of a fluid. It is defined as

$$Pr = \frac{\nu}{\alpha}. \quad (2.12)$$

A low Pr means heat conducts quickly through the fluid compared to the rate of momentum transfer by viscous effects. This typically indicates effective thermal transport. Thus for heat transfer applications low Pr fluids, such as liquid metals, are desirable [87].

The Nusselt number Nu describes the ratio between convective heat transfer and conductive heat transfer at a fluid boundary. It is defined as

$$Nu = \frac{h_c\mathcal{L}}{\lambda}, \quad (2.13)$$

where h_c is the heat transfer coefficient and λ is the thermal conductivity. For a static fluid $Nu = 1$, as there are no convective mechanisms other than pure conduction. A high value of Nu indicates that convective mechanisms are effectively transporting heat away from the surface, indicating efficient heat transfer [88].

These three non-dimensional quantities lie at the heart of heat transfer correlations. With Re describing the flow properties and Pr describing the molecular transport capabilities of the fluid, heat transfer correlations are generally of the form $Nu = f(Re, Pr)$. This allows correlations to be used independent of physical dimensions across different fields of physics and engineering.

Now that the most important non-dimensional numbers for heat transfer in fluid flows have been discussed, the next section is dedicated to the description of the flow regime characterised by high Re values: turbulence.

2.3. Turbulence

To discuss heat transfer in turbulent flows it is important to have an understanding of the phenomenology of turbulence and the methods to measure it. This section will give a brief introduction to the characteristics of turbulence and the statistics used to quantify it.

Turbulence is a so-called 'ill-posed problem': tiny variations in the initial and boundary conditions can lead to radically different behaviour of the system. Thus to completely predict the flow, these conditions need to be known with infinite accuracy. Because realistic initial and boundary conditions can only be known with finite accuracy, it is not possible to exactly know the flow behaviour at any given time. This unpredictability comes from the nonlinearity of the governing equations, specifically the $(\mathbf{u} \cdot \nabla)\mathbf{u}$ term, and is one of the defining properties of turbulent flows [83]. This section goes deeper into the properties of turbulence. First a phenomenological description of turbulence is given, after which the statistical tools used to measure and quantify turbulence are discussed.

2.3.1. Characteristics of Turbulence

Besides their unpredictable nature, turbulent flows have several characteristics that distinguish them from laminar flows. This section will explore these characteristics from a phenomenological point of view.

It is readily observed that turbulent flows consist of many swirling structures called vortices. The largest vortices are referred to as the macrostructure. The macrostructure is determined by the flow geometry and velocity and scales with \mathcal{U} and \mathcal{L} . As $Re \gg 1$ for turbulent flows, viscous effects are negligible at this scale and the macrostructure is independent of Re . The swirling motions of the macrostructure make turbulent flows highly effective in lateral transport. Quantities such as concentration and temperature are transported by the macrostructure and mixed quickly throughout the fluid domain [81]. Turbulence can therefore be a very effective mechanism for heat transfer.

Another characteristic of turbulent flows is that they dissipate kinetic energy. From experience we know that a stirred cup of coffee will quickly transition from highly energetic chaotic flow to more regular motions and ultimately to no motion. This dissipation happens through a process called the energy cascade. The biggest vortices become unstable and break up into smaller vortices. This process repeats itself until finally the vortices become so small that viscous effects become dominant and kinetic energy is dissipated through friction into heat. This scale is referred to as the microstructure. The length, time and velocity scales of the microstructure can be constructed from the viscosity ν and the rate of kinetic energy dissipation ϵ . These scales are known as the Kolmogorov scales and are given by

$$\eta = \left(\frac{\nu^3}{\epsilon}\right)^{1/4}, \quad \mathcal{T}_k = \left(\frac{\nu}{\epsilon}\right)^{1/2}, \quad \mathcal{V} = (\nu\epsilon)^{1/4}. \quad (2.14)$$

The micro- and macrostructure scales are related through [83]

$$\frac{\mathcal{L}}{\eta} \sim Re^{3/4}. \quad (2.15)$$

2.3.2. Turbulence Statistics

Because of the chaotic and fluctuating nature of turbulence a detailed description of the flow is of little importance, since it is impossible to reproduce. Luckily, for most applications it is not necessary to know these details and statistics are used to describe turbulence and its effects. This section will give an overview of the turbulence statistics used in this research to assess the accuracy of simulation results.

Averages and Non-dimensionalisation

A central tool in the statistical description of turbulence is Reynolds decomposition. A quantity ϕ is decomposed in an average and fluctuations

$$\phi = \overline{\phi} + \phi', \quad (2.16)$$

where the overline denotes the average and the apostrophe denotes fluctuations. The average is the so-called ensemble average. This average is taken over multiple realisations of the same turbulence experiment. It is defined as

$$\overline{\phi} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \phi^{(i)}, \quad (2.17)$$

where the i denotes a single experiment.

Fully developed turbulence is a statistically stationary process. This means average quantities do not change with time. Based on the ergodicity hypothesis it can be assumed that for a fully developed turbulent flow, the time average is the same as the ensemble average [83]. To collect the statistics needed for the current research, the ensemble average can be constructed from a set of instantaneous flow fields sufficiently separated in time. Several quantities are calculated in this manner to assess the reliability of the simulation results. The ensemble average of a quantity ϕ can then be calculated as

$$\bar{\phi} = \frac{1}{N_p} \sum_p^{N_p} \phi^{(p)}, \quad (2.18)$$

where p denotes a *probe*: a single instantaneous flow field where the quantity is measured and saved.

For the comparison to simulation results from literature, the flow quantities are non-dimensionalised using the wall shear velocity (2.10). The non-dimensional length, time and velocity are defined as

$$z^+ = \frac{zu_\tau}{\nu}, \quad t^+ = \frac{tu_\tau}{\mathcal{H}}, \quad u^+ = \frac{u}{u_\tau}. \quad (2.19)$$

Statistical Quantities

The first quantity used in this research is the mean streamwise velocity as a function of height. It is calculated as

$$\bar{u} = \frac{1}{N_p N_x N_y} \sum_p^{N_p} \sum_x^{N_x} \sum_y^{N_y} u. \quad (2.20)$$

For this quantity approximate analytical solutions exist in 2 regions near the wall. In the viscous sublayer, the region closest to the wall where viscous effects dominate and turbulent stresses are negligible, the average streamwise velocity follows

$$\bar{u}^+ = z^+. \quad (2.21)$$

In a region closer to the core the average streamwise velocity follows

$$\bar{u}^+ = 2.5 \ln(z^+) + 5.5. \quad (2.22)$$

The region where this holds is called the logarithmic layer [83].

To quantify the flow fluctuations two quantities are used: the turbulent intensities and the Reynolds stress. The turbulent intensities are the average root-mean-squared (RMS) fluctuations of the velocity fields, defined as

$$u'_{i,RMS} = \sqrt{\overline{u'_i u'_i}}. \quad (2.23)$$

The Reynolds stress is defined as

$$\tau_{ij}^{Re} = \overline{u'_i u'_j}. \quad (2.24)$$

This quantity arises when performing Reynolds decomposition (2.16) on the Navier-stokes equation (2.4) and can be interpreted as the transport in the j -direction of momentum per unit mass in the i -direction.

Finally, the RMS vorticity fluctuations are calculated, which provide a measure for the intensity of the swirling motions in the flow. Similar to the RMS velocity fluctuations they are calculated using

$$\omega'_{\alpha,RMS} = \sqrt{\overline{\omega'_\alpha \omega'_\alpha}}. \quad (2.25)$$

Having discussed the characteristics of turbulence and the statistical tools to measure it, the focus now shifts to a different method of describing fluids: kinetic theory, which is the origin of the numerical method used in this research.

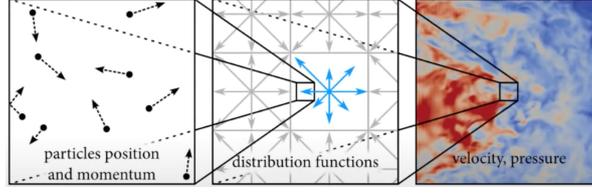


Figure 2.1: Three scales of describing fluids. From left to right: microscopic, mesoscopic and macroscopic

2.4. Kinetic Theory

The Lattice Boltzmann Method has its origins in the kinetic theory of gasses. This section gives a brief overview of the basic principles and central quantities within kinetic theory. It is a method of describing fluid flow on the mesoscopic scale (see figure 2.1). The central quantity is the particle distribution function $f(\mathbf{x}, \boldsymbol{\xi}, t)$, which describes the number of molecules at each point in space \mathbf{x} , moving with velocity $\boldsymbol{\xi}$, at each moment in time t . It can be seen as the generalisation of macroscopic density, taking into account microscopic particle velocity [89]. By taking the moments of f macroscopic properties of a system can be recovered:

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}, \quad (2.26)$$

$$\rho(\mathbf{x}, t)\mathbf{u}(\mathbf{x}, t) = \int \boldsymbol{\xi} f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}. \quad (2.27)$$

An important concept in kinetic theory is that of the equilibrium distribution function $f^{eq}(\mathbf{x}, \boldsymbol{\xi}, t)$. This captures the fact that a system will always evolve toward thermodynamic equilibrium if left undisturbed for a long enough time.

2.4.1. Boltzmann Equation

Time evolution of the distribution function is described by the Boltzmann equation:

$$\frac{\partial f}{\partial t} + \xi_\beta \frac{\partial f}{\partial x_\beta} + \frac{F_\beta}{\rho} \frac{\partial f}{\partial \xi_\beta} = \Omega(f), \quad (2.28)$$

where index notation is used and F_β/ρ is the specific body force. The first two terms on the left hand side represent the convection of f , the third term on the left hand side represents f being affected by external forces. $\Omega(f)$ is the collision operator. This term represents the change in f due to inter-particle collisions. Collision operators must respect conservation constraints such as mass and momentum conservation, and ensure f evolves towards local equilibrium f^{eq} .

By taking the moments of equation (2.28) the macroscopic conservation equations, such as equation (2.2) and equation (2.3), can be recovered. This shows that the Boltzmann equation can be used to describe macroscopic behaviour of a fluid[61].

This concludes the discussion of Kinetic Theory. The next section moves away from physics and goes into the specifics of parallel programming on Graphical Processing Units.

2.5. Parallel Programming on Graphical Processing Units

Direct Numerical Simulations of turbulent flows as performed in this research are notoriously computationally expensive. The small length and time scales of the microstructure require large numbers of spatial nodes and time steps, making such simulations highly memory and time consuming. Parallel programming is an effective and widely employed method to increase simulation speeds. A Graphical Processing Unit (GPU) is a type of processor specifically designed for this purpose. Originating from specialized graphics processors for video games, nowadays GPU's are widely used in many fields of science for high performance computing [90]. This section serves as an introduction to parallel programming on GPU's. It focuses on the Compute Unified Device Architecture (CUDA) programming language specific for NVIDIA GPU's. Section 2.5.1 covers the basics of CUDA programming, after which section 2.5.2 discusses three important hierarchies inherent to CUDA and NVIDIA GPU's.

2.5.1. Parallel Programming using CUDA

Parallel programming denotes algorithms written in such a way that they can be broken down into parts that can be executed simultaneously on multiple processor cores. When executed correctly this greatly increases computation speed. GPU's are systems specifically designed for parallel computations. They consist of thousands of cores, connected through several different forms of memory. Individually these cores are less powerful and versatile compared to conventional CPU cores and only have a limited amount of fast access memory available. An algorithm that efficiently makes use of GPU's parallel capabilities is therefore essential for high performance computing purposes.

The CUDA programming language is specifically designed for writing algorithms for NVIDIA GPU's. The algorithms are launched from the CPU, also called the host. It is broken down into kernels, which represent segments of code to be run in parallel on the GPU, also called the device. The host and the device have separate memory spaces. Because kernels can only access and change values stored in the device memory, the host and the device have to communicate information back and forward. This communication is very slow and should be minimized [91]. The general outline for a CUDA algorithm is then:

1. Initiate the system on host.
2. Store the initial state in device memory.
3. Perform system evolution steps on the device.
4. Send system state back to host for data processing.

Besides minimizing host-device communication, there are several factors important for writing fast code. The next section discusses three hierarchies of NVIDIA GPU's that are important to understand for algorithm efficiency.

2.5.2. Three Hierarchies of NVIDIA GPU's

An algorithm that efficiently makes use of GPU's parallel capabilities is essential for high performance computing purposes. To this end it is important to understand the way CUDA code interacts with the GPU hardware. This section discusses three hierarchies that play an important role in this process.

Software Hierarchy

As discussed earlier a CUDA kernel represents a segment of code that can be run in parallel. The computational domain on which the kernel is to be executed, e.g. a spatial domain, is divided into three parts: threads, blocks and grids (figure 2.2). This is done in order to control the memory access pattern and the order of execution [91].

- **Threads:** a single execution unit of a kernel
- **Blocks:** a group of threads that are executed in parallel
- **Grid:** an ensemble of blocks representing the entire computational domain on which the kernel is executed

When invoking a kernel, the number of threads per block, t_{pb} , and the number of blocks per grid, b_{pg} , have to be specified. Inefficient block and gridsizing can have a negative effect on computation speed [92].

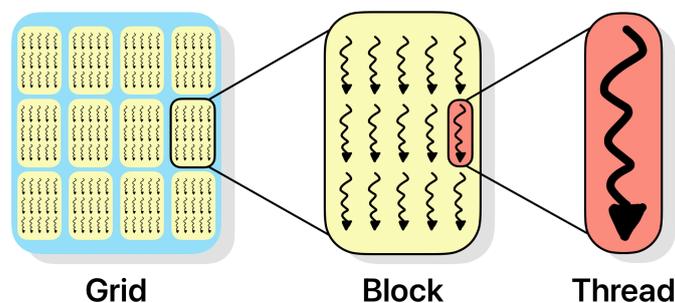


Figure 2.2: Schematic depiction of the division of a computational domain into a grid, blocks and threads.

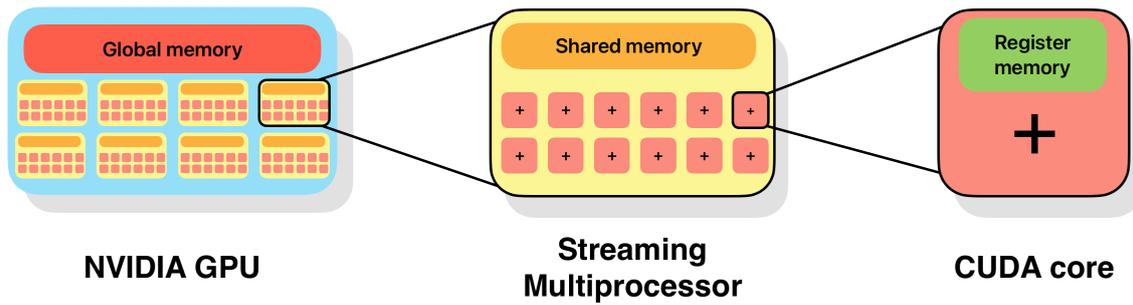


Figure 2.3: Schematic depiction of an NVIDIA GPU, showing division into streaming multiprocessors and cores and indicating locations of different forms of memory memory.

Memory Hierarchy

Within NVIDIA GPU's, different forms of memory with different characteristics exist. The different forms of memory are presented here ordered by thread access speed, starting with the fastest form of memory [91]:

- **Register memory:** Thread specific cache memory. It is located right next to the core and therefore very fast to access. It is however very limited in size.
- **Shared memory:** Block specific memory, accessible for reading and writing data by every thread in the same block. Comparable in speed to register memory. It is generally limited in size.
- **Constant memory:** Fast, read-only memory accessible by all threads in a grid. Typically on the same speed as register and shared memory.
- **Global memory:** Main memory store of a GPU, accessible for reading and writing by all threads across different kernels. Large amount of storage but slow to access by threads.
- **Local memory:** Thread specific part of global memory. Same operating speed as global memory. This part of memory is used by threads when register memory is full.

The implementation of these different forms of memory has significant influence on the speed of an algorithm. Caution should be taken when using shared or global memory, as these are accessible by all threads for reading and writing. Because threads are executed without a specific order, incorrect implementation can lead to so-called race conditions. This problem arises when a thread overwrites a value that still had to be used by another thread or when the results of computations are combined while one of the parts was not yet finished.

Another important concept in memory use is memory coalescence. This refers to the fact that consecutive threads in a block access consecutive memory locations. When this is implemented correctly it allows several memory access requests to be grouped into one big request, significantly increasing the memory throughput of a GPU algorithm [72].

Hardware Hierarchy

Where threads, blocks and grids are the programmers perspective, there exists a similar hierarchy from the hardware perspective, which is visualised in figure 2.3. An NVIDIA GPU consists of multiple Streaming Multiprocessors (SM's). These key components are responsible for the parallel execution and data management. Several blocks are assigned to one SM. Each SM contains warp schedulers. These components divide the threads in a block into groups of 32, called warps, and schedules them for execution. The warp scheduler takes warps and divides the threads over the available CUDA cores contained in the SM, until all cores are occupied. All threads in a warp are executed at the same time. For this reason blocks should always contain a multiple of 32 threads, such that maximum core occupancy is achieved [92]. The analogy to the software hierarchy is then:

- **Cores:** execute single threads.
- **Streaming Multiprocessors:** execute multiple blocks. A SM contains up to several thousand cores.

- **GPU:** Complete hardware system. A GPU contains several SM's.

From these hierarchies four guidelines for writing fast algorithms can be distilled:

1. Minimize communication between the host and the device. Generally this communication should only occur when data needs to be stored for post processing.
2. Wherever possible register or shared memory should be used in favor of local and global memory. Due to the limited size and potential race conditions, this can however prove to be difficult.
3. Data should be structured in such a way that memory accesses are coalesced.
4. The number of threads per block should always be a multiple of 32. This ensures the blocks can be divided into an integer number of warps which in turn ensures maximum core occupancy.

How these guidelines are implemented in the present code will be further discussed in the next chapter.

This concludes the theoretical background for this thesis. Having started with the fundamental governing equations describing fluid flows and thermal flows in section 2.1, section 2.2 moved on to cover the most important non-dimensional numbers. In section 2.3 the turbulent flow regime was discussed, touching on characteristics and statistical tools. From this the focus was shifted to an overview of Kinetic Theory from which the LBM originates in section 2.4. Finally the chapter concludes with section 2.5, where the fundamentals of parallel programming on NVIDIA GPU's were discussed. The topic of the next chapter is the specifics of the numerical method used in the present work for simulating thermal flows and phase change processes in turbulent flows.

3

Numerical Methods

The Lattice Boltzmann Method was chosen as the numerical method for simulating turbulent thermal flows incorporating freezing effects. This chapter covers the basic theory of the LBM for momentum flow and thermal flow, and the implementation of the specific LBM schemes used. In section 3.1 the transition is made from kinetic theory to LBM and the key aspects of the LBM algorithm will be introduced. Sections 3.2 and 3.3 then cover the specific collision operators and boundary conditions implemented in this research. Next the adjustments needed for the GPU implementation are discussed in section 3.4. Finally section 3.5 gives an overview of the entire algorithm developed for this study.

3.1. Lattice Boltzmann Basics

This section covers the basics of the Lattice Boltzmann Method. The step from kinetic theory to LBM is explained, stepping from the physical into the numerical domain and arriving at the Lattice Boltzmann equation. First the discretizations and velocity sets, together with basics of LBM applied to momentum flow are discussed. After this the two fundamental steps of the LBM algorithm are discussed. Finally the application of LBM to thermal flow problems is discussed.

3.1.1. Discretizations & Velocity Sets

In section 2.4 the particle distribution function f , and the Boltzmann equation (2.28) were introduced. This allows one to track the evolution of f through time under influence of a flow. To solve equation (2.28) numerically, f must be discretised in Cartesian space, time and velocity space. This creates discrete spatial nodes separated by Δx , discrete time steps Δt , and a set of discrete velocities $\{c_i\}$. The standard choice for lattice units in LBM is $\Delta x = 1 \text{ } ls$, $\Delta t = 1 \text{ } lt$ and $[c_i] = ls/lt$, where ls and lt are 'lattice spacing' and 'lattice time step' respectively. The resulting scheme is denoted by $DdQq$, where d is the number of spatial dimensions and q is the number of discrete velocities [94]. The velocity sets D3Q7 and D3Q19, which are used in this research, are shown in figure 3.1.

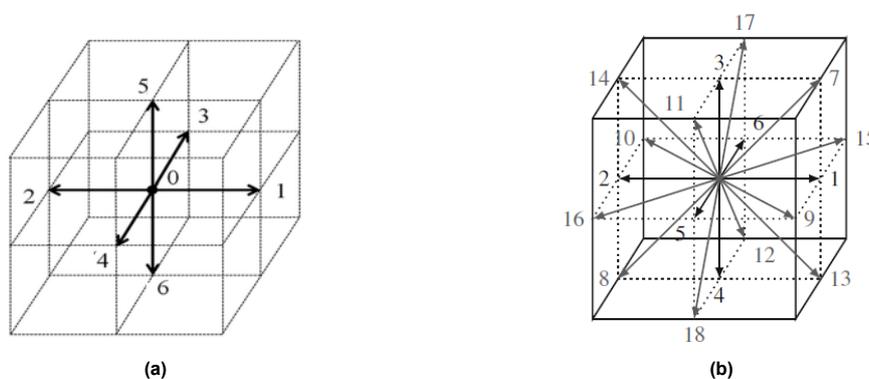


Figure 3.1: Velocity vectors for the D3Q7 (a) and the D3Q19 (b) velocity schemes. Obtained from [93] and [61] respectively.

Each velocity vector has a weight ω_i associated with it. The weights of the D3Q7 and D3Q19 are given by

$$\omega_i = \begin{cases} 1/4 & \text{if } i = 0, \\ 1/8 & \text{if } i \neq 0, \end{cases} \quad (3.1)$$

$$\omega_i = \begin{cases} 1/3 & \text{if } i = 0, \\ 1/18 & \text{if } 0 < i < 7, \\ 1/36 & \text{if } i \geq 7. \end{cases} \quad (3.2)$$

Another important quantity that can be derived from the velocity vectors and weights is the lattice speed of sound c_s , which is calculated with

$$\sum_i \omega_i c_{ia} c_{ib} = c_s^2 \delta_{ab}, \quad (3.3)$$

where a and b are the spatial dimensions and δ_{ab} is the Kronecker delta. This gives a value of $c_s = 1/2$ l_s/l_t for the D3Q7 set and $c_s = 1/\sqrt{3}$ l_s/l_t for the D3Q19 set. The speed of sound links the pressure to the density as

$$P = c_s^2 \rho. \quad (3.4)$$

The discretised distribution function with a $DdQq$ velocity set now describes a total of q *populations* at each lattice point. The particle distribution function corresponds to the macroscopic density and momentum as

$$\sum_i f_i(\mathbf{x}, t) = \rho(\mathbf{x}, t), \quad (3.5)$$

$$\sum_i \mathbf{c}_i f_i(\mathbf{x}, t) = \rho \mathbf{u}(\mathbf{x}, t). \quad (3.6)$$

The equilibrium distribution function for momentum flow is also discretised

$$f_i^{eq} = \omega_i \rho \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right). \quad (3.7)$$

3.1.2. Lattice Boltzmann Equation - Collision & Streaming

Performing these discretisations on equation (2.28) leads to the Lattice Boltzmann equation for momentum flow:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) + \Delta t \Omega_i(\mathbf{x}, t). \quad (3.8)$$

Equation (3.8) describes two processes that happen during each time step. The collision operator Ω_i models the effect of collisions between particles and locally redistributes the particles among the populations. This is called the collision process. These collisions must obey the conservation laws, such that

$$\sum_i \Omega_i = 0, \quad \sum_i \mathbf{c}_i \Omega_i = \mathbf{f}. \quad (3.9)$$

This process contains the computationally expensive calculations. After this, populations propagate from one lattice node \mathbf{x} to a neighbouring node $\mathbf{x} + \mathbf{c}_i \Delta t$. This is called the streaming, or propagation process [61]. This step is computationally inexpensive, as it only consists of redistributing populations over the lattice. These two processes make up the core of the LBM algorithm. A schematic depiction of the LBM time step is given in figure 3.2. The split between locality (collision) and non-locality (streaming) and the fact that the computationally expensive processes all happen locally makes the LBM very suitable for implementation on parallel computation systems [95].

Through a method known as the Chapman-Enskog analysis [97] it can be shown that equation (3.8) corresponds to the Navier-Stokes equations up to second order accuracy. The key is to find the non-equilibrium contribution to the distribution function

$$f_i = f_i^{eq} + f_i^{neq}. \quad (3.10)$$

Central to this method is the perturbation expansion of f_i around f_i^{eq}

$$f_i = f_i^{eq} + \epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} + h.o.t., \quad (3.11)$$

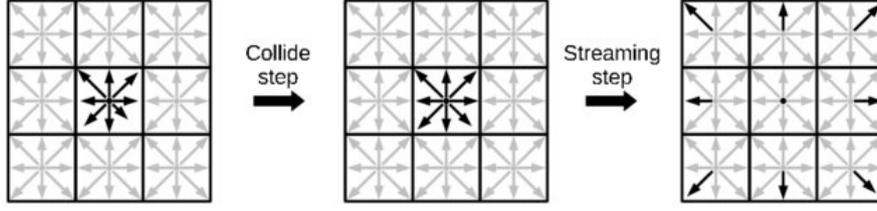


Figure 3.2: Schematic depiction of the two core steps of the LBM algorithm [96]. Particles are locally redistributed among populations in the collision step and then propagated to neighboring lattice nodes in the streaming step.

where ϵ is a kind of order of smallness parameter, and the first order Taylor expansion of equation (3.8)

$$\frac{\partial f_i}{\partial t} + \mathbf{c}_i \cdot \nabla f_i = \Omega_i. \quad (3.12)$$

Expanding the time and spatial derivatives up to second order and substituting (3.11), an expression for f_i is found

$$f_i = f_i^{eq} - \rho\nu\omega_i \left(\frac{(\mathbf{c}_i \cdot \nabla)(\mathbf{c}_i \cdot \mathbf{u})}{c_s^4} - \left(1 + \frac{2}{D} - B\right) \frac{\nabla \cdot \mathbf{u}}{c_s^2} \right), \quad (3.13)$$

where D is the number of spatial dimensions and $B = \zeta/\nu$ represents the ratio of the lattice bulk viscosity to the lattice kinematic viscosity. Using this expression for f_i and the conservation constraints on the collision operator (3.9) the Navier-Stokes equations can be recovered up to second order accuracy [98]. Having neglected the higher order terms places a limit on the maximum velocity in LBM, as these h.o.t. cease to be negligible for high velocities. In general the maximum velocity in simulations must not exceed $0.12 \text{ } l_s/lt$ [61].

3.1.3. Thermal Lattice Boltzmann Methods

There are essentially two ways of solving heat flow with LBM. The first method is the Multi-Speed (MS) model, where an extra velocity component is added so that the hydrodynamic and thermodynamic governing equations are recovered from a single distribution function [99]. The advantage of this approach is that the coupling between the governing equations, i.e. viscous dissipation and compression work, are inherently included. A disadvantage is instability [100]. Also, the standard cubic lattices, such as the D3Q19 lattice, are no longer sufficiently accurate [61]. For this reason these models are not often used.

A more successful technique in terms of accuracy and stability is the Double Distribution Function (DDF) method [101]. As the name suggests the idea is to introduce a second distribution function $g_i(\mathbf{x}, t)$ to describe the flow of heat through the domain, which obeys the Lattice Boltzmann equation for energy flow

$$g_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = g_i(\mathbf{x}, t) + \Delta t \Omega_i(\mathbf{x}, t). \quad (3.14)$$

The energy distribution function is collided and streamed separately from f_i and any coupling processes must be explicitly defined in the collision step. In the present work this method is used and the enthalpy approach for simulating thermal flow was chosen, such that g_i corresponds to the total enthalpy via

$$\sum_i g_i = H. \quad (3.15)$$

The advection-diffusion equation (2.8) can again be recovered using the Chapman-Enskog analysis. A simple linear equilibrium distribution function can be assumed [61]

$$g_i^{eq} = \omega_i H \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} \right). \quad (3.16)$$

Finding the expression for the non-equilibrium distribution lead to the following expression for the total distribution function [102]

$$g_i = g_i^{eq} - \omega_i \alpha \frac{\mathbf{c}_i \cdot \nabla H}{c_s^2}. \quad (3.17)$$

As discussed in section 2.1.2 the enthalpy formulation inherently tracks both the temperature and the phase transition. The temperature and liquid fraction are calculated from the total enthalpy with

$$T = \begin{cases} H/C_{p,s} & \text{if } H < H_s, \\ T_s + \frac{T_l - T_s}{H_l - H_s}(H - H_s) & \text{if } H_s \leq H \leq H_l, \\ T_l + \frac{H - H_l}{C_{p,l}} & \text{if } H_l < H, \end{cases} \quad (3.18)$$

and

$$f_l = \begin{cases} 0 & \text{if } H < H_s, \\ \frac{H - H_s}{H_l - H_s} & \text{if } H_s \leq H \leq H_l, \\ 1 & \text{if } H_l < H, \end{cases} \quad (3.19)$$

where T_s, T_l are the solid and liquid temperatures respectively. For pure materials these have the same value. $H_s = C_{p,s}T$ and $H_l = H_s + L$ are the solid and liquid enthalpy. Thus 3 phases exist: solid, mushy and liquid. Material properties differ for these phases and need to be specified. In the mushy zone the thermal conductivity and specific heat are calculated with

$$\lambda_m = (1 - f_l)\lambda_s + f_l\lambda_l, \quad (3.20)$$

and

$$C_{p,m} = (1 - f_l)C_{p,s} + f_lC_{p,l}. \quad (3.21)$$

Because this study deals with eutectic materials, the mushy zone is a very thin numerical region.

The DDF method is employed in the current study with the D3Q19 scheme for the particle distribution function f_i and the D3Q7 scheme for the enthalpy distribution function g_i . Now that the basics of LBM have been covered, the next section dives deeper into the specifics of the collision operators that were implemented for the purpose of simulating turbulent thermal flows.

3.2. Collision Operators

This section goes deeper into the collision operators used in this research. As discussed in the previous section, collision operators model the effects of local inter particle collisions. The simplest form is the so-called BGK operator, named after Bhatnagar, Gross and Krook who introduced it in 1954 [103]. For momentum flow it is formulated as

$$\Omega_i = -\frac{1}{\mathcal{T}}(f_i - f_i^{eq}), \quad (3.22)$$

where \mathcal{T} is called the relaxation-time. Physically it describes the tendency of the populations to evolve towards the equilibrium distribution f_i^{eq} . This operator is also called the single-relaxation-time (SRT) operator as it relaxes all populations at the same rate [61]. The relaxation time is related to the fluid viscosity when used for the density distribution or thermal diffusivity when used for the energy distribution. Through the Chapman-Enskog analysis it can be shown for the BGK operator that

$$\mathcal{T}_f = \frac{\nu}{c_s^2} + \frac{1}{2}, \quad (3.23)$$

and

$$\mathcal{T}_g = \frac{\alpha}{c_s^2} + \frac{1}{2}. \quad (3.24)$$

Several authors have shown that the stability of the BGK operator deteriorates when \mathcal{T} approaches 0.5 [104] [105] [106]. This happens for low values of the viscosity or thermal diffusivity. Furthermore the accuracy of the BGK operator is dependent on \mathcal{T} [61]. For simulating turbulent flows, it is necessary to simulate low ν and α fluids to achieve high Re , while keeping the maximum lattice velocity below 0.12 ls/lt and limiting the channel height in ls . The BGK operator is therefore ill-suited to the task. Other collision operator have been proposed to overcome these unwanted effects, such as the Two-Relaxation-time (TRT) operator, the Multiple-Relaxation-Time (MRT) operator and the Filter Matrix (FM) operator.

Implementation of the MRT operator involves the selection free parameters that significantly influence the simulation accuracy and stability, for which there is no straightforward method. This is a major

downside of the MRT operator [61]. Implementation of the TRT operator is similar in simplicity to the BGK operator, while improving stability. The FM operator chooses a different approach, also eliminating the selection of many free parameters and improving stability in low ν simulations. Therefore the latter two of these improved operators are chosen in favor of the MRT operator in the present work. Their performance to simulate heat transfer in turbulent channel flow is investigated.

3.2.1. Filter-Matrix Operator for Momentum Flow

A collision operator that shows superior stability compared to the SRT operator is the filter-matrix operator. It was first introduced by Somers in [76] in 2 dimensions for the D2Q9 scheme. Eggels and somers [77] later showed that it can also be used for scalar transport. More recently it was extended to 3 dimensions for the D3Q19 scheme by Zuo and Zhong [78].

The FM method is defined on a staggered grid,

$$f_i \left(\mathbf{x} + \frac{\mathbf{c}_i \Delta t}{2}, t + \frac{\Delta t}{2} \right) - f_i \left(\mathbf{x} - \frac{\mathbf{c}_i \Delta t}{2}, t - \frac{\Delta t}{2} \right) = \Omega_i(f_i), \quad (3.25)$$

which gives a pre-collision and a post-collision distribution function. By performing a Taylor expansion around $f_i(\mathbf{x}, t)$ and substituting the expression for f_i from the Chapman-Enskog analysis (3.13) an expression for the collision operator in terms of physical parameters is found

$$\Omega_i = \rho \omega_i \left(\frac{(\mathbf{c}_i \cdot \nabla)(\mathbf{c}_i \cdot \mathbf{u})}{c_s^2} - \nabla \cdot \mathbf{u} + \frac{\mathbf{c}_i \cdot \mathbf{g}}{c_s^2} \right). \quad (3.26)$$

From the expansions of the regular LB equation (3.12) and the staggered LB equation the following expression for the distribution function on the staggered grid can be found

$$f_i \left(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2} \right) = f_i(\mathbf{x}, t) \pm \frac{\Delta t}{2} \Omega_i. \quad (3.27)$$

Filling in the expressions for f_i (3.13) and Ω_i (3.26) gives an expression for the pre- and post collision distribution functions in terms of physical parameters and lattice velocity vectors

$$\begin{aligned} f_i \left(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2} \right) &= \omega_i \rho \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right) - \dots \\ &\quad \rho \nu \omega_i \left(\frac{(\mathbf{c}_i \cdot \nabla)(\mathbf{c}_i \cdot \mathbf{u})}{c_s^4} - \left(1 + \frac{2}{D} - B \right) \frac{\nabla \cdot \mathbf{u}}{c_s^2} \right) \pm \dots \\ &\quad \frac{\Delta t}{2} \rho \omega_i \left(\frac{(\mathbf{c}_i \cdot \nabla)(\mathbf{c}_i \cdot \mathbf{u})}{c_s^2} - \nabla \cdot \mathbf{u} + \frac{\mathbf{c}_i \cdot \mathbf{g}}{c_s^2} \right). \end{aligned} \quad (3.28)$$

This expression can be rewritten as a matrix multiplication

$$f_i \left(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2} \right) = \sum_k \omega_i E_{ik}^{19} \alpha_k^\pm(\mathbf{x}, t), \quad (3.29)$$

where E_{ik}^{19} is the filter matrix and α_k^\pm denotes the pre- and post-collision solution vector in moment space. The pre- and post-collision distribution functions can be transformed to moment space via the reverse operation

$$\alpha_k^\pm(\mathbf{x}, t) = \sum_i E_{ki}^{19} f_i \left(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2} \right), \quad (3.30)$$

where $\omega_i E_{ik}^{19} = (E_{ki}^{19})^{-1}$. The filter matrix and solution vectors for the density distribution function in the D3Q19 scheme are given by [78]

$$E_{ki}^{19} = \begin{bmatrix} 1, c_{ix}, c_{iy}, c_{iz}, 3c_{ix}^2 - 1, 3c_{iy}^2 - 1, 3c_{iz}^2 - 1, 3c_{iy}c_{iz}, 3c_{ix}c_{iz}, \\ 3c_{ix}c_{iy}, 3c_{ix}(c_{iy}^2 - c_{iz}^2), 3c_{iy}(c_{iz}^2 - c_{ix}^2), 3c_{iz}(c_{ix}^2 - c_{iy}^2), \\ c_{ix}(3c_{iy}^2 + 3c_{iz}^2 - 2), c_{iy}(3c_{ix}^2 + 3c_{iz}^2 - 2), \\ c_{iz}(3c_{ix}^2 + 3c_{iy}^2 - 2), 3(2c_{ix}^2 - c_{iy}^2 - c_{iz}^2)(|c_i|^2 - 1.5) \\ 3(c_{iy}^2 - c_{iz}^2)(|c_i|^2 - 1.5), 3|c_i|^2(|c_i|^2 - 2) + 1 \end{bmatrix}^T, \quad (3.31)$$

$$\alpha_k^\pm = \begin{bmatrix} \rho \\ \rho(u \pm \Delta t g_x / 2) \\ \rho(v \pm \Delta t g_y / 2) \\ \rho(w \pm \Delta t g_z / 2) \\ 3\rho u^2 + \rho(-6\nu \pm \Delta t)\partial_x u + (2 - 3B)\rho\nu\nabla \cdot \mathbf{u} \\ 3\rho v^2 + \rho(-6\nu \pm \Delta t)\partial_y v + (2 - 3B)\rho\nu\nabla \cdot \mathbf{u} \\ 3\rho w^2 + \rho(-6\nu \pm \Delta t)\partial_z w + (2 - 3B)\rho\nu\nabla \cdot \mathbf{u} \\ 3\rho vw + \rho(-3\nu \pm \Delta t/2)(\partial_y w + \partial_z v) \\ 3\rho uw + \rho(-3\nu \pm \Delta t/2)(\partial_x w + \partial_z u) \\ 3\rho uv + \rho(-3\nu \pm \Delta t/2)(\partial_x v + \partial_y u) \\ T_k^\pm, k = 10, \dots, 15 \\ F_k^\pm, k = 16, 17, 18 \end{bmatrix}. \quad (3.32)$$

The high-order terms are determined by

$$\alpha_{10-15}^+ = -\gamma_1 \alpha_{10-15}^-, \quad (3.33)$$

$$\alpha_{16-18}^+ = -\gamma_2 \alpha_{16-18}^-. \quad (3.34)$$

B is set to 3 and $\gamma_{1,2}$ are chosen as 0.8 and 0.95 respectively [78].

3.2.2. Collision Operators for Thermal Flow

For the enthalpy distribution function, two collision operators are used and their performance is compared. The first operator is the Optimal Two-Relaxation-Time (OTRT) operator proposed by Lu et al. [75] to deal with solid-liquid phase change in natural convection. This operator has not yet been implemented in forced convective flows. The second operator is the FM-operator for thermal flows. This operator has been implemented by Eggels and Somers [77] for scalar transport in the D2Q9 scheme. To the authors knowledge this is the first time it is applied to enthalpy transport in the D3Q7 scheme.

Optimal Two-Relaxation-Time Operator

Lu et al. [75] proposed an optimal two-relaxation-time (OTRT) operator specifically for the energy distribution function in solid-liquid phase change problems, to increase the accuracy of the SRT operator but maintain its simplicity of implementation. They based it on an SRT operator proposed in [107] and showed its increased performance for conduction melting and natural convection melting.

The main idea is to split the distribution function and equilibrium distribution function into a symmetric and anti-symmetric part. These are relaxed at different rates, introducing a second relaxation time:

$$\Omega_i(\mathbf{x}, t) = -\frac{1}{\mathcal{T}_s} [g_i^s(\mathbf{x}, t) - g_i^{seq}(\mathbf{x}, t)] - \frac{1}{\mathcal{T}_a} [g_i^a(\mathbf{x}, t) - g_i^{aeq}(\mathbf{x}, t)], \quad (3.35)$$

where

$$g_i^{seq} = \frac{g_i^{eq} + g_j^{eq}}{2}, \quad g_i^{aeq} = \frac{g_i^{eq} - g_j^{eq}}{2}. \quad (3.36)$$

j is defined as the opposite velocity direction as i , i.e. $c_j \equiv -c_i$. The antisymmetric relaxation time \mathcal{T}_a is related to the fluid properties

$$\frac{\lambda}{\rho C_{p,r}} = c_s^2 (\mathcal{T}_a - 0.5) \Delta t. \quad (3.37)$$

Here $C_{p,r}$ is a reference specific heat capacity introduced to account for the difference in specific heat capacity for different phases. It's value is set as the harmonic mean of specific heat of the solid phase $C_{p,s}$ and the liquid phase $C_{p,l}$:

$$C_{p,r} = \frac{2C_{p,s}C_{p,l}}{C_{p,s} + C_{p,l}}. \quad (3.38)$$

Tuning the symmetric relaxation time allows for better stability and accuracy, with the optimal stability achieved when [108]

$$\frac{1}{\mathcal{T}_s} + \frac{1}{\mathcal{T}_a} = 2. \quad (3.39)$$

The equilibrium distribution function is altered such that the latent heat is contained only in the static population

$$g_i^{eq} = \begin{cases} H - C_{p,r}T + \omega_i C_{p,r}T & \text{if } i = 0, \\ \omega_i C_p T \left(\frac{C_{p,r}}{C_p} + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} \right) & \text{if } i \neq 0 \end{cases} \quad (3.40)$$

The collision step is now given by

$$g_i^* = g_i - \frac{1}{\mathcal{T}_s} (g_i^s - g_i^{seq}) - \frac{1}{\mathcal{T}_a} (g_i^a - g_i^{aeq}), \quad (3.41)$$

where * denotes the post-collision distribution function.

Filter-Matrix operator for Solid-Liquid Phase Change

To perform the collision step on the enthalpy distribution function, a method is employed which was first used in the master thesis of Besseling [79] for phase change in natural convective flows and later in the master thesis of Bus [80] for simulating non-eutectic phase change in laminar forced convective flows.

The distribution function is split into the sensible part g_i^s and a latent part g_i^l . The latent heat is considered to only be a part of the static population:

$$g_i = g_i^s + g_i^l, \quad (3.42)$$

$$g_i^l = [f_L L, 0, 0, 0, 0, 0]^\top. \quad (3.43)$$

Because latent heat does not diffuse, this can be done without introducing significant errors. The collision step is then performed only on the sensible enthalpy. The macroscopic variables are calculated before the collision step from the total enthalpy distribution function g_i using equations (3.18) and (3.19).

After the macro variables are extracted and the latent heat is subtracted from the static population, g_i^s is collided in the same manner as the filter matrix method for momentum flow. The collision operator for the enthalpy distribution takes the form [102]

$$\Omega_i = \omega_i \mathbf{c}_i \cdot \nabla h. \quad (3.44)$$

This then leads to the following expression for the pre- and post-collision distribution functions in terms of the physical parameters

$$g_i^s \left(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2} \right) = \omega_i \left[h \left(1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} \right) - \left(\frac{\alpha}{c_s^2} \pm \frac{\Delta t}{2} \right) (\mathbf{c}_i \cdot \nabla h) \right], \quad (3.45)$$

which is again rewritten as a matrix multiplication

$$g_i^s \left(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2} \right) = \sum_k \omega_i E_{ik}^\top \beta_k^\pm(\mathbf{x}, t), \quad (3.46)$$

with the reverse operation

$$\beta_k^\pm(\mathbf{x}, t) = \sum_i E_{ki}^\top g_i \left(\mathbf{x} \pm \frac{\mathbf{c}_i \Delta t}{2}, t \pm \frac{\Delta t}{2} \right). \quad (3.47)$$

The matrices and solution vectors for the D3Q7 scheme are

$$E_{ki}^\top = [1, c_{ix}, c_{iy}, c_{iz}, 4c_{ix}^2 - 1, 4c_{iy}^2 - 1, 4c_{iz}^2 - 1]^\top, \quad (3.48)$$

$$\beta_k^\pm = \begin{bmatrix} h \\ hu + \frac{-8\alpha \pm \Delta t}{8} \partial_x h \\ hv + \frac{-8\alpha \pm \Delta t}{8} \partial_y h \\ hw + \frac{-8\alpha \pm \Delta t}{8} \partial_z h \\ T_4^\pm, T_5^\pm, T_6^\pm \end{bmatrix}. \quad (3.49)$$

The higher order terms in (3.49) are set to zero. After collision the latent heat is re-added to the static population. Now that all collision operators implemented for the present study have been discussed. The next section will go into detail about the treatment of the boundaries of the computational domain.

3.3. Boundary Conditions

Consistent boundary conditions are of key importance for any numerical simulation. In LBM boundary conditions come down to finding consistent formulations for the unknown populations that stream into the domain. It was observed that inlet and outlet effects had a significant influence on the fluid and energy flow results and therefore several different techniques were implemented. In this section different methods that were investigated in this research are described. First the conditions at the wall are described in section 3.3.1, after which the inflow conditions are treated in 3.3.2 and 3.3.3. Then the outflow conditions are treated in 3.3.4 and finally the treatment of the phase interface is discussed in 3.3.5.

3.3.1. Solid Walls

The hydrodynamic boundary condition at any solid-liquid interface is the no-slip condition, which states that the fluid velocity at the interface must be zero. The so-called bounce back (BB) method is the most widely adopted method for this in LBM. The half-way bounce back (HBB) method is a second-order accurate variation of the BB method and is used in this research.

In the HBB method populations that point at the wall before the streaming step are reversed in direction during the streaming step:

$$f_i(\mathbf{x}_b, t + \Delta t) = f_j^*(\mathbf{x}_b, t), \quad (3.50)$$

where $c_j = -c_i$ and * denotes the post-collision population. This effectively puts the solid boundary half way in between the last domain node \mathbf{x}_b and the first virtual solid node, making the HBB method a so-called link-wise method [61].

For the energy distribution a Dirichlet condition was employed, specifying a value for the enthalpy at the wall and leading to a constant wall temperature. This is achieved by the anti-bounce-back (ABB) method, another link-wise method. The ABB method for a static wall is given by the following formula [61]

$$g_i(\mathbf{x}_b, t + \Delta t) = -g_j^*(\mathbf{x}_b, t) + 2\omega_j h_{wall}. \quad (3.51)$$

The sensible enthalpy h_{wall} is used here because in the current FMLB method the latent heat is contained in the 0-population only, which is not streamed to other nodes.

3.3.2. Hydrodynamic Inlet

Perhaps the simplest inlet-outlet combination are periodic conditions. The populations leaving over the outlet boundary re-enter the domain at the inlet. This effectively makes an infinitely long channel. The advantage of this technique is the lack of unknown populations. Because every population was calculated using the physical models inside the domain, inlet and outlet effects are not present. The downside is the need to match the inlet and outlet to avoid discontinuities in the domain. This means the channel cannot have an entrance region.

A Dirichlet condition on the velocity at the inlet can be implemented using an altered form of the BB method. It corresponds to implementing a moving wall at the inlet and is implemented as

$$f_i(\mathbf{x}_b, t + \Delta t) = f_j^*(\mathbf{x}_b, t) - 2\omega_j \rho_{wall} \frac{c_j \cdot \mathbf{u}_{wall}}{c_s^2}. \quad (3.52)$$

The wall density can be approximated by the boundary density or obtained from extrapolation. No notable difference was observed in this research.

3.3.3. Thermal Inlet

Two methods for establishing a constant inlet temperature were implemented. The first one is the ABB method also used for the constant wall temperature in section 3.3.1. For a boundary with non-zero flow velocity, this method is implemented as follows [61]

$$g_i(\mathbf{x}_b, t + \Delta t) = -g_j^*(\mathbf{x}_b, t) + 2g_j^{eq}(\mathbf{x}_{wall}, t + \Delta t). \quad (3.53)$$

The second method that was adopted was proposed by Liu et al. in [109]. The unknown populations at the boundary are made functions of known populations and correction terms

$$g_i(\mathbf{x}_b, t) = g_i^\bullet(\mathbf{x}_b, t) + \omega_i G, \quad (3.54)$$

where the superscript \bullet indicates a known distribution and G is the corrector to enforce the specified value for the inlet enthalpy. Liu et al. tested several options for g_i^\bullet and observed no differences in accuracy. In the current research the post-collision distribution was used. The value for the correctors G can then be calculated from the current enthalpy H^\bullet , the wanted enthalpy H_{in} and equation (3.54). For the D3Q7 scheme and an inlet at $x = 0$ this is done using

$$H^\bullet = g_0 + g_1^\bullet + g_2 + g_3 + g_4 + g_5 + g_6, \quad (3.55)$$

$$G = (H_{in} - H^\bullet)/\omega_1. \quad (3.56)$$

3.3.4. Outlet

The outflow condition is applied after the streaming step and should ensure a smooth outflow that does not have influence on what happens inside the domain. There are several ways to achieve this. The simplest is the Neumann outflow condition [110], which sets the derivative of the macroscopic variables to zero at the outflow boundary. In practice this is achieved by copying the populations from the boundary node

$$\begin{aligned} f_i(N_x, y, z, t) &= f_i(N_x - 1, y, z, t), \\ g_i(N_x, y, z, t) &= g_i(N_x - 1, y, z, t). \end{aligned} \quad (3.57)$$

Another technique is the extrapolation method [110]. It is based on linear extrapolation of the last two nodes inside the domain

$$\begin{aligned} f_i(N_x, y, z, t) &= 2f_i(N_x - 1, y, z, t) - f_i(N_x - 2, y, z, t), \\ g_i(N_x, y, z, t) &= 2g_i(N_x - 1, y, z, t) - g_i(N_x - 2, y, z, t). \end{aligned} \quad (3.58)$$

The ABB method can be used to set a constant pressure at the outlet, similarly to its use for constant wall temperature with equation (3.51). For constant pressure the implementation takes the form [61]

$$f_i(\mathbf{x}_b, t + \Delta t) = -f_j^*(\mathbf{x}_b, t) + 2\omega_j \rho_{wall} \left[1 + \frac{(\mathbf{c}_j \cdot \mathbf{u}_{wall})^2}{2c_s^4} - \frac{\mathbf{u}_{wall}^2}{2c_s^2} \right], \quad (3.59)$$

choosing ρ_{wall} to give the desired pressure. The velocity at the wall \mathbf{u}_{wall} must be approximated. It can be set to the velocity at the boundary node or by extrapolation.

3.3.5. Phase Interface

Inside the solid phase and on the solid-liquid interface the velocity should be zero. To achieve this the immersed boundary method was employed. In this method the collision operator is altered [111] to be a function of the liquid fraction

$$f_i \left(\mathbf{x} + \frac{\mathbf{c}_i \Delta t}{2}, t + \frac{\Delta t}{2} \right) - f_i \left(\mathbf{x} - \frac{\mathbf{c}_i \Delta t}{2}, t - \frac{\Delta t}{2} \right) = \Delta t ((1 - B_f) \Omega_i + B_f \Omega_i^S), \quad (3.60)$$

with

$$B_f = \frac{(1 - f_l) \epsilon}{f_l + \epsilon}, \quad (3.61)$$

where ϵ is a small constant to avoid division by zero. Ω_i is the standard FM collision operator and the extra operator is given by

$$\Omega_i^S = f_j \left(\mathbf{x} - \frac{\mathbf{c}_i \Delta t}{2}, t - \frac{\Delta t}{2} \right) - f_i \left(\mathbf{x} - \frac{\mathbf{c}_i \Delta t}{2}, t - \frac{\Delta t}{2} \right) + f_i^{eq}(\rho, \mathbf{u}, s) - f_j^{eq}(\rho, \mathbf{u}), \quad (3.62)$$

with the solid velocity $\mathbf{u}_s = 0 \text{ ls/lt}$.

Now that the boundary conditions have been discussed, the next section covers the changes to the standard LBM scheme needed for its implementation on a GPU.

3.4. GPU Accelerated Lattice Boltzmann Method

As discussed in the previous chapter, performing coupled fluid and thermal flow simulations in the turbulent flow regime is highly computationally expensive. For this reason parallel programming on GPU's is a widely employed method to increase simulation speeds for LBM simulations [72] [112] [113]. This section will elaborate on the adjustments to the data structures and the algorithm applied for the present study.

Numba

The algorithm for this research is written in the programming language Python. This language was chosen for its relative ease of implementation and the large amount of open source packages and documentation produced by an active user base. As Python is a high-level programming language, it is generally slow compared to low-level languages such as FORTRAN or C++. To increase simulation speed, the just-in-time compiler Numba is used. During the first run of the algorithm Numba compiles a subset of the code to machine code. Effectively this means the first run takes a long time, but every consecutive run becomes much faster [114]. For the GPU implementation specifically, the Numba-CUDA package is used. This package allows Python algorithms to run on comparable speeds as C-CUDA code [115].

Host-Device Communication

As stated in section 2.5 communication between the host and the device should be minimized. To achieve this, all arrays for reading and writing are stored in the device memory before the start of the time step algorithm. There is no further sending of data from host to device. Because kernels can only create arrays that are terminated after kernel execution, and only alter values in arrays that already exist in device memory, two data structures need to be initialized for each distribution function. The first structure contains the post stream distribution of the previous time step. During a time step the post collision distribution is saved in the second structure after which the post stream distribution is saved in the first structure. This is depicted in figure 3.3. Data is only sent from the device to the host when it needs to be saved in host memory and only macroscopic variable fields are saved.

Memory Usage

Efficient memory usage is an essential part of fast GPU algorithms. Several steps are undertaken to increase memory handling efficiency:

- The data structures for the distribution functions and scalar fields are saved as one-dimensional arrays, as these are handled with less computational overhead by GPU architectures than multi-dimensional arrays [61]. The combination (x, y, z, i) for populations is transformed to a single index using

$$idx = i + N_c x + N_c N_x y + N_c N_x N_y z, \quad (3.63)$$

with N_c the number of populations, and the scalar field indices are calculated as

$$idx_s = x + N_x y + N_x N_y z. \quad (3.64)$$

- Register memory is used inside the collision operators as much as possible. For arrays needed in the collision process local memory was used.
- A 'pull-in' stream method is used, where a thread fills up the post-stream populations of a single spatial node. This involves non-local reads and local writes. It has been shown that this is faster than a 'push-out' method, where a thread redistributes the post-collision populations of a single spatial node. This involves local reads and non-local writes [72] [116].

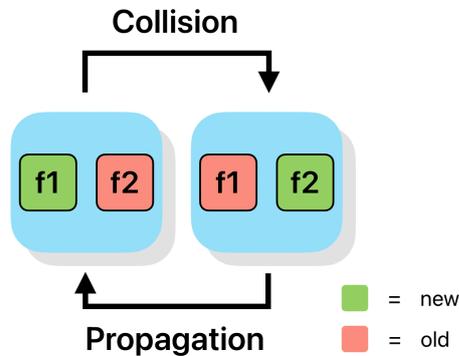


Figure 3.3: Schematic depiction of a time step using two data structures for one distribution function. The 'old' structure contains data which can be overwritten. The 'new' structure contains the most recently generated data.

Maximizing Core Occupancy

The block and grid structures should be chosen in such a way that core occupancy is maximized. A single grid is defined for the spatial domain, giving it dimensions (N_x, N_y, N_z) . The `bpg` are then set to $(1, N_y, N_z)$ and the `tpb` to $(N_x, 1, 1)$. Each domain is chosen such that N_x is a multiple of 32. This ensures each block can always be divided in an integer number of warps.

Race Conditions

As previously discussed, race conditions can occur when two threads have access to the same memory space. To avoid this, kernel synchronisation is used. The Numba CUDA function `synchronize()` is implemented after each execution of the collision and propagation kernels. This ensures all threads are executed before another kernel execution is queued.

With these adjustments to the data structures and algorithms, the efficiency of the GPU LBM implementation was increased. It should be noted that this implementation can be further improved. The data structure could be further optimized to increase memory coalescence. Compared to the single 1D array used in the current structure, the so-called 'Structure of arrays' layout generally leads to faster simulation speeds [72] [117]. This entails splitting the distribution functions into i separate arrays, where each array contains a single population at each grid point. This structure was omitted as this increases computational complexity and the current data structure lead to satisfactory speeds for the current purpose.

The current scheme does not make use of shared memory. In [118] and [119] an implementation of the propagation step using shared memory is described. The same references combine the collision and propagation step into a single kernel, also increasing simulation speed. These methods significantly increase the complexity of the algorithm and are prone to producing race conditions. Thus they have been omitted in the current study. The next section gives an overview of the entire algorithm.

3.5. The Algorithm

This section gives a step wise overview of the GPU-DDF-LB algorithm. Section 3.5.1 covers the initialisation phase and the main loop, omitting details of the collision and propagation kernels. These are covered in sections 3.5.2 and 3.5.3 respectively. Numba CUDA functions are denoted with `cd.function()`.

3.5.1. Initialisation and Main Loop

A flow chart of the algorithm is shown in figure 3.4, where the yellow boxes are executed on the host, the blue boxes are executed on the device and the green boxes indicate host-device communication steps. Here a textual description is given, where for each step is specified whether it's executed on the host or the device.

Initialisation

- (host) Specify the input parameters. Which parameters are chosen and how the other parameters are calculated differs for the laminar and turbulent cases. These will be discussed in their respective chapters.
- (host) Initialize the $E_{ki}^7, \omega_i E_{ik}^7, E_{ki}^{19}, \omega_i E_{ik}^{19}$ in host memory.
- (host) Create zarr files for data storage.
- (host) Choose initial ρ, \mathbf{u} and H fields and use (3.29) and (3.46) or (3.16) (depending on collision operator) to calculate the initial distribution functions. Save these in host memory as 1D Numpy arrays.
- (host - device) Send all arrays to global device memory using the Numba CUDA function `cd.to_device()`

Main Loop

1. (host) Check if $t > Nt$
 - yes: terminate simulation
 - no: continue
2. (host) Check if data should be saved in the current step (a) or not (b)

3. (device) Collide.
 - (a) save scalar fields to global device memory
 - (b) continue
4. (device) Synchronise threads.
5. (device) Propagate and apply boundary conditions.
6. (device) Synchronise threads.
7. (a) continue with 8. - 10.
(b) return to 1.
8. (device - host) Send scalar fields to host memory using `cd.copy_to_host()`.
9. (host) Save scalar fields to zarr files
10. (host) Check stability
 - unstable: terminate simulation
 - stable: return to 1.

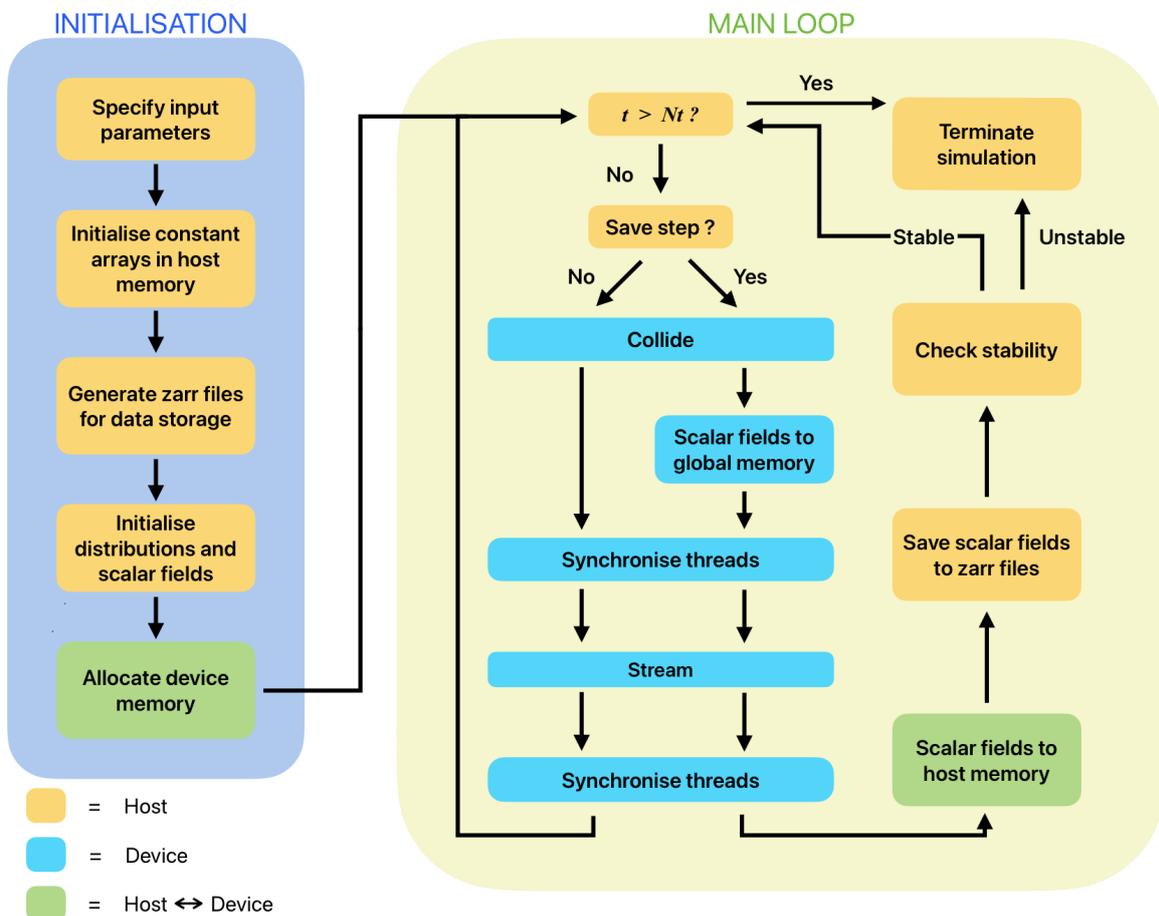


Figure 3.4: Flow chart of the full GPU-DDF-LBM algorithm. The yellow boxes are executed on the host and the blue boxes are executed on the device. The green boxes indicate a host-device communication step.

3.5.2. Collision

This section gives a step-by-step overview of the collision kernels for the GPU-DDF-LBM algorithm. Two kernels were written. One combining the FM algorithm for momentum flow with the OTRT algorithm for enthalpy flow and one combining the two FM algorithms. The IBM for correcting the velocity in solidified sections was included in these kernels.

FM-OTRT kernel

1. Calculate H with (3.15).
2. Calculate T, f_i from H with (3.18) and (3.19).
3. Construct α^- using (3.30)
4. Determine ρ, \mathbf{u} and relevant velocity derivatives from α^-
5. Update α^- to α^+
6. Calculate pre-IBM density distribution function using (3.29).
7. Calculate B_f and Ω_i^s for IBM using (3.61) and (3.62)
8. Calculate post-collision density distribution function with (3.60)
9. Calculate relaxation times for thermal collision using (3.37) and (3.39)
10. Calculate thermal equilibrium distribution using (3.40).
11. Determine post-collision distribution with (3.41).
12. Save scalar variables to global memory if needed.

FM-FM kernel

1. Calculate H with (3.15).
2. Calculate T, f_i from H with (3.18) and (3.19).
3. Subtract L from g_0 .
4. Construct pre-collision vectors α^- and β^- using (3.30) and (3.47)
5. Determine ρ, \mathbf{u} and relevant velocity derivatives from α^-
6. Determine h and $\nabla \cdot h$ from β^- and \mathbf{u} .
7. Update solution vectors, α^- to α^+ and β^- to β^+ .
8. Calculate pre-IBM density distribution function using (3.29).
9. Calculate B_f and Ω_i^s for IBM using (3.61) and (3.62)
10. Construct the post-collision density distribution function with (3.60)
11. Construct the post-collision sensible enthalpy distribution function with (3.46)
12. Add L to g_0^s
13. Save scalar variables to global memory if needed.

3.5.3. Propagation and Boundary Conditions

In this section a step-by-step overview of the propagation kernel is presented. The boundary conditions are included in this kernel. The steps are valid for both the density and the enthalpy distribution function

- Pull in the post-collision populations from global memory
- Apply solid wall conditions
- Apply inlet and outlet conditions

This concludes the general numerical methods employed in this study to create a versatile model capable of simulating turbulent, thermal channel flows with freezing. Starting from the basics of DDF-LBM in section 3.1, section 3.2 moved on to describe the FM operator for momentum flow and thermal flow and the OTRT operator for thermal flow. Subsequently, section 3.3 described the different boundary conditions implemented. The adjustments to the general algorithm needed for the GPU implementation were presented in section 3.4 and finally a step-by-step overview of the algorithm was given in section 3.5. The next chapter is dedicated to the validation of the model in the laminar flow regime.

4

Laminar Flow

The main goal of this research is to test the capabilities of the GPU-DDF-LB models for simulating heat transfer in turbulent channel flows under the influence of transient freezing. To assess the validity of the simulation results, the model is first tested in the laminar flow regime. This is the topic of this chapter. First an overview is given of the combination of boundary conditions that produced the best results in section 4.1. Next section 4.2 will cover the performance of the D3Q19 FM scheme for momentum transfer in laminar flow, where the simulation results are compared to a well-known analytical solution and a grid convergence test is performed. Section 4.3 considers the addition of the enthalpy distribution function. Asymmetrical heating is simulated and the performance of the OTRT operator and the thermal FM operator are compared. Section 4.4 covers transient freezing in forced convection. The chapter concludes with a note on stability limits observed when performing the simulations in section 4.5.

The simulation results in the laminar flow regime are compared to literature results using the Relative Root-Mean-Squared error defined as

$$\mathcal{E} = \sqrt{\frac{1}{N} \sum_i^N \frac{(\phi_i^{lit} - \phi_i^{lb})^2}{(\phi_i^{lit})^2}} \times 100\%. \quad (4.1)$$

4.1. Boundary Conditions

Boundary conditions are of utmost importance to numerical simulations and thus finding BC's that work best is the topic of this section. The boundary conditions were tested in the case of laminar flow with uniform inlet and wall temperature.

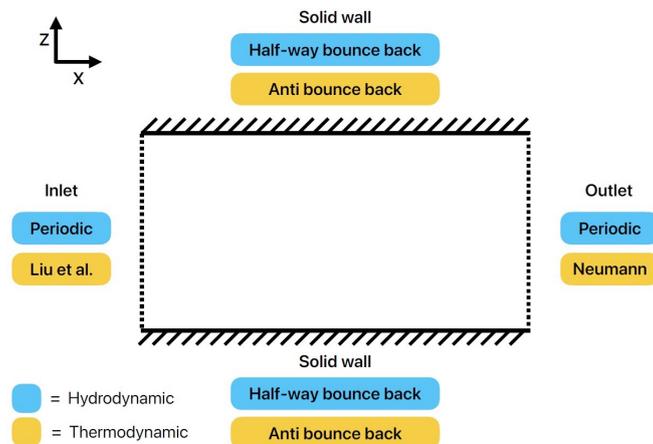


Figure 4.1: Schematic depiction of the simulation domain showing the boundary conditions implemented at the inlet, outlet and solid walls. On the spanwise boundaries periodic conditions are implemented

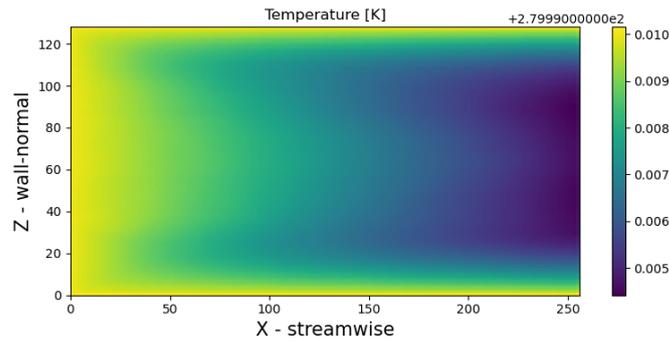


Figure 4.2: Converged temperature profile resulting from the implemented set of boundary conditions.

For the solid wall boundaries the HBB method for the density distribution function f_i and the ABB method for the enthalpy distribution function g_i showed good results without unphysical behaviour. They are therefore adopted in all simulations performed in the current research.

The different inlet and outlet BC's showed to have different effects on the hydrodynamic and thermodynamic flow solutions. Preliminary investigation showed that employing periodic boundary conditions for f_i was the only way to avoid unphysical effects at the inlet and outlet. For the thermodynamic in- and outlet a uniform inlet temperature is needed and thus periodic in- and outlet conditions are not applicable. It was found that a combination of the inlet condition by Liu (equation (3.54) - (3.56)) and the Neumann outlet condition (3.57) showed the least amount of inlet effects.

The full set of boundary conditions implemented in the simulations is

- Hydrodynamic in- and outlet: Periodic BCs,
- Thermodynamic inlet: BC by Liu et al.,
- Thermodynamic outlet: Neumann condition,
- Hydrodynamic spanwise boundaries: Periodic BCs,
- Thermodynamic spanwise boundaries: Periodic BCs,
- Hydrodynamic solid walls: HBB method,
- Thermodynamic solid walls: ABB method.

A schematic depiction of the domain showing the \hat{x} and \hat{z} boundaries is given in figure 4.1. Figure 4.2 shows a slice of a converged temperature profile from a laminar flow simulation with constant wall and inlet temperature of $280K$. The inlet effects are on the order of $10^{-3} K$, which was deemed acceptable.

4.2. Poiseuille Flow

This section covers the performance of the FM operator for momentum flow. First, the analytical solution and the determination of the flow parameters is explained together with the initialisation of the distribution function. Then the results of a grid convergence test and the final steady state flow profile are discussed. The model showed a very good agreement to the analytical solution.

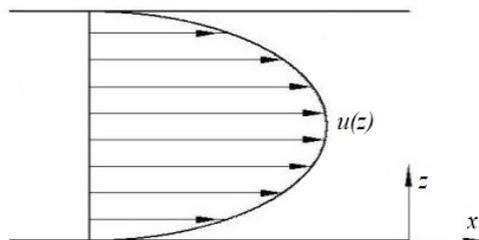


Figure 4.3: Schematic depiction of Poiseuille flow.

Quantity	Value in Lattice Units	Value in Physical Units
$N_x \times N_y$	10×3	$[10 \times 3; 5 \times 1.5; \dots; 0.77 \times 0.23] \text{ cm}$
N_z	$[10, 20, \dots, 130]$	10 cm
\mathcal{H}	$[5, 10, \dots, 65]$	5 cm
Re_τ	10	
U_{max}	0.1	1.434 mm/s
ν	$0.002\mathcal{H}$	$1.434 \text{ mm}^2/\text{s}$
g_x	$0.004/\mathcal{H}$	0.01645 mm/s^2
ρ_0	1	10^3 kg/m^3

Table 4.1: Input parameters for the grid convergence test. Lattice quantities were calculated via the methods described in section 4.2.1.

4.2.1. Analytical Solution and Initialisation

To initiate the simulation the fluid viscosity ν and body force acceleration g_x must be determined in lattice units. This must be done in such a way that the desired flow behaviour is achieved without exceeding the limit on the maximum flow velocity in lattice units. The analytical solution together with input parameters Re_τ , \mathcal{H} and U_{max} are used to calculate ν and g_x . The method is given here.

Laminar body force driven channel flow has a well known analytical solution. Given a body force $\mathbf{F} = \rho g_x \hat{x}$ and $\mathbf{u} = \mathbf{0}$ on the boundaries $z = 0, z = 2\mathcal{H}$, the analytical solution to this 1D flow problem is [84]

$$\mathbf{u} = -\frac{g_x z}{2\nu}(z - 2\mathcal{H})\hat{x}, \quad (4.2)$$

which is known as Poiseuille flow (figure 4.3). Because $u = U_{max}$ for $z = \mathcal{H}$, this gives an expression for U_{max} in terms of flow parameters

$$U_{max} = \frac{g_x \mathcal{H}^2}{2\nu}. \quad (4.3)$$

Using this expression and the definition of Re_τ (2.11), an expression for ν can be formulated as function of the input parameters

$$\nu = \frac{2\mathcal{H}U_{max}}{Re_\tau^2}. \quad (4.4)$$

Finally (4.3) and (4.4) can be combined to find g_x as function of the input parameters

$$g_x = \frac{4U_{max}^2}{\mathcal{H}Re_\tau}. \quad (4.5)$$

Now that the flow parameters have been found, the initial pre-collision solution vector can be constructed from any initial velocity field and an initial density field using equation (3.32). The initial density in LBM research is typically set to $\rho_0 = 1$. This is employed in all simulations. Finally equation (3.29) is used to calculate the initial distribution function.

4.2.2. Results

To test the performance of the density distribution function, the flow was developed from a static state. The streamwise and spanwise dimensions of the domain were kept small, since this is essentially a 1D problem. To study the effect of wall-normal resolution, the channel height was varied. The viscosity and body force acceleration were calculated via (4.4) and (4.5). An overview of the input parameters is given in table (4.1).

The results of the grid convergence test are shown in figure 4.4a. The decrease in error starts strong, but quickly tends to zero for increasing the height. The result for $N_z = 130$ is shown in figure 4.4. A very good agreement to the analytical solution is visible. The obtained error was $\mathcal{E} = 0.053\%$. Now that the performance of the momentum flow model has been benchmarked for laminar flow, the next section goes into the performance of the two different models for the thermal flow.

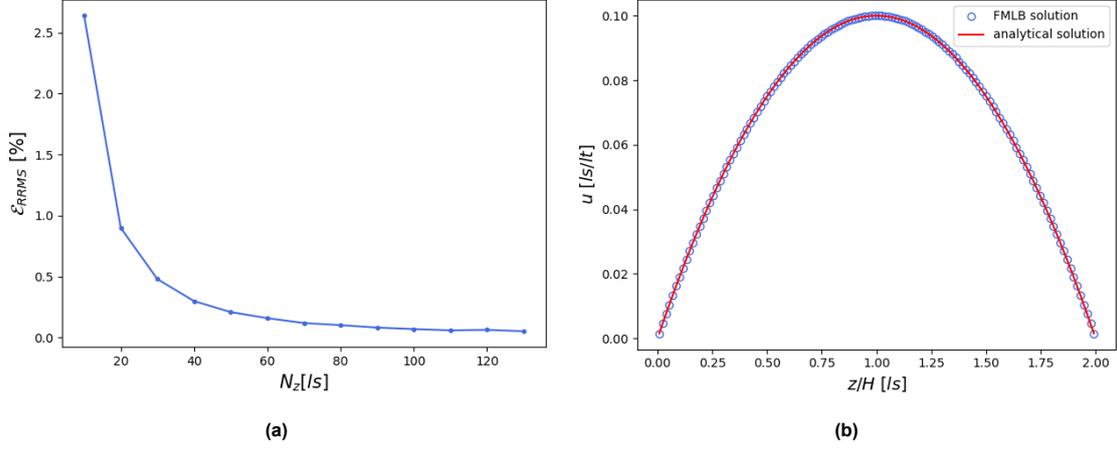


Figure 4.4: Laminar flow results. (a) shows \mathcal{E} compared to the analytical solution after steady state is reached, as function of channel height for the grid convergence test. (b) shows the the streamwise velocity as function of the channel height for $N_z = 130$ which resulted in an error of 0.053%. References to the physical quantities are found in appendix A.

4.3. Local Nusselt Number for Heating in Forced Convection

This section covers the performance of the model in laminar heating situations. The performance of the OTRT operator is compared with that of the thermal FM operator. First the determination of the thermal flow parameters from input and physical parameters, and the initialisation of the simulation is discussed. Then the comparison to analytical solutions is presented and the differences in performance are discussed. Both operators showed a reasonable agreement to the analytical solutions.

4.3.1. Initialisation

Using the method from the previous section the flow parameters ν and g_x are calculated from Re_τ , \mathcal{H} and U_{max} . To simulate thermal flows, additional parameters are needed in lattice units. The quantities needed are the thermal diffusivity α , the specific heat capacity C_p , the thermal conductivity λ , the latent heat L . When dealing with phase change, α , C_p and λ should be known for both the liquid and the solid phase. Finally, the solid and liquid temperatures T_s, T_l should be known. These quantities in lattice units are determined from a combination of non-dimensional numbers and physical parameters. The procedure to do so is described here. The physical quantities are denoted as $\tilde{\phi}$. Lattice quantities are denoted without tilde.

Since ν and g_x are determined from Re_τ , \mathcal{H} and U_{max} , these are set. Using the definition of Pr (2.12), α_l can then be determined

$$\alpha_l = \frac{\nu}{Pr}. \quad (4.6)$$

From the definition of α_l [81] and the choice of $\rho_0 = 1$ in lattice units, which is standard in LBM, the ratio of λ_l to $C_{p,l}$ set to be

$$\frac{\lambda_l}{C_{p,l}} = \alpha_l. \quad (4.7)$$

Since there are no stability constraints on λ_l or $C_{p,l}$, one of these lattice quantities can be chosen freely, which in turn sets the value of the other. In the present work the choice is made for $\lambda_l = 1$. Now several lattice quantities can be calculated using λ_l , physical ratio's and definitions:

$$C_{p,l} = \frac{\lambda_l}{\alpha_l \rho_0}, \quad (4.8)$$

$$\lambda_s = \lambda_l \frac{\tilde{\lambda}_s}{\lambda_l}, \quad (4.9)$$

$$C_{p,s} = C_{p,l} \frac{\tilde{C}_{p,s}}{\tilde{C}_{p,l}}, \quad (4.10)$$

Quantity	Entrance Region Simulation		Entrance Region Simulation	
	Lattice Units	Physical Units	Lattice Units	Physical Units
$N_x \times N_y \times N_z$	$256 \times 3 \times 128$	$20 \times 0.23 \times 10 \text{ cm}$	$256 \times 3 \times 128$	$20 \times 0.23 \times 10 \text{ cm}$
\mathcal{H}	64	5 cm	64	5 cm
Re_τ	10		5	
U_{max}	0.1	1.434 mm/s	0.1	0.3585 mm/s
ν	0.128	1.434 mm ² /s	0.512	1.434 mm ² /s
g_x	6.25×10^{-5}	0.01645 mm/s ²	1.25×10^{-4}	$2.056 \times 10^{-3} \text{ mm/s}^2$
ρ_0	1	10 ³ kg/m ³	1	10 ³ kg/m ³
T_0	280	280 K	280	280 K
T_{inlet}	280	280 K	280	280 K
T_{wall}	282	282 K	282	282 K
T_f	273.15	273.15 K	273.15	273.15 K
α_l	0.01210	0.1456 mm ² /s	0.05199	0.1456 mm ² /s
Pr	9.849		9.849	
λ_l	1	0.607 W/mK	1	0.607 W/mK
$C_{p,l}$	76.94	4182 J/kgK	19.24	4182 J/kgK

Table 4.2: Input parameters for the single sided heating in laminar flow simulations. The lattice quantities are calculated via the methods described in sections 4.2.1 and 4.3.1. References to the physical quantities are found in appendix A.

$$\alpha_s = \frac{\lambda_s}{\rho C_{p,s}}. \quad (4.11)$$

Using the ratio between the latent heat and the solid enthalpy in physical to lattice units, the latent heat in lattice units can be calculated with

$$L = \tilde{L} \frac{H_s}{\tilde{H}_s}. \quad (4.12)$$

Using these equations in combination with the material properties of the fluid gives all the needed quantities in lattice units. For the thermal FM method, the initial distribution function can be found from an initial enthalpy field and flow field using equation (3.46). The distribution function for the OTRT simulations is initialised as the equilibrium distribution function using equation (3.40).

4.3.2. Results

To test the performance of the OTRT and the thermal FM collision operators without phase change, the Nusselt relation was calculated for single sided heating with a constant wall temperature and a fully developed Poiseuille flow profile.

Analytical solution

Approximate analytical solutions for this configuration are presented in [120]. The expressions used to benchmark the current code are

$$Nu(x) = \begin{cases} 1.233Gz(x)^{1/3} + 0.4 & \text{if } Gz(x)^{-1} \leq 0.001, \\ 7.541 + 6.874(0.001Gz(x))^{0.488} e^{-245/Gz(x)} & \text{if } Gz(x)^{-1} > 0.001, \end{cases} \quad (4.13)$$

where $Gz(x)^{-1}$ is inverse of the local Graetz number, defined as

$$Gz(x) = \frac{Re_D Pr D_h}{x}, \quad (4.14)$$

with $D_h = 4\mathcal{H}$ the hydraulic diameter of the channel with height $2\mathcal{H}$, and Re_D the Reynolds number computed with the mean velocity and the hydraulic diameter. $Gz(x)^{-1} \leq 0.001$ defines the thermal entrance region. $Gz(x)^{-1} > 0.001$ defines the thermally developed region.

Two simulations were performed for each collision operator to test the performance in both regions. The input parameters are given in table 4.2. The flow was initialized as a fully developed laminar flow and given some time to stabilize before the heating from the lower wall was initiated. $Nu(x)$ is dependent on the heat transfer coefficient h_c via equation (2.13). h_c can also be written as

$$h_c(x) = \frac{q''(x)}{T_{wall} - T_{mc}(x)} = -\frac{\lambda}{T_{wall} - T_{mc}(x)} \left. \frac{\partial T(x, z)}{\partial z} \right|_{z=0}, \quad (4.15)$$

giving the following expression for local Nusselt number

$$Nu(x) = -\frac{D_h}{T_{wall} - T_{mc}(x)} \left. \frac{\partial T(x, z)}{\partial z} \right|_{z=0}, \quad (4.16)$$

where T_{mc} is the so-called mix-cup temperature, defined as [121]

$$T_{mc}(x) = \frac{\int_0^{2H} \rho C_p u T dz}{\int_0^{2H} \rho C_p u dz}. \quad (4.17)$$

Since the HBB and ABB BC's put the boundary at $0.5 l_s$ from the wall, the temperature derivative at the wall was determined by first using linear interpolation between $k = 0, 1$ and $k = 1, 2$, where k is the index in z -direction, to obtain the temperature at $z = 1$ and $z = 2$. Then the 3-point forward difference formula was used to obtain the derivative

$$\left. \frac{\partial T}{\partial z} \right|_{z=0} = -\frac{3}{2}T(z=0) + 2T(z=1) - \frac{1}{2}T(z=2). \quad (4.18)$$

The temperature at $z = 0$ is T_{wall} .

Simulation Results

The results are shown in figures 4.5 and 4.6. 4.5a shows the $Re_\tau = 10$ FM simulation where the whole domain lies in the thermal entrance region. An error of $\mathcal{E} = 6.0\%$ was found, where it is visible the error is mainly due to the first few nodes of the domain. In this region of the domain, the heat has only travelled a very short distance into the domain. For this reason the resolution in the heated region is low, possibly leading to a slightly larger error. This becomes more evident in figure 4.5b which shows the simulation where only part of the domain lies in the thermal entrance region. Here the resolution in the thermal entrance region is even lower and the error increases to $\mathcal{E} = 25.28\%$. In the thermally developed region an error of $\mathcal{E} = 2.42\%$ is found.

The same seems to be true for the OTRT operator. However the error in the thermal entrance region is higher compared to the FM operator, with $\mathcal{E} = 9.48\%$ in figure 4.6a and $\mathcal{E} = 31.74\%$ in figure 4.6b. The error in the thermally developed region is only 0.43% for the OTRT operator. It seems that although the effects of the low resolution in the entrance are more prominent in the OTRT results, they are corrected more quickly by the OTRT than the FM operator.

Overall both operators show good agreement to the approximate analytical solutions presented in [120]. The FM operator matches the analytical solution in the thermal entrance region slightly better, whereas the OTRT operator produces a lower error in the developed region. The next section goes into the performance of the different collision operators for transient freezing in laminar flow.

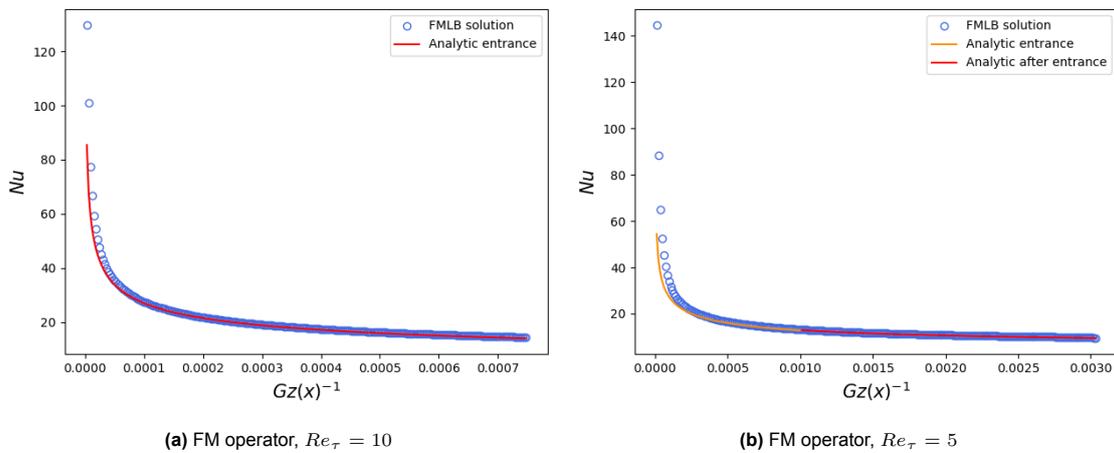


Figure 4.5: Local Nusselt number for asymmetrical heating in laminar flow calculated with the FM operator. (a) shows the thermal entrance simulation. (b) shows the thermally developed simulation.

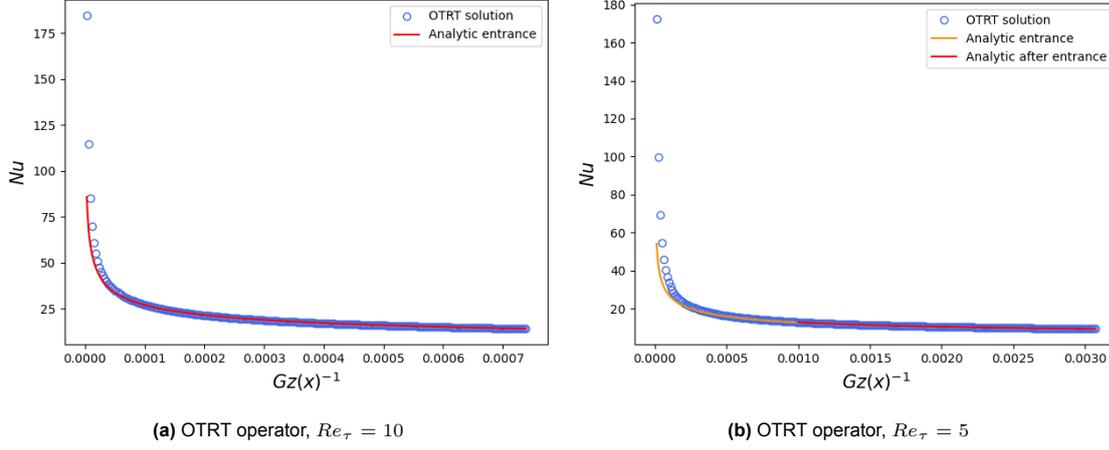


Figure 4.6: Local Nusselt number for asymmetrical heating in laminar flow calculated with the OTRT operator. (a) shows the thermal entrance simulation. (b) shows the thermally developed simulation.

4.4. Transient Freezing

To test the capabilities of the model to simulate freezing under forced convection, a simulation was performed of laminar flow of water between parallel plates with a constant wall temperature below the freezing temperature. The results of this simulation are compared with an analytical solution of transient ice growth presented in by Weigand en Beer in [122]. The solution they proposed is for fully developed laminar flow entering the cooled section of the channel. It was found that the FM operator performed better than the OTRT operator.

Analytical Solution

The solution presented in [122] for the normalised ice thickness $\tilde{\delta} = \delta/\mathcal{H}$ is the solution of the equation

$$\frac{\mathcal{T}^+}{1 - \delta_s} = \frac{1}{2}(\tilde{\delta}^2 - 1) + (\delta_s - 1) \left[\tilde{\delta} - 1 + \delta_s \ln \left(\frac{\tilde{\delta} - \delta_s}{1 - \delta_s} \right) \right], \quad (4.19)$$

where \mathcal{T}^+ is a dimensionless time given by

$$\mathcal{T}^+ = FoSte = \frac{t\alpha_s Cp, s(T_f - T_w)}{\mathcal{H}^2 L}, \quad (4.20)$$

and $\delta_s = \frac{f(\xi)}{f(\xi) - B_f}$ is the steady state distance between the channel centerline and the solid-liquid interface. $f(\xi)$ is the dimensionless temperature gradient at the solid-liquid interface given by

$$\left. \frac{\partial \theta}{\partial \tilde{z}} \right|_{\tilde{z}=\tilde{\delta}} = \sum_{n=0}^{\infty} c_n Y'_n(1) e^{-\frac{2}{3}\lambda_n^2 \xi}, \quad (4.21)$$

where c_n , $Y'_n(1)$ and λ_n are constants, eigenfunctions and eigenvalues given literature [34]. B_f is a freezing parameter given by

$$B_f = \frac{\lambda_s(T_f - T_w)}{\lambda_l(T_0 - T_f)}. \quad (4.22)$$

For a given time \mathcal{T}_1^+ , equation (4.19) yields $\tilde{\delta}$ as a function of ξ . With the transformation relation

$$\frac{d\xi}{d\tilde{x}} = \frac{1}{\tilde{\delta}} \quad ; \quad \tilde{x} = 0 \quad ; \quad \xi = Gz(x)^{-1} = 0 \quad ; \quad \mathcal{T}^+ = \mathcal{T}_1^+, \quad (4.23)$$

$\tilde{\delta}$ is found as function of \tilde{x} with solution

$$\tilde{x} = \xi \tilde{\delta}. \quad (4.24)$$

This equation is solved iteratively using the `scipy.optimize.fsolve` function in Python.

Quantity	Value in Lattice Units	Value in Physical Units
$N_x \times N_y \times N_z$	$512 \times 3 \times 192$	$26.67 \times 0.16 \times 10 \text{ cm}$
\mathcal{H}	96	5 cm
Re_τ	10	
U_{max}	0.1	1.434 mm/s
ν	0.192	1.434 mm ² /s
g_x	4.1667×10^{-4}	0.01645 mm/s ²
ρ_0	1	10 ³ kg/m ³
T_0	275	275 K
T_{inlet}	275	275 K
T_{wall}	270	270 K
T_f	273.15	273.15 K
α_l	0.01949	0.1456 mm ² /s
Pr	9.849	
λ_l	1	0.607 W/mK
λ_s	3.722	2.259 W/mK
$C_{p,l}$	51.30	4182 J/kgK
$C_{p,s}$	25.64	2090 J/kgK
L	4097	$334 \times 10^3 \text{ J/kg}$

Table 4.3: Input parameters for laminar freezing simulation. The lattice quantities are calculated via the methods described in sections 4.2.1 and 4.3.1. References to the physical quantities are found in appendix A.

Initialisation

The simulation is performed with the standard boundary conditions used in all laminar simulations (figure 4.1). For comparison of the simulation results to the analytical solution, the periodic flow conditions introduce a problem. Cooling the entire length of solid wall would create a discontinuity of ice height across the periodic boundary. The ice layer would have zero height at the inlet, due to the inlet temperature $T_{inlet} > T_{wall}$, but non-zero height at the outlet due to the wall temperature. To avoid this only the first quarter of the channel is cooled. As the ice layer grows however, it affects the flow behind it. In combination with the mass flow rate dropping due to the freezing of mass, the flow velocity gradually drops. This is schematically depicted in figure 4.7. Because of this choice the flow impacting the cooled region has a lower velocity than fully developed laminar flow. As this violates one of the assumptions of the analytical solution, this will have a negative impact on the comparison to the analytical solution. To minimize this effect the domain size is increased compared to the laminar heating simulations.

The input parameters are given in table 4.3. The flow was initialized as a fully developed laminar flow at constant temperature T_0 and given some time to stabilize before the cooling was initiated.

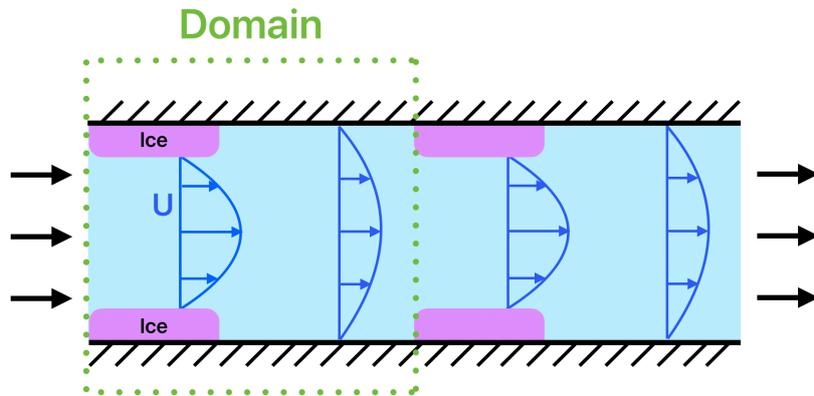


Figure 4.7: Schematic depiction of the domain for the laminar freezing simulations. The periodicity of the domain is highlighted by showing two repetitions of the domain.

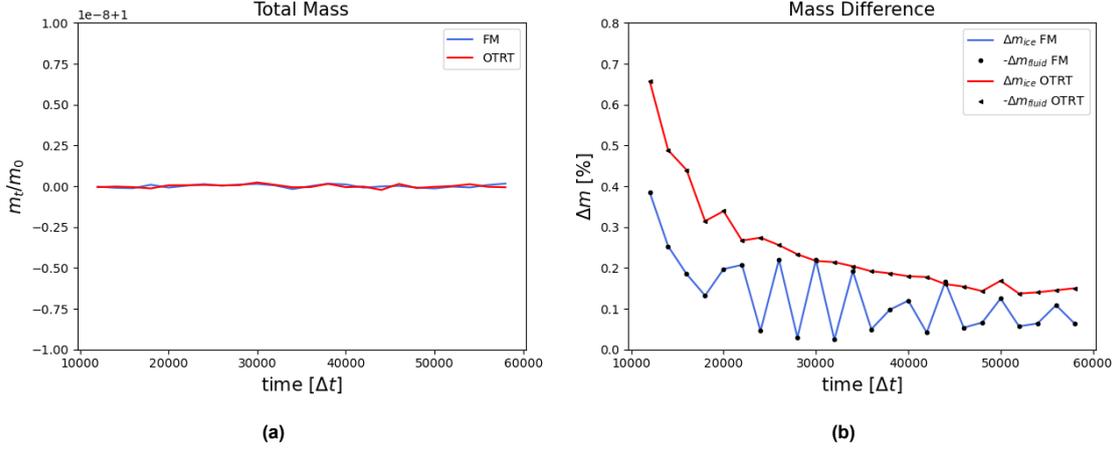


Figure 4.8: Results of the mass conservation tests for laminar freezing. (a) shows the total mass normalised by the initial mass m_0 . Fluctuations are of order 10^{-10} . (b) shows the relative change in mass compared to the previous probe.

Results

To test the mass conservation of the current model in freezing simulations, the total mass and the relative change of frozen and flowing mass were calculated from the moment the freezing started. The results are shown in figure 4.8. Figure 4.8a shows the total mass normalised by the initial mass m_0 , where fluctuations on the order of 10^{-10} are observed, indicating good total mass conservation. Figure 4.8b shows the relative change in frozen and flowing mass. Each data point shows the difference in mass compared to the previous probe. It can be seen that the reduction in flowing mass $-\Delta m_{fluid}$ very closely matches the growth in frozen mass Δm_{ice} , also indicating good mass conservation for both collision operators. The OTRT operator shows a more smooth solidification compared to the FM operator.

The results for an instantaneous ice layer are shown in figure 4.9. 4.9a shows the instantaneous ice layer height at $\mathcal{T}^+ = 0.0139$ for the FM operator, where an error of $\mathcal{E} = 7.16\%$ compared to the analytical solution was found. Compared to the analytical solution the FMLB solution overestimates the height at the bend. Where after it is more flattened out and underestimates the height in the second half. The mismatch in ice profile can be attributed to the slight deviation from the analytic case in terms of inflow. As the flow the velocity reduces due to ice growth, causing less convection, a sharper corner is feasible. This effect was further examined by plotting the ice layer and velocity profiles at later times, where the same behaviour was observed but more pronounced. As the velocity became lower, the overestimation of the ice height grew. These graphs are included in appendix B.

Figure 4.9b shows the instantaneous ice layer height at $\mathcal{T}^+ = 0.00928$ for the OTRT operator, where an error of $\mathcal{E} = 40.04\%$ was found. The difference between the two operators is striking, as their performance was very similar in the heating experiments. The ice layer produced by the OTRT operator shows opposite characteristics compared to the FM operator. The ice height is underestimated at the bend but keeps increasing more steeply, leading to an overestimation further in the domain. It is also visible the ice layer is more ragged and shows small local dips. Because the results for the heating case were much better, it seems that the inaccuracy stems from heat transfer through the solid phase and at the phase interface. Because Lu et al. did not report these findings in natural convection melting, the error seems to be brought up by the higher convection in the current simulations.

4.5. A Note on the Stability of the Thermal Flow Models

While performing the simulations, instabilities were encountered for different values of Pr while increasing Re_τ . Due to the calculation of the hydrodynamic and thermodynamic parameters, raising Re_τ and Pr effectively lowers ν and α . Lowering these values generally leads to more unstable behaviour for LBM simulations [61]. For the OTRT operator this is especially true. Small values of α lead to a big difference between \mathcal{T}_a and \mathcal{T}_s through equations (3.37) and (3.39). As \mathcal{T}_a approaches 0.5, \mathcal{T}_s becomes very large and the symmetric term of the collision operator goes to zero. Thus the OTRT operator approaches an antisymmetric SRT operator.

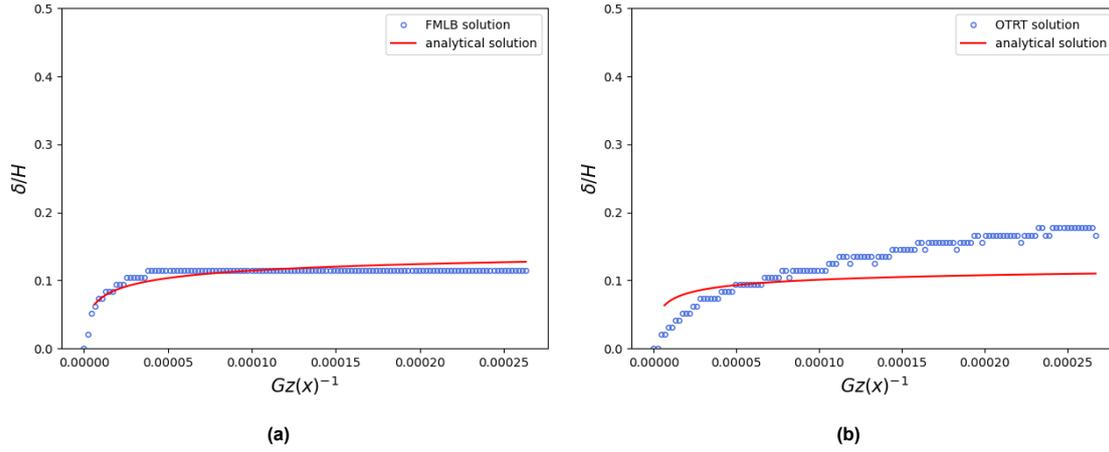


Figure 4.9: Instantaneous ice height compared to the analytical solution for the FM operator (a) and the OTRT operator (b).

To map the observed instabilities for both operators, several simulations were performed at different Re_τ with periodic inflow and outflow conditions for both distributions, eliminating any inlet effects, and constant wall temperatures $T_{wall} = T_0$. The Pr were chosen as 10, 1, 0.1 ranging from water to the order of magnitude of liquid metals. The results are depicted in table 4.4, where S indicates stable results, F indicates fluctuations smaller than $1K$, LF indicates large fluctuations on the order of multiple kelvin and D indicates diverging simulation results.

The FM operator exhibits small fluctuations on the order of $0.1K$. These generally are smoothed out as the simulations settle or remain on this order. However for higher Re_τ and Pr these fluctuations become unstable and cause the results to diverge. It was found that the thermal FM-operator diverges for $Pr = 10$ and $Re_\tau \geq 30$. For $Pr = 1$ the results were stable in the laminar flow regime. However, preliminary tests for turbulent flow showed unstable behaviour for $Pr = 1$ at $Re_\tau = 180$. $Pr = 0.1$ simulations did not show unstable fluctuations both laminar or turbulent.

The OTRT operator also showed fluctuations. These did not cause the simulations to diverge, but did become larger for higher Re_τ and higher Pr . For certain combinations, the fluctuations grow to the order of several degrees and the frequency increases leading to erratic fluctuations. This was found for $Pr = 10$ and $Re_\tau \geq 50$ and for $Pr = 1$ and $Re_\tau \geq 60$. For $Pr = 0.1$ the fluctuations and their frequency remained small.

Thus it was found that higher Pr simulations become more unstable for higher Re_τ . Decreasing the maximum lattice velocity had a positive effect on stability. This can be explained by the fact that decreasing the lattice velocity also decreases the lattice time step size in physical units. To simulate the same amount of physical time the total number of simulated time steps must thus also be increased by the same factor. By implementing a maximum lattice velocity of $U_{max} = 0.005 \text{ ls/lt}$, the FM operator was able to simulate $Pr = 10$ at $Re_\tau = 50$ without significant instabilities. This requires the total number of time steps to be increased by a factor 20.

These findings correspond to the general notion that higher Pr flows require a higher resolution [123], because of the different timescales the thermal structures evolve. However, the fact that fluctuations arise in simulations where both wall and temperatures are kept constant indicates stability limits inherent to the collision operators. The influence of these fluctuations on the accuracy of the results in the turbulent flow regime will be discussed in the next chapter.

FM Pr	Re							OTRT Pr	Re						
		20	30	40	50	60	180			20	30	40	50	60	180
	10	S	D					10	S	F	F	LF			
	1	S	S	S	S	S	D	1	S	S	F	F	LF		
	0.1	S	S	S	S	S	F	0.1	S	S	S	S	F	F	

Table 4.4: results of the stability analysis. S , F , LF and D indicate 'stable', 'fluctuations', 'large fluctuations' and 'diverging results' respectively.

4.6. Laminar Benchmark Conclusion

The GPU-FMLB model for momentum flow shows accurate performance in the laminar regime, with an error of only 0.053% with respect to the analytical Poiseuille profile. Two operators for the thermal flow based on the enthalpy formulation were tested and compared in two benchmark cases. In the benchmark case of single-sided heating the operators performed similarly and both gave good agreement to the analytical expressions for $Nu(x) = f(Re, Pr)$. The FM operator showed slightly better performance in the thermal entrance region. The OTRT showed better performance in the thermally developed region. These differences were small however and both methods were able to give good results.

In the benchmark case of transient ice layer development, the differences were more striking. The FM operator produced a more stable ice layer with an error of 7% with respect to the analytical solution. This deviation can be attributed to the chosen set of boundary conditions that did not fully match the benchmark case. Thus the FM operator produces satisfactory results. The OTRT operator produced more ragged ice layers and a significant overestimation of the ice layer thickness with an error of over 40% with respect to the analytical solution. The shape and ragged profile did not match the expected behaviour, thus it was concluded the OTRT operator in combination with the immersed boundary method does not produce satisfactory results for simulating transient freezing in laminar forced convective flow.

The stability tests indicate the accuracy and stability of the thermal models deteriorate for higher Re, Pr flows. This matches expectation, as higher Re and Pr correspond to lower numerical values of ν and α , which generally leads to instabilities for LBM models. Decreasing the maximum lattice velocity had a positive effect on stability. However, this also requires an increase of the number of time steps with the same factor, which significantly increases total simulation time. The effects of the restrictions on Pr for the turbulence simulations will be discussed in the next chapter.

5

Turbulent Flow

The main goal of this thesis is to investigate the capability of the GPU-DDF-LB models to simulate transient freezing in turbulent channel flow. Now that the model has been benchmarked in the laminar flow regime, the next step is to investigate its capabilities in the turbulent flow regime. First, the performance of the model in terms of simulation speed and parallel efficiency will be discussed in section 5.1. Section 5.2 covers the benchmarks of the momentum flow model, where two simulations at different Re_τ values are compared to well-established benchmark solutions. After this the results of the thermal flow models in isothermal flow are assessed, followed by a comparison to another benchmark case from literature in section 5.3. Finally a preliminary calculation for the heat transfer correlation of asymmetrically heated lead-bismuth eutectic in turbulent channel flow is presented in section 5.4.

5.1. Parallel Performance

A DNS of a turbulent flow generally requires a great amount of computational resources. To this end the code for the present research was optimized for parallel executions on a GPU. This section discusses the performance of the GPU-DDF-LB models in terms of speed and parallel efficiency. The GPU used for the simulations was the NVIDIA Tesla V100S GPU, part of the DelftBlue High Performance Computer Cluster [124]. The hardware specifications of this GPU are given in table 5.1.

The speed tests for the different GPU-DDF-LB models were performed on cubic lattices and scalar values were saved every $10^4 \Delta t$. The performance of the models is measured in million lattice updates per second (MLUPS), where a single iteration over a domain of size $N_x \times N_y \times N_z$ accounts for $N_x * N_y * N_z$ lattice updates. The results are shown in figure 5.1. For grids larger than 2×10^6 nodes, performance stays approximately stable. Above this limit the single distribution function model reaches

NVIDIA Tesla V100S		
CUDA cores		5120
Max. grid dimensions	$2^{31} \times 2^{16} \times 2^{16}$	
Max. grid dimensions	$2^{10} \times 2^{10} \times 2^6$	
Number of SM's		4
Max. threads per block		1024
Max. blocks per SM		32
Max. threads per SM		2048
Register mem. per SM (KiB)		64
Shared mem. per block (KiB)		64
Local mem. per thread (KiB)		512
Global memory (GB)		32
Memory bandwidth (GB/s)		1134

Table 5.1: GPU Hardware specification [124] [125]

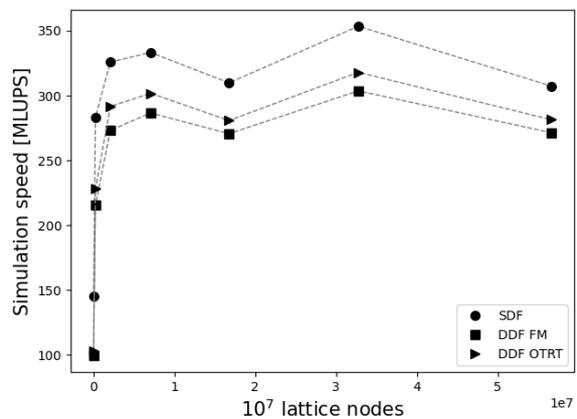


Figure 5.1: Performance of the different GPU-DDF-LB models for a square channel in MLUPS.

a speed of approximately 320 MLUPS on average. The addition of the second distribution function does not drastically decrease the speed. Because g_i only has 7 components as opposed to the 19 components of f_i , the thermal collision step is computationally less expensive than the momentum collision step. The OTRT model achieves a speed of approximately 290 MLUPS on average, making it slightly faster than the FM model that reaches 280 MLUPS on average.

The performance of the models is measured against the maximum theoretical performance. Because GPU-LBM models are constraint by memory usage, the maximum theoretical performance can be calculated using

$$LUPS = \frac{BW_{SDK}}{N_A}, \quad (5.1)$$

where BW_{SDK} is the GPU's memory bandwidth and N_A the number of memory accesses per node [72]. This calculation only considers reading from and writing to GPU memory and omits any performed calculations, thus 100% efficiency is not feasible. One iteration for the momentum distribution function consists of 19 reads and 19 writes for both the collision and streaming step. The thermal distribution adds another 7 reads and 7 writes for the populations in both collision and streaming. Because each element consists of 4 bytes of memory, this gives a total of $N_A = 4 \cdot (4 \cdot 19 + 4 \cdot 7) = 416$ bytes for a single node for the current DDF models. As reported by Delbosc et al. [72] and Forslund et al [126] the practical bandwidth of a GPU is typically 70% of the theoretical bandwidth. Using equation 5.1 to compare the maximum theoretical performance to the performance of the current models, both the FM and the OTRT models achieve approximately 15% efficiency.

Other GPU LBM implementations in literature have reported efficiencies around 80% [72] [126] [117]. There are several improvements possible to the current algorithm to increase the efficiency.

- **Shared memory** - The current scheme does not make use of shared memory in the collision or the stream step. Because shared memory is much faster than global memory, this will have a significant impact on computational speed. Especially the streaming step can be optimized by writing a communication scheme between thread blocks [72] [118] or by using the Open Graphics Library [126].
- **Combining the stream and collision kernel** - By changing the order of the collision and the stream step and combining these processes in a single kernel, the computational speed can be further increased as demonstrated in [72] [119]. This was attempted and speeds approaching 1000 MLUPS were found. However, race conditions arose that could not be readily resolved and simulation results became inaccurate.
- **Memory layout** - The memory layout can be further optimized by implementing the 'Structure of Arrays' layout [72] [117]. A first attempt at this layout lead to a speed-up of roughly 50%, but again unresolved race conditions arose, making the simulation results unusable.

Although the present models are inefficient when measured against other GPU-LBM implementations, a speed-up of about 300% was achieved compared to a FM-LBM implementation by Zhuo and Zhong [78] optimized to run in parallel on a 16-core CPU. The current models allowed the largest simulations to run within several days and the computational speeds were deemed acceptable for the goal of this research.

5.2. Turbulent Channel Flow

To calculate heat transfer correlations in turbulent flow, it is important that the hydrodynamic model accurately simulates the turbulent flow over a range of Re -values. To this end two simulations were performed at different Re_τ and the results compared to literature. Section 5.2.1 covers the initialisation of the turbulence simulations. In section 5.2.2 the results for $Re_\tau = 180$ will be compared to a widely accepted benchmark case and a CPU-FMLB simulation. Section 5.2.3 then compares the results for $Re_\tau = 395$ to another widely accepted benchmark case.

The error norm used to compare results of the turbulent simulations is the Root-mean-squared error defined as

$$\mathcal{E} = \sqrt{\frac{1}{N} \sum_i^N (\phi_i^{lit} - \phi_i^{lb})^2}. \quad (5.2)$$

A different error norm than for the laminar flow benchmarks is used. This is because it was observed that the very small values of quantities near the edges of the domain produce very large errors where the actual difference is orders of magnitude smaller than the values in the bulk of the domain. The RMS error gives a better view of the absolute difference between simulation data and literature and as such should be viewed with respect to the order of magnitude of the quantity that is compared. Before the results are discussed, section 5.2.1 discusses the initialisation method used for the simulations.

5.2.1. Initialisation

Flow Parameters

To keep the maximum flow velocity below the previously mentioned stability limit of 0.12 ls/lt , care must be taken in the initialisation. The flow parameters ν and g_x are calculated to produce a turbulent flow at the wanted Re_τ .

First the non-dimensionalisation of the velocity (2.19) is rewritten to obtain an expression for u_τ in terms of \bar{u} and z^+ . Using equation (2.22), the wall shear velocity in the logarithmic layer can be written as

$$u_\tau = \frac{\bar{u}}{2.5 \ln(z^+) + 5.5}. \quad (5.3)$$

Substituting the channel half-height into z^+ recovers Re_τ

$$z^+|_{z=\mathcal{H}} = \frac{\mathcal{H}u_\tau}{\nu} = Re_\tau. \quad (5.4)$$

Substituting this into equation (5.3), together with the maximum flow velocity U_{max} as the mean velocity \bar{u} , leads to the following expression for the wall shear velocity in the center of the channel

$$u_\tau = \frac{U_{max}}{2.5 \ln(Re_\tau) + 5.5}, \quad (5.5)$$

which can be calculated. This is an approximation because the logarithmic layer doesn't reach all the way up to \mathcal{H} . Furthermore u_τ is directly coupled to the shear stress at the wall, thus defining it at the channel wall would be more intuitive. Nevertheless this method was found to produce satisfactory results. Now the definitions of Re_τ (2.11) and u_τ (2.10) can be used to find ν and g_x as function of the input parameters with

$$\nu = \frac{u_\tau \mathcal{H}}{Re_\tau}, \quad (5.6)$$

and

$$g_x = \frac{u_\tau^2}{\mathcal{H}}. \quad (5.7)$$

Domain Size

As equation (2.15) shows, the ratio of the macrostructure length scale to the microstructure length scale is approximately $Re^{3/4}$. In order to have the same grid resolution with respect to the microstructure, the size of the domain must be adjusted to the Re . The domain for the $Re_\tau = 180$ simulation was based on two LB simulations performed by Amati [127] and Rohde [128], who reported it was sufficient to resolve the first order turbulent statistics.

To obtain the domain for the higher Re_τ simulation, the ratio of the macro- to the microstructure (2.15) was used together with the restriction on block sizing for GPU efficiency. Using equation (2.15) together with the fact that the smallest resolved structures will be of lattice spacing size, i.e. $\eta \approx \Delta x$, an approximate relation can be found for the domain sizes at different Re

$$\frac{\mathcal{H}_1/\Delta x}{\mathcal{H}_2/\Delta x} = \left(\frac{Re_1}{Re_2} \right)^{3/4}. \quad (5.8)$$

From this the channel half-height for the higher Re simulation was calculated. Additionally it was required that the streamwise length to height ratio was at least as large as the $Re_\tau = 180$ domain and that the number of lattice nodes in the streamwise direction was a multiple of 32 for GPU thread occupation efficiency. The resulting domain sizes are shown in table 5.2, together with the rest of the input parameters.

Quantity	Low Turbulence Simulation		High turbulence Simulation	
	Lattice Units	Physical Units	Lattice Units	Physical Units
$N_x \times N_y \times N_z$	$256 \times 128 \times 128$	$20 \times 10 \times 10 \text{ cm}$	$480 \times 232 \times 232$	$20 \times 10 \times 10 \text{ cm}$
\mathcal{H}	64	5 cm	116	5 cm
Re_τ	180		395	
U_{max}	0.1	95.41 mm/s	0.1	231.6 mm/s
ν	1.923×10^{-3}	1.434 mm ² /s	1.4362×10^{-3}	1.434 mm ² /s
g_x	4.5741×10^{-7}	0.5330 mm/s ²	2.0619×10^{-7}	2.567 mm/s ²
ρ_0	1	10 ³ kg/m ³	1	997 kg/m ³

Table 5.2: Input parameters for the turbulence benchmark simulations. The lattice quantities are calculated via the methods described in sections 5.2.1. References to the physical quantities are found in appendix A.

Initial Velocity Field

To initialize the turbulent flow field, a velocity field is constructed from a field of divergence free fluctuations superimposed on a laminar velocity profile. The density field is taken as uniform with $\rho_0 = 1$. The initial pre-collision distribution function is constructed from the velocity density fields using equation (3.29). The simulation is run for $t^+ \approx 54$ to let the flow develop, as was done in [78].

Saving

After the start-up phase, instantaneous velocity fields are saved every $10^4 \Delta t$, which corresponds to $t^+ \approx 0.85$, for a total of 100 saves. The final distribution function is saved to be used as a starting point for further simulations with the respective Re_τ values.

5.2.2. Low Re Turbulence

This subsection covers the results of the turbulent flow benchmark at $Re_\tau = 180$. The DNS of fully developed turbulent channel flow at $Re_\tau = 180$ performed by Kim, Moin and Moser [129] (KMM) is one of the most widely used benchmark cases in turbulence research. Using an uneven gridspacing they showed that they could resolve all turbulent scales in their simulation. More recently Zhuo and Zhong [78] (ZZ) performed an LES-based FMLB simulation of channel flow at the same Re_τ . The results of these two papers will be used to compare the current GPU-FMLB model.

The statistical quantities discussed in section 2.3 were calculated from the 100 instantaneous flow fields after the simulation finished. The results are shown in figure 5.2.

A good agreement of the mean streamwise velocity (figure 5.2a) to both the results of KMM and the analytical expressions in the viscous sublayer and the logarithmic layer can be seen in figure 5.2a. An error of $\mathcal{E} = 0.053$ was found between the current GPU-FMLB model and the data of KMM. This is 3 orders of magnitude lower than \bar{u}^+ itself. This difference is an indication that the current grid spacing is sufficient to accurately calculate the mean velocity profiles.

Figure 5.2b also shows a good agreement. Total shear stress can be calculated with [129]

$$\tau_t^+ = -\tau_{Re}^+ + Re_\tau^{-1} \frac{\partial \bar{u}}{\partial z}, \quad (5.9)$$

and for fully developed channel flow this should be straight line, i.e. $\tau_{conv}^+ = 1 - z/H$. KMM use this as a convergence norm for their simulations. From the figure it is clear that this is satisfied to a good extend by the current model. An error of $\mathcal{E} = 0.0028$ was obtained between the current results and the line τ_{conv}^+ , which is 2 orders of magnitude lower than the simulation result, indicating the simulation did indeed statistically converge. Deviations are largest near the channel wall, where the turbulent structures become smaller. The error between the current results and τ_{Re}^+ calculated by KMM is $\mathcal{E} = 0.0075$ where again deviations are most prominent near the wall of the channel.

The turbulent intensities (figure 5.2c) show slight deviations from the results of KMM and ZZ. Compared to KMM the error for the x , y and z components are $\mathcal{E} = 0.10, 0.022$ and 0.020 respectively. The discrepancy between the current simulation and literature can in part be attributed to the lack of resolution in the wall region. KMM use a variable grid spacing with smaller cells near the walls. Their first grid cell has a length of $z^+ \approx 0.05$, where the grid spacing for the current simulation is $\Delta z^+ \approx 2.8$, making the first cell of the current simulation 28 times longer. The error compared to ZZ for the x , y and z components are $\mathcal{E} = 0.074, 0.014$ and 0.017 respectively, which is a better agreement than to KMM.

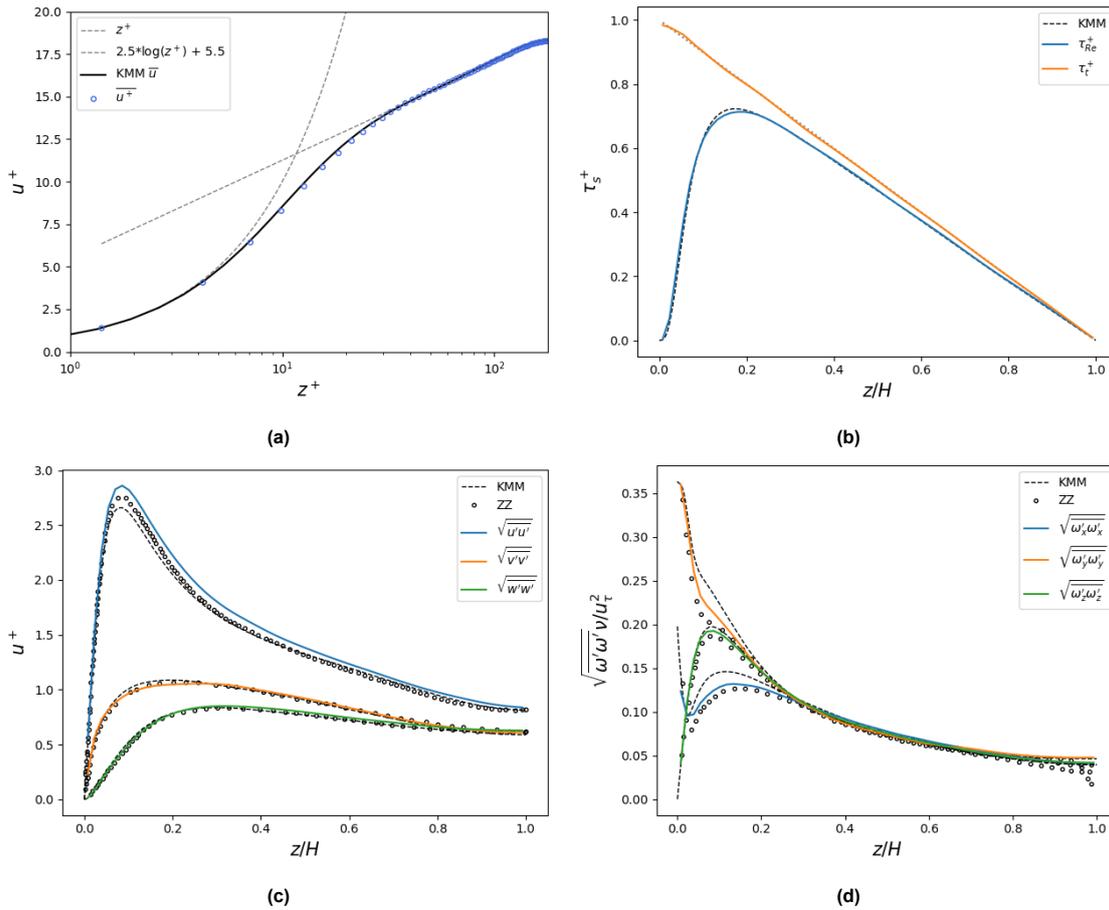


Figure 5.2: Results of the turbulence benchmark at $Re_\tau = 180$. (a) shows the mean streamwise profile, (b) shows the Reynolds stress and total shear stress, (c) shows the turbulence intensities and (d) shows the RMS vorticity fluctuations.

ZZ also attribute their overshoot for the x -component and the undershoot for the y -component in the near-wall region to the lack of resolution.

Finally the RMS vorticity fluctuations (figure 5.2d) also deviate somewhat from KMM and ZZ. The errors of the x, y, z components with respect to KMM are $\mathcal{E} = 0.0066, 0.0071$ and 0.0021 respectively. The errors of the x, y, z components with respect to ZZ are $\mathcal{E} = 0.0039, 0.0079$ and 0.0053 respectively. The deviations are clearly most prominent near the wall, which can again be attributed to insufficient grid sizing in this region. Overall the present results show better agreement to KMM than the results of ZZ, indicating that the present GPU-FMLB scheme better captures the rotating structures than the LES-based FMLB scheme of ZZ.

5.2.3. High Re Turbulence

To test the performance of the hydrodynamic model at higher Re , a simulation at $Re_\tau = 395$ was performed. Kim, Moser and Mansour (KMM) [130] performed a DNS for fully developed channel flow at $Re_\tau = 395$, which is used as a benchmark for the performance of the current model. Initialisation and saving was done analogously to the low Re case. The results are shown in figure 5.3.

The mean streamwise velocity for the high Re simulation (figure 5.3a) shows very good agreement to both the analytical expressions and the results of KMM. The error with KMM is $\mathcal{E} = 0.076$. This shows that the current model is successful in producing the expected mean behaviour for both low and high Re turbulence.

The Reynolds and total shear stress (figure 5.3b) also show good agreement to the benchmark results. Although τ_{Re}^+ shows a slight underestimation nearing the wall, the result still closely matches KMM with an error of $\mathcal{E} = 0.011$. The total shear stress, calculated with equation (5.9), shows a linear profile with small deviations. These deviations were also observed by KMM. They point out these

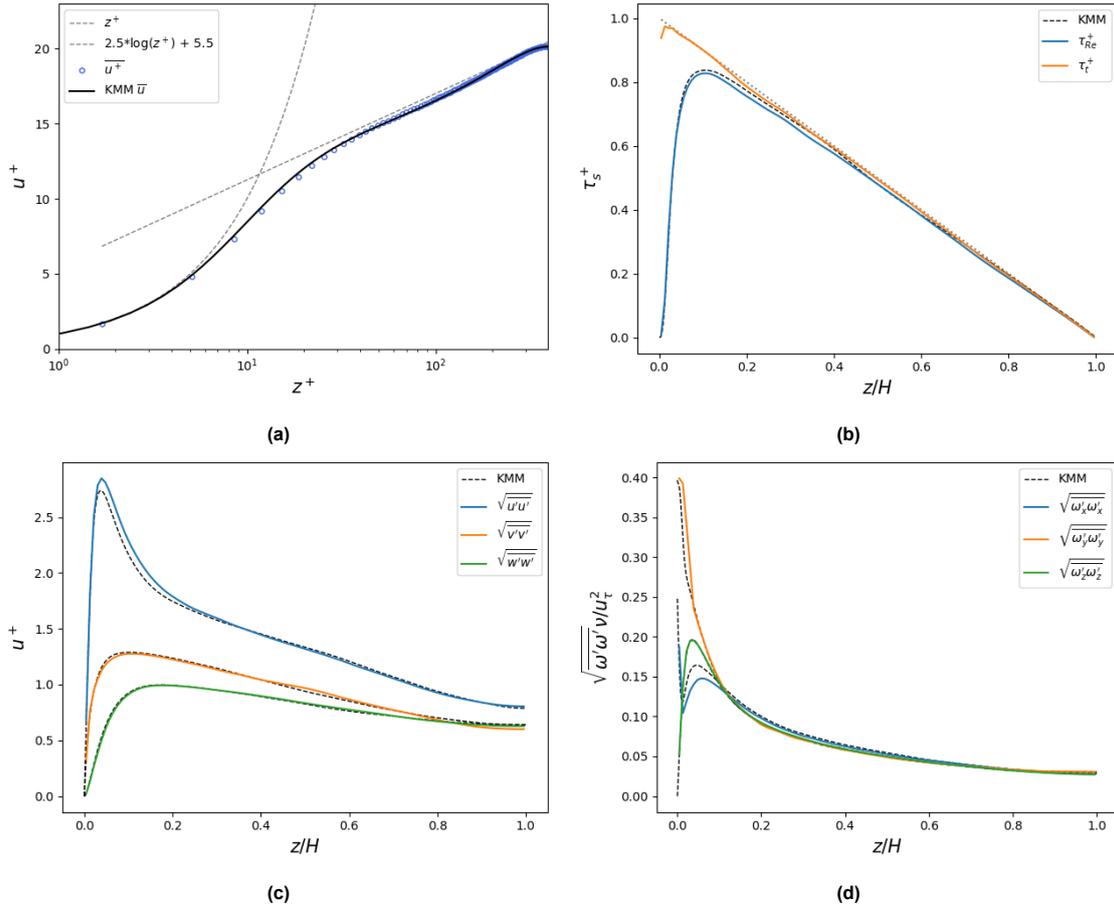


Figure 5.3: Results of the turbulence benchmark at $Re_\tau = 395$. (a) shows the mean streamwise profile, (b) shows the Reynolds stress and total shear stress, (c) shows the turbulence intensities and (d) shows the RMS vorticity fluctuations.

deviations are consequence of the finite number samples for constructing the averages. Near the wall a local dip can be observed, which can be attributed to the lack of resolution in the near wall region and numerical errors introduced by the derivative near the edge of the domain. The error of τ_t^+ with the straight line is $\mathcal{E} = 0.0083$.

The turbulent intensities produced in the high Re simulation (figure 5.3c) more closely match the results from KMM than for the $Re_\tau = 180$ case, with errors of $\mathcal{E} = 0.046, 0.020, 0.020$ for the x, y, z -components respectively. The channel wide overshoot of the x -component, observed in the $Re = 180$ case, is not present here. The overshoot is still visible in the near-wall region. As the size of the grid cells relative to the size of the microstructure is approximately equal for the high and low Re case, this overshoot can again be attributed to a lack of resolution in this region. The difference between the two cases could be connected to the slight difference in the streamwise domain length. As Re increases and the microstructure becomes smaller, the correlation between two points separated over a certain distance also becomes smaller. Because of the periodic boundary conditions and insufficient domain size each point can, very marginally, influence itself. This combined with the slightly larger N_x/N_z ratio for the high Re can lead to the somewhat better results.

Finally, the RMS vorticity fluctuations (figure 5.3d) also show very good agreement to the results of KMM. The errors for the x, y, z -components are $\mathcal{E} = 0.0055, 0.0093, 0.00081$ respectively. The trends are very similar to the low Re case, where the x -component exhibits the most deviation from KMM with an undershoot near the wall, which can again be attributed to a lack of resolution. The y -component shows slight overestimation compared to KMM, while the z -component shows very close resemblance all the way to the wall.

5.2.4. Turbulence Benchmark Conclusion

This concludes the benchmarking of the GPU-FMLB model for momentum flow in the turbulent regime. All calculated statistics show very good qualitative agreement to benchmark results for both $Re_\tau = 180$ and $Re_\tau = 395$. The first order statistics generated by the current GPU-FMLB model match the results of literature to a good degree. The mean streamwise velocity and the Reynolds stress show very good quantitative agreement to KMM. The RMS fluctuations of the velocity and the vorticity fields show more quantitative discrepancies. The streamwise velocity fluctuations show an overshoot of roughly 10% near the wall, which becomes smaller closer to the channel, for $Re_\tau = 180$. The vorticity fluctuations of the current algorithm actually show better agreement to KMM than the data produced by ZZ for $Re_\tau = 180$. The x and y components show the most deviations and all deviations are most prominent in the near wall region. As pointed out by ZZ these discrepancies can be attributed to a lack of resolution compared to the benchmark case. The next section covers the benchmarking of the GPU-LBM schemes for thermal flow in turbulent channel flow.

5.3. Thermal Benchmark in Turbulent Channel Flow

This section covers the benchmark of the performance of the two collision operators for the total enthalpy distribution functions in the turbulent flow regime. During the stability tests presented in section 4.5 it was found the current models can only simulate turbulent flows of fluids with $Pr \leq 0.1$, i.e. liquid metals. Even at low Pr , unexpected temperature fluctuations were observed. To study these fluctuations in more detail, simulations at constant temperature were performed. Section 5.3.1 discusses the results of these simulations with constant and equal wall temperature in a $Re_\tau = 180$ and $Pr = 0.025$ channel flow. Section 5.3.2 then compares the current results to a DNS performed by Kawamura, Abe and Shingai [131] for single sided heating in a $Re_\tau = 180$ and $Pr = 0.025$ channel flow.

5.3.1. Constant Temperature

From the stability analysis discussed in section 4.5 it became apparent that both the OTRT and the thermal FM operator produce unphysical temperature fluctuations. For high Pr these fluctuations became very large, but for lower Pr they kept below the order of $1K$. This section aims to show the effect of these fluctuations on the temperature profile in a $Re_\tau = 180$ channel flow with constant and equal wall temperatures and periodic in- and outflow conditions. Because no temperature differences are introduced, all deviations from T_0 are unphysical.

Initialisation

The flow field was initialized as a fully developed turbulent channel flow at $Re_\tau = 180$, taken from the saved final distribution function of the flow benchmark. To stay below the found stability limit, Pr was set to 0.025. The initial temperature was set to T_0 everywhere and the wall temperatures were kept constant at $T_0 = 280$. Periodic in- and outflow conditions were implemented for both distributions to

Quantity	Value in Lattice Units	Value in Physical Units
$N_x \times N_y \times N_z$	$256 \times 128 \times 128$	$20 \times 10 \times 10 \text{ cm}$
\mathcal{H}	64	5 cm
Re_τ	180	
U_{max}	0.1	11.66 mm/s
ν	1.923×10^{-3}	$0.1752 \text{ mm}^2/\text{s}$
g_x	4.5741×10^{-7}	$7.956 \times 10^{-3} \text{ mm/s}^2$
ρ_0	1	$1.043 \times 10^5 \text{ kg/m}^3$
T_0	280	280 K
Pr	0.025	
α_l	0.07695	$7.008 \text{ mm}^2/\text{s}$
λ_l	1	10.76 W/mK
$C_{p,l}$	13.00	146 J/kgK
L	3427	$3.85 \times 10^4 \text{ J/kg}$

Table 5.3: Input parameters for the turbulent constant temperature simulation. The lattice quantities are calculated via the methods described in sections 5.2.1 and 4.3.1. References to the physical quantities are found in appendix A.

eliminate inlet effects. The input parameters were calculated using the methods of section 4.3.1 and 5.2.1. The flow was run for $t^+ \approx 17$ to let the simulation settle. The size of this interval was determined from preliminary simulations. The scalar fields were saved every $t^+ \approx 0.85$ for a total of 100 saves. The input parameters are shown in table 5.3

Results

The deviation of the mean temperature field from T_0 along the channel height is shown in figure 5.4a. Because the initial temperature was set to T_0 and wall temperatures are kept constant at T_0 , any deviation from T_0 is unexpected. It can be seen that for both the OTRT and the FM model the mean temperature shows an overshoot in the channel center. For the FM operator this increase has a maximum of $0.00578K$, while the increase for the OTRT operator is $0.0249K$. In the case where temperature differences on the order of $1K$ are introduced, these unphysical overshoots are not expected to have a large influence on the mean temperature profile. This will be further demonstrated in the next section.

The RMS temperature fluctuations of both models are shown in figure 5.4b. Because no temperature differences were introduced, no temperature fluctuations are expected. Thus the observed fluctuations are unphysical. For both operators the fluctuations are highest near the wall and have a minimum in the channel center. The shape qualitatively matches the shape of the RMS fluctuations of the velocity fields. For the FM operator the maximum is $0.0656K$, while the maximum is $0.297K$ for the OTRT operator. In the case where temperature differences on the order of $1K$ are introduced, these fluctuations can become quite significant, as will become apparent in the next section.

Both operators show deviations from the expected behaviour, where the deviation of the OTRT operator is roughly 5 times as large as for the FM operator. Because the periodic boundary conditions ensure the elimination of inlet effects and the deviation from T_0 at the wall goes to zero it seems these fluctuations are a direct consequence of the collision operators. For both the mean temperature and the RMS temperature fluctuations the deviations resemble the shape of the corresponding velocity field. This indicates the velocity field influences the temperature field to a higher degree than expected. More research is needed to pinpoint the exact nature of this interaction. Another thing to note is the slight asymmetry of the mean temperature profile in the channel center, visible for both operators. Asymmetries like this can be an indication that race conditions arose during the calculations. Careful deconstruction of the algorithm and non-parallel tests on a CPU should be conducted to completely eliminate any suspicion of race conditions.

The next section discusses the results of simulations performed of asymmetrical heating in turbulent channel flow. Results will be compared to a benchmark case from literature. The effect of the observed, unphysical fluctuations on the expected temperature profiles will be discussed.

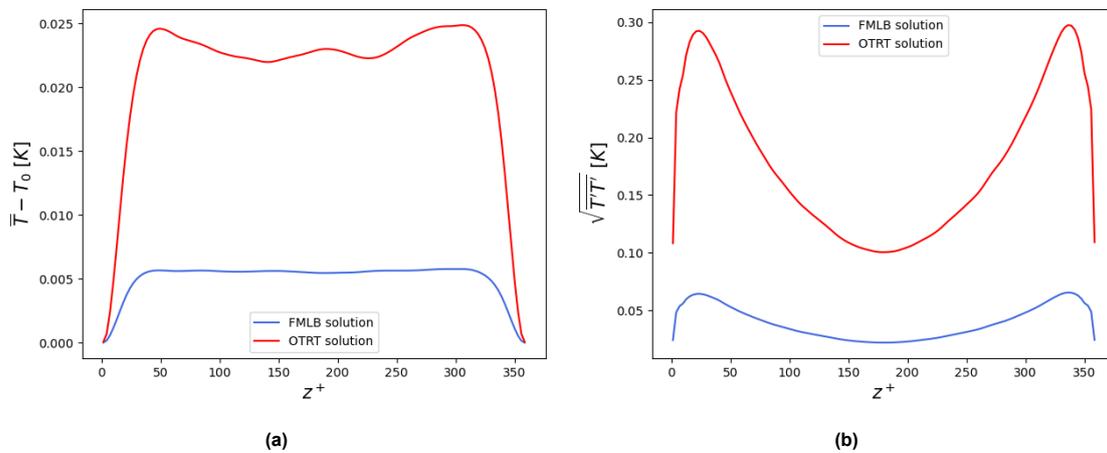


Figure 5.4: Temperature profiles generated by the simulations run with constant and equal wall temperature T_0 . (a) shows the difference between the wall temperature and the mean temperature and (b) shows the RMS temperature fluctuations.

Quantity	Value in Lattice Units	Value in Physical Units
$N_x \times N_y \times N_z$	$256 \times 128 \times 128$	$20 \times 10 \times 10 \text{ cm}$
\mathcal{H}	64	5 cm
Re_τ	180	
U_{max}	0.1	11.66 mm/s
ν	1.923×10^{-3}	$0.1752 \text{ mm}^2/\text{s}$
g_x	4.5741×10^{-7}	$7.956 \times 10^{-3} \text{ mm}/\text{s}^2$
ρ_0	1	$1.043 \times 10^5 \text{ kg}/\text{m}^3$
T_0	280	280 K
T_{wall}	282	282 K
Pr	0.025	
α_l	0.07695	$7.008 \text{ mm}^2/\text{s}$
λ_l	1	$10.76 \text{ W}/\text{mK}$
$C_{p,l}$	13.00	$146 \text{ J}/\text{kgK}$
L	3427	$3.85 \times 10^4 \text{ J}/\text{kg}$

Table 5.4: Input parameters for the single sided heating in turbulent channel flow simulations.. The lattice quantities are calculated via the methods described in sections 5.2.1 and 4.3.1. References to the physical quantities are found in appendix A.

5.3.2. Single Sided Heating

The thermal flow model in turbulent flow is further benchmarked in the case of single sided heating. The results were compared to the results obtained by Kawamura, Abe and Shingai (KAS) [131], who performed a DNS of single sided heating of a fluid with $Pr = 0.025$ in a $Re_\tau = 180$ channel flow with a constant wall temperature.

Initialisation

The flow field was again initialized as a fully developed turbulent channel flow at $Re_\tau = 180$, taken from the saved final distribution function of the flow benchmark. Periodic in- and outlet conditions were implemented for both distributions. After a period of $t^+ \approx 17$ the bottom wall temperature is set to $T_{wall} = 282K$. The top wall is kept constant at $T_0 = 280K$. The flow is left to develop for another $t^+ \approx 17$ after which the scalar fields are saved every $t^+ \approx 0.85$ for a total of 80 saves. The input parameters are shown in table 5.4.

Results

The results for the mean temperature are shown in figure 5.5a. A reasonable agreement can be seen for both operators. Note that on linear axes, these profiles would give approximately straight lines. The FM operator shows an overshoot of the mean temperature in the region very close to the wall, after which it shows a slight undershoot in the rest of the channel. Overall an error of $\mathcal{E} = 0.078$ is achieved

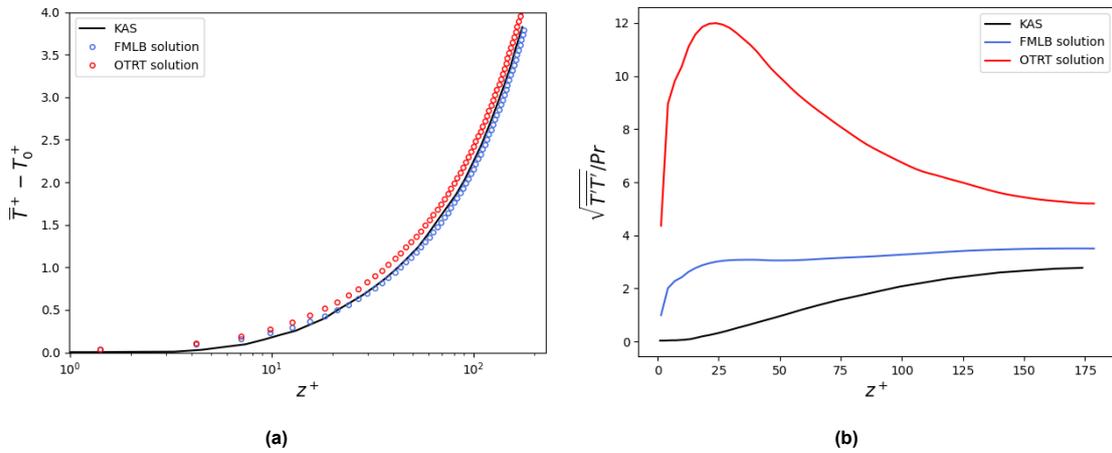


Figure 5.5: The mean temperature (a) and the RMS temperature fluctuations (b) as function of channel height for the OTRT and the thermal FM models compared to the results of [131] the case of single sided heating.

by the FM operator. The OTRT operator shows an overshoot along in the entire channel. An error of $\mathcal{E} = 0.17$ was found for the OTRT operator. The influence of the unphysical fluctuations presented in the previous section is visible here, leading to a larger error for the OTRT operator.

The influence of these extra, unphysical fluctuations becomes more pronounced when analysing the RMS temperature fluctuations (figure 5.5b). Here both operators show a significant over estimations, where the shape of the unphysical fluctuations is clearly visible in the graphs for both operators. The FM operator shows an error of $\mathcal{E} = 1.64$ and the OTRT $\mathcal{E} = 6.89$. Both of these errors are on the same order of magnitude as the RMS fluctuations themselves.

To better see the influence of the erroneous fluctuations, a correction was applied. The unphysical fluctuations were subtracted from the RMS temperature fluctuations produced by the single sided heating simulations

$$\left(\frac{\sqrt{T'T'}}{Pr}\right)_{correction} = \left(\frac{\sqrt{T'T'}}{Pr}\right) - \left(\frac{\sqrt{T'T'}}{Pr}\right)_{unphysical} \quad (5.10)$$

These results are shown in figure 5.6. A large improvement with respect to the graph in figure 5.5b can be seen for the FM operator. In the region near the wall the FM model still overestimates the RMS temperature fluctuations. Nearing the center of the channel the fluctuations are now slightly underestimated. Qualitatively the FM operator now shows reasonable resemblance to the benchmark results. The error is reduced to $\mathcal{E} = 0.13$. The result of the OTRT operator still shows significant deviation from the benchmark result both qualitatively and quantitatively. An error of $\mathcal{E} = 1.21$ was found.

5.3.3. Thermal Benchmark in Turbulent Flow Conclusion

From the thermal benchmark results it can be concluded that both operators work reasonably when calculating mean quantities and satisfactory results can be achieved. The unphysical fluctuations produced by the FM operator seem to have less effect on the solution and by correcting for them, a reasonable agreement of the RMS temperature fluctuations to KAS is achieved. The unphysical fluctuations produced by the OTRT operator have a bigger influence on the physical behaviour produced by the operator and even after correction the RMS fluctuations are inaccurate. To accurately simulate the complex interactions of transient freezing in turbulent flows, further research is necessary to eliminate the unphysical fluctuations from the models. However, preliminary calculations of heat transfer correlations can be performed based on average flow fields. The next section covers such a calculation for lead-bismuth eutectic flowing turbulently in an asymmetrically heated channel.

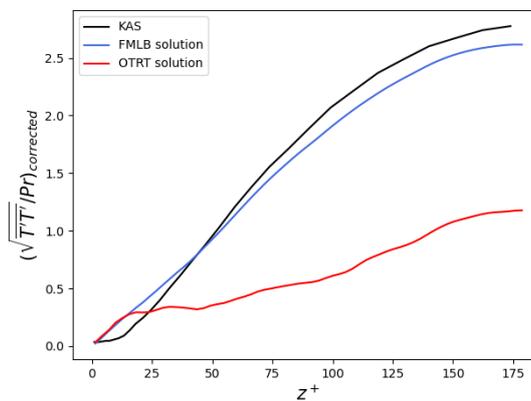


Figure 5.6: The corrected RMS temperature fluctuations for single sided heating. The RMS fluctuations obtained from the simulation at constant temperature (section 5.3.1) were subtracted from the RMS fluctuations found for the case of single sided heating.

5.4. Heat Transfer Correlation

Now that the benchmark cases have been discussed and the performance of the model for thermal flows in turbulent regime has been analysed, the results of a preliminary heat transfer correlation calculation for single sided heating of lead-bismuth eutectic with constant wall temperature is discussed. Three simulations were performed for each collision operator, with $Re_\tau = [180, 288, 395]$.

Heat transfer correlations for liquid metals are generally of the form [54]

$$Nu = a + bPe^c, \quad (5.11)$$

where $Pe = RePr$ is the Peclet number, and the average Nu is implied. As noted by Pacio et al. [132], very few reliable experimental data are available for heating of liquid metals under constant wall temperature boundary conditions. In their 2015 review on pipeflows they suggested the correlation presented by Seban and Shimazaki [133] as an upper limit below which all experimental data they found, lay

$$Nu = 5.0 + 0.025Pe^{0.8}. \quad (5.12)$$

This correlation was later also obtained by Subbotin et al. [134], who conducted experiments with liquid sodium in pipes for the range $40 \leq Pe \leq 1150$. This relation is used to compare the results of the current simulations.

\overline{Nu} is calculated from the mean temperature field similarly as for the laminar flow case

$$\overline{Nu} = \frac{D_h}{T_{wall} - T_{mc}} \left. \frac{\partial \overline{T}(z)}{\partial z} \right|_{z=0}. \quad (5.13)$$

The mix-cup temperature and the derivative at the wall are calculated using the same procedure as described in section 4.3, i.e. equation (4.17) and (4.18) respectively. The mean temperature field was used, which showed little influence from the unphysical fluctuations. Therefore no corrections were applied. The obtained \overline{Nu} values were fitted to the form of equation (5.11) in MATLAB using the non-linear least squares method, to obtain a heat transfer correlation for LBE flowing turbulently in an asymmetrically heated channel under a constant wall temperature boundary condition. The overline on \overline{Nu} will be omitted for the remainder of this chapter.

5.4.1. Initialisation

The flows are initialised at constant temperature $T_0 = 500K$ with a momentum flow profile saved from earlier non-thermal simulations. The flow parameters are derived using the methods discussed in section 4.3.1 and 5.2.1. Periodic boundary conditions were implemented for inlet and outlet of both distributions. The flow was again left to settle for $t^+ \approx 17$, after which the bottom wall was set to $T_{wall} = 510K$. The top wall was kept constant at T_0 . After the initialisation interval, the scalar fields are saved every $t^+ \approx 0.85$ for a total of 80 saves. The input parameters are shown in table 5.5.

5.4.2. Results

The results of the simulations are shown in figure 5.7, where Nu is plotted against Pe . The correlation presented by Subbotin is plotted as the gray dotted line. The heat transfer correlations calculated by the current models are shown in the colored dashed lines. The correlation fitted to the data from the FM operator is given by

$$Nu = 4.685 + 1.091 \times 10^{-9} Pe^{3.571}. \quad (5.14)$$

The obtained Nu values from the FM operator have an error of $\mathcal{E} = 0.03046$ compared to the fit. The data from the OTRT simulations did not allow a good fit with the form of equation (5.11). The correlation fitted to the data from the OTRT operator is given by

$$Nu = 4.45 + 1.142 \times 10^{-10} Pe^{3.976}, \quad (5.15)$$

where an error of $\mathcal{E} = 0.4584$ of the data compared to the fit is found.

It is readily observed that the correlations obtained from the present models differ from the correlation presented by Subbotin et al. Compared to the correlation by Subbotin et al. for pipeflow (5.12), the present results give Nu values which are 15 – 30% lower. Moreover, the correlations from the current

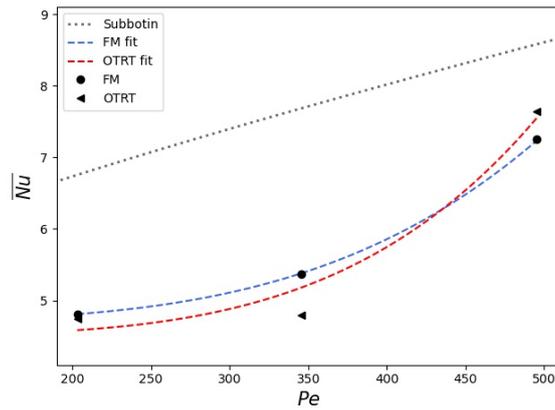


Figure 5.7: Averaged Nusselt number calculated for single sided heating of LBE in turbulent channel flow. The colored dashed lines show the present correlations. The gray dotted line shows the correlation by Subbotin et al. [134].

LB models show a much stronger dependence on Pe , which is expressed in the correlations as a larger exponent on Pe .

The fact that the current results are lower compared to (5.12), can be attributed to the difference between the pipe and channel geometry. Pipe flows exhibit more intense interactions between turbulence structures. This leads to larger turbulence intensities in the wall-normal and spanwise directions and a larger mean streamwise velocity for pipe flows compared to channel flows [135]. Thus pipe flow exhibits a stronger convective transport of heat away from the wall and thus higher Nu . This has also been reported for the case of constant wall heat flux boundary conditions. Seban [136] proposed an analytical expression for single sided heating of liquid metals in turbulent flow between parallel plates. As stated by Lyon in [137] this expression gives values for Nu which are $\sim 20\%$ lower compared to values for turbulent pipe flow. This roughly coincides with the results of the present simulations compared to (5.12).

The stronger dependence on Pe of the present results can not be readily explained. Correlations proposed for turbulent channel flow with constant heat flux typically show the same dependence on Pe compared to pipe flows, i.e. $Nu = f(Pe^{0.8})$. It is possible that trends shown by the current results, i.e. $Nu = f(Pe^{3.571})$ for FM and $Nu = f(Pe^{3.976})$ are a consequence of the unphysical fluctuations discussed in the previous sections. As these have only been analysed for $Re_\tau = 180$, further research is needed to reveal the magnitude and the effects of these fluctuations in higher Re flows.

Quantity	Low Turbulence		Medium Turbulence		High Turbulence	
	Lattice Units	Physical Units	Lattice Units	Physical Units	Lattice Units	Physical Units
$N_x \times N_y \times N_z$	$256 \times 128 \times 128$	$20 \times 10 \times 10 \text{ cm}$	$384 \times 182 \times 182$	$20 \times 10 \times 10 \text{ cm}$	$480 \times 232 \times 232$	$20 \times 10 \times 10 \text{ cm}$
\mathcal{H}	64	5 cm	91	5 cm	116	5 cm
Re_τ	180		288		396	
U_{max}	0.1	14.23 mm/s	0.1	24.22 mm/s	0.1	34.86 mm/s
ν	1.923×10^{-3}	$0.2139 \text{ mm}^2/\text{s}$	1.607×10^{-3}	$0.2139 \text{ mm}^2/\text{s}$	1.424×10^{-3}	$0.2139 \text{ mm}^2/\text{s}$
g_x	4.574×10^{-7}	$0.01186 \text{ mm}/\text{s}^2$	2.844×10^{-7}	$0.03036 \text{ mm}/\text{s}^2$	2.089×10^{-7}	$0.05889 \text{ mm}/\text{s}^2$
ρ_0	1	$1.043 \times 10^4 \text{ kg}/\text{m}^3$	1	$1.043 \times 10^4 \text{ kg}/\text{m}^3$	1	$1.043 \times 10^4 \text{ kg}/\text{m}^3$
T_0	500	510 K	500	510 K	500	510 K
T_{wall}	510	510 K	510	510 K	510	510 K
α_l	0.06302	$7.008 \text{ mm}^2/\text{s}$	0.05266	$7.008 \text{ mm}^2/\text{s}$	0.04664	$7.008 \text{ mm}^2/\text{s}$
Pr	0.03054		0.03054		0.03054	
λ_l	1	$10.76 \text{ W}/\text{mK}$	1	$10.76 \text{ W}/\text{mK}$	1	$10.76 \text{ W}/\text{mK}$
$C_{p,l}$	15.87	$146 \text{ J}/\text{kgK}$	18.99	$146 \text{ J}/\text{kgK}$	21.44	$146 \text{ J}/\text{kgK}$
L	4184	$3.85 \times 10^4 \text{ J}/\text{kg}$	5008	$3.85 \times 10^4 \text{ J}/\text{kg}$	5654	$3.85 \times 10^4 \text{ J}/\text{kg}$

Table 5.5: Input parameters of the LBE heating simulations. The lattice quantities are calculated via the methods described in sections 5.2.1 and 4.3.1. References to the physical quantities are found in appendix A.

6

Conclusion

Generation IV nuclear reactors are a vital tool for meeting the future demand of renewable energy. For the functioning of these systems, heat exchangers are a key component. Heat transfer correlations are a widely employed tool in safety analysis and reactor design. Experimental and numerical research to find these correlations for different geometries and working fluids remains an active area of research. Due to the properties the coolants and possible conditions within the heat exchangers, turbulent flows and freezing of coolant are feasible. Thus accurate heat transfer correlations covering both turbulence and freezing are needed. The present work aims to contribute to the current state of knowledge on these topics by developing a GPU accelerated double distribution function Lattice Boltzmann based model capable of performing direct numerical simulations of turbulent channel flows incorporating transient freezing. Two collision operators for the total enthalpy distribution function were implemented and tested in laminar and turbulent flow regimes. This chapter covers the conclusions drawn from the conducted research and provides recommendations for further work.

6.1. Conclusion

GPU-DDF-LB Model Implementation

A GPU-DDF-LB model was implemented in Python using the Numba CUDA library. To increase the efficiency several adjustments were made to the standard CPU-based LB scheme:

- The data structure for the distribution functions and scalar fields were changed to one dimensional arrays.
- Two data structures were implemented for each distribution function to minimize communication between the CPU and GPU.
- A pull-in stream method was used to minimize non-local data writing in GPU memory.
- Core occupancy was maximized through adequate grid and block sizing.

It was found that the following boundary conditions produced the best results:

- Periodic boundary conditions on the hydrodynamic inlet and outlet.
- The combination of the inlet condition by Liu et al. [109] and Neumann outlet condition or periodic conditions for the thermal inlet and outlet.
- The half-way bounce back method for the no-slip condition at the solid walls.
- The anti bounce back method for constant temperature at the solid walls.

The Filter Matrix operator with a D3Q19 velocity scheme was implemented for the density distribution function. The Optimal-Two-Relaxation-Time operator and the Filter Matrix operator were implemented with a D3Q7 velocity scheme for the total enthalpy distribution function. The Immersed Boundary Method was used to achieve zero velocity within the solidified region.

The models ran on an NVIDIA Tesla V100S GPU and achieved speeds of 290 MLUPS and 280 MLUPS for the OTRT and FM implementations respectively. An efficiency of 15% was achieved compared to theoretical maximum performance. These speeds allowed the model to run turbulent simulations within a time frame of several hours up to several days and were thus sufficient for the present

work. However, the low efficiency indicates that much improvement is still possible. An increase of simulation speed of 300% was found compared to a FM-LB model from literature [78] optimized to run in parallel on a 16-core CPU .

Laminar Flow

In the laminar flow regime the models showed accurate results compared to analytical solutions. The models were tested for single sided heating of water in fully developed laminar channel flow. The well-known Poiseuille flow profile was achieved with a very small error and approximate analytical expressions for Nu as a function Re and Pr were achieved with small errors by both collision operators for total enthalpy.

The models were also tested for double sided transient freezing of water in laminar channel flow. The FM operator produced a stable ice layer that matched analytical solution reasonably well. The OTRT operator produced a more ragged ice layer and unexpected profile. The error with respect to the analytical solution was significant. Thus it was concluded the FM operator better captures the transient freezing behaviour of water in a laminar channel flow compared to the OTRT operator.

Stability tests showed that the accuracy and stability of the thermal models deteriorates for higher values of Re and Pr . Due to the numerical constraints on the velocity and size of the domain, raising Re and Pr entails lowering the viscosity and thermal diffusivity. This generally leads to loss of stability in LBM simulations. Lowering the maximum lattice velocity has a positive effect on stability, but requires the increase of simulation time. It was found that the current models were only able to simulate thermal flows nearing the turbulent regime within a reasonable time frame for Pr on the order of liquid metals.

Turbulent Flow

The benchmarks in the turbulent flow regime showed the GPU-FMLB model for momentum flow was able to reproduce the first order turbulent statistics with reasonable accuracy. Quantities deviated slightly from results by Kim et al. [129] [130] in the near wall region, which can be attributed to a lack of resolution.

The thermal models were found to produce unphysical temperature fluctuations in the turbulent flow regime. Simulations of turbulent flows in an infinite channel at constant temperature showed that mean temperature and root-mean-squared temperature fluctuations qualitatively match the shape of the corresponding velocity fields, where no fluctuations are expected. The fluctuations were approximately 5 times larger for the OTRT operator compared to the FM operator. In the case of single sided heating at low Pr , the unphysical fluctuations were found to have only a small effect on the mean temperature profile, which matched results from literature reasonably well. The RMS temperature fluctuations however were significantly influenced and results from the present simulations did not match results from literature. Thus it was concluded the current models only produce accurate results for mean temperature profiles, and the FM model performs better in terms of accuracy and stability compared to the OTRT operator for simulating turbulent thermal flows.

A preliminary calculation of heat transfer correlations for heating of lead-bismuth eutectic without freezing was conducted and expressions for Nu as a function of Pe were proposed for both collision operators. The FM operator was able to be fitted to the expected form with small error. The OTRT operator did not follow the expected form and only a fit with a large error could be produced. The obtained correlations lead to lower Nu values and a larger dependence on Pe compared to a correlation from literature for pipe flow. The first difference can be attributed to the difference in geometry. Similar results have been reported in literature. The second difference is not readily explained and differs from results from literature. The effect of the unphysical fluctuations present in the current simulations should be examined in further detail to make definitive statements on this matter. Due to the inaccuracies found in the simulation results, simulations of freezing in turbulent channel flow were omitted in the present work.

6.2. Recommendations

Due to found inaccuracies and stability limits, the current GPU-DDF-LB models were not able to simulate transient freezing in turbulent channel flows. The current models are only able to simulate thermal turbulent flows of fluids with very low Pr , i.e. liquid metals. For higher Pr values simulation results diverge or large unphysical fluctuations render the results meaningless. Further research is necessary to improve the models ability to simulate transient freezing in turbulent flows for both liquid metals and

molten salts. Several possible improvements to the current models are presented here, with the goal of increasing the computational efficiency of the models and the accuracy of the results.

Computational efficiency

Tests indicate lowering the maximum lattice velocity improves stability. More research is needed to assess the influence of lattice velocity on the stability of the simulations and the accuracy of the results. Because this necessitates the increase of total simulation time, future work will benefit from increased computational efficiency. To this end the following recommendations are made:

- **Shared Memory** - The implementation of shared memory in the streaming and collision step is expected to lead to a significant increase in computational speed.
- **Structure of Arrays** - By changing the data structure of the distribution functions to a separate array for each population, the memory coalescence can be improved, leading to increase in computational efficiency.
- **Combining Streaming & Collision** - By changing the order of streaming and collision and combining the two processes into a single kernel, the number of calls to global GPU memory is decreased and computational speed is increased. Care should be taken to avoid race conditions.
- **Local Grid Refinement** - Implementing local grid refinement near the walls will improve the flow results and lower the memory requirements of the model, thereby increasing simulation speed.

Further Improvements

The results obtained from the OTRT model indicate this operator is not suited for simulating turbulent thermal flows. Although the FM operator performs better in terms of accuracy and stability, further improvements to the thermal scheme are needed. To this end, the following possible improvements are proposed:

- **Large Eddy Simulation** - By reducing the resolution and implementing an appropriate subgrid model, the stability and accuracy of the present work could be improved, while also reducing computational cost by reducing the required resolution. Subgrid models are readily implemented in LBM [138]. Examples from literature include an FM-based LES-LB model for non-thermal flow [78] and several MRT-based LES-LB models for thermal turbulent flows [139] [140]. An FM-based thermal LES-LB model has to the authors knowledge not yet been reported and could prove an improvement to the current models.
- **Multiple Relaxation Time Operator** - A variety of thermal MRT-based LB models have been reported in literature, e.g. [141] [142] [143]. Although implementation of the MRT operator is typically not straightforward, it could improve the accuracy of the current thermal model.
- **Alternative Velocity Sets** - Increasing the number of populations at each lattice node can increase the accuracy and stability of the models. In their review on LBM applied to nuclear reactors [138] Cifuentes et al. mention several works illustrating improved accuracy of the D3Q27 set over the D3Q19 set for momentum flow and in [139] an LES-LB model with a D3Q19 temperature scheme is reported with satisfactory results. These schemes significantly increase the computational cost per iteration compared to the D3Q7 scheme for thermal flow.

References

- [1] *Energy Production and Consumption*. <https://ourworldindata.org/energy-production-consumption>. Accessed: 2022-05-22.
- [2] V Masson-Delmotte et al. *Summary for Policymakers. In: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by T K Maycock et al. Cambridge, United Kingdom; New York, NY, USA: Cambridge University Press, pp. 3–32.
- [3] *9 Notable Facts about the World's First Nuclear Power Plant - EBR-I*. <https://www.energy.gov/ne/articles/9-notable-facts-about-worlds-first-nuclear-power-plant-ebr-i>. Accessed: 2022-05-25.
- [4] *Generation IV International Forum*. Accessed: 2023-06-22. URL: https://www.gen-4.org/gif/jcms/c_9260/public.
- [5] Andrea Di Ronco, Antonio Cammi, and Stefano Lorenzi. "Preliminary analysis and design of the heat exchangers for the Molten Salt Fast Reactor". In: *Nuclear Engineering and Technology* 52.1 (2020), pp. 51–58.
- [6] Martin David, Adrien Toutant, and Françoise Bataille. "Numerical development of heat transfer correlation in asymmetrically heated turbulent channel flow". In: *International Journal of Heat and Mass Transfer* 164 (2021), p. 120599.
- [7] *Generation IV International Forum - Lead-cooled Fast Reactor*. Accessed: 2023-06-22. URL: https://www.gen-4.org/gif/jcms/c_42149/lead-cooled-fast-reactor-lfr.
- [8] *Generation IV International Forum - Molten Salt Reactor*. Accessed: 2023-06-22. URL: https://www.gen-4.org/gif/jcms/c_42150/molten-salt-reactor-msr.
- [9] *Generation IV International Forum - Sodium-cooled Fast Reactor*. Accessed: 2023-07-10. URL: https://www.gen-4.org/gif/jcms/c_9361/sfr.
- [10] Alessandro Alemberti et al. "Overview of lead-cooled fast reactor activities". In: *Progress in Nuclear Energy* 77 (2014), pp. 300–307.
- [11] Alan E Waltar, Donald R Todd, and Pavel V Tsvetkov. *Fast spectrum reactors*. Springer, 2011.
- [12] Yu G Dragunov et al. "BREST-OD-300 reactor facility: development stages and justification". In: (2017).
- [13] Alessandro Alemberti. "ELFR: The European Lead Fast Reactor. Design, Safety Approach and Safety Characteristics". In: (2012).
- [14] Craig F Smith et al. "SSTAR: The US lead-cooled fast reactor (LFR)". In: *Journal of Nuclear Materials* 376.3 (2008), pp. 255–259.
- [15] E. Merle. *Advanced Nuclear Power Reactors*. https://www.gen-4.org/gif/upload/docs/application/pdf/2017-05/07_elsa_merle_france.pdf. Accessed: 2022-05-25.
- [16] *Time Warp: Molten salt reactor experiment-Alvin Weinberg's magnum opus: ORNL*. Accessed: 2023-06-22. June 2016. URL: <https://www.ornl.gov/blog/ornl-review/time-warp-molten-salt-reactor-experiment-alvin-weinberg-s-magnum-opus>.
- [17] *Molten Salt Reactors - World Nuclear Association*. Accessed: 2023-06-22. URL: <https://www.world-nuclear.org/information-library/current-and-future-generation/molten-salt-reactors.aspx>.
- [18] *SAMOSAFER*. accessed: 2023-06-22. Dec. 2022. URL: <https://samosafer.eu/>.
- [19] *Thorizon*. Accessed: 2022-05-25. URL: <https://www.thorizon.com/>.
- [20] *Copenhagen Atomics*. Accessed: 2022-05-25. URL: <https://www.copenhagenatomics.com/>.

- [21] Xiaohan Yu and Hongjie Xu. "Update on SINAP TMSR research". In: *MSR Workshop, ORNL*. 2016.
- [22] World Nuclear News. *Operating permit issued for Chinese molten salt reactor*. Accessed: 2022-05-25. URL: <https://world-nuclear-news.org/Articles/Operating-permit-issued-for-Chinese-molten-salt-re>.
- [23] RR Matthews et al. *Performance and operation of the Dounreay fast reactor*. Tech. rep. United Kingdom Atomic Energy Authority, Caithness, Scotland (United Kingdom ..., 1964.
- [24] SE Jensen and PL Ølgaard. "Description of the prototype fast reactor at Dounreay". In: (1995).
- [25] Kazumi Aoto et al. "A summary of sodium-cooled fast reactor development". In: *Progress in Nuclear Energy* 77 (2014), pp. 247–265.
- [26] Jaewoon Yoo et al. "Status of Sodium Cooled Fast Reactor Development Program in Korea". In: (2017).
- [27] Hiroki Hayafune et al. "Advanced sodium-cooled fast reactor development regarding GIF safety design criteria". In: (2017).
- [28] Hidemasa Yamano et al. "Activities of the GIF Safety and Operation project of sodium-cooled fast reactor systems". In: *The Proceedings of the International Conference on Nuclear Engineering (ICONE) 2019.27*. The Japan Society of Mechanical Engineers. 2019, p. 1025.
- [29] Hiroyuki Ohshima and Shigenobu Kubo. "Sodium-cooled Fast Reactors (SFRs)". In: *Handbook of Generation IV Nuclear Reactors*. Elsevier, 2023, pp. 173–194.
- [30] Brandon M Chisholm, Steven L Krahn, and Andrew G Sowder. "A unique molten salt reactor feature—The freeze valve system: Design, operating experience, and reliability". In: *Nuclear Engineering and Design* 368 (2020), p. 110803.
- [31] N Le Brun, GF Hewitt, and CN Markides. "Transient freezing of molten salts in pipe-flow systems: Application to the direct reactor auxiliary cooling system (DRACS)". In: *Applied Energy* 186 (2017), pp. 56–67.
- [32] Hong Liu et al. "Numerical study on the heat transfer characteristics of a liquid lead–bismuth eutectic in a D-type channel". In: *Frontiers in Energy Research* 10 (2022), p. 1041900.
- [33] Allan P Colburn. "A method of correlating forced convection heat-transfer data and a comparison with fluid friction". In: *International Journal of Heat and Mass Transfer* 7.12 (1964), pp. 1359–1384.
- [34] Ramesh K Shah and Alexander Louis London. *Laminar flow forced convection in ducts: a source book for compact heat exchanger analytical data*. Academic press, 1978.
- [35] E Nn Sieder and Go E Tate. "Heat transfer and pressure drop of liquids in tubes". In: *Industrial & Engineering Chemistry* 28.12 (1936), pp. 1429–1435.
- [36] Karl Stephan. "Wärmeübergang und Druckabfall bei nicht ausgebildeter Laminarströmung in Rohren und in ebenen Spalten". In: *Chemie Ingenieur Technik* 31.12 (1959), pp. 773–778.
- [37] Karl Stephan and K Nesselmann. "Gleichungen für den Wärmeübergang laminar strömender Stoffe in ringförmigen Querschnitten". In: *Chemie Ingenieur Technik* 33.5 (1961), pp. 338–343.
- [38] W Roetzel, X Luo, and D Chen. "Basic thermal design theory for heat exchangers". In: *Design and Operation of Heat Exchangers and Their Networks* (2020), pp. 13–69.
- [39] Frederick William Dittus. "Heat transfer in automobile radiators of the tubular type". In: *Univ. of California Pub., Eng.* 2.13 (1930), pp. 443–461.
- [40] JF Barnes and JD Jackson. "Heat transfer to air, carbon dioxide and helium flowing through smooth circular tubes under conditions of large surface/gas temperature ratio". In: *Journal of Mechanical Engineering Science* 3.4 (1961), pp. 303–314.
- [41] Lyle D Dailey, Ning Meng, and Richard H Pletcher. "Large eddy simulation of constant heat flux turbulent channel flow with property variations: quasi-developed model and mean flow results". In: *J. Heat Transfer* 125.1 (2003), pp. 27–38.

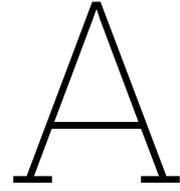
- [42] Volker Gnielinski. "Ein neues Berechnungsverfahren für die Wärmeübertragung im Übergangsbereich zwischen laminarer und turbulenter Rohrströmung". In: *Forschung im Ingenieurwesen* 61 (1995), pp. 240–248.
- [43] Volker Gnielinski. "G1 heat transfer in pipe flow". In: *VDI heat atlas*. Springer, 2010, pp. 691–700.
- [44] Volker Gnielinski. "New equations for heat and mass transfer in turbulent pipe and channel flow". In: *Int. Chem. Eng.* 16.2 (1976), pp. 359–368.
- [45] Dawid Taler. "Simple power-type heat transfer correlations for turbulent pipe flow in tubes". In: *Journal of Thermal Science* 26 (2017), pp. 339–348.
- [46] E Battista and HC Perkins. "Turbulent heat and momentum transfer in a square duct with moderate property variations". In: *International Journal of Heat and Mass Transfer* 13.6 (1970), pp. 1063–1065.
- [47] Alberto Ghione et al. "Assessment of thermal–hydraulic correlations for narrow rectangular channels with high heat flux and coolant velocity". In: *International Journal of Heat and Mass Transfer* 99 (2016), pp. 344–356.
- [48] Daeseong Jo et al. "Experimental investigation of convective heat transfer in a narrow rectangular channel for upward and downward flows". In: *Nuclear Engineering and Technology* 46.2 (2014), pp. 195–206.
- [49] Jian Ma et al. "Experimental studies on single-phase flow and heat transfer in a narrow rectangular channel". In: *Nuclear Engineering and Design* 241.8 (2011), pp. 2865–2873.
- [50] VP Smirnov et al. "Thermohydraulic research for the core of the BREST-OD-300 reactor". In: (2003).
- [51] Konstantin Mikityuk. "Heat transfer to liquid metal: review of data and correlations for tube bundles". In: *Nuclear Engineering and Design* 239.4 (2009), pp. 680–687.
- [52] Zaiyong Ma et al. "An innovative method for prediction of liquid metal heat transfer rate for rod bundles based on annuli". In: *Annals of Nuclear Energy* 47 (2012), pp. 91–97.
- [53] Xu Cheng and Nam-il Tak. "Investigation on turbulent heat transfer to lead–bismuth eutectic flows in circular tubes for nuclear applications". In: *Nuclear Engineering and Design* 236.4 (2006), pp. 385–393.
- [54] Dawid Taler. "Heat transfer in turbulent tube flow of liquid metals". In: *Procedia Engineering* 157 (2016), pp. 148–157.
- [55] Wadim Jaeger. "Heat transfer to liquid metals with empirical models for turbulent forced convection in various geometries". In: *Nuclear Engineering and Design* 319 (2017), pp. 12–27.
- [56] Fangxian Wang et al. "A comprehensive review on phase change material emulsions: Fabrication, characteristics, and heat transfer performance". In: *Solar Energy Materials and Solar Cells* 191 (2019), pp. 218–234.
- [57] Takashi Morimoto and Hiroyuki Kumano. "Flow and heat transfer characteristics of phase change emulsions in a circular tube: Part 2. Turbulent flow". In: *International Journal of Heat and Mass Transfer* 117 (2018), pp. 903–911.
- [58] Pedro Galione et al. "Solid-liquid phase change with turbulent flow". In: *THMT-12. Proceedings of the Seventh International Symposium On Turbulence Heat and Mass Transfer*. Begel House Inc. 2012.
- [59] RR Gilpin, T Hirata, and KC Cheng. "Wave formation and heat transfer at an ice-water interface in the presence of a turbulent flow". In: *Journal of Fluid Mechanics* 99.3 (1980), pp. 619–640.
- [60] Ali A Shibani and MN Özisik. "Freezing of liquids in turbulent flow inside tubes". In: *The Canadian Journal of Chemical Engineering* 55.6 (1977), pp. 672–677.
- [61] Timm Krüger et al. *The lattice Boltzmann method*. Springer International Publishing, 2017.
- [62] Haihu Liu et al. "Multiphase lattice Boltzmann simulations for porous media applications: A review". In: *Computational Geosciences* 20 (2016), pp. 777–805.

- [63] Chongxun Pan, Li-Shi Luo, and Cass T Miller. "An evaluation of lattice Boltzmann schemes for porous medium flow simulation". In: *Computers & fluids* 35.8-9 (2006), pp. 898–909.
- [64] Mohsen Eshraghi, Bohumir Jelinek, and Sergio D Felicelli. "Large-scale three-dimensional simulation of dendritic solidification using lattice Boltzmann method". In: *Jom* 67 (2015), pp. 1786–1792.
- [65] Nanqiao Wang et al. "Phase-field-lattice Boltzmann method for dendritic growth with melt flow and thermosolutal convection–diffusion". In: *Computer Methods in Applied Mechanics and Engineering* 385 (2021), p. 114026.
- [66] Y Kuwata and K Suga. "Lattice Boltzmann direct numerical simulation of interface turbulence over porous and rough walls". In: *International Journal of Heat and Fluid Flow* 61 (2016), pp. 145–157.
- [67] Kurnchul Lee, Dazhi Yu, and Sharath S Girimaji. "Lattice Boltzmann DNS of decaying compressible isotropic turbulence with temperature fluctuations". In: *International Journal of Computational Fluid Dynamics* 20.6 (2006), pp. 401–413.
- [68] Saeed Jafari and Mohammad Rahnama. "Shear-improved Smagorinsky modeling of turbulent channel flow using generalized Lattice Boltzmann equation". In: *International Journal for Numerical Methods in Fluids* 67.6 (2011), pp. 700–712.
- [69] M Weickert et al. "Investigation of the LES WALE turbulence model within the lattice Boltzmann framework". In: *Computers & Mathematics with Applications* 59.7 (2010), pp. 2200–2214.
- [70] H Sajjadi et al. "Turbulent indoor airflow simulation using hybrid LES/RANS model utilizing Lattice Boltzmann method". In: *Computers & Fluids* 150 (2017), pp. 66–73.
- [71] Xiao Xue, Hua-Dong Yao, and Lars Davidson. "Synthetic turbulence generator for lattice Boltzmann method at the interface between RANS and LES". In: *Physics of Fluids* 34.5 (2022), p. 055118.
- [72] Nicolas Delbosc et al. "Optimized implementation of the Lattice Boltzmann Method on a graphics processing unit towards real-time fluid simulation". In: *Computers & Mathematics with Applications* 67.2 (2014), pp. 462–475.
- [73] Wang Xian and Aoki Takayuki. "Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster". In: *Parallel Computing* 37.9 (2011), pp. 521–535.
- [74] Xiong Aokui. "Intrinsic instability of the lattice BGK model". In: *Acta Mechanica Sinica* 18 (2002), pp. 603–607.
- [75] JH Lu, HY Lei, and CS Dai. "An optimal two-relaxation-time lattice Boltzmann equation for solid-liquid phase change: The elimination of unphysical numerical diffusion". In: *International Journal of Thermal Sciences* 135 (2019), pp. 17–29.
- [76] JA Somers. "Direct simulation of fluid flow with cellular automata and the lattice-Boltzmann equation". In: *Applied Scientific Research* 51 (1993), pp. 127–133.
- [77] JGM Eggels and JA Somers. "Numerical simulation of free convective flow using the lattice-Boltzmann scheme". In: *International Journal of Heat and Fluid Flow* 16.5 (1995), pp. 357–364.
- [78] Congshan Zhuo and Chengwen Zhong. "LES-based filter-matrix lattice Boltzmann model for simulating fully developed turbulent channel flow". In: *International Journal of Computational Fluid Dynamics* 30.7-10 (2016), pp. 543–553.
- [79] Thorben Besseling. *Investigating Adaptive Mesh Refinement Criteria for a Double-Distribution FMLB Scheme in Melting and Solidification*. Master Thesis at Technical University Delft. 2022.
- [80] Celeke Bus. *Simulating the Transient Freezing in cooled Non-eutectic Molten Salt Channel Flow*. Master Thesis at Technical University Delft. 2022.
- [81] Pijush K Kundu, Ira M Cohen, and David R Dowling. *Fluid mechanics*. Academic press, 2015.
- [82] Junuthula Narasimha Reddy. *Principles of Continuum Mechanics: Conservation and Balance Laws with Applications*. Cambridge University Press, 2017.
- [83] Jerry Westerweel, Bendiks J Boersma, and Frans TM Nieuwstadt. *Turbulence: Introduction to Theory and Applications of Turbulent Flows*. Springer International Publishing, 2016.

- [84] HEA Van den Akker and RF Mudde. *Fysische transportverschijnselen I*. Delft University Press, 1998.
- [85] Vaughan R Voller, M Cross, and NC Markatos. "An enthalpy method for convection/diffusion phase change". In: *International journal for numerical methods in engineering* 24.1 (1987), pp. 271–284.
- [86] Celeke Bus and Martin Rohde. "A Filter Matrix lattice-Boltzmann method for melting and solidification". 2023.
- [87] Bastian E Rapp. *Microfluidics: modeling, mechanics and mathematics*. Elsevier, 2022.
- [88] Jack Philip Holman. *Heat transfer*. McGraw Hill, 1986.
- [89] AA Mohamad. *Lattice boltzmann method*. Vol. 70. Springer, 2011.
- [90] Alexander S. Gillis and Colin Steele. *What is a GPU?* Dec. 2020. URL: <https://www.techtarget.com/searchvirtualdesktop/definition/GPU-graphics-processing-unit>.
- [91] Shane Cook. *CUDA programming: a developer's guide to parallel computing with GPUs*. Newnes, 2012.
- [92] TheBeard. *Cuda - threads, blocks, grids and synchronization*. May 2020. URL: <http://thebeardsage.com/cuda-threads-blocks-grids-and-synchronization/>.
- [93] Yuankai Yang and Moran Wang. "Pore-scale modeling of chloride ion diffusion in cement microstructures". In: *Cement and Concrete Composites* 85 (2018), pp. 92–104.
- [94] Yue-Hong Qian, Dominique d'Humières, and Pierre Lallemand. "Lattice BGK models for Navier-Stokes equation". In: *Europhysics letters* 17.6 (1992), p. 479.
- [95] Kevin R Tubbs and Frank T-C Tsai. "GPU accelerated lattice Boltzmann model for shallow water flow and mass transport". In: *International Journal for Numerical Methods in Engineering* 86.3 (2011), pp. 316–334.
- [96] Philipp Neumann et al. "A coupled approach for fluid dynamic problems using the PDE framework peano". In: *Communications in Computational Physics* 12.1 (2012), pp. 65–84.
- [97] Sydney Chapman, Thomas George Cowling, and D Burnett. *The Mathematical Theory of Non-uniform Gases: An Account of Kinetic Theory of Viscosity, Thermal Conduction, and Diffusion*. Cambridge University Press, 1952.
- [98] Uriel Frisch et al. *Lattice gas hydrodynamics in two and three dimensions*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States); Observatoire ..., 1986.
- [99] Frank J Alexander, Shiyi Chen, and JD Sterling. "Lattice boltzmann thermohydrodynamics". In: *Physical Review E* 47.4 (1993), R2249.
- [100] Guy R McNamara, Alejandro L Garcia, and Berni J Alder. "Stabilization of thermal lattice Boltzmann models". In: *Journal of Statistical Physics* 81 (1995), pp. 395–408.
- [101] Qing Li et al. "Lattice Boltzmann methods for multiphase flow and phase-change heat transfer". In: *Progress in Energy and Combustion Science* 52 (2016), pp. 62–105.
- [102] Congshan Zhuo, Chengwen Zhong, and Jun Cao. "Filter-matrix lattice Boltzmann model for incompressible thermal flows". In: *Physical Review E* 85.4 (2012), p. 046703.
- [103] Prabhu Lal Bhatnagar, Eugene P Gross, and Max Krook. "A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems". In: *Physical review* 94.3 (1954), p. 511.
- [104] Xiong Aokui. "Intrinsic instability of the lattice BGK model". In: *Acta Mechanica Sinica* 18 (2002), pp. 603–607.
- [105] XD Niu et al. "Investigation of stability and hydrodynamics of different lattice Boltzmann models". In: *Journal of statistical physics* 117 (2004), pp. 665–680.
- [106] James D Sterling and Shiyi Chen. "Stability analysis of lattice Boltzmann methods". In: *Journal of Computational Physics* 123.1 (1996), pp. 196–206.
- [107] Rongzong Huang, Huiying Wu, and Ping Cheng. "A new lattice Boltzmann model for solid-liquid phase change". In: *International Journal of Heat and Mass Transfer* 59 (2013), pp. 295–301.

- [108] Irina Ginzburg, Dominique d’Humières, and Alexander Kuzmin. “Optimal stability of advection-diffusion lattice Boltzmann models with two relaxation times for positive/negative equilibrium”. In: *Journal of Statistical Physics* 139 (2010), pp. 1090–1143.
- [109] Chih-Hao Liu et al. “Thermal boundary conditions for thermal lattice Boltzmann simulations”. In: *Computers & Mathematics with Applications* 59.7 (2010), pp. 2178–2193.
- [110] Qin Lou, Zhaoli Guo, and Baochang Shi. “Evaluation of outflow boundary conditions for two-phase lattice Boltzmann equation”. In: *Physical review E* 87.6 (2013), p. 063301.
- [111] Benjamin K Cook, David R Noble, and John R Williams. “A direct simulation method for particle-fluid systems”. In: *Engineering Computations* 21.2/3/4 (2004), pp. 151–168.
- [112] Yaqi Guo et al. “A GPU-accelerated 3D PF-LBM modelling of multi-dendritic growth in an under-cooled melt of Fe–C binary alloy”. In: *Journal of Materials Research and Technology* 17 (2022), pp. 2059–2072.
- [113] Aimon Rahman et al. “Natural convection and entropy generation of non-Newtonian nanofluids with different angles of external magnetic field using GPU accelerated MRT-LBM”. In: *Case Studies in Thermal Engineering* 30 (2022), p. 101769.
- [114] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. “Numba: A llvm-based python jit compiler”. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. 2015, pp. 1–6.
- [115] Lena Oden. “Lessons learned from comparing C-CUDA and Python-Numba for GPU-Computing”. In: *2020 28th Euromicro international conference on parallel, distributed and network-based processing (PDP)*. IEEE. 2020, pp. 216–223.
- [116] Gerhard Wellein et al. “On the single processor performance of simple lattice Boltzmann kernels”. In: *Computers & Fluids* 35.8-9 (2006), pp. 910–919.
- [117] Shane Ryou et al. “Optimization principles and application performance evaluation of a multi-threaded GPU using CUDA”. In: *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming*. 2008, pp. 73–82.
- [118] Jonas Tölke. “Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA”. In: *computing and Visualization in Science* 13.1 (2010), p. 29.
- [119] Qingang Xiong et al. “Efficient 3D DNS of gas–solid flows on Fermi GPGPU”. In: *Computers & Fluids* 70 (2012), pp. 86–94.
- [120] Wilfried Roetzel, Xing Luo, and Dezhen Chen. *Design and operation of heat exchangers and their networks*. Academic Press, 2019.
- [121] Arunn Narasimhan. *Concept of Bulk Temperature*. URL: <https://srirangamarunn.files.wordpress.com/2016/11/thermofluids-notes-05-bulk-temp-arunn.pdf>.
- [122] Bernhard Weigand and Hans Beer. “Transient freezing of liquids in forced laminar flow inside a parallel plate channel”. In: (1992).
- [123] Hiroshi Kawamura et al. “DNS of turbulent heat transfer in channel flow with low to medium-high Prandtl number fluid”. In: *International Journal of Heat and Fluid Flow* 19.5 (1998), pp. 482–491.
- [124] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 1)*. <https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1>. 2022.
- [125] *NVIDIA Tesla V100 GPU Architecture*. Aug. 2017. URL: <https://www.nvidia.com/en-gb/data-center/tesla-v100/>.
- [126] TOM Forslund et al. “A dual-lattice hydrodynamic-thermal MRT-LBM model implemented on GPU for DNS calculations of turbulent thermal flows”. In: *International Journal of Numerical Methods for Heat & Fluid Flow* ahead-of-print (2022).
- [127] G Amati, S Succi, and R Piva. “Preliminary analysis of the scaling exponents in channel flow turbulence”. In: *Fluid dynamics research* 24.4 (1999), pp. 201–209.

- [128] Martin Rohde et al. "A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes". In: *International journal for numerical methods in fluids* 51.4 (2006), pp. 439–468.
- [129] John Kim, Parviz Moin, and Robert Moser. "Turbulence statistics in fully developed channel flow at low Reynolds number". In: *Journal of fluid mechanics* 177 (1987), pp. 133–166.
- [130] Robert D Moser, John Kim, and Nagi N Mansour. "Direct numerical simulation of turbulent channel flow up to $Re_{\tau} = 590$ ". In: *Physics of fluids* 11.4 (1999), pp. 943–945.
- [131] Hiroshi Kawamura, Hiroyuki Abe, and Kenji Shingai. "DNS of turbulence and heat transport in a channel flow with different Reynolds and Prandtl numbers and boundary conditions". In: *Turbulence, Heat and Mass Transfer* 3 (2000), pp. 15–32.
- [132] Julio Pacio, LUCA Marocco, and Th Wetzel. "Review of data and correlations for turbulent forced convective heat transfer of liquid metals in pipes". In: *Heat and Mass Transfer* 51 (2015), pp. 153–164.
- [133] Ralph Alois Seban and Thomas Tamotsu Shimazaki. "Heat transfer to a fluid flowing turbulently in a smooth pipe with walls at constant temperature". In: *Transactions of the American Society of Mechanical Engineers* 73.6 (1951), pp. 803–807.
- [134] VI Subbotin et al. "A study of heat transfer to molten sodium in tubes". In: *Soviet Atomic Energy* 13.4 (1963), pp. 991–994.
- [135] Carmine Di Nucci and Rafik Absi. "Comparison of Mean Properties of Turbulent Pipe and Channel Flows at Low-to-Moderate Reynolds Numbers". In: *Fluids* 8.3 (2023), p. 97.
- [136] RA Seban. "Heat transfer to a fluid flowing turbulently between parallel walls with asymmetric wall temperatures". In: *Transactions of the American Society of Mechanical Engineers* 72.6 (1950), pp. 789–794.
- [137] Richard N Lyon and H Poppendiek. "Liquid-metal heat transfer". In: *Liquid-metals Handbook* (1951), p. 184.
- [138] Johan Augusto Bocanegra Cifuentes et al. "Lattice Boltzmann method applied to nuclear reactors—a systematic literature review". In: *Sustainability* 12.18 (2020), p. 7835.
- [139] Yoshiaki Kuwata and Kazuhiko Suga. "Wall-modeled large eddy simulation of turbulent heat transfer by the lattice Boltzmann method". In: *Journal of Computational Physics* 433 (2021), p. 110186.
- [140] Tong-Miin Liou and Chun-Sheng Wang. "Large eddy simulation of rotating turbulent flows and heat transfer by the lattice Boltzmann method". In: *Physics of Fluids* 30.1 (2018).
- [141] Kazuhiko Suga et al. "A D3Q27 multiple-relaxation-time lattice Boltzmann method for turbulent flows". In: *Computers & Mathematics with Applications* 69.6 (2015), pp. 518–529.
- [142] El B Lahmer et al. "Double constricted channel for laminar flow and heat transfer based on double MRT-LBM". In: *Double constricted channel for laminar flow and heat transfer based on double MRT-LBM* 16 (2019).
- [143] H Sajjadi et al. "The effect of indoor heating system location on particle deposition and convection heat transfer: DMRT-LBM". In: *Computers & Mathematics with Applications* 86 (2021), pp. 90–105.



Thermodynamic Properties

A.1. Water

Property	value		source
ρ	10^3	kg/m^3	https://www.engineeringtoolbox.com/
ν	1.434×10^{-6}	m^2/s	https://www.omnicalculator.com for water at 280K
α	1.4558×10^{-7}	m^2/s	https://www.omnicalculator.com for water at 280K
λ_l	0.607	W/mK	https://www.omnicalculator.com for water at 280K
λ_s	2.259	W/mK	https://www.engineeringtoolbox.com/ice-thermal-properties-d_576.html for water at 268.15K
$C_{p,l}$	4182	J/kgK	https://www.omnicalculator.com for water at 280K
$C_{p,s}$	2090	J/kgK	https://spacemath.gsfc.nasa.gov/weekly/92Mod11Prob2.pdf
L	334×10^3	J/kg	https://www.omnicalculator.com
$T_l = T_s = T_f$	273.15	K	https://www.engineeringtoolbox.com

A.2. Lead-bismuth Eutectic

Property	value		source
ρ	1.0434×10^4	kg/m^3	[A]
ν	2.1393×10^{-7}	m^2/s	[A]
α	7.008×10^{-6}	m^2/s	[A]
λ_l	10.75975	W/mK	[A]
λ_s	4.0	W/mK	[C]
$C_{p,l}$	146	J/kgK	[B]
$C_{p,s}$	140	J/kgK	[B]
L	38.5×10^3	J/kg	[A]
$T_l = T_s = T_f$	398.15	K	[A]

- A : Fazio, C., Sobolev, V. P., Aerts, A., Gavrilo, S., Lambrinou, K., Schuurmans, P., ... & Hwang, I. S. (2015). Handbook on lead-bismuth eutectic alloy and lead properties, materials compatibility, thermal-hydraulics and technologies-2015 edition (No. NEA-7268). Organisation for Economic Co-Operation and Development.
- B : Wang, L., Zhang, Y., Huang, R., Li, Q., Peng, T., & Hong, G. (2020). Measurement and analysis of specific heat capacity of lead-bismuth eutectic. Progress in Nuclear Energy, 123, 103284.
- C : <https://www.americanelements.com/bismuth-lead-alloy>

B

Ice Layer Development

This appendix shows the ice layer at different times for the laminar freezing simulations with the FM operator. Together with the ice height the flow profile 10 lattice nodes before the inlet is shown.

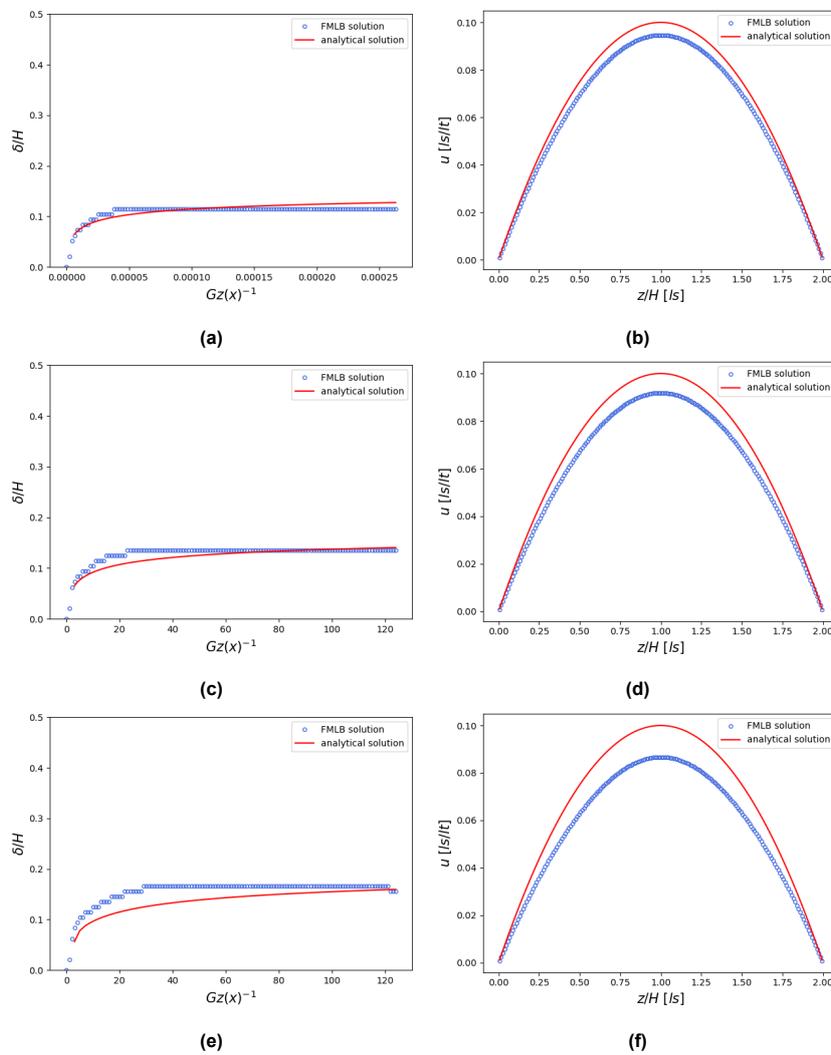


Figure B.1