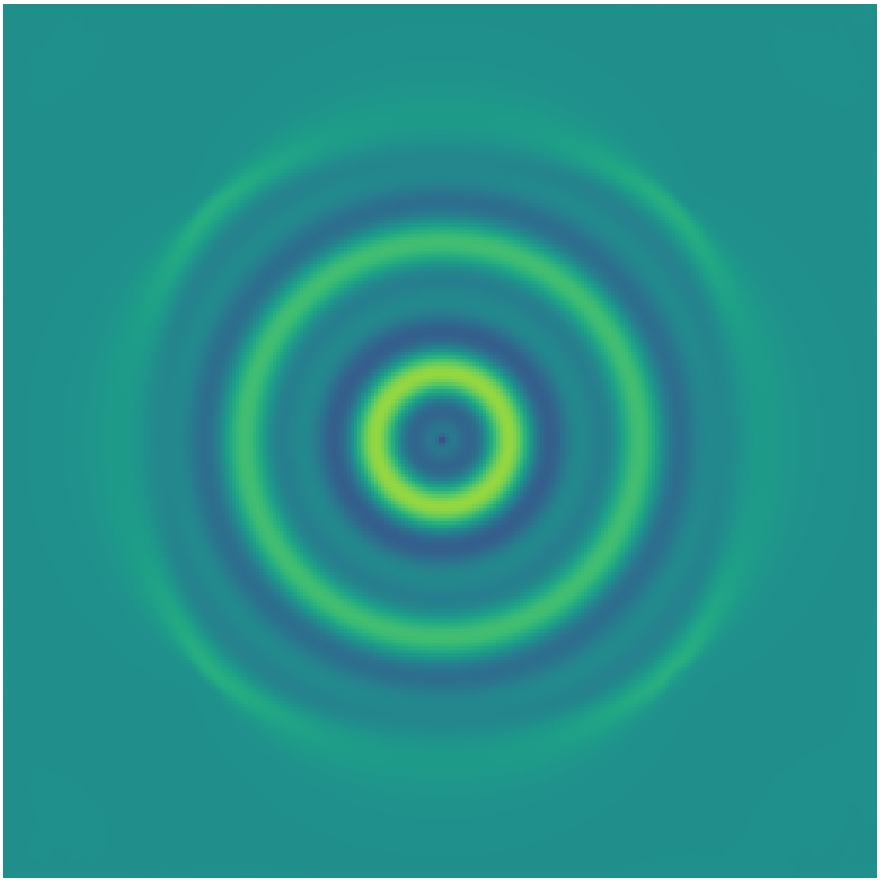

The lattice-Boltzmann method and its applications in acoustics

Developing code for applications in acoustics using the lattice-Boltzmann method

PONTUS NILSSON

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's Thesis in Applied Mechanics

The lattice-Boltzmann method and its applications in acoustics

Developing code for applications in acoustics using the lattice-Boltzmann method

PONTUS NILSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Marine Technology
Chalmers University of Technology
Gothenburg, Sweden 2025

The lattice-Boltzmann method and its applications in acoustics
Developing code for applications in acoustics using the lattice-Boltzmann method
PONTUS NILSSON

Supervisor: Xiao Xue, University College London
Examiner: Huadong Yao, Chalmers University of Technology, Department of Mechanics and Maritime Sciences

Master's Thesis 2025
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Density field for an acoustic wave with one overtone propagating from a monopole source.

Typeset in LaTeX
Gothenburg, Sweden 2025

The lattice-Boltzmann method and its applications in acoustics
Developing code for applications in acoustics using the lattice-Boltzmann method
PONTUS NILSSON
Department of Mechanics and Maritime Sciences
Division of Marine Technology
Chalmers University of Technology

Abstract

In the field of acoustics there is a need to accurately and efficiently simulate the propagation of acoustic waves. Furthermore, as sound generation is a fluid mechanical phenomena, there is a need to study how different flow cases affect sound propagation and generate sound. In order to do this there is a need for a computational fluid dynamics (CFD) method that accurately covers the relevant phenomena.

The lattice-Boltzmann method (LBM) simulates the evolution of particle distributions that can be used to compute the necessary macroscopic field variables. As LBM allows for weak compressibility, it models both acoustic wave propagation and sound generation from fluid-structure interaction without any additions to the original model, which makes it a good simulation tool for these applications.

In this master's thesis a Python-based code has been developed following the current standard LBM procedures. The models used as well as the code structure and validation tests are described. The results show that although LBM works as a simulation method for Couette flow and monopole sound generation test cases, there is a need to improve several parts of the code in order to be properly accurate for a Poiseuille flow and yield the right frequency response from vortex shedding. It is also concluded that several acoustic phenomena are not yet commonly modeled using LBM and will need future research in order to use LBM to yield more physically accurate results.

Keywords: Lattice-Boltzmann method, acoustics, two-relaxation time, sponge layers, monopole.

Preface

This report presents the outcome of our master's thesis project carried out at the Department of Mechanics and Maritime Sciences at Chalmers University of Technology during the spring of 2025.

Acknowledgements

I would first and foremost like to thank my examiner Huadong Yao and my supervisor Xiao Xue for the excellent support and guidance I have received during this project. I would also like to thank Lars Davidson for recommending me for this opportunity, my colleague Erik Hasselwander for many new ideas on how to structure and implement the code, and my opponent Viktor Sundström for helping me better understand the theory of LBM. Furthermore, I would like to thank the people at the division of fluid mechanics for good company during lunch hours, and my friend Viktor Skott for lending me extra computational power.

Pontus Nilsson, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ABB	Anti-bounce-back
BC	Boundary condition
BGK	Bhatnagar-Gross-Krook
CPU	Central processing unit
csv	Comma-separated values
DdQq	A d-dimensional velocity set containing q velocities
FVM	Finite volume method
GPU	Graphics processing unit
HBB	Half-way bounce-back
HP	Hermite polynomial
LBE	Lattice Boltzmann equation
LBM	Lattice Boltzmann method
MRT	Multi-relaxation time
NRBCs	Non-reflecting boundary conditions
NSE	Navier-Stokes equations
PDF	Particle distribution function
TRT	Two-relaxation time

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices and superscripts

i, j	Indices for lattice directions
$\bar{i}$	Opposite direction to i
α, β	Index for spacial dimension
x, y	Vector component in spacial direction
eq	Equilibrium value
c	Post-collision value
$*$	Non-dimensional value
max	A maximum value
0	Value at reference level
’	Deviation from reference value
$+$	Symmetric part
$-$	Anti-symmetric part

Variables

$\mathbf{F} = [F_\alpha]$	Body force [$\mathrm{N\,m^{-3}}$]
$\mathbf{x} = [x_\alpha]$	Position [m]
$\mathbf{x}_i = [x_{i\alpha}]$	A set of discrete points in the Gauss-Hermite quadrature rule [-]
$\boldsymbol{\xi} = [\xi_\alpha]$	Particle velocity [$\mathrm{m\,s^{-1}}$]
$\mathbf{c} = [c_{i\alpha}]$	Lattice velocity [$\mathrm{m\,s^{-1}}$]
$\mathbf{u} = [u_\alpha]$	Local mean velocity [$\mathrm{m\,s^{-1}}$]
$\mathbf{v} = [v_\alpha]$	Relative velocity [$\mathrm{m\,s^{-1}}$]

x, y	Spatial directions [m]
r	A distance to a point [m]
t	Time [s]
T	Period time [s]
ζ	A variable change analogous to t used for integration
φ	Frequency [Hz]
ω	Angular velocity [$\mathrm{rad\,s^{-1}}$]
k	Wave number [$\mathrm{m^{-1}}$]
ρ	Density [$\mathrm{kg\,m^{-3}}$]
p	Pressure [Pa]
F_i	Force term [$\mathrm{kgs^2m^{-6}}$]
S_i	Forcing source term [$\mathrm{kgs^2m^{-6}}$]
E	Specific total energy [$\mathrm{J\,kg^{-1}}$]
e	Specific internal energy [$\mathrm{J\,kg^{-1}}$]
s	Specific entropy [$\mathrm{J\,kg^{-1}\,K^{-1}}$]
n, N, m	Integer [-]
n_x, n_y	Number of nodes in directions x or y [-]

Parameters

c_s	Speed of sound [$\mathrm{m\,s^{-1}}$]
c_p	Phase speed of sound [$\mathrm{m\,s^{-1}}$]
η	Dynamic viscosity [Pa s]
η_B	Bulk dynamic viscosity [Pa s]
ν	Kinematic viscosity [$\mathrm{m^2\,s^{-1}}$]
ν_B	Bulk kinematic viscosity [$\mathrm{m^2\,s^{-1}}$]
τ	Relaxation time [s]
τ_{vi}	Viscous relaxation time [s]
h	The height of a channel [m]
L	A reference length [m]
D	The diameter of a circle or cylinder [m]
V	A reference velocity [$\mathrm{m\,s^{-1}}$]
Δx	Space between two nodes [m]
Δt	Time step [s]

R	Specific gas constant [$\mathrm{J\,kg^{-1}\,K^{-1}}$]
C_V	Heat capacity at constant volume [$\mathrm{J\,kg^{-1}\,K^{-1}}$]
γ	Adiabatic index [-]
θ	Temperature [K]
d	Integer representing the number of spatial dimensions [-]
q	Integer representing the number of lattice dimensions [-]
Λ	The magic parameter in TRT [-]

Constants

e	Euler's constant [-]
w_i	The weights in the Gauss-Hermite quadrature rule [-] or [$\mathrm{s^3\,m^{-3}}$]

Functions

$\mathbf{H}^{(n)}(\mathbf{x})$	A d-dimensional Hermite polynomial of order n [-]
$\omega(\mathbf{x})$	The weight function to a d-dimensional Hermite polynomial [-]
$\mathbf{a}^{(n)} = a^{(n)}_{\alpha_1 \dots \alpha_n}$	The Hermite series expansion coefficient of order n for a d-dimensional function [-]
$P^{(N)}(\mathbf{x})$	A d-dimensional polynomial of order N [-]
$\mathbf{P}^{(N)}(\mathbf{x})$	A d-dimensional vector of d-dimensional polynomials of order N [-]
$g(\mathbf{x})$	An arbitrary function of the d-dimensional vector $\mathbf{x}$ [-]
$f(\mathbf{x}, \boldsymbol{\xi}, t)$	Particle distribution function [$\mathrm{kgs^3m^{-6}}$]
$\hat{f}(\mathbf{x}, \boldsymbol{\xi}, t)$	Truncated series expansion of f [$\mathrm{kgs^3m^{-6}}$]
$\overline{f}(\mathbf{x}, \boldsymbol{\xi}, t)$	A variable change of f [$\mathrm{kgs^3m^{-6}}$]
$\Omega(f)$	Collision operator [$\mathrm{kgs^2m^{-6}}$]
$Q(t)$	A function describing density fluctuations in a point source [$\mathrm{kg\,m^{-3}}$]
$j_i = w_i Q(t)$	A source term for Q in LBM [$\mathrm{kgs^3m^{-6}}$]

Non-dimensional numbers

$\mathrm{Ma} = \frac{u}{c_s}$	Mach number
$\mathrm{Re} = \frac{uL}{\nu}$	Reynolds number
$\kappa = \frac{\varphi D}{u}$	Strouhal number

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

The Lattice Boltzmann method (LBM) is a technique for simulations of fluids that provides an alternative to traditional finite volume methods (FVM). Obvious advantages with LBM include the ability to treat complex boundaries and its potential for parallelization due to the method being explicit and local [1].

The inherent weak compressibility of LBM makes it a suitable method for simulation of acoustic waves [2]. This could make it a useful tool within several fields of research and industry, as minimizing noise pollution is needed within a variety of applications.

The current operation of the shipping industry produces a lot of underwater noise, particularly due to propeller cavitation. A growing field of research is how to minimize this noise in order to lower the negative effects on marine life [3]. Other parts of the transport industry also try to minimize their noise contribution for the comfort of passengers and bystanders [4]. The generation and propagation of noise is also important due to the negative health effects associated with noise. Examples of fields where this is studied includes urban planning [5] and the indoor climate industry [6].

Within any of these applications of acoustics there is a need for accurate and efficient simulation software, and LBM is a reasonable method for this due to the aformentioned advantages. However, as LBM is a recent method (compared to other computational methods) there are many open questions regarding how to best utilize LBM as an acoustic solver, and what sub-models might be needed. This master's thesis is thus part of a larger project researching these questions.

1.2 Purpose

The purpose of this is for the writer to start a research project within the field of LBM as a simulation method for acoustics, and thus within the limitations described later in this report learn the fundamentals of LBM and how it can be applied to acoustics while also showcasing this as a research field of interest for industrial applications.

1.3 Goals

The primary goal of the project is to build software based on the fundamental theory of LBM. More specifically, the goal is to achieve 2D software that can simulate acoustics in simple geometries and export the results for visualization. The aim was that this code would successfully reproduce basic validation cases.

1.4 Limitations

To limit the scope of the project, turbulence modeling was excluded, as this can be implemented in a later stage. It is also decided to exclude physical phenomena that is not of common acoustic relevance, and to thus only study flows of $\mathrm{Ma} \leq 0.1$. This is in particular due to the known difficulty of using LBM for high Mach number flows. Implementation of the so called characteristic boundary conditions for LBM has also been excluded.

2

Theory

This chapter will explain what the lattice-Boltzmann is by showing a summary of its derivation and implementation as well as showing the fundamentals of acoustic wave propagation and how LBM can be used as a model for these. Most of this chapter is based on the book *The Lattice Boltzmann Method: Principles and Practice* by Timm Krüger et al. [1], and as such there will only be citations in the text when a reference is to any other source.

The beginning of this chapter mentions the analytical solutions to several canonical fluid dynamics problems that will later be used as reference cases for LBM. There is then a section discussing how a compressible fluid model needs to be completed with an *equation of state* relating pressure and density through other variables, as this provides further understanding of the assumptions on which LBM is based.

Later a brief derivation of LBM is presented, aiming to give the reader a general background as to how and why LBM works as a fluid description. This derivation is followed by an explanation of particular models and algorithms utilised for this thesis. The last part of this chapter discusses the theory of the field of acoustics, and in particular how LBM can be used within that field.

2.1 Notation

This chapter follows tensor notation using the Einstein summation convention. The choice of variable names is done following the book by Krüger et al. [1], which implies that the indices for spacial dimensions are α, β etc., with i, j etc. being reserved for lattice directions.

2.2 Continuous fluid dynamics

Traditional fluid mechanics is part of the field of continuum mechanics, where all matter is assumed to be in a continuous distribution. These assumptions lead to the famous *Navier-Stokes equations* (NSE). When the viscosity is assumed to be constant in a fluid they read

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial \rho u_\alpha}{\partial x_\alpha} &= 0 \\ \frac{\partial (\rho u_\alpha)}{\partial t} + \frac{\partial (\rho u_\alpha u_\beta)}{\partial x_\beta} &= -\frac{\partial p}{\partial x_\alpha} + \eta \frac{\partial^2 u_\alpha}{\partial x_\beta \partial x_\beta} + \left(\eta_B + \frac{\eta}{3}\right) \frac{\partial^2 u_\beta}{\partial x_\alpha \partial x_\beta} + F_\alpha, \end{aligned} \tag{2.1}$$

where we have one equation describing the continuity of mass and one equation per spacial dimension describing momentum in that direction. However, there is always one more unknown than the number of equations, as u_α, ρ and p are unknowns. For an incompressible flow, ρ is assumed to be a constant and this problem disappears, while for a compressible flow there is a need of a further equation, normally referred to as an *equation of state*.

Other concepts from traditional fluid mechanics often used in this report are the Reynolds number $\mathrm{Re} = \frac{uL}{\nu}$ and the Mach number $\mathrm{Ma} = \frac{u}{c_s}$.

2.2.1 Analytical solutions to the Navier-Stokes equations

There are several known analytical solutions to the incompressible NSE with simple geometries. A *Couette flow* is an incompressible shear flow driven by the velocity difference of two infinitely large moving walls. The result is a linear velocity profile between the velocity of the two walls.

A two-dimensional *Poiseuille flow* is a flow in a channel between two no-slip boundaries that is driven by a pressure gradient or a force. The analytical solution is

$$u_x(y) = \left(\frac{\mathrm{d}p}{\mathrm{d}x} - F_x\right) \frac{1}{2\rho\nu} \left(y^2 - \frac{h^2}{4}\right) \tag{2.2}$$

where h is the width of the channel and y is measured from the symmetry line. Both Couette and Poiseuille flow are typical of low Reynolds numbers.

2.3 Equations of state

The most famous equation of state is the ideal gas law,

$$p = \rho R\theta, \tag{2.3}$$

which relates pressure and density through the *specific gas constant* R and the temperature θ. However, as the temperature can be a variable as well and as such a function of further equations this does not necessarily close the equation system.

An equivalent equation of state is

$$\frac{p}{p_0} = \left(\frac{\rho}{\rho_0}\right)^\gamma e^{(s-s_0)/C_V} \tag{2.4}$$

which instead relates pressure and density to entropy s using a constant reference state (p_0, ρ_0, s_0) and the heat capacity at constant volume C_V with $\gamma = \frac{C_V+R}{C_V}$. In acoustics we further assume entropy to be constant and are thus left with the *isentropic equation of state*

$$p = p_0 \left(\frac{\rho}{\rho_0}\right)^\gamma. \tag{2.5}$$

By linearising the isentropic equation of state around the reference values and noting that by definition the speed of sound is defined

$$c_s^2 = \frac{\partial p}{\partial \rho} \tag{2.6}$$

at a constant entropy it is found that $p \approx p_0 + c_s^2 \rho'$ where $\rho = \rho_0 + \rho'$. Acknowledging that the reference values are constant, and thus p_0 never affects the NSE, as only the gradient of p is a contributing factor, we can set $p_0 = c_s^2 \rho_0$ and thus end with

$$p = c_s^2 \rho. \tag{2.7}$$

It should be noted that this equation is the same result as the isothermal equation of state that would result from setting the temperature to a constant in the ideal gas law, albeit with a different definition of the speed of sound. The key conclusion from this derivation is that equation 2.7 can be used as an equation of state whenever entropy is nearly constant and the fluctuations in density and pressure are small enough for the linear assumption.

2.4 The mesoscopic nature of LBM

Continuum mechanics are derived based on the assumption that the scale of the system is much larger than the scale of the molecules that the fluid is made of, such that the molecules themselves can be considered fully irrelevant. In LBM literature this is called a *macroscopic* description. The opposite of this would be a *microscropic* description, where the fluid instead is modeled as a collection of individual molecules. An example of a microscopic fluid description is *molecular dynamics*. In between the microscopic and macroscopic realms lies the *mesoscopic* realm. In a mesoscopic model the scales are large enough to not model every molecule, yet small enough for a continuum description to be invalid. That is, a mesoscopic description of a fluid models distributions of particles. LBM is a model for macroscopic quantities done through mesoscopic theory, and there is thus a need to understand which equations are important at that scale in order to understand how LBM is derived. The field of *statistical mechanics* describes the properties of molecular distributions at the mesoscopic scale.

2.5 Derivation of LBM

This section outlines a common derivation method of LBM in a brief manner explaining its core concepts. First the statistical description of particle distributions and how it can be used to describe a macroscopic fluid is introduced. Then the field equation that describes such a particle distribution is studied to create a discritised version where information only is stored for nodes on a regular lattice. The concept of non-dimensionalisation will be briefly described, as will *Hermite polynomials*.

2.5.1 Particle distribution function

The *particle distribution function* (PDF), $f(\mathbf{x}, \xi, t)$, represents the density of particles with a certain velocity ($\boldsymbol{\xi}$) at a specific location in space-time. Similar to a statistical probability distribution, the integral of f has a clearer physical meaning. In particular its integrals through velocity space, called moments, are

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}, \tag{2.8}$$

$$\rho(\mathbf{x},t)\mathbf{u}(\mathbf{x},t) = \int \boldsymbol{\xi} f(\mathbf{x},\boldsymbol{\xi},t)d\boldsymbol{\xi} \tag{2.9}$$

and

$$\rho(\mathbf{x},t)E(\mathbf{x},t) = \int |\boldsymbol{\xi}|^2 f(\mathbf{x},\boldsymbol{\xi},t)d\boldsymbol{\xi} \tag{2.10}$$

where E is the total energy. It is however possible to remove the energy from the bulk flow and only retain the internal energy e by introducing the relative velocity $\mathbf{v} = \boldsymbol{\xi} - \mathbf{u}$, where $\mathbf{u}$ represents the local mean velocity. We thus retain

$$\rho(\mathbf{x},t)e(\mathbf{x},t) = \int |\mathbf{v}|^2 f(\mathbf{x},\boldsymbol{\xi},t)d\boldsymbol{\xi}. \tag{2.11}$$

Knowing macroscopic quantities can be computed from f we know that a differential equation describing the evolution of f would describe a fluid flow. Knowing that f only depends on $\mathbf{x}$, $\boldsymbol{\xi}$ and t, we know that

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial t}\frac{\mathrm{d}t}{\mathrm{d}t} + \frac{\partial f}{\partial x_\alpha}\frac{\mathrm{d}x_\alpha}{\mathrm{d}t} + \frac{\partial f}{\partial \xi_\alpha}\frac{\mathrm{d}\xi_\alpha}{\mathrm{d}t} \tag{2.12}$$

where $\mathrm{d}t/\mathrm{d}t = 1$, $\mathrm{d}x_\alpha/\mathrm{d}t = \xi_\alpha$ and from Newtons second law for a volumetric body force F_α, $\mathrm{d}\xi_\alpha/\mathrm{d}t = F_\alpha/\rho$.

Introducing $\Omega(f) = \mathrm{d}f/\mathrm{d}t$ as an operator describing the local redistribution of f, we have can rewrite equation 2.12 as the *Boltzmann equation*

$$\frac{\partial f}{\partial t} + \xi_\alpha\frac{\partial f}{\partial x_\alpha} + \frac{F_\alpha}{\rho}\frac{\partial f}{\partial \xi_\alpha} = \Omega(f). \tag{2.13}$$

This can be interpreted as a transport equation for f, where the first two terms on the left hand side describe it's advection with the velocity ξ_β, and the third left hand side term represents forces affecting the advection. The $\Omega(f)$ operator must therefore describe the change in f due to particle collisions, which is why it's normally referred to as the *collision operator*. This collision operator can be showed to represent viscosity and diffusion constants in the Boltzmann equation.

Allowing the gas to settle for a long enough time would lead to the velocity distribution of molecules behaves isotropically around an average fluid velocitu u. It can be showed through statistical mechanics that this *equilibrium distribution* is

$$f^{\mathrm{eq}} = \frac{\rho}{(2\pi R\theta)^{d/2}} e^{-(\boldsymbol{\xi}-\mathbf{u})^2/(2R\theta)} \tag{2.14}$$

where d is the spacial dimension the derivation is done in (i.e normally $d = 3$), R is the specific gas constant and θ is the temperature of the fluid.

As the equilibrium distribution is what internal collisions lead to if the behaviour settles we know that the collision operator will bring f towards f^{eq}. We can then list a set of demands on any collision operator. Since we know collisions conserve mass and momentum, those must also be conserved by the collision operator. In LBM it is also assumed that the translational energy is conserved, which is only true for a monoatomic gas. A simple collision operator that follows these conditions that is a common foundation for LBM collision operators is the the *BGK operator*, named after its inventors Bhatnagar, Gross and Krook.

$$\Omega(f) = -\frac{1}{\tau}(f - f^{\mathrm{eq}}) \tag{2.15}$$

The BGK operator introduces the *relaxation time* τ, representing the time it takes for the system to return to the equilibrium distribution locally.

It should be noted that the BGK operator leads to a *Prandtl number* (the ratio of viscosity and heat conduction) of 1, while more complex collision operators can return values closer to what's found in experiments.

2.5.2 Non-dimensionalisation

Any physical quantity can be written in a non dimensional form by normalisation with a reference value. The equations in LBM theory only use the quantities length, time, mass and temperature, which can be normalised using any four dimensionally independent reference variables only containing these four quantities. For a fluid these are normally a reference length scale L, a reference velocity V, a reference density ρ_0 and in the case of LBM also the specific gas constant R. This allows us to define e.g. non-dimensional length $x^* = x/L$, velocity $\xi^* = \xi/V$, time $t^* = Vt/L$, and temperature $\theta^* = R\theta/V^2$. Here we use starred quantities to denote that they are non-dimensional, but in this report this will be dropped unless specifically stated to make the notation more compact.

The non-dimensional Boltzmann equation can then be found to be of the exact shape as the normal Boltzmann equation (equation 2.13) with the non-dimensional equilibrium distribution

$$f^{\text{eq}} = \frac{\rho}{(2\pi\theta)^{d/2}} e^{-|\boldsymbol{\xi}-\mathbf{u}|^2/(2\theta)} \tag{2.16}$$

2.5.3 Hermite Polynomials

A *Hermite polynomial* (HP) in a multidimensional space of dimension $d \geq 1$ is a polynomial of order $n \geq 0$ defined as

$$\mathbf{H}^{(n)}(\mathbf{x}) = (-1)^n \frac{1}{\omega(\mathbf{x})} \boldsymbol{\nabla}^n \omega(\mathbf{x}) \tag{2.17}$$

where

$$\omega(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} e^{-|\mathbf{x}|^2/2} \tag{2.18}$$

is a so called *weight function*. The tensors $\mathbf{H}^{(n)}$ and $\boldsymbol{\nabla}^{(n)}$ are of rank n and dimension d and thus containing d^n elements. The elements of $\boldsymbol{\nabla}^{(n)}$ all correspond to n sequential derivatives, and the elements of $\mathbf{H}^{(n)}$ thus correspond to a one dimensional polynomial of order n with up to d different variables.

It is possible to reformulate any sufficiently well-behaved function $g(\mathbf{x})$ as a series of HPs in the same multidimensional space. This is done by constructing a set of expansion coefficients

$$\mathbf{a}^{(n)} = \int g(\mathbf{x}) \mathbf{H}^{(n)}(\mathbf{x}) \mathrm{d}\mathbf{x}, \tag{2.19}$$

which leads to

$$g(\mathbf{x}) = \omega(\mathbf{x}) \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{a}^{(n)} \cdot \mathbf{H}^{(n)}(\mathbf{x}). \tag{2.20}$$

An important property of HPs is the *Gauss-Hermite quadrature rule* for any polynomial of order N denoted $P^{(N)}$ with $N \leq 2n-1$, which in d dimensions read

$$\begin{aligned} \int \omega(\mathbf{x}) P^{(N)}(\mathbf{x}) \mathrm{d}\mathbf{x} &= \sum_{i=1}^{n^d} w_i P^{(N)}(\mathbf{x}_i) \\ w_i &= \prod_{\alpha=1}^{d} \frac{n!}{\left(nH^{(n-1)}(x_{i\alpha})\right)^2}. \end{aligned} \tag{2.21}$$

where $\mathbf{x}_i$ is are a set of all the n^d d-dimensional points that can be made from components that are roots to the one dimensional HP of order n.

2.5.4 Discretisation of the Boltzmann equation

In order to solve the Boltzmann equation on a regular lattice it needs to be discretised in all its variables (i.e. velocity, space, and time). The discretisation in space allows for the assumption that the particle distributions move exactly between the spacially discrete nodes during every discrete time step.

2.5.4.1 Discretisation of the equilibrium distribution

As $f^{\mathrm{eq}} = \frac{\rho}{\theta^{d/2}} \omega(\boldsymbol{\eta})$ for $\boldsymbol{\eta} = \frac{\boldsymbol{\xi}-\mathbf{u}}{\sqrt{\theta}}$ it is simple to exactly compute its Hermite series expansion. The result shows that the first three series coefficients are directly linked to the conserved properties of the collision, i.e. mass, momentum, and energy.

$$\begin{aligned} a^{(0),eq} &= \rho \\ a^{(1),eq}_{\alpha} &= \rho u_\alpha \\ a^{(2),eq}_{\alpha\beta} &= \rho(u_\alpha u_\beta + (\theta-1)\delta_{\alpha\beta}) \end{aligned} \tag{2.22}$$

It can further be shown that only the first three HPs will be necessary to recover all macroscopic behaviour of a fluid. We find

$$\begin{aligned} f^{\mathrm{eq}} &\approx \omega(\boldsymbol{\xi}) \sum_{n=0}^{2} \frac{1}{n!} \mathbf{a}^{(n),\mathrm{eq}} \cdot \mathbf{H}^{(n)}(\boldsymbol{\xi}) = \\ &= \omega(\boldsymbol{\xi}) \rho \Big[1 + \xi_\alpha u_\alpha + (u_\alpha u_\beta + (\theta-1)\delta_{\alpha\beta})(\xi_\alpha \xi_\beta - \delta_{\alpha\beta}) \Big] = \hat{f}^{\mathrm{eq}}. \end{aligned} \tag{2.23}$$

By utilizing the Gauss-Hermite quadrature rule when computing $\mathbf{a}^{(n),\mathrm{eq}}$ (which can be computed using the truncated Hermite series $\hat{f}^{\mathrm{eq}}$) we find that

$$\mathbf{a}^{(n),\mathrm{eq}} = \int \hat{f}^{\mathrm{eq}}(\boldsymbol{\xi}) \mathbf{H}^{(n)}(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \int \omega(\boldsymbol{\xi}) \mathbf{P}^{(N)}(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \rho \sum_{i=1}^{m^d} w_i Q(\boldsymbol{\xi}_i) \mathbf{H}^{(n)}(\boldsymbol{\xi}_i) \tag{2.24}$$

where $\mathbf{P}^{(N)}$ is a polynomial of order $N \leq 4$ when $n \leq 2$, which includes all Hermite polynomials included in $\hat{f}^{\mathrm{eq}}$, where $\boldsymbol{\xi}_i$ are all combinations of roots to $H^{(m)}$. As we then have $4 \leq 2m-1$ we need $m \geq 3$, which leads to the roots 0 and $\pm\sqrt{3}$. By then changing the scaling of the velocity using $c_s = \frac{1}{\sqrt{3}}$, substituting $c_{i\alpha}/c_s = \xi_{i\alpha}$

(the notation c_s is chosen as this constant will correspond to the physical quantity of the speed of sound), while also adding an isothermal assumption of $\theta = 1$ we get the discrete (in velocity space) equilibrium distribution

$$f_i^{\mathrm{eq}}(\mathbf{x}, t) = w_i \rho \left(1 + \frac{c_{i\alpha} u_\alpha}{c_s^2} + \frac{u_\alpha u_\beta (c_{i\alpha} c_{i\beta} - c_s^2 \delta_{\alpha\beta})}{2 c_s^4} \right) \tag{2.25}$$

which satisfies the same conservation laws for density, momentum and energy as the full equilibrium distribution. Other ways to discretise the the truncated equilibrium disribution also exist [1] and can lead to different values for c_s [7].

It should be noted that the isothermal assumption allows us to use the linear equation of state from equation 2.7.

2.5.4.2 Discretisation in velocity space

In order to assure that the same moments are conserved for f as for f^{eq} the discretisation is done in an analogous manner. In order to satisfy the Gauss-Hermite rule we discretise $\frac{\omega(\mathbf{c})}{\omega(\mathbf{c})} f(\mathbf{c})$ therefore end up with

$$\mathbf{a}^{(n)} = \int \frac{\omega(\mathbf{c})}{\omega(\mathbf{c})} f(\mathbf{c}) \mathbf{H}^{(n)}(\mathbf{c}) \mathrm{d}\mathbf{c} \approx \sum_{i=1}^{q} \frac{w_i}{\omega(\mathbf{c}_i)} f(\mathbf{c}_i) \mathbf{H}^{(n)}(\mathbf{c}_i), \tag{2.26}$$

yielding the discrete (in velocity space) particle distribution function

$$f_i(\mathbf{x}, t) = \frac{w_i f(\mathbf{x}, \mathbf{c}_i, t)}{\omega(\mathbf{c}_i)}. \tag{2.27}$$

The discretisation in velocity space of the forcing term $\frac{F_\alpha}{\rho} \frac{\partial f}{\partial \xi_\alpha}$ is done by moving the derivative into the Hermite series expansion of f. The result is to use the force F_α to define the term

$$F_i = w_i \left(\frac{c_{i\alpha}}{c_s^2} + \frac{(c_{i\alpha} c_{i\beta} - c_s^2 \delta_{\alpha\beta}) u_\beta}{c_s^4} \right) F_\alpha. \tag{2.28}$$

The discrete velocity Boltzmann equation will then be

$$\frac{\partial f_i}{\partial t} + c_{i\alpha} \frac{\partial f_i}{\partial x_\alpha} = \Omega_i + F_i, \tag{2.29}$$

where Ω_i is a collision operator acting on f_i.

2.5.4.3 Discretisation in space and time

In order to have an explicit time evolution scheme for the discrete velocity Boltzmann equation there is a need to discretise the equation in space and time. The main step is to replace any differential operators with discrete difference operators with the time step Δt and the spacial step $\mathbf{c}_i \Delta t$. This can be done exactly using the so called *method of characteristics* for the left hand side, assuming there is a variable ζ such that

$$\frac{\mathrm{d}t}{\mathrm{d}\zeta} = 1, \quad \frac{\mathrm{d}x_\alpha}{\mathrm{d}\zeta} = c_{i\alpha}. \tag{2.30}$$

We then find that

$$\frac{\mathrm{d}f_i}{\mathrm{d}\zeta} = \frac{\partial f_i}{\partial t}\frac{\mathrm{d}t}{\mathrm{d}\zeta} + \frac{\partial f_i}{\partial x_\alpha}\frac{\mathrm{d}x_\alpha}{\mathrm{d}\zeta} = \frac{\partial f_i}{\partial t} + c_{i\alpha}\frac{\partial f_i}{\partial x_\alpha}, \tag{2.31}$$

which implies that

$$\int_t^{t+\Delta t}\int_{\mathbf{x}}^{\mathbf{x}+\mathbf{c}_i\Delta t}\left(\frac{\partial f_i}{\partial t} + c_{i\alpha}\frac{\partial f_i}{\partial x_\alpha}\right)\mathrm{d}\mathbf{x}\mathrm{d}t = \int_t^{t+\Delta t}\frac{\mathrm{d}f_i}{\mathrm{d}\zeta}\mathrm{d}\zeta = f_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) - f_i(\mathbf{x},t). \tag{2.32}$$

The integral of the right hand side of equation 2.29 will however have to be approximated. In order to obtain second order accuracy space and time the integration is normally approximated using the trapezoidal rule:

$$\begin{aligned}\int_t^{t+\Delta t}(\Omega_i + F_i)\,\mathrm{d}\zeta = \Bigg(&\frac{\Omega_i(\mathbf{x},t) + \Omega_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t)}{2} \\ &+ \frac{F_i(\mathbf{x},t) + F_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t)}{2}\Bigg)\Delta t \\ &+ \mathcal{O}(\Delta t^3).\end{aligned} \tag{2.33}$$

Unfortunately this result depends on the collision operator in both time steps, and is therefore a time-implicit scheme. But by performing the variable change

$$\overline{f}_i = f_i - \frac{(\Omega_i + F_i)\Delta t}{2} \tag{2.34}$$

we instead regain the time-explicit lattice Boltzmann equation (LBE)

$$\overline{f}_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) - \overline{f}_i(\mathbf{x},t) = (\Omega_i(\mathbf{x},t) + F_i(\mathbf{x},t))\Delta t, \tag{2.35}$$

where $\overline{f}_i$ can be used to compute macroscopic density and momentum density as

$$\begin{aligned}\rho &= \sum_i \overline{f}_i + \frac{\Delta t}{2}\sum_i F_i, \\ \rho u_\alpha &= \sum_i \overline{f}_i c_{i\alpha} + \frac{\Delta t}{2}\sum_i F_i c_{i\alpha},\end{aligned} \tag{2.36}$$

where it should be noted that $\sum_i F_i = 0$ for a body force, as this otherwise would represent a mass source. If F_α is a function of u_α (such as for a corilois force) this leads to a situation where F_i can't be computed exactly.

Using the BGK operator there needs to be a shift in the collision operator as well such that equation 2.35 can be written only in terms of $\overline{f}_i$, using a redefined relaxation time $\overline{\tau} = \tau + \Delta t/2$. For notational simplicity the bar notation is dropped and f_i is normally used instead. Using the redefined variables without the bar notation we find

$$f_i(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t) = f_i(\mathbf{x},t) - \left(\frac{f_i - f_i^{\mathrm{eq}}}{\tau} - S_i\right)\Delta t \tag{2.37}$$

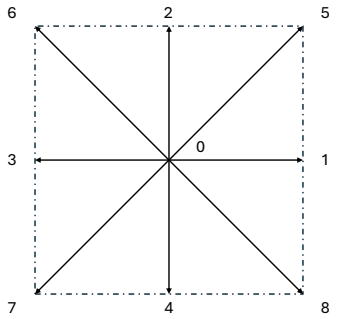

Figure 2.1: A convention for numbering lattice directions in the D2Q9 lattice.

where

$$S_i = \left(1 - \frac{\Delta t}{2\tau}\right) F_i. \tag{2.38}$$

It should be noted that it's possible to derive several different forcing schemes resulting in different versions of equation 2.36 and expressions of S_i. The version presented here is often called *Guo forcing*, as it was first presented by Guo et al. [8].

2.5.4.4 Choice of velocity set

The vectors $c_{i\alpha}$ describe the discrete particle velocities, and thus how the computational nodes are related to each other in the physical space. The choice of physical lattice is normally notated DdQq, with d being the number of spacial dimensions and q being the amount of velocity vectors. Using the full set of vectors described by the roots of $H^{(3)}$ yields the velocity sets D1Q3, D2Q9, and D3Q27.

It is possible to use diffrent mathematical arguments as to why certain vectors in the velocity sets can be discarded in order to create simpler lattices, such as D3Q15 and D3Q19. However, in these cases the expression for the weights also differ [7]. This process is referred to as *pruning*.

Lattices with an odd number of velocity vectors contain a zero velocity vector. This velocity normally has index $i = 0$, with the non-zero vectors having $0 < i \leq q - 1$. For example D2Q9, which is the only one that is relevant in this report, normally has its lattice directions defined as shown in figure 2.1.

2.5.5 The lattice Boltzmann equation as a fluid description

The field of *Chapman-Enskog analysis* relates the Boltzmann equation to macroscopic fluid descriptions, such as the NSE. The result is that the Boltzmann equation is an equivalent description to the a weakly compressible version of the NSE that behaves like the incomrpessible NSE if $\mathrm{Ma}^2 \ll 1$. It is also possible to see that LBM has second order accuracy in both node spacing and time step.

Another important part of Chapman-Enskog analysis is that it is possible to find how the choice of collision operator affects the behaviour of the fluid description. Using the BGK operator we find that the physical viscosity relates to the time step and (redefined) relaxation time as:

$$\nu = c_s^2 \left(\tau - \frac{\Delta t}{2} \right), \quad \eta_B = \frac{2}{3}\eta. \tag{2.39}$$

2.6 The lattice-Boltzmann method

The lattice-Boltzmann method is any scheme used to solve the lattice-Boltzmann equation on a regular lattice. While equation 2.35 allows for f_i to be computed explicitly, it is common to split the computational steps into a collision step

$$f_i^{\mathrm{c}}(\mathbf{x}, t) = f_i(\mathbf{x}, t) + (\Omega_i(\mathbf{x}, t) + F_i(\mathbf{x}, t))\Delta t \tag{2.40}$$

and a streaming step

$$f_i(\mathbf{x} + \mathbf{c_i}\Delta t, t + \Delta t) = f_i^{\mathrm{c}}(\mathbf{x}, t). \tag{2.41}$$

The algorithm is usually set up as follows:

1. Initialise the simulation by computing f_i^{eq} from the initial condition field variables using equation 2.25 and setting $f_i = f_i^{\mathrm{eq}}$. There are also other ways to set up the initialisation.
2. Compute f_i^{c} from equation 2.40.
3. Compute the next time step f_i from equation 2.41. As this equation is not defined for a node located at a boundary of some sorts there is also a need to have different equations for boundary conditions.
4. Compute the next time step ρ and u_α from equations 2.36 and use these to update f_i^{eq} with equation 2.25.
5. Export any variable of interest.
6. Repeat steps 2-5 until the last time step or until convergence has been reached (if the simulation is of a steady flow).

2.6.1 Lattice units

It is practical to avoid including the lattice scale Δx and time step Δt when computing any quantities in LBM. There is a set of units called *lattice units* containing a length unit lu and time unit tu such that $\Delta x = 1\,\mathrm{lu}$ and $\Delta t = 1\,\mathrm{tu}$. This choice of units allows expressions such as (here using superscript $*$ for dimensionless values normalised by the lattice units)

$$\tau = \tau^* \Delta t, \quad c_s = c_s^* \frac{\Delta x}{\Delta t} \tag{2.42}$$

with $c_s^* = \frac{1}{\sqrt{3}}$, and

$$\nu = \nu^* \frac{\Delta x^2}{\Delta t} = c_s^{*2} \frac{\Delta x^2}{\Delta t^2} \left(\tau^* \Delta t - \frac{\Delta t}{2} \right). \tag{2.43}$$

Thus, by defining all physical quantities in lattice units they can be used directly in the non-dimensional discretised LBE, while all factors of Δt can be assumed to be unity. The transformation of units between the physical system and the lattice units are done solely by setting the spacing between the nodes and the simulation time step.

For LBM using the BGK operator the parameters that control the system are thus Δx, Δt and $\tau^* = \tau/\Delta t$. In order to have model similarity with the physical system the Reynolds number have to be preserved between the physical system and the LBM model in lattice units. This implies that for a given spacial resolution it is possible to choose viscosity and velocity freely as long as equation 2.43 is fulfilled. This same argument can be made no matter which lattice scale is decided first. Hence there is normally one degree of freedom left when setting up an LBM simulation with BGK. This degree of freedom will be directly linked to the stability of the simulation.

2.6.2 Stability considerations

Through Chapman-Enskog analysis it is possible to find several variables that affect the stability of a simulation, most notably the maximum occuring velocity. It has been found that the maximum velocity where the simulations are stable for a D2Q9 lattice is increasing as a linear function of τ^* with $0 < |\mathbf{u}_{\max}| \leq 0.4$ for $0 < \tau^* \leq 0.55$ [9]. This can be written

$$|\mathbf{u}| \leq 8\left(\tau^* - \frac{1}{2}\right)\frac{\Delta x}{\Delta t} \tag{2.44}$$

or, using equation 2.43

$$\mathrm{Ma} \leq \frac{8\nu\Delta x}{c_s^3 \Delta t^2} = 24\sqrt{3}\nu^*. \tag{2.45}$$

It important to note how this shows the effect of τ^* on the stability of LBM. As $\nu^* \geq 0$ we know that $\tau^* \geq \frac{1}{2}$ from equation 2.43 is necessary, but for values of τ^* close to $1/2$ there are still stability issues, especially at higher velocities. This directly affects which choices can be done when setting up an LBM simulation using BKG, as the value of τ^* is a direct consequence of the viscosity and the lattice units.

In order to work around this problem, and to decouple viscosity from stability, there are several other possible collision operators. Most notably the *multi-relaxation time* (MRT) operators, which use more than one relaxation time. The simplest MRT operator is the *two-relaxation time* operator (TRT), used in this report.

2.6.2.1 The TRT collision operator

In the TRT collision operator we split the populations and equilibrium populations in their symmetric and anti-symmetric parts and let these have different relaxation times. That is, for every population index i there is a population with the exact opposite direction with index $\bar{i}$ (for $i = 0$ we have $\bar{i} = 0$), and can define

$$f_i^+ = (f_i + f_{\bar{i}})/2 \tag{2.46}$$

$$f_i^- = (f_i - f_{\bar{i}})/2 \tag{2.47}$$

and use this as

$$\Omega_i = -\frac{\Delta t}{\tau^+}(f_i^+ - f_i^{\mathrm{eq}+}) - \frac{\Delta t}{\tau^-}(f_i^- - f_i^{\mathrm{eq}-}). \tag{2.48}$$

Since $f_i^+ = f_{\bar{i}}^+$ and $f_i^- = -f_{\bar{i}}^-$ these extra computations compared with BGK add less computational effort than could be naïvely expected.

The two relaxation times are related through the so called *magic parameter* Λ as

$$\Lambda = \left(\frac{\tau^+}{\Delta t} - \frac{1}{2}\right)\left(\frac{\tau^-}{\Delta t} - \frac{1}{2}\right), \tag{2.49}$$

where different values of Λ cancels different truncation errors in the model. It can also be shown that Λ affects the stability. Krüger et al. [1] states that $\Lambda = \frac{1}{4}$ produces the most stable simulations, and this does in general improve stability compared to BGK for the same viscosity.

Only the symmetric relaxation time affects the viscosity when using TRT, and it is related to viscosity exactly like the BGK relaxation time. That is

$$\nu^* = c_s^{*2}\left(\tau^{+*} - \frac{1}{2}\right), \tag{2.50}$$

with τ^{-*} being a consequence of the choice of Λ.

For TRT it's possible to use a forcing scheme similar to the Guo forcing scheme. As F_i can be split in symmetric and anti-symmetric parts [10]

$$\begin{aligned} F_i^+ &= w_i \frac{(c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta})u_\beta F_\alpha}{c_s^4} \\ F_i^- &= w_i \frac{c_{i\alpha}F_\alpha}{c_s^2}, \end{aligned} \tag{2.51}$$

it is possible to construct a full source term as

$$S_i = \left(1 - \frac{\Delta t}{2\tau^+}\right)F_i^+ + \left(1 - \frac{\Delta t}{2\tau^-}\right)F_i^-. \tag{2.52}$$

2.6.3 Boundary conditions

As equation 2.41 only works for a lattice node that is solely connected to other lattice nodes symbolizing the bulk of a fluid, called fluid nodes, there is a need to have algorithms for the edge of the domain, as well as a way to handle internal geometry. The latter can be handled by marking any internal nodes not included in the fluid due to internal geometry and excluding them from the simulation algorithm. This type of excluded node is normally referred to as a solid node. Any node located next on or next to a boundary such that updating it necessitates considering the BC is normally referred to as a boundary node. The approach to internal geometries leads to arbitrary geometries not aligned with the coordinate directions being approximated by a staircase approach due to Δx being finite. Any shape that's not planar and aligned with the lattice will be full of corners.

It is important to note that there are two radically different methods for implementing boundary conditions at the boundary nodes. There are the wet-node

boundary conditions, where the boundary is assumed to be located at the boundary nodes, and there are the link-wise boundary conditions, where the boundary is located approximately halfway between the boundary nodes and the solid node next to them (i.e on the lattice links). Both types of BCs are commonly used, with the wet-node BC proposed by Zou and He [11] among the most popular. However, Krüger et. al [1] argues that all wet-node BCs lack an accurate way to treat convex corners, and thus only the link-wise boundary conditions are described in this report.

2.6.3.1 Velocity boundary conditions

The *half-way bounce back* (HBB) BC is based on the assumption that a population that would stream into a wall would change direction and stream back to where it came from. It can be shown that this behaviour is equivalent to the no-slip BC commonly used for the NSE. The implementation is similar to the streaming step described in equation 2.41, but with the difference that any population f_i that points out from a HBB BC gets its new population streamed from the population $f_{\bar{i}}$ in that same node. That is, $f_i(\mathbf{x}, t + \Delta t) = f^{\mathrm{c}}_{\bar{i}}(\mathbf{x}, t)$ if $f_{\bar{i}}(\mathbf{x})$ would stream across a HBB boundary. The algorithm is illustrated in figure 2.2.

If the wall would have a non zero velocity u^{w}_α the entire system could be coordinate transformed the the wall's reference frame using a Galilean transform and then transformed back after reversing the population direction, in order to find the momentum change that comes from interacting with the wall. This leads to the full HBB expression

$$f_i(\mathbf{x}, t + \Delta t) = f^{\mathrm{c}}_{\bar{i}}(\mathbf{x}, t) - 2w_{\bar{i}}\rho^{\mathrm{w}}\frac{c_{\bar{i}\alpha}u^{\mathrm{w}}_\alpha}{c_s^2}, \tag{2.53}$$

where ρ^{w} is the local density at the boundary. Normally the wall density is approximated as either $\rho(\mathbf{x}, t)$ or as the average density in the entire domain as any local density fluctuation will be of $\mathcal{O}(\mathrm{Ma}^2)$ and therefore small.

Since a Dirichlet type velocity BC such as an inlet is equivalent to a moving wall at a consistent location, equation 2.53 can be used both for moving walls and general Dirichlet velocity BCs. The main drawback for HBB is that a boundary not aligned with a coordinate direction that has a non-zero prescribed velocity will not be capable of exact mass conservation.

2.6.3.2 Pressure boundary conditions

The link-wise pressure BC is called the *anti-bounce-back* method (ABB), as the sign of the bounced back population is changed. ABB is a BC controlling density, or pressure using equation 2.7. The expression reads

$$f_i(\mathbf{x}, t + \Delta t) = -f^{\mathrm{c}}_{\bar{i}}(\mathbf{x}, t) + 2w_{\bar{i}}\rho^{\mathrm{w}}\left(1 + \frac{(c_{\bar{i}\alpha}u^{\mathrm{w}}_\alpha)^2}{2c_s^4} - \frac{u^{\mathrm{w}}_\alpha u^{\mathrm{w}}_\alpha}{2c_s^2}\right), \tag{2.54}$$

where ρ^{w} is known parameter but u^{w}_α is unknown. This unknown velocity is normally extrapolated from the nodes in the inward normal direction from the boundary, and as it's only used as a square (or to a higher power) even a linear extrapolation is enough for second order accuracy.

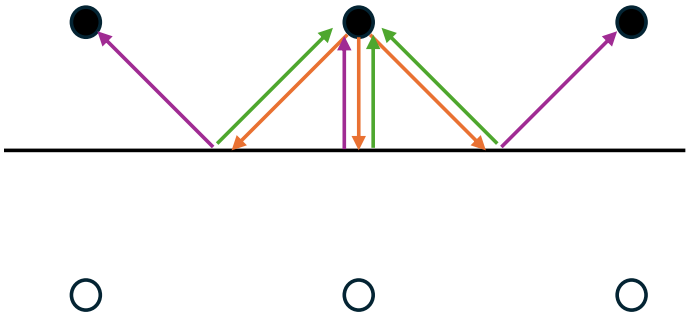

Figure 2.2: Streaming from a fluid node (filled circle) towards a node not part of the fluid domain, e.g. a solid node, (empty circle) along the red vectors intersecting a link-wise boundary (vertical line). Half-way bounce-back (and anti-bounce-back) would lead to streaming back to the original node (green vectors), while a slip BC would lead to propagation in the wall parallel direction (purple vectors).

2.6.3.3 Slip boundary conditions

A slip boundary condition is a common BC for symmetry planes using NSE, as it allows for any velocity in the boundary tangential direction with zero velocity in the boundary normal direction. Slip BCs can also be used together with a wall friction model as a wall BC for a turbulence model [12]. The implementation of a slip BC as a link-wise LBM condition is similar to HBB at a stationary wall, but streaming is allowed in the tangential direction. For a visual description see figure 2.2

When this slip BC is implemented in a concave corner this erroneously leads to the exact same behaviour as for a no-slip BC. This can be avoided by adding a diagonal slip boundary algorithm at a 45° angle [13].

2.6.3.4 Periodic boundaries

Periodic boundaries can be implemented trivially by allowing streaming to take place from one edge of the domain to another. That is, f_i^c is streamed such that it's moved back into the domain in the other end.

2.7 Acoustics

An acoustic wave is a small perturbation in density, pressure and velocity that propagate through the fluid. By inserting

$$\begin{aligned} \rho &= \rho_0 + \rho' \\ p &= p_0 + p' \\ u_\alpha &= u_{0,\alpha} + u'_\alpha \end{aligned} \tag{2.55}$$

in the NSE together with the isothermal equation of state (equation 2.7) and discard all terms that aren't linear we get the *viscous wave equation*

$$\frac{1}{c_s^2}\frac{\partial^2 p}{\partial t^2} - \left(1 + \tau_{vi}\frac{\partial}{\partial t}\right)\nabla^2 p = 0 \tag{2.56}$$

where the *viscous relaxation time* is defined as

$$\tau_{vi} = \frac{1}{c_s^2}\left(\frac{4}{3}\nu + \nu_B\right). \tag{2.57}$$

By assuming $\tau_{iv} = 0$ one finds the *ideal wave equation* where the propagation speed of sound waves is unaffected by the viscosity.

The steady state solutions to equation 2.56 take the shape of a wave with the angular velocity ω and the wavenumber k, moving with the phase speed c_p, while the amplitude slowly attenuates as the waves get dispersed. Due to these attenuation effects $c_p \neq c_s$. In an inviscid world we would instead have waves moving with phase speed c_s, angular velocity ω_0 and wave number $k_0 = \frac{\omega_0}{c_s}$.

It can be showed that the ratio between the speed of sound and the phase speed is

$$\frac{c_p}{c_s} = \frac{k_0}{k} = 1 + \frac{3}{8}(\omega_0\tau_{vi})^2 \tag{2.58}$$

for a forced wave (a wave with a source) and

$$\frac{c_p}{c_s} = \frac{\omega}{\omega_0} = 1 - \frac{1}{8}(\omega_0\tau_{vi})^2 \tag{2.59}$$

for a free wave moving by itself. Here there should be noted that the *acoustic viscosity number* $\omega_0\tau_{vi}$ is a dimensionless number describing the viscous attenuation and dispersion (both increasing with $\omega_0\tau_{vi}$) and their effect on wave propagation.

2.7.1 Sound generation

As sound is nothing else than waves of pressure fluctuation they can be generated in a multitude of ways. A *monopole* sound source is a sound source that behaves like a small sphere with fluctuating walls. This behaviour is equivalent to a local density change in a point. Two monopoles of the same frequency but opposite phase located closely together is called a *dipole* sound source. The analytical solution for the monopole changes depending on how many dimensions the wave propagates through. Without any viscous attenuation we have no amplitude change in 1D, amplitude decreasing with $\frac{1}{\sqrt{r}}$ where r is the distance to the sound source in 2D [14], and amplitude decreasing with $\frac{1}{r}$ in 3D.

Another source of sound would be any flow scenario that leads to an oscillating pressure field. For example, a *von Kárman vortex street* is a famous example where flow around a cylinder starts oscillating periodically at certain Reynolds numbers. It has been shown that this behaves like a dipole sound source. The sound generated from this can be heard e.g. from electrical power lines hanging in the wind [15]. The relation between the cylinder diameter D, the frequency φ, and the flow velocity u is the Strouhal number

$$\kappa = \frac{\varphi D}{u}, \tag{2.60}$$

which can be assumed to be $\kappa \approx 0.2$ at $300 < \mathrm{Re} < 3 \cdot 10^5$ [16].

2.7.2 Acoustic phenomena covered by LBM

Most of what's described above are already included in the derivation of LBM, and thus acoustic propagation at phase velocity c_p is already a part of standard LBM. However, due to discretisation errors short wavelengths that get resolved too coarsely will propagate in an erroneous anisotropic way. This is true from the perspective of spatial resolution for any discretised solver, but as LBM also is discretised in velocity space there will be further errors for large acoustic viscosity numbers ($\omega_0\tau_{vi} > 0.1$. It should also be noted that as LBM is based on the Boltzmann equation instead of the NSE there is an $\mathcal{O}((\omega_0\tau_{vi})^2)$ discrepancy between this models when it comes to dispersion.

In fact, it is common for acoustic waves to appear in LBM simulations. For example due to the initial conditions being a poor match with the solution to the given BCs. Any pressure change due to e.g. a BC will propagate similarly to a free wave through the domain. This behaviour allows for LBM to capture the generation of aerodynamic noise due to fluid structure interactions.

It is also possible to create artificial sound sources in LBM. A monopole can be implemented directly in equation 2.35 by adding an extra source term

$$j_i(t) = w_i Q(t), \tag{2.61}$$

where $Q(t)$ is a function describing the density fluctuation for a point source, on the right hand side for any nodes that should be used as a point source. In a similar fashion, a infinitesimal dipole source can be implemented as an oscillating body force only acting on one node.

In order for an acoustic simulation to match a physical system there is a need to not only match the scaling of the geometry and the viscosity, but to also match the behaviour of the acoustic waves. The approximation $\tau_{vi}^*\Delta t = \tau_{vi}$ would imply that a preserved wave propagation speed compared to the rest of the lattice scales $c_p^*\frac{\Delta x}{\Delta t} = c_p$ is a direct consequence of keeping the wave angular velocity and the speed of sound between lattice units and physical units, i.e. $\omega_0^*/\Delta t = \omega_0$ and $c_s^*\frac{\Delta x}{\Delta t} = c_s$. As $\nu_B^* = 2/3\nu^*$ for BGK (being the same in TRT [17]), $\tau_{vi}^*\Delta t \neq \tau_{vi}$ for a general fluid with a different bulk viscosity to viscosity ratio using any of these collision operators. Thus either ω_0^* would have to be changed compared with the physical case in order to preserve the acoustic viscosity number, or the phase speed and viscous attenuation would be slightly incorrect. It has been noted that acoustic waves in LBM propagate with a velocity of approximately c_s, deviating up to 5% [2].

Keeping the speed of sound between the physical domain means that the ratio between Δx and Δt is constant for any physical system. A direct consequence of this condition is that ν^* will be directly linked to resolution, with ν^* increasing as both Δx and Δt decreases (following equation 2.43). This is also equivalent to using *acoustic scaling* when changing the lattice size, and to conserve the Mach number between the physical domain and the lattice units.

2.7.3 Acoustic phenomena not covered by LBM

Sound waves normally also affect the temperature of the fluid they pass through. This in turn leads to energy loss and further attenuation of the sound waves. However, by constructing a thermal version of LBM, including an advection-diffusion equaiton for the temperature field this acoustic phenomena is accurately modeled.

Further attenuation of the sound waves also happens due to molecular effects, where energy from the sound waves can be absorbed by the molecules as non-translational energy. This type of attenuation is dominant in both air and water at audible frequencies, although not at very high frequencies where viscous and thermal attenuation is dominant. Little research has been done regarding implementing this in LBM models.

Another real world acoustic effect that isn't commonly implemented in LBM is dampening due to absorption of the waves at a solid wall, especially if the surface is soft. At a harder wall the wave will be reflected without much absorption, which is similar to the absorption-less reflection at a HBB BC.

2.7.4 Boundary conditions

As mentioned in section 2.6.3.1 a HBB BC will lead to a somewhat physical reflection. However, as velocity inlets can be modeled using the HBB BC, there will be non-physical reflections at velocity inlets, as these physically should allow the wave to propagate out of the domain.

In a similar fasion, the ABB BC used for pressure BCs that are open and therefore should allow propagation outside of the domain also reflect the incoming wave, albeit with its phase inverted. There is thus a need for *non-reflecting boundary conditions* (NRBC) to be used at any non-wall boundary in an acoustic simulation.

There are several different ways to create NRBCs, with *characteristic boundary conditions* among the more common. These attempt reflecting as little of the incoming wave as possible through modification of the BC, and will not be described further in this report. A simpler method is to create an absorbing layer. That is, changing the fluid behaviour close to the boundary such that any incoming waves get as attenuated as possible before encountering the BC. One way to do this is through a *viscous sponge layer*, where the viscosity of the fluid is increased in the near boundary region in order to increase viscous attenuation and dispersion. A downside of viscous sponge layers is that the wave might get reflected from the sponge layer itself. This problem is somewhat mitigated by having the viscosity being a smooth function of the coordinates in the domain [2].

3
Methods

As the main purpose of this project has been to implement LBM in Python-based code, most of the work has been the programming and debugging of said code. The latter parts of the thesis work has also been spent testing the code. In order to convey the full picture of the thesis project, this chapter will therefore be split in two sections, the first of which will focus on how the code is structured and briefly motivate the key design choices, while the second part will explain the validation procedure and the chosen test cases. Chapter 4 will then present the results from these test cases.

3.1 Code overview and general structure

The code was made in several layers in order to help readability. In the top layer the user defines a set of instructions for the flow case that should be simulated, as well as instructions for how to export the data. This is then used to create an instance of a class named `lbmParent`, the purpose of which is to act as an intermediate layer between the case definition and the main simulation loop. The parent class creates an instance of yet another class (simply named `lbm`) for the main simulation, forwarding all case definitions. The simulations are run by calling the function `lbmParent.run()`, which leads to the simulations initialising before entering a loop of exporting and running a user defined number of time steps of the main algorithm between each export. A graphical representation can be seen in figure 3.1.

The `lbmParent`-specific exporting function exports the values of any user defined function of ρ, $\mathbf{u}$ or f_i as a numpy file, a .csv file, or a plot, also allowing the user to define separate directories for these files.

A single time step is done by the `lbm` class calling calling functions from as these functions are defined using numba no-python just in time compilation with parallelisation in order to have the loops of the domain necessary done as efficiently as possible.

3.1.1 The main code

When the parent class initialises an instance of the main class, empty numpy arrays for any necessary variable with the size matched to the domain size are created. For example, as $i = 0, 2, ..., 8$ in a D2Q9 lattice as used in this code, the $f_i(x, y)$ array is size $n_x, n_y, 9$. The class initialisation also stores which collision function

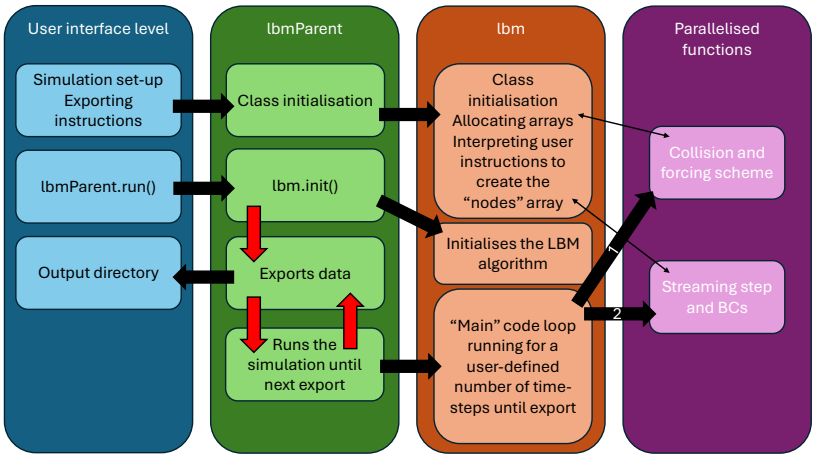

Figure 3.1: A graphical representation of the code structure. The black arrows indicate that one action triggers another while the red arrows show that how exporting and time stepping happens sequentially after initialisation. The class initialisation of the `lbm`-class also creates the `nodes`-array that's used as instruction for streaming and sets which collision and forcing scheme is used during the collision and forcing step.

and forcing scheme should be used, such that the collision step later just calls the function that is stored without checking which one is in use. Both BGK and TRT are implemented, along with a Guo forcing scheme for each version.

Thereafter a map of the domain geometry is created, by creating a n_x by n_y array marking the location of any solid nodes. This map is then used to set up the boundary conditions, where the code is set up such that the domain edge and any internal solid nodes are assumed to be a stationary wall BC.

This part of the code also decodes the user information about where to put sponge layers and using which viscosity at the boundary. The viscosity is then assumed to increase parabolically throughout the domain as this leads to a smooth transition. The changes in relaxation time this leads to is stored as a field variable.

When everything is set up properly and the instance of the main class is created, the parent class calls a routine `lbm.init()`, which performs the initialisation described in section 2.6. It is also possible to use a manual set-up of f_i as the initialisation, allowing for continuation of saved simulation results.

3.1.1.1 Boundary condition set-up

Any boundary condition that is not a stationary wall on the edge of the domain or around a solid node needs to be specified in detail. The user creates a list of arrays, where every array represents a BC in one lattice direction at one particular location. These arrays contain the coordinates of a point, a lattice direction that acts from a boundary towards that node, an integer representing what type of BC should be used, and any prescribed values used for that BC (e.g. density or a velocity vector, or in the case of a periodic boundary information describing which other periodic

boundary it's connected to).

All of this BC information is used to create a $n_x \cdot n_y$ by 11 integer array named `nodes`, where every row has the format

$$[x, y, \text{node type}, \text{BC normal to direction } 1, ..., \text{BC normal to direction } 8].$$

where the node type is 0 for a fluid node and 1 for a solid node (or a node not part of the domain), while the BC type is 0 if there is no BC, and has a different integer value for any BC type implemented in the code. Meanwhile, the prescribed values for the BCs get stored in separate field arrays, i.e. one for all prescribed velocities and one for all prescribed densities. Here there are two possibilities, as the information regarding the boundary could be stored both at the fluid node and at the solid node. The difference is subtle and can be explained as the following question: should all boundaries of the same type around a fluid node have the same value or should all boundaries of the same type around a solid node have the same value? The first version allows for the implementation of an infinitesimal wall as part of a geometry using the HBB BC, as described in Krüger et al. [1], while the latter version allows for a corner intersection between an inlet and a wall. Thus both versions are implemented in the code for HBB, while only the first version is implemented for ABB, as this allows for a pressure BC to be implemented directly at the edge of the domain. If there would be a need to have two different pressure BCs meet at a corner the second type would need to be implemented as a separate type of BC.

3.1.1.2 The main loop

The `lbm` class runs its main loop a user defined number of times between every data exportation done by the parent class. This loop contains the collision and streaming steps described in section 2.6, followed by the update of the macroscopic variables and the equilibrium distribution, as described in that same section. As all of these algorithms require a loop through all nodes in the domain, they were implemented using numba's `@njit(parallel = True)` routine, using parallel loops of all rows in the `nodes` array.

3.1.2 The parallelised functions

The bottom layer of the code contains parallelised functions for the algorithms done each and every time step. The code is split in many different functions in order to improve readability, but the main functions are implemented as follows. The collision step is done in accordance with equation 2.40 for all fluid nodes. Then j_i is added to f_i^c for any node that has a monopole sound source. The streaming step is then done by iterating trough all rows in the `nodes` array again and checking weather they are fluid nodes or not. For any fluid node all 9 lattice directions are iterated through. In the zero-velocity direction we have $f_0 = f_0^c$, while the algorithm for the other lattice directions is to use the f_i^c value in the $\bar{i}$ direction as f_i. Here the structure of the rows in the `nodes` array are of importance, as it is possible for any i to check whether it is possible to conduct streaming in this way. If there

is a boundary condition at the location in the array corresponding to the i lattice direction the algorithm corresponding to that BC is used instead to compute f_i.

3.1.3 Summary of code features

As a summary the code has the following functionality:

- collision operators
  - BGK
  - TRT
- Boundary conditions
  - Half-way bounce-back (Dirichlet velocity) in two versions (homogenous domain edge and elsewhere respectively)
  - Anti bounce-back (Dirichlet pressure, Neumann velocity) for a domain edge
  - Slip boundary conditions aligned with the coordinate axii
- Arbitrary stationary geometry
- Arbitrary constant force-field for body forces
- Rectangular sponge layers with a parabolic change of viscosity along any lattice direction
- Monopole sound sources with arbitrary wave form and period time of integer time step length

3.2 Validation methods

Several different validation methods were used in order to test different parts of the code. Poiseuille and Couette flows at Re = 1 allow tests of the code as an alternative to a NSE solver, while monopole simulations allow for tests of acoustic wave propagation and wave absorption and von Karman vortices allow for study of an aeroacoustic flow at higher Reynolds numbers than the other test cases.

3.2.1 Couette flow

In this test case, the velocity of the bottom wall was assumed to be zero, while the top wall has a velocity at Ma = 0.025, i.e. $u_x = \frac{0.025\Delta x}{\sqrt{3}\Delta t}$. The viscosity was set such that the Reynolds number was 1 in a 400 by 200 node domain ($\nu^* \approx 2.89$). The test case was performed using TRT with $\Lambda = 0.25$, thus $\tau^{+*} \approx 9.16$ and $\tau^{-*} \approx 0.529$. The main purpose of this test case was to validate the periodic boundary conditions used at the left and right edges of the domain.

3.2.2 Poiseuille flow

A 2D Poiseuille flow with Re = 1 was implemented in a 400 by 100 domain, using a slip BC as a symmetry line and HBB with a zero velocity as the one wall that's modeled (thus, $\frac{H}{2} = 100\Delta x$). The maximum velocity was chosen to be at Ma = 0.025 or Ma = 10^{-3} respectively for the two cases that were simulated. The viscosity was then defined from the Reynolds number, which leads to the same viscosity as for the

Couette case, $\nu^* \approx 2.89$, for the Ma $= 0.025$ case and $\nu^* \approx 0.115$ for the Ma $= 10^{-3}$ case. The first case gives us $\tau^* \approx 9.16$ using BGK and $\tau^{+*} \approx 9.16$ and $\tau^{-*} \approx 0.529$ using TRT with $\Lambda = 0.25$, while the second case gives us $\tau^* \approx 0.846$ using BGK and $\tau^{+*} \approx 0.846$ and $\tau^{-*} \approx 1.22$ using TRT with $\Lambda = 0.25$. Both of these collision operators where used for both cases.

As the flow case could be driven both by a pressure gradient or by a body force (or any combination of them both) it was decided to test the accuracy of both the pressure (density) BCs and the forcing scheme. The initial density for the simulations, as well as what was used to compute the driving force or driving pressure gradient was $\rho(\mathbf{x}, t) = 1$.

The force field was implemented along the x-axis, while the pressure gradient was implemented with in the opposite direction, by raising the inlet (left hand side) density by $\frac{\rho}{c_s^2} 200\Delta x$, and lowering the outlet (right hand side) by the same amount, leading to the expression

$$\frac{\rho_{\text{in}} - \rho_{\text{out}}}{400\Delta x} = \frac{\mathrm{d}\rho}{\mathrm{d}x} = \frac{1}{c_s^2}\frac{\mathrm{d}p}{\mathrm{d}x}, \tag{3.1}$$

following equation 2.7. In the force driven case, ABB BCs were still used for the inlet and outlet, albeit with their values set to $\rho = 1$, leading to a non-existent pressure gradient. In summary, both Mach numbers where simulated using both collision operators and both pressure driven and force driven flow.

3.2.3 Monopole sound source

A 2D acoustic monopole with a sinusoidal wave was also used as a validation case. This was compared with a sinusoidal wave that dampened with the square root of the distance to the origin, as this would have the same basic shape as the analytical solution. However, this is not a true reference case in that the analytical solution is based on a different equation (no viscous attenuation) and that the amplitude of the reference wave was normalised to match the simulation amplitude instead of analytically derived from the strength of the sound source.

The simulations where set up in a 2000 node wide square domain with ABB BCs of homogeneous pressure in the entire domain as well as on the boundaries ($\rho^* = 1$). Here it was assumed that this domain approximated a 4m square of air at room temperature ($c_s \approx 343\,\text{m/s}$ [18], $\nu \approx 1.48 \cdot 10^{-5}\,\text{m}^2/\text{s}$ [19]). As the acoustic viscosity number in air is less than $2 \cdot 10^-5$ for a frequency of 10 kHz [1], while the reference solution doesn't include viscous attenuation, it was decided to neglect the contribution of $(\omega_0 \tau_{vi})^2$ to the phase speed and attenuation when setting up the case, and thus directly assume $\frac{1}{\sqrt{3}}\frac{\Delta x}{\Delta t} = 340\,\text{m/s}$ to define Δt. All monopole simulations were performed using TRT with $\Lambda = 0.25$, thus $\nu^* = 1.25 \cdot 10^-5$, $\tau^{+*} = 0.500038$ and $\tau^{-*} = 6632$. Another common setting for all monopole simulations was a duration of exactly ten period times.

It was quickly discovered that the wave reflections at the ABB boundaries was the dominating phenomenon, and as such the main purpose of the simulations were to tune the sponge layers to minimise these reflections. The sponge layers were set with a thickness of 500 (decreasing the part of the domain that were without

a sponge layer to 1000 by 1000) and several different wall viscosities were tested. These were $\nu^* = 20$, $\nu^* = 35$, $\nu^* = 50$, $\nu^* = 75$ and $\nu^* = 100$. This test was done with a periodtime of $T = 675\Delta t$, which corresponds to a frequency of 440.07 Hz in the physical domain. The amplitude of the sine wave was a maximum density change 10^-5, which should be compared with the homogeneous initial density of $\rho^* = 1$. This corresponds to a sound level of 100 dB at the source.

As $\nu^* = 50$ seemed to lead to reflections that affected the simulations the least (of the tested values) this one was quickly chosen to be used in the further tests. See section 4.3.1. With these settings a wider range of frequencies were tested, with a shorter wavelength than the original waves, as this would lead to more waves passing before interference from the boundaries affect the result. These frequencies were chosen to be $T = 100\Delta t$, $T = 200\Delta t$, $T = 300\Delta t$, $T = 400\Delta t$ and $T = 500\Delta t$. This corresponds to frequencies of 2944.5 Hz, 1472.2 Hz, 981.50 Hz, 736.12 Hz and 588.90 Hz. The amplitude of these where again set to 10^{-5}.

There was also a test case where the wave was a superposition of two different frequencies, in order to test that the code can recover a more complex shape than a pure sinusoidal. It should be noted here that all periodic signals can be decomposed as Fourier series in sinusoidal waves [20]. This test case was the sum of a sine wave with frequency 588.90 Hz with an amplitude of 10^{-5} and a sine wave with frequency 1177.8 Hz with an amplitude of $5 \cdot 10^{-6}$. The sampling frequency was 25 samples per period time for all simulations.

3.2.4 Von Kárman vortex street

Aerodynamical sound generation was tested using the von Kárman vortex street. The vortex shedding frequency was investigated using a 400 by 200 node domain with a circle with diameter $D = 10\Delta x$ located at $x = 100\Delta x$, $y = 100\Delta x$. The inlet had a 50 nodes thick sponge layer with a maximum viscosity of $\nu^* = 0.1$ and the outlet had a 100 nodes thick sponge layer with a maximum viscosity of $\nu^* = 10$. These sponge layers were needed to dampen the waves dominating the initialisation part of the simulation. Both y-normal walls used HBB no-slip BCs with 0 velocity. The simulation was run for 60000 time steps and sampled every 25 time steps. The density and velocity fluctuations at $x = 150\Delta x$, $y = 50\Delta x$ were analysed. Reynolds number 400 was used with an inlet maximum Mach number of 0.05. It was observed that the inlet velocity profile needed to obey the wall BCs, and as such a symmetrical quartic function was used. The viscosity in the middle of the domain was computed from the Reynolds number. TRT was used with $\Lambda = 0.25$.

As the vortex street also is a higher Reynolds number case than the other test cases it was also used to test the stability of the TRT operator using $\Lambda = 0.25$. Using two different Mach numbers (0.1 and 0.05) the same general domain was used but with the cylinder size doubled. Three different resolutions was used for these stability tests, the dimensions of which can be seen in table 3.1. The sponge layer thicknesses was scaled with n_x, but the sponge layer viscosity was kept constant. Figure 3.2 shows the general domain shape for all von Kárman simulations.

The procedure was to start at a coarse resolution $D = 10\Delta x$ with Re = 100 for both Mach numbers, and to then increase the Reynolds number in steps of 100

Table 3.1: Values of n_x, n_y, and D for the different resolutions of the stability tests done with the von Kárman set-up.

n_x	n_y	D
200	100	10
400	200	20
800	400	40

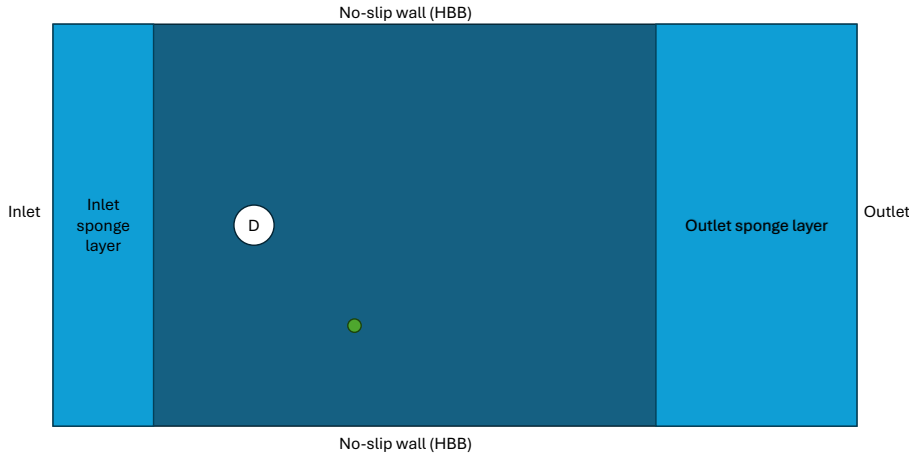

Figure 3.2: The von Kárman vortex shedding domain. The diameter D was doubled from $n_y/20$ to $n_y/10$ for the stability tests. The green circle marks the sampling point for measuring velocity and pressure fluctuations for frequency analysis.

until the simulations crashed before consistent vortex shedding appeared. The same procedure was done at $D = 20\Delta x$ and $D = 40\Delta x$ as well. Table 4.1 shows the exact settings for the maximum stable Reynolds numbers for each Mach number and resolution.

4

Results and discussion

This chapter presents and discusses the results of the validation and stability tests described in chapter 3 in the same order they are described in that chapter. The last section of the chapter discusses how the code could be developed further in order to improve on the results from the validation tests, as well as improve its applicability on real world phenomena.

4.1 Couette flow

The Couette flow simulation compared with the analytical solution for the same case can be seen in figure 4.1. Figure 4.2 shows the density change and velocity change between time steps. This shows that although that the velocity hasn't converged fully it's approaching some converged state, while the density residual has stabilised at a constant value, indicating that the simulation isn't mass conserving. This suggests further investigation of the boundary conditions used needs to be done. As the simulation and analytical results match perfectly in a visual inspection of figure 4.1 we see that the code correctly predicts the velocity profile of Couette flow.

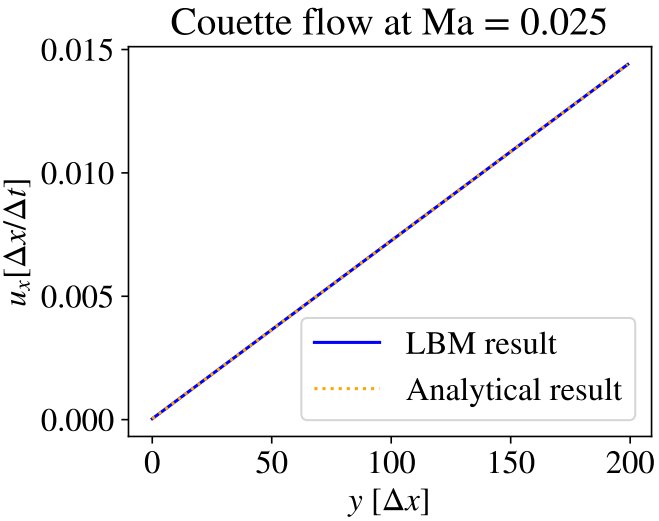

Figure 4.1: A comparison of the simulation result and the analytical solution for Couette flow at Re = 1 and M = 0.025. Note the match.

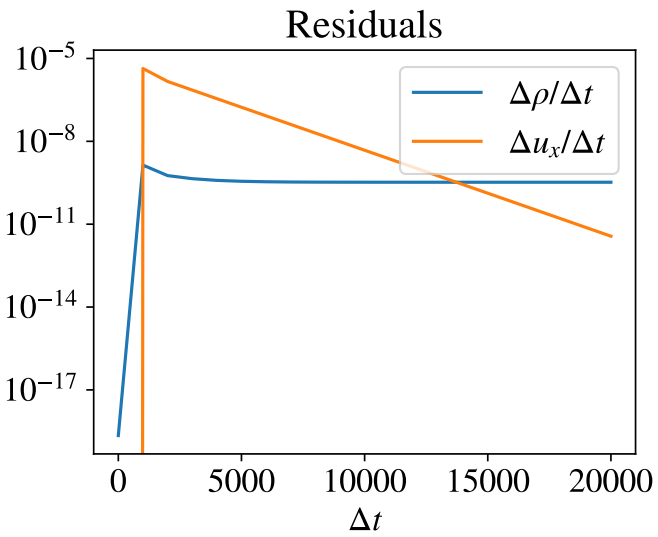

Figure 4.2: The differences in density and velocity between each time showing that the velocity is converging but mass isn't conserved

4.2 Poiseuille flow

The results for the Poiseuille flow simulations at Ma = 0.025 using both TRT and BGK, with both pressure gradient and force as the driving part of the simulation, can be seen in figure 4.3. All models produce a parabolic velocity distribution, like the analytical solution does. The simulations do, however, mostly overpredict the velocity, with the exeption of TRT using a body force. Figure 4.4 shows the results of the Poiseuille flow at Ma = 10^{-3}, where the simulation using TRT and a body force is widely different compared to the other models and the analytical solution, indicating that the forcing scheme implemented for TRT somehow is wrong. This is most likely due to this scheme being implemented as a combination of information from both Krüger et al. [1] and [10] (specifically, parts of [10] contradicting [1] were assumed to be misprints), raising the likelihood that something went wrong with the implementation. Figures 4.5 and 4.7 show that the velocity in the simulations hasn't fully converged yet at the end time step, but indicating that they are close to the final convergence value with the residuals being small, suggesting that any errors compared to the analytical solution are unrelated to convergense. Figures 4.6 and 4.8 show a declining density residual, suggesting that there is a chance that the simulations are reaching a mass conserving state.

Studying the Ma = 0.025 case we find a clear difference between the performance of TRT and BGK, with both BGK simulations resulting in the exact same result, with the TRT simulation using a pressure gradient being slightly closer to the original solution. This behaviour might be explained by the large lattice viscosity in the set-up, as errors increase with τ^* such that $\tau^* \gg 1$ are not recommended to use. This is mitigated somewhat when using TRT [1]. The Ma = 10^{-3} case has a 25 times lower viscosity and should not encounter this issue, which could explain why pressure driven TRT and pressure driven BGK yielded such similar results.

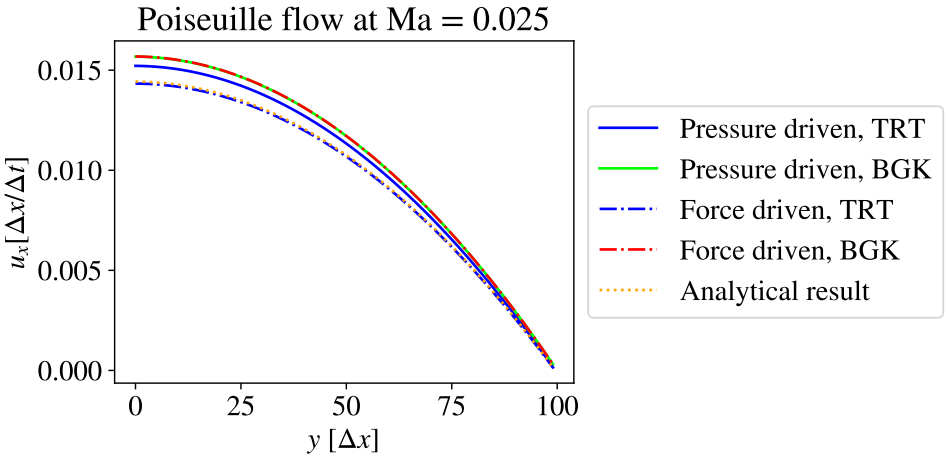

Figure 4.3: A comparison between the results from different collision operators and ways to set up a Poiseuille flow at Re = 1 and Ma = 0.025.

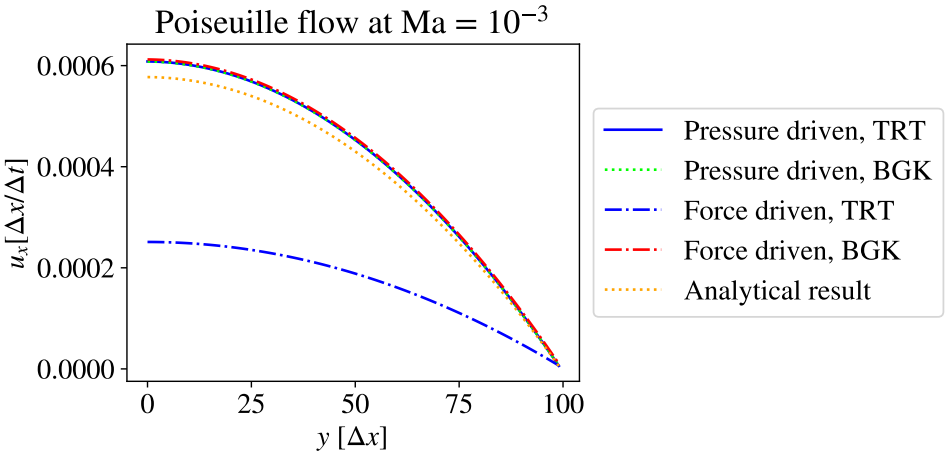

Figure 4.4: A comparison between the results from different collision operators and ways to set up a Poiseuille flow at Re = 1 and Ma = 10^{-3}.

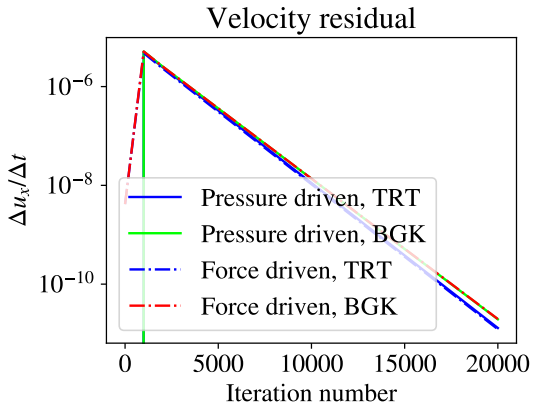

Figure 4.5: The difference in velocity between each time step for the Ma = 0.025 Poiseuille flow showing that the simulations are converging.

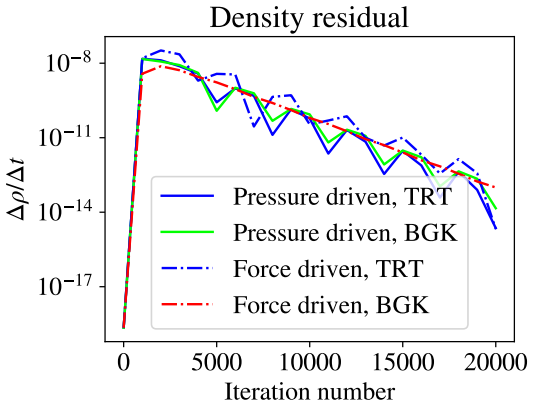

Figure 4.6: The difference in density between each time step for the Ma = 0.025 Poiseuille flow indicating that the simulations are converging to a mass conserving state.

It is known that the force driven case in LBM is the more accurate description of the incompressible Poiseuille flow, as the pressure gradient driven case also has a density gradient due to 2.7 [1]. It is thus unexpected that both BGK simulations perform equally in the Ma = 0.025 case and that the pressure driven simulation performed better in the Ma = 10^{-3} case. This might indicate a wrongful choice of forcing scheme, or a bad implementation of the one used.

The general overestimation of the Poiseuille flow from all simulations (except the innacurate force driven TRT set-up) might be due to the reflectiveness of the ABB BCs, and unavoidable without a non-reflective BC [13].

4.3 Monopole sound source

This section starts with a discussion on the results from the 440 Hz simulations with different sponge layers, motivating why $\nu^* = 50$ was chosen at the boundary for the other simulations. Then the results for other frequencies are presented and discussed.

4.3.1 Tuning the sponge layer

Figures 4.9-4.11 show the density fluctuations of a 440 Hz monopole as a function of time at a few different points in the domain with the time scale and coordinates represented in SI units, using different strengths of sponge layer. The results with $\nu^* = 35$ and $\nu^* = 75$ were omitted for brevity, but can be found in appendix A. Their results lie as expected in between the other results. As the point $x = 0.6$ m, $y = 0$ m and the point $x = 0.36$ m, $y = 0.48$ m lie on exactly the same radius from the sound source, the density fluctuation should have the exact same amplitude and phase at these two locations. This can clearly be seen to be the case for the first few period times, before the wave reaches the domain boundary. Shortly after the

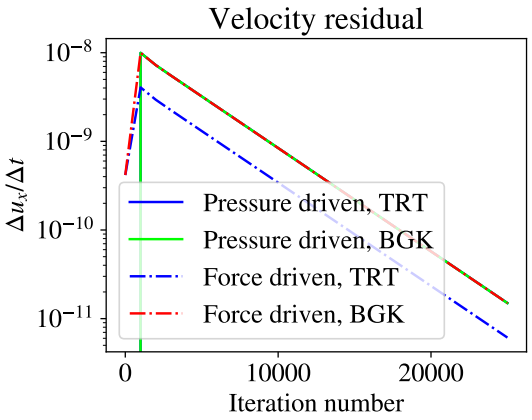

Figure 4.7: The difference in velocity between each time step for the Ma = 0.025 Poiseuille flow showing that the simulations converge.

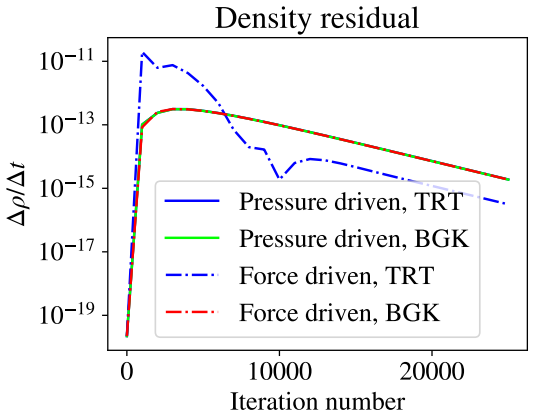

Figure 4.8: The difference in density between each time step for the Ma = 10^{-3} Poiseuille flow indicating that the simulations are converging to a mass conserving state.

wave reaches the wall the density fluctuations at these two locations with the same radius drift apart in both amplitude and phase, albeit differently for the different sponge layer settings. This, as well as the fact that a small wave reaches the domain boundary in all tests, indicate that none of the sponge layers completely dampened the reflection. The simulation where $\nu^* = 50$ at the wall (figure 4.10) contains no clear phase separation to the naked eye and was therefore assumed to be the most effective sponge layer. Hence $\nu^* = 50$ was used at the wall in all further monopole simulations. Figure 4.12 show how the waves are mostly dampened by the sponge layer with these settings with no reflections directly visible.

It is possible that the square shape of the domain and sponge layer interface lead to an anisotropic propagation of the wave towards the BC, which in turn lead to the wave being reflected in a different fashion than if it was isotropically dampened. This can be hinted at in the corners of figure 4.12.

4.3.2 Monopole simulations at different frequencies

With the shorter wavelengths expected from the longer period times used for the rest of the monopole simulations it was assumed that there would be less interference from reflections at the boundary, as the waves would encounter the boundary later in the simulation. This was already noticeable in the 588.90 Hz case seen in figure 4.13. Here it can also be seen that even after the wave has encountered the boundary it has little to no effect on the waves in the rest of the domain. This could be due to the increased viscous attenuation in the sponge layer due to a larger acoustic viscosity number stemming from an increased frequency compared to the 440 Hz case.

For the 2944.5 Hz case all ten wavelengths fit inside of the domain, allowing for a direct comparison with a normalized analytical case can be done. We see in figure 4.14 that the amplitude decays in the same pace from the origin outwards, as

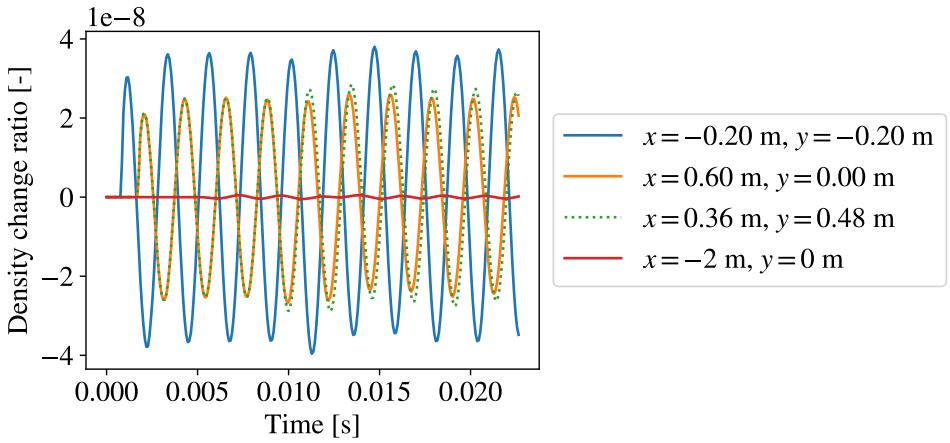

Figure 4.9: The density difference from the initial state of $\rho^* = 1$ as a function of time for a 440 Hz sinusoidal wave originating from the origin using a 500 node sponge layer with maximum $\nu^* = 20$, measured at four points in the domain. Note that $x = 0.60\,\text{m}, y = 0\,\text{m}$ (orange) and $x = 0.36\,\text{m}, y = 48\,\text{m}$ (green) have the exact same radii from the monopole sound source, while the $x = -2\,\text{m}, y = 0\,\text{m}$ point (red) is located at the boundary, where the sponge layer has dampened most of the amplitude.

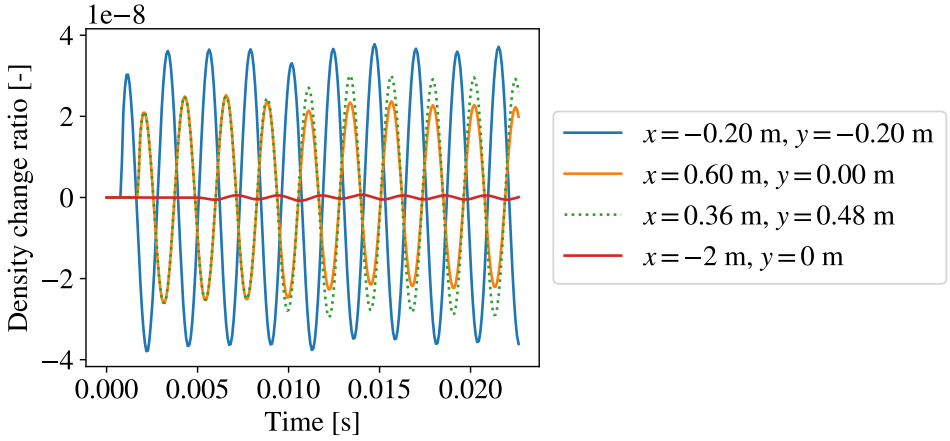

Figure 4.10: The density difference from the initial state of $\rho^* = 1$ as a function of time for a 440 Hz sinusoidal wave originating from the origin using a 500 node sponge layer with maximum $\nu^* = 50$, measured at four points in the domain. Note that $x = 0.60\,\text{m}, y = 0\,\text{m}$ (orange) and $x = 0.36\,\text{m}, y = 48\,\text{m}$ (green) have the exact same radii from the monopole sound source, while the $x = -2\,\text{m}, y = 0\,\text{m}$ point (red) is located at the boundary, where the sponge layer has dampened most of the amplitude.

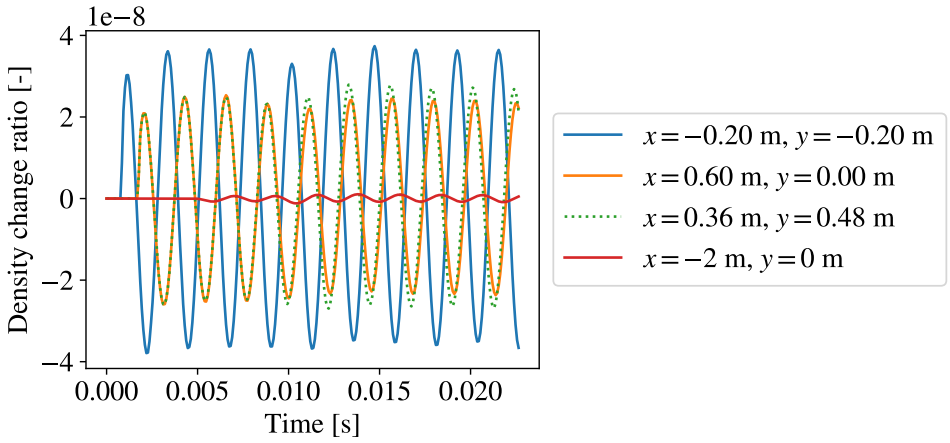

Figure 4.11: The density difference from the initial state of $\rho^* = 1$ as a function of time for a 440 Hz sinusoidal wave originating from the origin using a 500 node sponge layer with maximum $\nu^* = 100$, measured at four points in the domain. Note that $x = 0.60\,\text{m}, y = 0\,\text{m}$ (orange) and $x = 0.36\,\text{m}, y = 48\,\text{m}$ (green) have the exact same radii from the monopole sound source, while the $x = -2\,\text{m}, y = 0\,\text{m}$ point (red) is located at the boundary, where the sponge layer has dampened most of the amplitude.

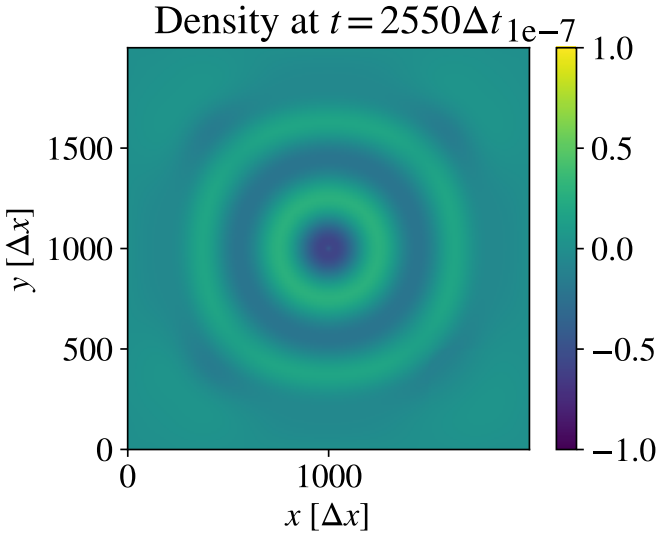

Figure 4.12: The density field at $t = 2550\Delta t$ using a 500 node sponge layer with maximum $\nu^* = 50$. The scale is limited with a threshold that the monopole source is stronger than. Note that the sponge layer dampens the wave as it approaches the boundary. Also note the changed coordinate system in lattice units, placing the origin at the lower left corner.

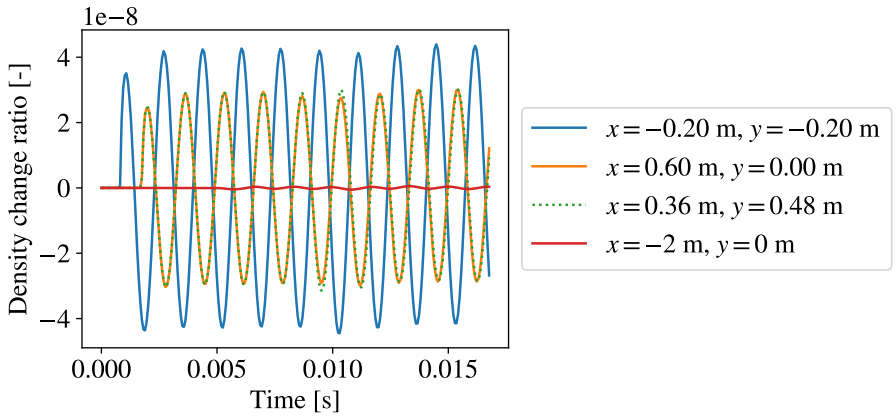

Figure 4.13: The density difference from the initial state of $\rho^* = 1$ as a function of time for a 588.90 Hz sinusoidal wave originating from the origin using a 500 node sponge layer with maximum $\nu^* = 50$, measured at four points in the domain. Note that $x = 0.60\,\text{m}, y = 0\,\text{m}$ (orange) and $x = 0.36\,\text{m}, y = 48\,\text{m}$ (green) have the exact same radii from the monopole sound source, and thus share phase and amplitude, while the $x = -2\,\text{m}, y = 0\,\text{m}$ point (red) is located at the boundary, where the sponge layer has dampened most of the amplitude.

would be expected from the negligible viscous attenuation. There is however little to be said about the amplitude or phase of the wave, as it is not clearly known which phase differences occur between the monopole source and its wave, nor is it clearly known how to relate the density fluctuation to the velocity fluctuation solution for the analytical monopole. The phase of the analytical solution in this comparison is directly in phase with the monopole source, suggesting any such phase difference is negligible. However, the amplitude is only normalized such that the analytical solution fits in the same graph.

Figure 4.15 show that we can draw the same type of conclusions for the longer wavelengths as well, here we see the 1472.2 Hz case. Figure 4.16 shows the frequency spectrum at the same location for each of the five test cases. There are multiple observations to be made here. The frequency of the original monopole source is preserved as the fundamental frequency for each wave, but the frequency spectrum doesn't go to zero apart from that. This implies that the shape of the waves are somewhat changed as they propagate, but part of the error could also stem from a too low sampling rate (25 points per wave).

It should also be noted that the amplitude differs between the simulation results, while the amplitude of the sources were exactly the same. This should not be due to viscous attenuation as that both should be negligible and should affect higher frequencies more than low for a constant τ_{vi} [1]. It is instead likely that the source term somehow creates waves with larger amplitude for higher frequencies.

4.3.3 A wave with harmonics

Figure 4.17 show the frequency spectrum result from having the source producing the sum of two different waves at different amplitude. It can clearly be seen that the

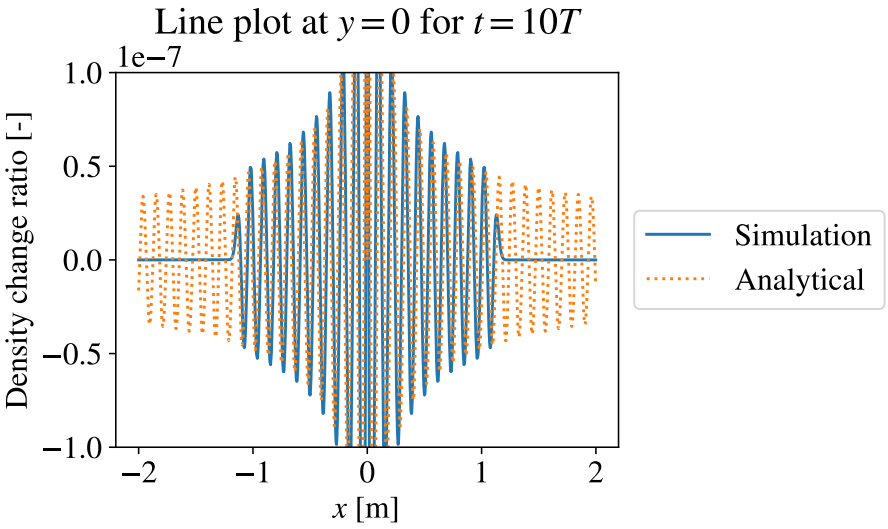

Figure 4.14: The density difference from the initial state of $\rho^* = 1$ as a function of x at $y = 0\,\mathrm{m}$, $t = 10T$ for a 2944.5 Hz sinusoidal wave originating from the origin compared with the solution for a two-dimensional inviscid acoustic monopole, adjusted such that the phase and amplitude somewhat match. Note that the first wave crest just has entered the sponge layer and therefore has dampened.

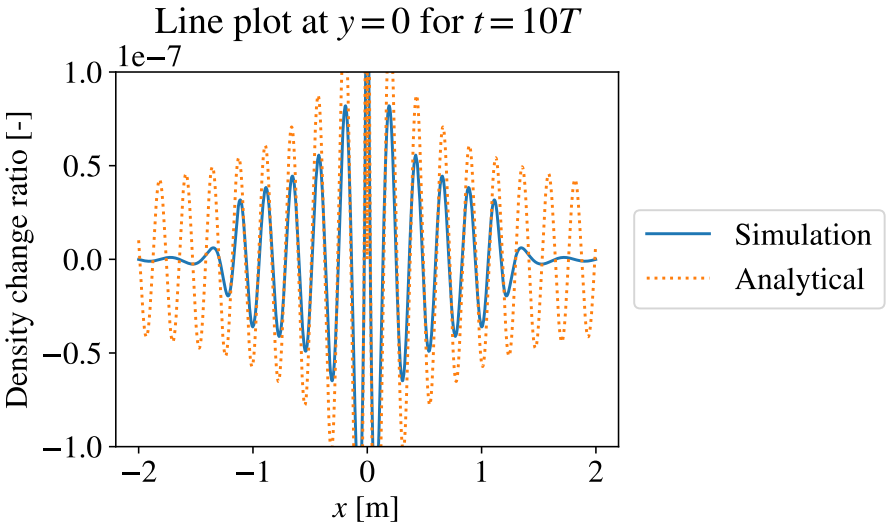

Figure 4.15: The density difference from the initial state of $\rho^* = 1$ as a function of x at $y = 0\,\mathrm{m}$, $t = 10T$ for a 1472.2 Hz sinusoidal wave originating from the origin compared with the solution for a two-dimensional inviscid acoustic monopole, adjusted such that the phase and amplitude somewhat match. Note that the simulated solution is dampened quickly by the sponge layer.

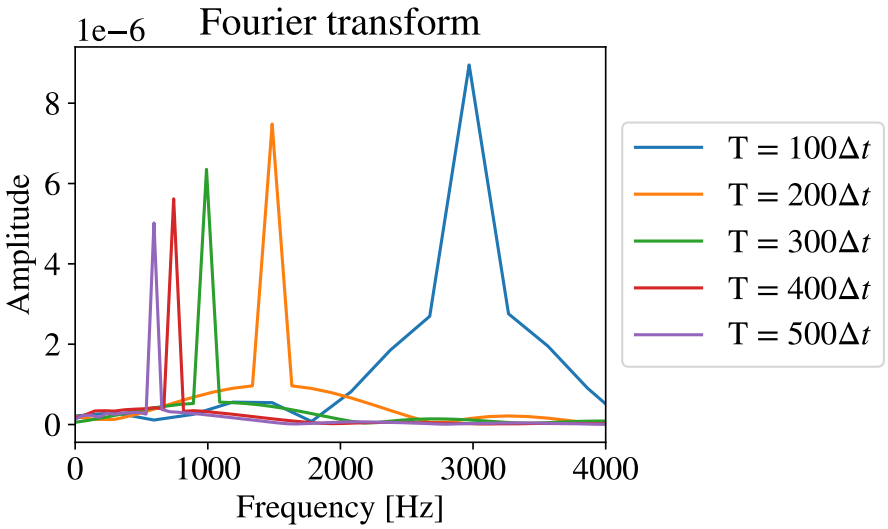

Figure 4.16: The frequency spectrum at $x = -0.2\,\mathrm{m}$, $y = -0.2\,\mathrm{m}$ for the different cases with different period lengths. Note the width of the peaks and the increasing amplitude with frequency.

amplitude ratio is wrong compared with the source function (the source function has a higher frequency wave with an amplitude of 0.5 times the amplitude of the lower frequency wave, while the amplitude ratio in the results is about 0.72). It is likely that this discrepancy stems from the same error that lead to the amplitude discrepancy in figure 4.16 and where discussed in section 4.3.2 This indicates that the source terms used for acoustic monopoles should be further investigated and understood.

The error in frequency spectrum can also be seen when the wave is compared to its analytical counterpart, see figure 4.18.

4.4 Von Kárman vortex street

This section first presents the results from a von Kárman vortex simulation with a high sample rate discussing the acoustic response from the vortex shedding. Then a short stability analysis of the von Kárman vortex simulations is presented. A few snapshots from the Re = 1400 simulations is available in appendix B for the reader who wants to see graphics of vortex shedding.

The Reynolds numbers and Mach numbers used in this section are somewhat physically relevant, as they roughly correspond to the flow case of a guitar string left out in a storm. A simulation of a phenomena larger than that in air would require even higher Reynolds numbers. The applicability to water is however limited, as the speed of sound in water is much larger than the speed of sound in air [21]. It was however noted that a lower velocity simulation has a proportionally lower time scale, and thus needs proportionally more time steps.

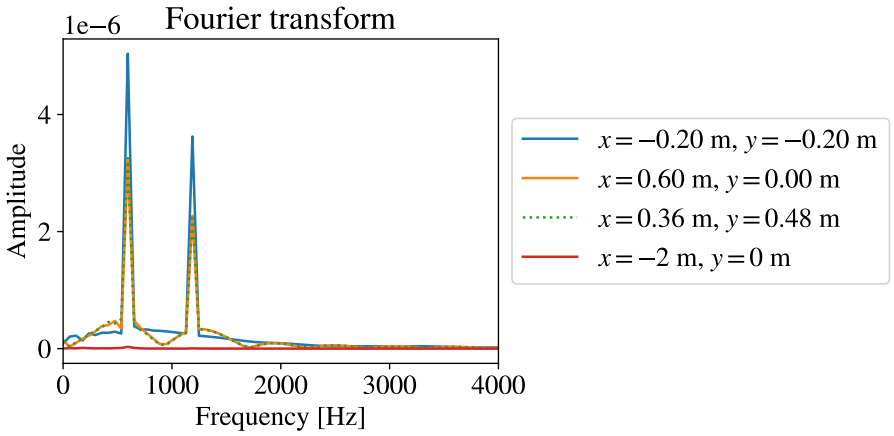

Figure 4.17: The frequency spectrum at a few points in a domain for the simulation with a $\varphi = 588.90\,\text{Hz}$ fundamental frequency as well as its first harmonic (1177.8 Hz) at half of the amplitude. Note that the amplitude ratio between the two frequencies is different to the amplitude ratio between the frequencies at the source.

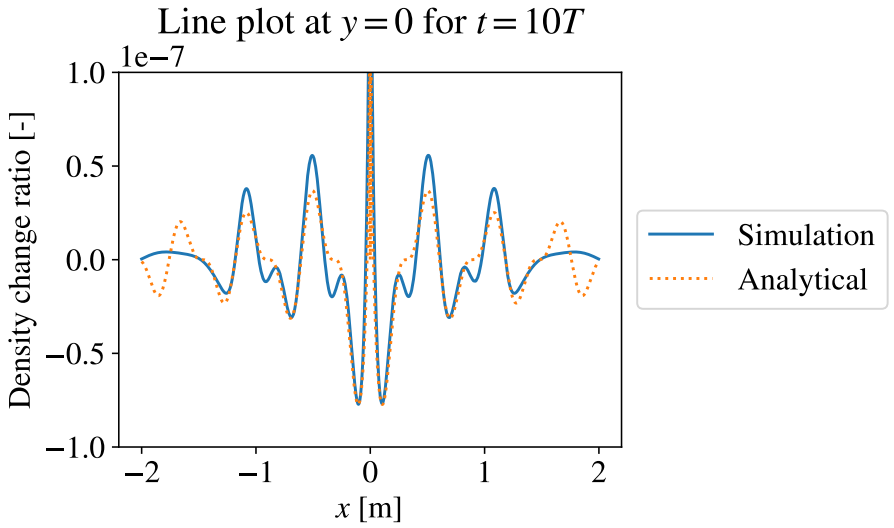

Figure 4.18: The density difference from the initial state of $\rho^* = 1$ as a function of x at $y = 0\,\text{m}$, $t = 10T$ for the case with a wave with an harmonic originating from the origin compared with the solution for a two-dimensional inviscid acoustic monopole, adjusted such that the phase and amplitude somewhat match. Note how the difference in frequency spectrum amplitudes compared to the source affects the shape of the wave, and also note the attenuation when the wave enters the sponge layer.

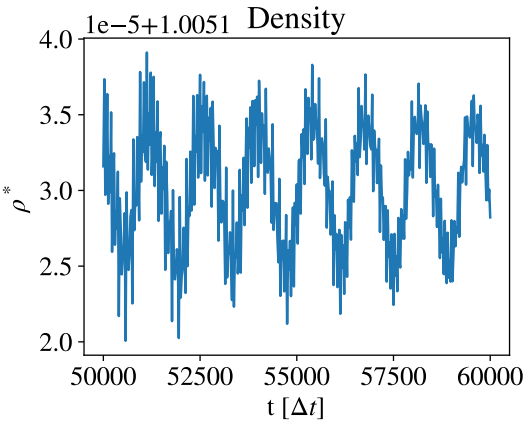

Figure 4.19: The density at $x = 150\Delta x$, $y = 50\Delta x$ as a function of time in the von Kárman vortex simulation where frequency was analysed.

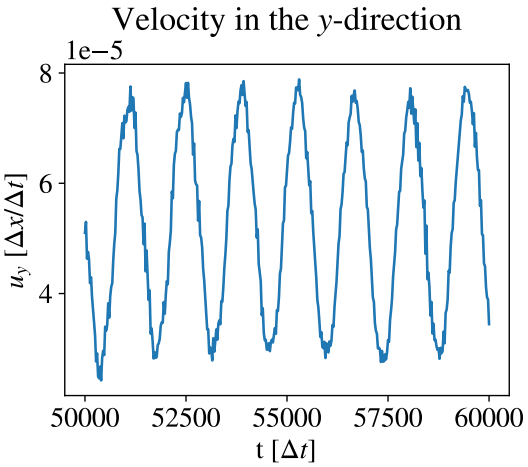

Figure 4.20: The velocity at $x = 150\Delta x$, $y = 50\Delta x$ as a function of time in the von Kárman vortex simulation where frequency was analysed.

4.4.1 Frequency analysis

The density and u_y fluctuations at $x = 150\Delta x$, $y = 50\Delta x$ from a 400 by 200 node domain used to study the frequency of the vortex shedding can be seen in figures 4.19 and 4.20. Based on $\kappa \approx 0.2$ we should find $\varphi \approx 5.77 \cdot 10^{-4}/\Delta t$ as $D = 10\Delta x$ and $u_x = 0.05 c_s^* \Delta x/\Delta t$. However, a Fourier transform of both the density fluctuations and the velocity fluctuations in the y-direction instead reveal $\varphi = 7 \cdot 10^{-4}$, a discrepancy of almost 18%. The synchronization in frequency between the two measurements clearly shows the acoustical behaviour of vortex shedding, with an acoustical tone being created with the same frequency as the velocity change due to the vortices. However, no direct explanation has been found for the frequency discrepancy compared to the theoretical value. Possible explanations that would need further investigations include wall interaction, reflections at the outlet or due to a poorly tuned sponge layer, and the staircase approximation of the cylinder with a small radius in lattice units.

4.4.2 Stability analysis

Table 4.1 shows the settings used for the stable simulations with the highest Reynolds number for all resolutions and Mach numbers. It can be seen that, as increasing the resolution allows for a raised Reynolds number with both ν^* and the Mach number preserved, the stability limit increases with resolution. However, it also seems that higher resolution leads to a narrower stability demand, since increasing D from $10\Delta x$ to $20\Delta x$ necessitates a higher ν^*, with $D = 40\Delta x$ not being stable at $\mathrm{Ma} = 0.1$ for any Reynolds number tested (that is, for any $\mathrm{Re} \geq 100$).

Equation 2.45 was also used to calculate the minimum ν^* required by the BGK model based on previous research. It is here clearly noted that TRT allows for a lower ν^* than BGK would have, but more data would be necessary in order to draw

any direct conclusions about the stability limits for BGK.

Table 4.1: The settings used for stable von Kárman vortex simulations, as well as the minimum ν^* for stable BGK simulations from equation 2.45. Lower Reynolds numbers than the ones showed here where also tested, but are omitted from the results as these also where stable.

Ma	Re	D	ν^*	Minimum BGK ν^*	τ^{*+}
0.05	400	$10\Delta x$	$7.21 \cdot 10^{-4}$	$1.20 \cdot 10^{-3}$	0.5022
0.10	400	$10\Delta x$	$1.44 \cdot 10^{-3}$	$2.40 \cdot 10^{-3}$	0.5043
0.05	700	$20\Delta x$	$8.25 \cdot 10^{-4}$	$1.20 \cdot 10^{-3}$	0.5025
0.10	600	$20\Delta x$	$1.92 \cdot 10^{-3}$	$2.40 \cdot 10^{-3}$	0.5058
0.05	1400	$40\Delta x$	$8.25 \cdot 10^{-4}$	$1.20 \cdot 10^{-3}$	0.5025

4.5 Future development and further tests

The results highlight a few issues with the code that need direct improvement. Clearly there is a need to improve the forcing scheme for TRT such that it can produce accurate results and it would be good to further understand the behaviour of the monopole source implementation such that different frequencies can be used at the same strength in an artificial sound source. As several errors might depend on reflections from the boundary (the frequency response of vortex shedding as well as the velocity profile of Poiseuille flow) there is also a need for better non-reflecting BCs. While more tuning tests of sponge layers might lead to better future results, there is most likely necessary to also implement the characteristic BCs briefly mentioned in section 2.7.4. The indications that the Couette flow simulations aren't mass conserving, while the Poiseuille flow might be, suggest that either the periodic BCs, or HBB with a tangential prescribed velocity has problems conserving mass, which would need to be investigated and improved upon.

Apart from these improvements that intend fixing errors there are also many opportunities for improvement. As high Reynolds numbers and low viscosities are typical for aero-acoustic flows in every-day fluids it is important to further investigate the limits of stability in the current code, implement other collision operators with better stability (such as MRT), and increase performance of the code at a certain resolution (such that the resolution can be raised when needed for stability considerations). To be able to accurately simulate flows with higher Reynolds numbers there is a need to implement turbulence models. The code should also be expanded to three dimensions, which will increase the necessity of better performance.

The code performance could most likely be heavily increased by changing some minor parts of the code that are memory-inefficient and changing the parallelization to a proper parallelization running on the GPU, instead of relying on numba's CPU parallelization. Other possible code improvements would be a wider choice of forcing schemes and BCs.

5
Conclusion

It can be concluded that the code written during this thesis project demonstrates that LBM is a working option for aero-acoustic simulations. However, as noted in section 4.5,he code produced during this project, is not in a state that it can be used efficiently in any applied application and further development would be necessary to achieve that. The most important improvements would probably include:

- A proper forcing scheme for the TRT collision operator.
- Non-reflective BCs, e.g. characteristic BCs.
- MRT collision operator and a related forcing scheme.
- Parallelization on the GPU.
- A 3D lattice.
- A turbulence model.

Another conclusion is that while LBM works as a way to simulate aero-acoustics, there are still many sub-fields that require more research. For example, section 2.7.3 discusses e.g. attenuation of sound waves due to molecular effects, which would have to be modeled for LBM to accurately describe real world acoustics.

To summarize the report LBM contains much of the physics of aero-acoustics already in its derivation, but real world acoustics can yet be tricky to simulate as the Reynolds numbers are high and the viscosities are low, and several physical phenomena are not yet commonly implemented in LBM.

Bibliography

[1] Timm Krüger, Halim Kusumaatmaja, Alexandr Kuzmin, Orest Shardt, Goncalo Silva, and Erlend Magnus Viggen. *The Lattice Boltzmann Method: Principles and Practice*. Springer International Publishing, 2017. ISBN: 9783319446493. DOI: 10.1007/978-3-319-44649-3. URL: http://dx.doi.org/10.1007/978-3-319-44649-3.

[2] Johan Augusto Bocanegra, Mario Misale, and Davide Borelli. "A systematic literature review on Lattice Boltzmann Method applied to acoustics". In: *Engineering Analysis with Boundary Elements* 158 (Jan. 2024), pp. 405–429. ISSN: 0955-7997. DOI: 10.1016/j.enganabound.2023.11.007. URL: http://dx.doi.org/10.1016/j.enganabound.2023.11.007.

[3] Leonie S. Föhring, Peter Møller Juhl, and Dietrich Wittekind. "Experimental Parameters Influencing the Cavitation Noise of an Oscillating NACA0015 Hydrofoil". In: *Journal of Marine Science and Engineering* 11.10 (Oct. 2023), p. 2023. ISSN: 2077-1312. DOI: 10.3390/jmse11102023. URL: http://dx.doi.org/10.3390/jmse11102023.

[4] Haidong Yuan, Zhigang Yang, Yigang Wang, Yajun Fan, and Yuhao Fang. "Experimental analysis of hydrodynamic and acoustic pressure on automotive front side window". In: *Journal of Sound and Vibration* 476 (June 2020), p. 115296. ISSN: 0022-460X. DOI: 10.1016/j.jsv.2020.115296. URL: http://dx.doi.org/10.1016/j.jsv.2020.115296.

[5] Erin Flanagan, Kristoffer Mattisson, Anna Oudin, Susanna Gustafsson, and Ebba Malmqvist. "Health impact assessment of exposure to road traffic noise and air pollution according to pre- and post-densification scenarios in Helsingborg, Sweden". In: *City and Environment Interactions* 24 (Dec. 2024), p. 100176. ISSN: 2590-2520. DOI: 10.1016/j.cacint.2024.100176. URL: http://dx.doi.org/10.1016/j.cacint.2024.100176.

[6] M. Ottersten, H.D. Yao, and L. Davidson. "Inlet gap effect on aerodynamics and tonal noise generation of a voluteless centrifugal fan". In: *Journal of Sound and Vibration* 540 (Dec. 2022), p. 117304. ISSN: 0022-460X. DOI: 10.1016/j.jsv.2022.117304. URL: http://dx.doi.org/10.1016/j.jsv.2022.117304.

[7] Xiaowen Shan, Xue-Feng Yaúan, and Hudong Chen. "Kinetic theory representation of hydrodynamics: a way beyond the Navier–Stokes equation". In: *Journal of Fluid Mechanics* 550.1 (Feb. 2006), p. 413. ISSN: 1469-7645. DOI: 10.1017/s0022112005008153. URL: http://dx.doi.org/10.1017/S0022112005008153.

[8] Zhaoli Guo, Chuguang Zheng, and Baochang Shi. "Discrete lattice effects on the forcing term in the lattice Boltzmann method". In: *Physical Review E*

65.4 (Apr. 2002). ISSN: 1095-3787. DOI: 10.1103/physreve.65.046308. URL: http://dx.doi.org/10.1103/PhysRevE.65.046308.

[9] X. D. Niu, C. Shu, Y. T. Chew, and T. G. Wang. "Investigation of Stability and Hydrodynamics of Different Lattice Boltzmann Models". In: *Journal of Statistical Physics* 117.3–4 (Nov. 2004), pp. 665–680. ISSN: 1572-9613. DOI: 10.1007/s10955-004-2264-x. URL: http://dx.doi.org/10.1007/s10955-004-2264-x.

[10] Goncalo Silva. "Discrete effects on the forcing term for the lattice Boltzmann modeling of steady hydrodynamics". In: *Computers amp; Fluids* 203 (May 2020), p. 104537. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2020.104537. URL: http://dx.doi.org/10.1016/j.compfluid.2020.104537.

[11] Qisu Zou and Xiaoyi He. "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model". In: *Physics of Fluids* 9.6 (June 1997), pp. 1591–1598. ISSN: 1089-7666. DOI: 10.1063/1.869307. URL: http://dx.doi.org/10.1063/1.869307.

[12] X. Xue, H.D. Yao, and L. Davidson. "Wall-modeled large-eddy simulation integrated with synthetic turbulence generator for multiple-relaxation-time lattice Boltzmann method". In: *Physics of Fluids* 35.6 (June 2023). ISSN: 1089-7666. DOI: 10.1063/5.0153526. URL: http://dx.doi.org/10.1063/5.0153526.

[13] Emil Ellénius. "Lattice-Boltzmann for Aeronautical Flows: An introduction to and evaluation of the Lattice-Boltzmann Method". MA thesis. City, Country: Chalmers University of Technology, 2024.

[14] L.D. Landau and E.M. Lifshitz. "SOUND". In: *Fluid Mechanics*. Elsevier, 1959, pp. 245–309. ISBN: 9780080291420. DOI: 10.1016/b978-0-08-029142-0.50013-1. URL: http://dx.doi.org/10.1016/B978-0-08-029142-0.50013-1.

[15] R.D. Blevins. "Review of sound induced by vortex shedding from cylinders". In: *Journal of Sound and Vibration* 92.4 (Feb. 1984), pp. 455–470. ISSN: 0022-460X. DOI: 10.1016/0022-460x(84)90191-3. URL: http://dx.doi.org/10.1016/0022-460X(84)90191-3.

[16] J.M. McCarthy, S. Watkins, A. Deivasigamani, and S.J. John. "Fluttering energy harvesters in the wind: A review". In: *Journal of Sound and Vibration* 361 (Jan. 2016), pp. 355–377. ISSN: 0022-460X. DOI: 10.1016/j.jsv.2015.09.043. URL: http://dx.doi.org/10.1016/j.jsv.2015.09.043.

[17] Li-Shi Luo, Wei Liao, Xingwang Chen, Yan Peng, and Wei Zhang. "Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations". In: *Physical Review E* 83.5 (May 2011). ISSN: 1550-2376. DOI: 10.1103/physreve.83.056710. URL: http://dx.doi.org/10.1103/PhysRevE.83.056710.

[18] *Air - Speed of Sound vs. Temperature — engineeringtoolbox.com*. https://www.engineeringtoolbox.com/air-speed-sound-d_603.html. [Accessed 22-05-2025].

[19] Engineers Edge. *Viscosity of Air, Dynamic and Kinematic — engineersedge.com*. https://www.engineersedge.com/physics/viscosity_of_air_dynamic_and_kinematic_14483.htm. [Accessed 2025-04-10].

[20] *The Fourier Series — setzeus.com*. https://www.setzeus.com/community-blog-posts/the-fourier-series. [Accessed 2025-05-22].

[21] Sharon Nieukirk. *Understanding Ocean Acoustics*. https://oceanexplorer.noaa.gov/explorations/sound01/background/acoustics/acoustics.html. [Accessed 2025-05-26].

A

Omitted monopole results

The monopole simulations with $\varphi = 440\,\text{Hz}$ and sponge layer maximum viscosities of $\nu^* = 35$ and $\nu^* = 75$ were omitted from chapter 4 in order to keep the total amount of graphs down. However, comparing these results to figure 4.10 is useful in finding the optimal viscosity in the sponge layer. The only method used to decide on $\nu^* = 50$ was optical inspection, but it is much clearer that the optimal ν^* isn't 20 or 100 than that it should be 50 instead of 35 or 75.

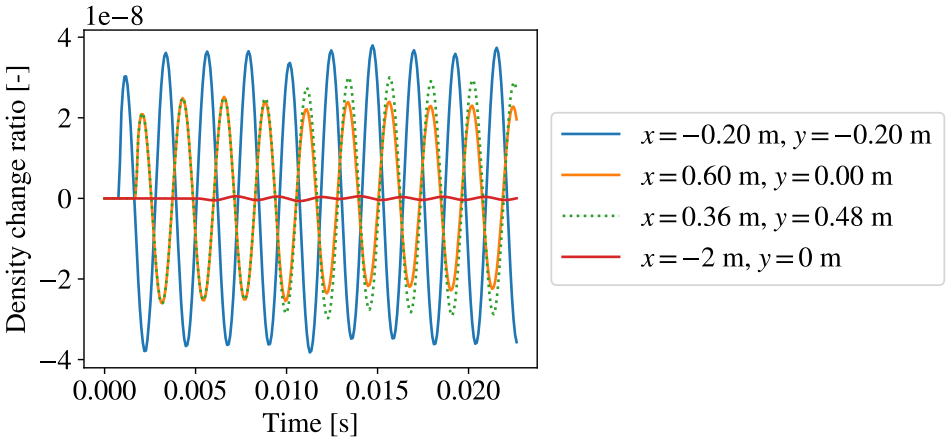

Figure A.1: The density difference from the initial state of $\rho^* = 1$ as a function of time for a 440 Hz sinusoidal wave originating from the origin using a 500 node sponge layer with maximum $\nu^* = 35$, measured at four points in the domain. Note that $x = 0.60\,\text{m}, y = 0\,\text{m}$ (orange) and $x = 0.36\,\text{m}, y = 48\,\text{m}$ (green) have the exact same radii from the monopole sound source, while the $x = -2\,\text{m}, y = 0\,\text{m}$ point (red) is located at the boundary, where the sponge layer has dampened most of the amplitude.

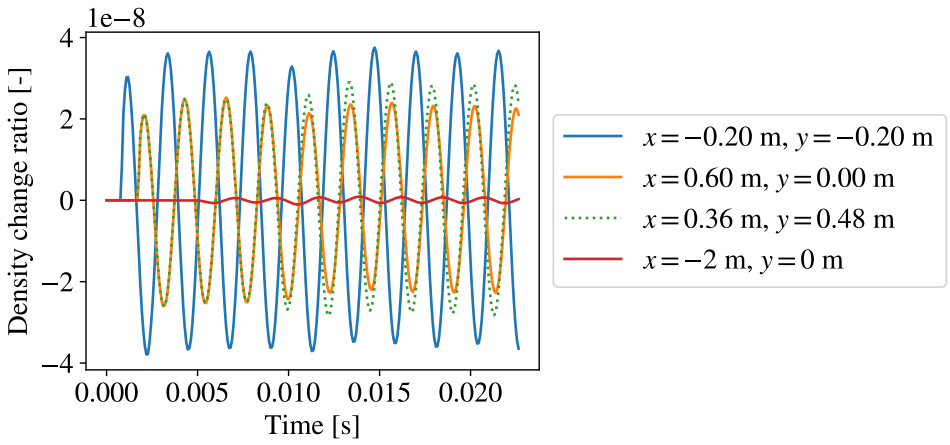

Figure A.2: The density difference from the initial state of $\rho^* = 1$ as a function of time for a 440 Hz sinusoidal wave originating from the origin using a 500 node sponge layer with maximum $\nu^* = 75$, measured at four points in the domain. Note that $x = 0.60\,\text{m}, y = 0\,\text{m}$ (orange) and $x = 0.36\,\text{m}, y = 48\,\text{m}$ (green) have the exact same radii from the monopole sound source, while the $x = -2\,\text{m}, y = 0\,\text{m}$ point (red) is located at the boundary, where the sponge layer has dampened most of the amplitude.

B

Vortex shedding images at Reynolds number 1400

This chapter showcases every 10000th time step of the Re = 1400 von Kárman vortex simulation with Ma = 0.05. The x-component of velocity is plotted in lattice units.

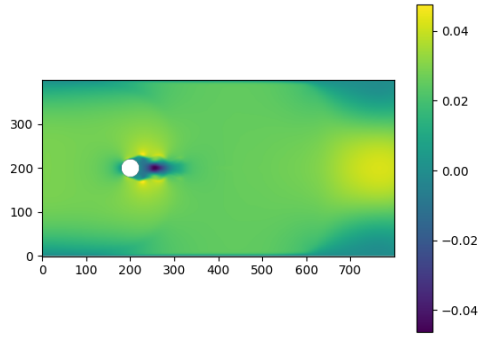

Figure B.1: Time step is $t = 10000\Delta t$

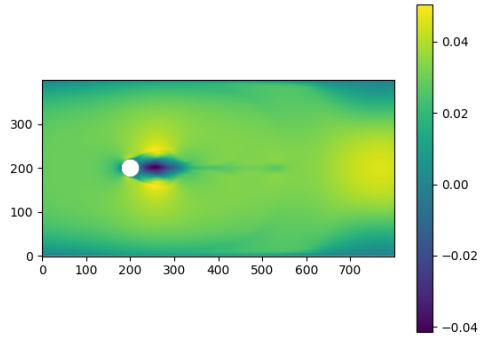

Figure B.2: Time step is $t = 20000\Delta t$

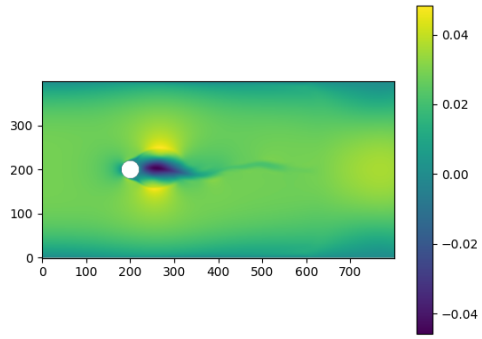

Figure B.3: Time step is $t = 30000\Delta t$

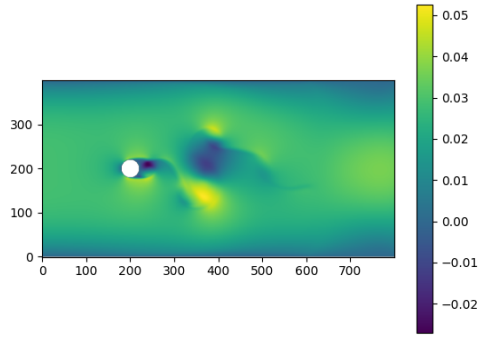

Figure B.4: Time step is $t = 40000\Delta t$

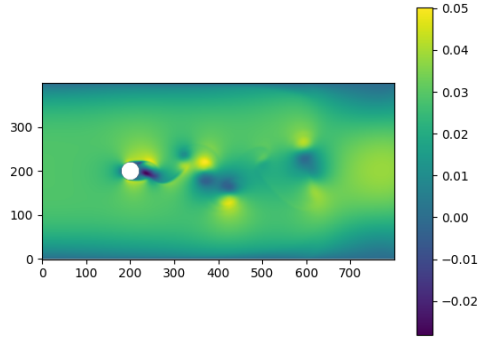

Figure B.5: Time step is $t = 50000\Delta t$

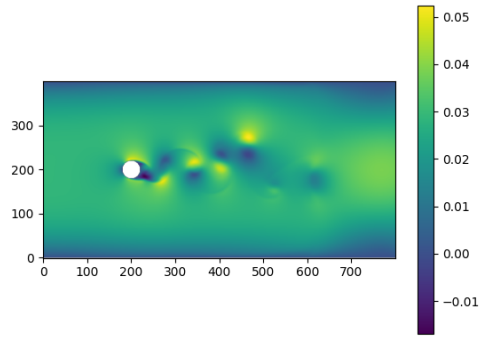

Figure B.6: Time step is $t = 60000\Delta t$

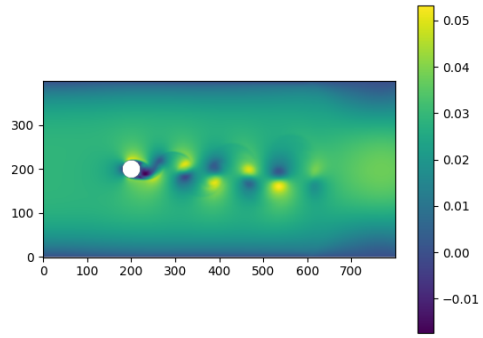

Figure B.7: Time step is $t = 70000\Delta t$

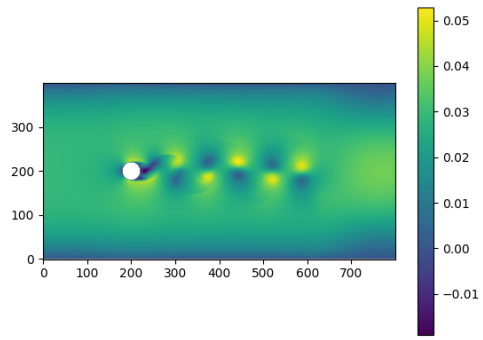

Figure B.8: Time step is $t = 80000\Delta t$

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY