The Lattice Boltzmann Method with Applications in Acoustics

Erlend Magnus Viggen

Department of Physics — NTNU 2009

Thesis description

Student:	Erlend Magnus Viggen
Written at:	Department of Physics, NTNU
Period:	January 20 – June 16, 2009
Internal supervisor:	Alex Hansen, Department of Physics, NTNU
External supervisor:	Ulf Kristiansen, Department of Electronics and Telecommunications, NTNU
English title:	The Lattice Boltzmann Method with Applications in Acoustics
Norwegian title:	Lattice Boltzmann-Metoden med Anvendelser innen Akustikk

Abstract in English

An introduction is given to the lattice Boltzmann method and its background, with a view towards acoustic applications of the method. To make a larger range of acoustic applications possible, a point source method is proposed. This point source is applied to simulate cylindrical waves and plane waves, and is shown to give a very good numerical result compared with analytic solutions of viscously damped cylindrical and plane waves. Good results are found for simulations of Doppler effect, diffraction, and viscously damped standing waves.

It is concluded that the lattice Boltzmann method could be suitable for simulating acoustics in complex flows, at ultrasound frequencies and very small spatial scales. The lattice Boltzmann method is shown to be unfeasible at lower frequencies or for larger systems.

Abstract in Norwegian

En introduksjon gis til lattice Boltzmann-metoden og dens bakgrunn, med henblikk på akustiske anvendelser av metoden. For å muliggjøre flere forskjellige akustiske anvendelser foreslås en punktkildemetode. Denne punktkilden anvendes for å simulere sylinder- og planbølger, og det vises at den gir et svært bra numerisk resultat sammenlignet med analytiske løsninger av viskøst dempede sylinder- og planbølger. Gode resultater finnes for simuleringer av Dopplereffekt, diffraksjon og viskøst dempede stående bølger.

Det konkluderes med at lattice Boltzmann-metoden kan være passende for å simulere akustikk i komplekse strømninger, ved ultralydfrekvenser og svært små romlige skalaer. Lattice Boltzmann-metoden vises å være upraktisk ved lavere frekvenser og for større systemer.

Preface

The thought of writing my master's thesis on the possibility of acoustics simulations in the lattice Boltzmann method came about when I was discussing possible thesis subjects with my advisor, Ulf Kristiansen. We had already agreed that I would write my master thesis on something in the field of numerical acoustics, but we had not decided on a final subject.

While we were both familiar with lattice gases, none of us were aware that the field of lattice gases had morphed into the field of lattice Boltzmann some time ago. When we discovered this, we decided that it would be interesting to see how well this method for fluid simulations would be suited to perform acoustics simulations. After all, sound waves are special cases of fluid behaviour, right?

I had not touched fluid mechanics since the introductory course I took on it three years ago. I had forgotten quite a lot since then, but this thesis gave me the opportunity to relearn a lot of interesting fluid mechanics theory.

As mentioned, I had no previous knowledge of the lattice Boltzmann method before starting with this thesis, but I quickly discovered that it was a very interesting field. One of the advantages of the method is its relative simplicity, which meant that I was up and running in a reasonably short time. Within the first week of learning about the method, I had working lattice Boltzmann code.

Still, I wasted some time in the beginning barking up the wrong tree. For instance, I was confused by several papers stating that the lattice Boltzmann method gives incompressible flow behaviour. If the method was restricted to incompressible flow, it would mean that simulations of sound wave propagation would be impossible, as sound is a phenomenon of compressible flow. It turned out I had misconstrued the statement — the method gives a behaviour consistent with compressible flow, of which incompressible flow is a special case. Incompressible flow is merely the most popular special case of the lattice Boltzmann method.

In writing this master's thesis, I have tried to comply with the guidelines from my department. Specifically, I have tried to make this text accessible to my peers, meaning other fifth-year university students of physics. The unfortunate side-effect of this is that I spend a lot of time explaining theory which should be familiar for people with particular experience with the subjects discussed here.

I would like to thank all the people who have helped me in this work. First, I would like to thank Joris Verschaeve, who is a PhD student at the fluids engineering group at

NTNU's Department of Energy and Process Engineering. Joris has been very helpful in explaining the finer points of the lattice Boltzmann method to me.

I would like to thank the community at LBMethod.org, in particular Jonas Lätt and Orestis Malaspinas. This site has been a very useful repository of knowledge for me, and I have received help at its forums when it was needed. Jonas has also been very helpful with questions about his PhD thesis, which has been one of my main sources.

Finally, I would like to thank my advisors, for discussions and follow-up questions, and for letting me write about such an interesting subject.

Contents

CHAPTER 1

Introduction

Everything should be made as simple as possible, but no simpler.

— *Attributed to Albert Einstein*

The lattice Boltzmann method is a new and promising method in computational fluid dynamics. The method has evolved from the older method of lattice gas automata, and was first proposed in 1988. Among the method's advantages are great ease of implementation, near-infinite potential for parallelisation, and a nature which makes expansions of the method quite simple.

Its ancestor, the method of lattice gas automata, was one of the more successful cellular automata, with a history going as far back as 1973. A cellular automaton is a discrete computer model of a grid system with an evolution described by mathematical rules. The lattice gas method was an attempt to model the motion of single particles in a fluid using behavioural rules which were as simple as possible (but no simpler — the rules still obeyed basic physics). Thus, lattice gas automata can be seen as an exceedingly simple method of molecular dynamics.

Restricting the particles' motion to nodes in a hexagonal grid and giving simple rules for how they should collide turned out to be enough to achieve behaviour conforming to the equation of continuity and the Navier-Stokes equation. These equations govern the flow of all dense fluids.

The lattice Boltzmann method was introduced in 1988 as a way to avoid certain weaknesses of lattice gases, while retaining their distinct advantages: Parallelisation and simplicity. Instead of handling single particles, the lattice Boltzmann method handles particle distributions and treats collision in a different manner than lattice gas automata.

Otherwise, the two methods are very similar. The most important similarity is that the lattice Boltzmann method also gives a behaviour corresponding to the equations governing fluid flow. When it comes to simplicity, lattice gases are simpler in principle, but the lattice Boltzmann method is more simple in implementation and usage.

While the basic lattice Boltzmann method is used to simulate the behaviour of compressible and incompressible isothermal fluids, many models exist which expand it in different ways, enabling it to simulate complex fluids, thermohydrodynamics, magnetohydrodynamics, and so forth.

Parallelisation

Many computational problems can be broken down into pieces which are run side-by-side on several processor cores simultaneously, instead of being run one-after-the-other on a single core. Problems which can be run in parallel in this manner are known as parallelisable problems.

The parallelisability of computer algorithms is becoming increasingly important today, as the last years' trend in processors has been to put more CPUs on each processor instead of drastically improving the speed of each CPU. In addition, the emergence of general-purpose graphics processing units, which can be looked at as massively parallel processors, offers exciting new opportunities for rapid calculation of parallelisable problems.

The lattice Boltzmann method can be classified as an *embarassingly parallel* computational problem. This is a term for problems which are particularly simple to parallelise, and where the speed-up in calculation is nearly linear with the number of processor cores used. The reason for the lattice Boltzmann's simple parallelisation is that operations on the grid are local, so that each node can be updated independently of others.

Scopes

Physical systems can be simulated at different scales. We order these scales into three different scopes:

Microscopic scope: Simulations of the behaviour of single particles interacting with each other through fields or collisions. Since simulations in this scope can only be of one specific configuration of particles at a time, care must be taken when attempting to predict larger-scale behaviour from a microscopic result. This is the scope used in molecular dynamics methods, which include the lattice gas method.

Mesoscopic scope: Simulations of particle distributions, averaged over a great number of particles. More fine detail is represented in this scope than in the macroscopic scope, but it avoids some of the problems inherent in the microscopic scope. This is the scope used in the lattice Boltzmann method.

Macroscopic scope: Simulations of a continuum. No physical systems are actual continuums, but with a great enough number of particles, the microstate of the system becomes irrelevant and the continuum approximation is a good one. This is the scope used in most branches of classical physics, such as acoustics and fluid mechanics. For this reason, the finite difference and finite element methods, which discretise physical equations, normally use a macroscopic scope.

One can learn about the larger-scale behaviour a system from simulations in smaller scopes, although this is not practical in many cases. This can be done for instance by averaging over space, time, or several similar but different initial conditions. It is impossible to go the other way, as information about particle movements is lost in the transition to a larger scope.

Lattice Boltzmann acoustics

So far, little work has been done on lattice Boltzmann acoustics. Some papers have been written analysing linear and non-linear acoustic propagation in the method by letting

pre-initialised sound waves propagate. This approach is quite limited in applicability, and cannot be used to estimate transient responses, to be used for instance in estimating transfer functions.

It should not be surprising that the lattice Boltzmann method gives behaviour according to the wave equation, and that it can be used to simulate acoustics. After all, the compressible Navier-Stokes equation can be simulated using the lattice Boltzmann method, and the wave equation can be derived from compressible Navier-Stokes.

Of course, for pure wave equation simulations, there will probably be more effective numerical methods. The lattice Boltzmann method's strength is being a full Navier-Stokes solver, which means that it can be used to simulate non-linear acoustics in complex flows.

1.1 Available resources

Literature

Although the field of lattice Boltzmann methods is still young, several books have been written on this method in general. As these vary quite a lot in their targets and presentation of the method, it is useful with an overview.

Cellular Automata Modelling of Physical Systems
by B. Chopard and M. Droz, 1998. [1]
While this book is about different kinds of cellular automata that can model physical systems, it also contains some information about lattice gas automata and the lattice Boltzmann method.

The Lattice Boltzmann Equation for Fluid Dynamics and Beyond
by S. Succi, 2001. [2]
This was the first book entirely dedicated to the lattice Boltzmann method. It gives a fairly complete picture of the state of the lattice Boltzmann field in 2001. Its style is somewhat unusual, as its notation and its explanations differ from most of the field.

Lattice-Gas Cellular Automata and Lattice Boltzmann Models
by D. Wolf-Gladrow, 2005. [3]
This book is part of Springer's "Lecture Notes in Mathematics" series. It gives approximately equal weight to lattice gas automata and the lattice Boltzmann method, and focuses on theoretical analysis of the method.

Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers *by M. C. Sukop and D. T. Thorne, 2006.* [4]
This book gives a very simple introduction to the lattice Boltzmann method, with later chapters describing how to implement more complex fluid models. Its scope is unfortunately somewhat limited, but it is an easily digested introduction to the field.

Review articles: In addition to these books, a number of review articles for the lattice Boltzmann method have been written, [5–9] with more to be published soon. [10]

Community

There is an active community around the documentation project at `http://lbmethod.org`. This is a wiki containing overviews of the lattice Boltzmann models, details of selected lattice Boltzmann aspects, lattice Boltzmann galleries, example codes in several

different programming languages, and an overview of important literature in the field. There is also an active forum connected to this site, with discussions on theory and implementation of the lattice Boltzmann method.

An actively-developed open-source lattice Boltzmann software library for C++ is available at `http://openlb.org`. This library is designed to be modular and well-optimized for both single-processor and parallel applications. It also incorporates many different variations of the lattice Boltzmann model.*

1.2 Thesis structure

This thesis attempts to follow a logical structure: First we have some theoretical background, which is used to build up the method's theory piece by piece. Then we apply the basic method, and discuss expansions of the method. In order, the chapters and appendices are:

Chap. 1 **Introduction:** This chapter.

Chap. 2 **Theory:** An overview of the notation used, together with some theoretical background of fluid mechanics and acoustics.

Chap. 3 **Lattice gas automata:** An overview of lattice gas automata, which is very useful for understanding the lattice Boltzmann method.

Chap. 4 **The lattice Boltzmann method:** Details of the lattice Boltzmann method.

Chap. 5 **Lattice Boltzmann boundary conditions:** More complex boundary conditions, which enable useful simulations.

Chap. 6 **Simulations:** Simulations of acoustic behaviour using the lattice Boltzmann method.

Chap. 7 **Alternative lattice Boltzmann models:** Variations of the lattice Boltzmann method which perform tasks that the basic model cannot.

Chap. 8 **Discussion:** A discussion of the results found and of what remains to be done.

Chap. 9 **Conclusion:** A summary of what has been achieved in this thesis.

App. A **Derivation of Navier-Stokes from LBM dynamics:** Derives the compressible Navier-Stokes equation from the lattice Boltzmann method.

App. B **Miscellaneous derivations:** Shorter derivations, placed in an appendix to avoid breaking the flow of the text elsewhere.

App. C **Code:** Code snippets used to prove points made elsewhere in the text.

If the reader wishes to skip straight to a succinct description of the lattice Boltzmann method, section 4.6 is written to serve as such.

Note that this thesis is intended to give a fairly complete overview of the lattice Boltzmann method. As its goal is to estimate how useful the method is for acoustics simulations while remaining accessible to laymen, this is a necessity. The *new* work that has been done is largely based around the acoustic point source introduced in section 5.4, the potential of which is demonstrated in Chapter 6.

*Note that the code used in simulations in this project is not based on OpenLB, as the author thought writing his own code would facilitate his understanding of the lattice Boltzmann method.

CHAPTER 2

Theory

For this text to be accessible to readers with a variety of backgrounds, it is essential to go through some theoretical fundamentals first.

2.1 Tensors and index notation

We define a tensor as a multidimensional generalisation of vectors and matrices. It can be seen as an array with n dimensions, which stores numbers which can be retrieved with n indices. For instance, a vector can be seen as a one-dimensional tensor, since all the elements in the vector can be retrieved with one index. Similarly, a matrix is a two-dimensional tensor, since all elements in the matrix can be retrieved with two indices.*

In the scientific field of the lattice Boltzmann method, a form of index notation reminiscent of Einstein notation is often used. Greek letters are used as indices to point to elements in a tensor, and *whenever a greek letter index is repeated in the same term, it implies a sum over that index.*

For instance, the scalar product between the vectors $\vec{a}$ and $\vec{b}$ can be written as

$$\vec{a} \cdot \vec{b} = a_\alpha b_\alpha.$$

The left side of this equation is the classical vector notation. The right side is in index notation. Index notation implies that since α is repeated,

$$a_\alpha b_\alpha = \sum_\alpha a_\alpha b_\alpha.$$

The right side of this equation is the familiar definition of the scalar product of two vectors.

In this text, a vector will be written with a vector arrow (e.g. $\vec{a}$), a higher-than-one-dimensional tensor in bold lettering (e.g. $\mathbf{A}$), and a scalar with no particular embellishments (e.g. a).

*Unfortunately, there is no universally accepted definition of a tensor. It is worth mentioning, though, that this definition of a n-dimensional tensor corresponds to another common definition of a rank 1 tensor of order n.

We will also write the partial derivative in a form which may not be familiar to all readers. We let

$$\frac{\partial}{\partial t} \to \partial_t,$$

and

$$\frac{\partial}{\partial x_\alpha} \to \partial_\alpha,$$

where x_α is the αth Cartesian coordinate.

A list of several common tensor operations is given in Table 2.1. Most of these should be familiar, but a few deserve particular mention.

Table 2.1: Several common tensor operations in tensor and index notation. Largely taken from Jonas Lätt's PhD thesis. [11]

Operation	**Tensor notation**	**Index notation**
Scalar product	$\lambda = \vec{a} \cdot \vec{b}$	$\lambda = a_\alpha b_\alpha$
Cross product	$\vec{c} = \vec{a} \times \vec{b}$	$c_\alpha = \varepsilon_{\alpha\beta\gamma} a_\beta b_\gamma$
Tensor contraction	$\lambda = \mathbf{A} : \mathbf{B}$	$\lambda = A_{\alpha\beta} B_{\alpha\beta}$
Dyadic vector product	$\mathbf{A} = \vec{a}\vec{b}$	$A_{\alpha\beta} = a_\alpha b_\beta$
Gradient	$\vec{a} = \overrightarrow{\nabla} \lambda$	$a_\alpha = \partial_\alpha \lambda$
Divergence of vector (1D tensor)	$\lambda = \overrightarrow{\nabla} \cdot \vec{a}$	$\lambda = \partial_\alpha a_\alpha$
Divergence of matrix (2D tensor)	$\vec{a} = \overrightarrow{\nabla} \cdot \mathbf{A}$	$a_\alpha = \partial_\beta A_{\alpha\beta}$
Divergence of 3D tensor	$\mathbf{A} = \overrightarrow{\nabla} \cdot \mathbf{T}$	$A_{\alpha\beta} = \partial_\gamma T_{\alpha\beta\gamma}$
Double-divergence of 3D tensor	$\vec{a} = \overrightarrow{\nabla}\overrightarrow{\nabla} : \mathbf{T}$	$a_\alpha = \partial_\beta \partial_\gamma T_{\alpha\beta\gamma}$

Tensor contraction when applied on two matrices means that the two matrices are multiplied together element for element and the sum over all elements of the resulting matrix is taken, resulting in a scalar.

The dyadic vector product is the matrix multiplication of a column vector and a row vector resulting in a matrix.

The index formulation of the cross product uses the Levi-Civita symbol, which can be seen as a three-dimensional tensor. It is constructed from the parity of permutation of the indices in such a way that

$$\varepsilon_{\alpha\beta\gamma} = \begin{cases} +1 & \text{if } (\alpha,\beta,\gamma) \text{ is } (1,2,3),(3,1,2) \text{ or } (2,3,1), \\ -1 & \text{if } (\alpha,\beta,\gamma) \text{ is } (3,2,1),(1,3,2) \text{ or } (2,1,3), \\ 0 & \text{otherwise: } \alpha=\beta \text{ or } \beta=\gamma \text{ or } \gamma=\alpha. \end{cases}$$

2.1.1 Useful examples

Index notation is useful for performing a variety of derivations in linear algebra. We will here show some linear algebra relations using index notation, both as examples and for use later.

The product of two scalar products can be rewritten as a tensor contraction like

$$\left(\vec{a} \cdot \vec{b}\right)\left(\vec{c} \cdot \vec{d}\right) = (a_\alpha b_\alpha)(c_\beta d_\beta) = (a_\alpha c_\beta)(b_\alpha d_\beta) = \vec{a}\vec{c} : \vec{b}\vec{d}. \tag{2.1}$$

Note that the parentheses in the index notation serve no purpose other than clarifying how we have moved between tensor and index notation. Note also how $\vec{a}\vec{d} : \vec{b}\vec{c}$ would be an equally valid result.

A constant vector $\vec{a}$ multiplied with the divergence of another vector $\vec{b}$ can be rewritten as

$$\vec{a}\left(\overrightarrow{\nabla} \cdot \vec{b}\right) = a_\alpha \left(\partial_\beta b_\beta\right) = \partial_\beta \left(a_\alpha b_\beta\right) = \overrightarrow{\nabla} \cdot \vec{a}\vec{b}. \tag{2.2}$$

The first order Taylor expansion of a function of a vector, $f = f(\vec{x}) = f(x_1, x_2, \ldots)$, can be written as

$$f\left(\vec{x} + \vec{c}\right) \approx f\left(\vec{x}\right) + c_\alpha \partial_\alpha f\left(\vec{x}\right) = f\left(\vec{x}\right) + \overrightarrow{\nabla} \cdot \vec{c} f\left(\vec{x}\right), \tag{2.3}$$

assuming that $\vec{c}$ does not vary with the variables x_i.

The identity matrix $\mathbf{I}$ has components $I_{\alpha\beta} = \delta_{\alpha\beta}$, where $\delta_{\alpha\beta}$ is the Kronecker delta. From this, we can find the value of a dyadic vector product $\vec{a}\vec{b}$ tensor contracted with the identity matrix, which is

$$\vec{a}\vec{b} : \mathbf{I} = a_\alpha b_\beta \delta_{\alpha\beta} = a_\alpha b_\alpha = \vec{a} \cdot \vec{b}. \tag{2.4}$$

2.2 Fluid mechanics and Navier-Stokes

We can describe the state of a simple isothermal fluid using the macroscopic variables of fluid density ρ, flow velocity $\vec{u}$, and pressure p.

Other important quantities are the medium's speed of sound c_s, its dynamic shear viscosity μ and its dynamic bulk viscosity μ' (sometimes also known as the volume viscosity or second viscosity). In monatomic ideal gases, it can be derived from kinetic theory that $\mu' = 0$. [12, 13]

In an isothermal ideal gas, the pressure is related to the density as [11, 12]

$$p = c_s^2 \rho. \tag{2.5}$$

The behaviour of a fluid always conforms to [11, 12, 14]

$$\boxed{\partial_t \rho + \overrightarrow{\nabla} \cdot (\rho \vec{u}) = 0,} \tag{2.6}$$

which is known as the *continuity equation*, and [14]

$$\boxed{\rho\left(\partial_t \vec{u} + \left(\vec{u} \cdot \overrightarrow{\nabla}\right)\vec{u}\right) + \overrightarrow{\nabla} p - \mu \nabla^2 \vec{u} - \left(\frac{\mu}{3} + \mu'\right)\overrightarrow{\nabla}\left(\overrightarrow{\nabla} \cdot \vec{u}\right) = \vec{f},} \tag{2.7}$$

which is known as the *compressible Navier-Stokes equation*. In this equation, $\vec{f}$ represents external body forces, given by

$$\vec{f} = \frac{d\vec{F}}{dV} = \rho \vec{a},$$

where $\vec{F}$ is the Newtonian force, $\vec{a}$ is the acceleration caused by the force, and V a volume. This force and acceleration can come from any source, but it is usually gravitational in nature.

Both equations express conservation of a quantity. The continuity equation expresses conservation of mass, while the the Navier-Stokes equation expresses conservation of momentum. [11]

It is important to note that the Navier-Stokes equation can be formulated in other ways. In some parts of the literature, different formulations are used. [11, 12] One such common formulation is shown in Appendix B.1 to be equivalent to the one in equation 2.7.

Equation 2.7 describes the behaviour of *compressible* flow. This means that the density ρ can vary a great deal. Wave propagation, for instance, is a phenomenon of compressible flow.

Some *incompressible* fluids, such as water, have a nearly constant density. This enables us to take the simplifying assumption that

$$\rho = \rho_0 = \text{const.},$$

which lets us simplify the continuity equation (2.6) to

$$\overrightarrow{\nabla} \cdot \vec{u} = 0, \tag{2.8}$$

which we can appropriately call the *incompressible continuity equation.*

We can use this to remove a term in equation 2.7, giving us

$$\rho_0 \left(\partial_t \vec{u} + \left(\vec{u} \cdot \overrightarrow{\nabla} \right) \vec{u} \right) + \overrightarrow{\nabla} p - \mu \nabla^2 \vec{u} = \vec{f}, \tag{2.9}$$

which is known as the *incompressible Navier-Stokes equation*, a useful simplification of the compressible Navier-Stokes equation. It is used for calculating flow in incompressible fluids such as water, where the effects of compression are negligible. Of course, if there can be no compression in the fluid, there can be no wave propagation.

So far, we've been using the dynamic viscosities μ and μ'. There are situations where it is more convenient to rescale the dynamic viscosities to

$$\nu = \frac{\mu}{\rho},$$
$$\nu' = \frac{\mu'}{\rho}.$$

These rescaled viscosities are called kinematic viscosities.

2.2.1 Bulk viscosity

One unfortunate fact commented on by several authors [12, 15] is that there is no universally accepted definition of the bulk viscosity. In many cases, the definition is as given in equation 2.7. [12, 14] In other cases, bulk viscosity is defined as $\mu' - 2\mu/3$, [15] $2\mu/3 - \mu'$, [11] or $\mu/3 - \mu'$. [16, 17]

This is quite a problem since different sources use different definitions without much explanation, but there are fortunately a few different ways to find out which definition has been used.

The first method is to check the source's definition of the Navier-Stokes equation. In many cases, it is written in a fashion similar to equation 2.7, 2.14, or B.3. The definition of bulk viscosity can then be found by comparison with the equations given here.

Another, simpler, method is to see if the source states the bulk viscosity of an ideal gas. As mentioned earlier, the bulk viscosity of an ideal gas is $\mu' = 0$ with the definition used in this text. If a source states another value for this quantity (for instance $\mu' = -2\mu/3$ or $2\mu/3$), the source's definition can be found.

2.3 Reynolds number

In an incompressible flow, or a compressible flow in the incompressible limit, the behaviour of the fluid is given by equations 2.8 and 2.9. Two new quantities are now introduced: The characteristic length, l_0, and the characteristic time, t_0.

The characteristic length is a length that defines the system, such as the diameter of a pipe in a pipe flow or the size of an obstacle in a fluid. The characteristic time is a period that similarly defines the system. It is not uncommon to relate these two through a characteristic speed, $u_0 = l_0/t_0$.

We can use these characteristic quantities to convert physical quantities to dimensionless units through dimensional analysis: [18]

$$t_\mathrm{p} = t_{0,\mathrm{p}} t_\mathrm{d}, \qquad \vec{x}_\mathrm{p} = l_{0,\mathrm{p}} \vec{x}_\mathrm{d}, \qquad u_\mathrm{p} = \frac{l_{0,\mathrm{p}}}{t_{0,\mathrm{p}}} u_\mathrm{d},$$
$$p_\mathrm{p} = \rho_0 \frac{l_{0,\mathrm{p}}^2}{t_{0,\mathrm{p}}^2} p_\mathrm{d}, \qquad \partial_{t_\mathrm{p}} = \frac{1}{t_{0,\mathrm{p}}} \partial_{t_\mathrm{d}}, \qquad \vec{\nabla}_\mathrm{p} = \frac{1}{l_{0,\mathrm{p}}} \vec{\nabla}_\mathrm{d},$$

where subscript p and d means physical and dimensionless units, respectively.

Inserting these into equation 2.9, neglecting external forces, and dividing by ρ_0, we get

$$\partial_{t_\mathrm{d}} u_\mathrm{d} + \left(u_\mathrm{d} \cdot \vec{\nabla}_\mathrm{d}\right) u_\mathrm{d} + \vec{\nabla}_\mathrm{d} p_\mathrm{d} - \frac{1}{\mathrm{Re}} \nabla_\mathrm{d}^2 u_\mathrm{d} = 0. \tag{2.10}$$

Re is the dimensionless Reynolds number, given by

$$\mathrm{Re} = \frac{l_{0,\mathrm{p}}^2}{t_{0,\mathrm{p}} \nu_\mathrm{p}}. \tag{2.11}$$

From equation 2.10 it is clear that any two systems with the same geometry and the same Reynolds number will behave identically for uncompressed flow. This has been widely used for scale model tests. For instance, a scale model of an airplane wing that is a quarter of a real wing's size must be tested with a air flow speed (l_0/t_0) that is four times the airplane's speed to get the same behaviour as on the airplane.

As a simple example, a football of diameter $l_0 = 0.7$ m flying through the air at a speed of $l_0/t_0 = 25$ m/s through air with a kinematic viscosity of $\nu = 1.5 \cdot 10^{-5}$ m^2/s (taken from Table 8.1) gives a Reynolds number of

$$\mathrm{Re} = \frac{0.7 \text{ m } \cdot 25 \text{ m/s}}{1.5 \cdot 10^{-5} \text{ m}^2/\text{s}} \approx 1.2 \cdot 10^6.$$

2.4 Navier-Stokes and the wave equation

As mentioned in the last section, wave propagation is a phenomenon of compressible flow. We will now derive the lossy wave equation from the compressible Navier-Stokes equation, mostly following *Fundamentals of Acoustics*.* [13]

In linear acoustics, we look at small compressions and rarefactions around the medium's *equilibrium density*, ρ_0. We introduce a quantity called the *condensation*,

$$s = \frac{\rho - \rho_0}{\rho_0}. \tag{2.12}$$

*Other derivations can be found in for instance section 2.3 of *Methods of Theoretical Physics*, [19] and section 6.4 of *Theoretical Acoustics*. [20]

This is the ratio between the deviation from the equilibrium density and the equilibrium density itself. Since these compressions and rarefactions are small, we can assume that $|s| \ll 1$.

We can also see the total pressure as

$$p = p_0 + p_a,$$

where p_0 is the constant equilibrium pressure, which can be seen as the atmospheric pressure, and p_a is the deviation from the equilibrium pressure, which we can call the *acoustic pressure*.

We use the definition of condensation to rewrite the equation of continuity (2.6), as

$$\partial_t \left[\rho_0 \left(1 + s\right)\right] + \overrightarrow{\nabla} \cdot \left[\rho_0 \left(1 + s\right) \vec{u}\right] = 0.$$

Since ρ_0 is constant and s is small and a weak function of space, this reduces to

$$\partial_t s + \overrightarrow{\nabla} \cdot \vec{u} = 0. \tag{2.13}$$

We now apply the mathematical relation [19]

$$\nabla^2 \vec{a} = \overrightarrow{\nabla}\left(\overrightarrow{\nabla} \cdot \vec{a}\right) - \overrightarrow{\nabla} \times \left(\overrightarrow{\nabla} \times \vec{a}\right)$$

to the compressible Navier-Stokes equation (2.7), giving

$$\rho\left(\partial_t \vec{u} + \left(\vec{u} \cdot \overrightarrow{\nabla}\right)\vec{u}\right) + \overrightarrow{\nabla} p - \left(\frac{4}{3}\mu + \mu'\right)\overrightarrow{\nabla}\left(\overrightarrow{\nabla} \cdot \vec{u}\right) + \mu \overrightarrow{\nabla} \times \left(\overrightarrow{\nabla} \times \vec{u}\right) = \vec{f}, \tag{2.14}$$

Since this is linear acoustics, we can assume that $\vec{u}$ is small, and therefore that $\partial_t \vec{u} \gg (\vec{u} \cdot \overrightarrow{\nabla})\vec{u}$. In most processes in linear acoustics, rotational effects are small and confined to the vicinity of boundaries, [13] so we can assume that $\overrightarrow{\nabla} \times (\overrightarrow{\nabla} \times \vec{u}) \to 0$. We also neglect body forces, saying $\vec{f} \to 0$. These approximations leave us with

$$\rho \partial_t \vec{u} + \overrightarrow{\nabla} p - \left(\frac{4}{3}\mu + \mu'\right)\overrightarrow{\nabla}\left(\overrightarrow{\nabla} \cdot \vec{u}\right) = 0.$$

By taking the divergence of this equation and using equation 2.13, we get

$$\rho \partial_t^2 s \nabla^2 p + \left(\frac{4}{3}\mu + \mu'\right)\nabla^2 \partial_t s = 0.$$

We can rewrite the pressure-density relation, equation 2.5, using the definitions of condensation and acoustic pressure to get

$$p = \underbrace{c_s^2 \rho_0}_{p_0} + \underbrace{c_s^2 \rho_0 s}_{p_a} \quad \Rightarrow \quad s = \frac{p_a}{c_s^2 \rho_0}.$$

We use this and the simple fact that $\nabla^2 p = \nabla^2 p_a$, since p_0 is constant, to get

$$\frac{\rho}{c_s^2 \rho_0}\partial_t^2 p_a - \nabla^2 p_a - \frac{1}{c_s^2 \rho_0}\left(\frac{4}{3}\mu + \mu'\right)\nabla^2 \partial_t p_a = 0.$$

Since the deviations from the equilibrium density are small, $\rho/\rho_0 \approx 1$. Using this and rewriting the equation, we get the *lossy wave equation*,

$$\boxed{(1 + \tau_s \partial_t) \nabla^2 p_a = \frac{1}{c_s^2} \partial_t^2 p_a,} \tag{2.15}$$

where

$$\tau_s = \frac{1}{c_s^2} \left(\frac{4}{3} \nu + \nu' \right).$$

In the limit of no viscosity, $\tau_s \to 0$ and we are left with the familiar wave equation

$$\boxed{\nabla^2 p_a = \frac{1}{c_s^2} \partial_t^2 p_a.} \tag{2.16}$$

In acoustics, this non-lossy equation is almost always used instead of the lossy wave equation, since τ_s is very small in most cases, and the analytic solution to the non-lossy wave equation is simpler.

2.4.1 Solutions to the wave equation

It can be shown [13] that the one-dimensional analytic solution to equation 2.16 in one dimension is

$$p_a = A \, \mathrm{e}^{j(\omega t - kx)}, \tag{2.17}$$

where A is a constant, ω is the angular frequency and k is the angular wavenumber, $k = \omega/c$.

Similarly, it can be shown [13] that the one-dimensional analytic solution to equation 2.15 is

$$p_a = A \, \mathrm{e}^{-\alpha_s x} \, \mathrm{e}^{j(\omega t - kx)}. \tag{2.18}$$

α_s is called the *spatial absorption coefficient* and is given by

$$\alpha_s = \frac{\omega}{c_s \sqrt{2}} \sqrt{\frac{\sqrt{1 + (\omega \tau_s)^2} - 1}{1 + (\omega \tau_s)^2}}. \tag{2.19}$$

Since τ_s tends to be of the order of magnitude 10^{-10} s for gases and 10^{-12} s for liquids, [13] it is safe to take the assumption $\omega \tau_s \ll 1$ for sonic frequencies, simplifying this equation to

$$\alpha_s = \frac{\omega^2 \tau_s}{2 c_s}. \tag{2.20}$$

From equations 2.18 and 2.20, we can calculate that for a 1000 Hz plane wave in air, it takes on the order of kilometers for the wave's amplitude to be absorbed by one percent. For a 20 000 Hz wave, this takes on the order of metres. This means that for low-frequency acoustics, absorption effects from viscosity are negligible, while they become significant for ultrasound.

For an outgoing cylindrical wave from an infinitely long line source, the analytic stationary solution to equation 2.15 in cylindrical coordinates is

$$p_a = A H_0^{(2)}(kr) \mathrm{e}^{j\omega t}. \tag{2.21}$$

k is here a complex angular wavenumber, given by

$$k = \frac{\omega}{c_s} - j\alpha_s, \tag{2.22}$$

while $H_0^{(2)}$ is a Hankel function given by

$$H_0^{(2)}(kr) = J_0(kr) - jY_0(kr), \tag{2.23}$$

where J_0 and Y_0 are Bessel functions of the first and second kinds, respectively. A short derivation of this is given in Appendix B.2.

CHAPTER 3

Lattice gas automata

As the lattice Boltzmann method is a direct descendant of the earlier model of lattice gas automata (LGA), an overview of these models is useful.

The purpose of lattice gas automata is to simulate the behaviour and interaction of many single particles in a gas as simply as possible. LGAs can be seen as very simple molecular dynamics methods. Macroscopic quantities such as particle density and velocity can be recovered from this microscopic scope, making it possible to study the macroscopic behaviour of a fluid in different geometries with this model.

The gas is modeled as a multitide of hard spheres moving along a regular grid, with a discrete set of possible velocities $\vec{c_i}$ for each particle. Collision between particles is handled by a set of elastic collision rules that must conserve the system's quantities of mass m and momentum $\vec{p}$.

As lattice gas automata can be more easily understood by examining the workings of an actual LGA instead of speaking in generalities, we will start by looking at the simplest LGA, the HPP model.

3.1 The HPP model

The HPP model was proposed by Hardy, Pomeau, and de Pazzis (hence the name of the model) as early as 1973. [21, 22] In the HPP model, the grid is two-dimensional and square, so that each node in the grid has four neighbours. The particles can have four possible velocities, $\vec{c_1} = (1, 0)$, $\vec{c_2} = (0, 1)$, $\vec{c_3} = (-1, 0)$, and $\vec{c_4} = (0, -1)$, as shown in Figure 3.1.

For each time step, each particle is moved forward one step in the direction of its velocity, as shown in Figure 3.2.

When two or more particles meet in the same node after a time step, a collision occurs. To conserve mass and momentum, the number of particles and the total velocity of all the particles in the node must be the same before and after the collision. When two particles collide head-on, they are thrown out at right angles to their original velocities, as shown in Figure 3.3. This conserves momentum, as the sum of the velocities of the two particles is zero in both configurations.

When three or four particles meet in the same node, the only configuration that satisfies conservation of mass and momentum is the same configuration as before the

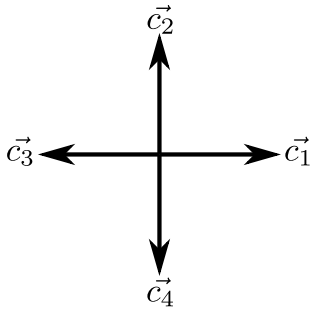


Figure 3.1: The velocity vectors of the HPP model.

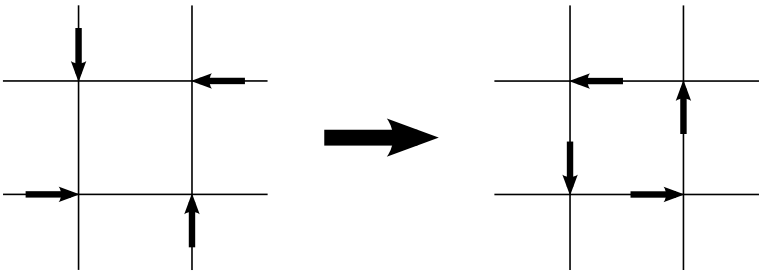


Figure 3.2: Four particles with different directions moving in the HPP model, from one time step to the next.

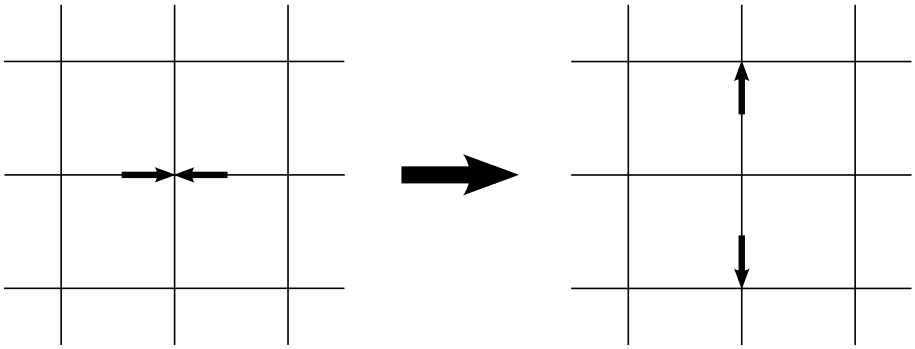


Figure 3.3: The head-on collision rule for the HPP model.

collision. Therefore, collision between three or four particles cannot and does not result in a change of configuration, and the particles stream on as if no collision occurred.

The collisions are entirely deterministic, meaning that each collision has one and only one possible result. Because of this, the HPP model has a property called *time reversal invariance*, which means that the model can be run in reverse to recover any earlier state.

An interesting example of this is given in the article *Cellular Automata and Lattice Boltzmann Techniques*. [23] Two rooms are separated by a wall with a small opening. The first room starts full of particles while the second room starts empty. When the simulation starts, particles escape from the first room to the second until the system reaches an equilibrium of roughly equal particle densities. The model is then run in reverse until all particles in the second room have retraced their steps and gone back to their initial positions in the first room.

The quantities used in the simulation are related to real quantites through the lattice spacing Δx and the time step Δt. For instance, the physical particle speeds $\vec{c}_{i,\mathrm{p}}$ in the lattice are given by

$$\vec{c}_{i,\mathrm{p}} = \frac{\Delta x}{\Delta t}\vec{c}_{i,\mathrm{l}}. \tag{3.1}$$

We say that the vector $\vec{c}_{i,\mathrm{l}}$ is in *lattice units*, which are units normalized by Δx and Δt. In lattice units, we have that $\left|\vec{c}_{i,\mathrm{l}}\right| = 1$ for all i in the HPP model, and that $\Delta t = 1$. For the vector $\vec{c}_{i,\mathrm{p}}$, which is in physical units, we have that $\left|\vec{c}_{i,\mathrm{p}}\right| = \Delta x/\Delta t$.

The particle density in each node can be calculated simply from the number of particles present in the node, or

$$\rho(\vec{x}, t) = \sum_i n_i(\vec{x}, t), \tag{3.2}$$

where $\rho(\vec{x}, t)$ is the particle density at the node with position $\vec{x}$ at time t, and n_i is the Boolean occupation number, meaning the number of particles present (0 or 1) at this node with velocity $\vec{c}_i$. The quantities $\vec{x}$ and t are in lattice units. Similarly, the total momentum in each node can be calculated from

$$\rho(\vec{x}, t)\vec{u}(\vec{x}, t) = \sum_i \vec{c}_i n_i(\vec{x}, t), \tag{3.3}$$

where $\vec{u}(\vec{x}, t)$ is the mean velocity of the particles at this node.

Hard walls can be implemented in the HPP model as special nodes which cause incoming particles to be reflected back. The boundaries of the simulated system can be hard walls, or they can be periodic. Periodic boundaries imply that a particle which exits the system at one edge will re-enter the system at the opposite edge. With hard boundaries, the behaviour of a fluid trapped in a box is simulated, while with periodic boundaries it is the behaviour of a fluid in a periodic system which is simulated. It is naturally possible to combine the two boundaries, for instance by having hard vertical boundaries on two sides and periodic boundaries on the other two.

Force can be simulated in an LGA by randomly changing some particles' velocity to the direction of the force with a given probability. [1, 4]

With all these rules known, it is possible to perform a simulation. First, a certain geometry is created using hard walls, and an initial distribution of particles with certain positions and velocities is created. The system is then left to run for a while until it reaches a sort of equilibrium. Then, the macroscopic particle density can be found by

averaging over space (several nodes) and/or time (several time steps) to find the average number of particles in each area. The macroscopic momentum can be found by a similar average of the nodes' momentum.

3.1.1 Advantages and disadvantages

The HPP model has some advantages compared to other numerical methods. The state of each node in the grid can be described completely by four bits: Bit i represents the presence or absence of one particle moving in the direction $\vec{c_i}$. This means that very little storage space and memory is required.

Another advantage is that due to the boolean nature of the system, no floating point numbers are used in the model, which means that all numbers are perfectly exact and no round-off errors occur. The exact and deterministic* nature of the model also means that it has a property called *time reversal invariance*, which means that the system can be run in reverse to reproduce an earlier state perfectly.

A third advantage is the inherent parallel nature of the system. The events in each node is not related to the simultaneous events in another node, since the only communication between nodes is through streaming of particles. In addition, the nature of the particle streaming is such that each particle can only have one origin.

This means that several processors can process the collisions or particle streaming in different sections of the grid simultaneously without a need for communication, apart from distributing the work among each other and sending back results.

There are also many inherent weaknesses in the HPP model. Due to the microscopic scope of the model, it can never be said to be in a complete equilibrium. The system will never by itself reach a state where the system is identical from one step to the next. This is obvious from the model's property of time reversal invariance: For the system to be in a state where the next state is identical to the current state, all previous states must have been equal to the current state. Therefore it is only possible to reach permanent states by setting them as the initial condition. A trivial example is a system with all possible particles present everywhere, but this can hardly be said to be a natural equilibrium.

For this reason, the microscopic state of the system is always changing. As mentioned earlier, the macroscopic quantities of the system can be found by averaging the microscopic quantities over space or time. But due to the constant change in the microscopic state, there will always be statistical noise in the macroscopic quantities. This problem can be reduced by broadening the average, but never avoided.

Possibly the greatest weakness of the HPP model is that it fails to achieve rotational invariance, which means that its behaviour becomes anisotropic. [1, 2, 23] For instance, in a system with particles evenly spread out, apart from a high concentration in the center, this high concentration will spread out from the centre in a diamond pattern. [1]

As a result of this, HPP systems fail to behave in accordance to the Navier-Stokes equations. This weakness alone is crippling to the point that the HPP model is not useful for fluid simulations. The HPP model was abandoned for this reason in the late 1980s for the FHP model, which gives isotropic Navier-Stokes behaviour by changing the shape of the lattice.

*The property of determinism is spoiled by the stochastic nature of force simulation in an LGA, if such a force simulation is performed. Still, it might be possible to change the particles' velocity in such a pseudo-random manner so that time reversal is still possible.

3.2 The FHP model

The FHP model was suggested by Frisch, Hasslacher and Pomeau (hence the name of the model) in 1986. [24] It was shown that by moving from a square lattice to a hexagonal lattice, the incompressible Navier-Stokes equations could be recovered due to the extra rotational invariance afforded by the hexagonal lattice. [1, 23–25]

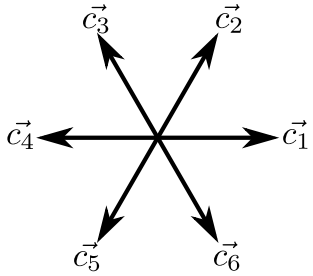


Figure 3.4: The velocity vectors of the FHP model.

The lattice vectors in the FHP model are $\vec{c_1} = (1, 0)$, $\vec{c_2} = (1/2, \sqrt{3}/2)$, $\vec{c_3} = (-1/2, \sqrt{3}/2)$, $\vec{c_4} = (-1, 0)$, $\vec{c_5} = (-1/2, -\sqrt{3}/2)$ and $\vec{c_6} = (1/2, -\sqrt{3}/2)$. These are shown in Figure 3.4.

A hexagonal lattice allows two possible resolutions for a head-on collision which conserve both mass and momentum, illustrated in Figure 3.5. This is different from the HPP model, where there is only one possible resolution. The resolution which is to occur is chosen randomly for every collision, with equal probability. Due to this stochastic element in the model, it is no longer fully deterministic, and does not have the time reversal property of the HPP model.*

The FHP model also has a resolution for a three-particle collision: such a collision reflects each particle back the way it came. This is illustrated in Figure 3.6. For all other collisions, no change in particle distributions should be performed. [1, 24, 25]

As mentioned, it can be derived that the FHP model gives a behaviour in accordance with the incompressible Navier-Stokes equations. From the derivation, one can show that the kinematic viscosity of the system must be [1, 23]

$$\nu = \frac{\Delta x^2}{\Delta t}\left(\frac{1}{2\rho_0\left(1-\rho_0/6\right)^3} - \frac{1}{8}\right), \tag{3.4}$$

where ρ_0 is the equilibrium density of the gas. This can be found as the average particle density of the entire system.

We see from this equation that the viscosity can become arbitrarily large in the limits $\rho_0 \to 0$ and $\rho_0 \to 6$, indicating no particles in the system and a maximum number of particles in the system, respectively. The minimal viscosity for the model is found when $\rho = 3/2$. The minimal viscosity is then

$$\nu_{\min} = \frac{431}{648}\frac{\Delta x^2}{\Delta t} = 0.665\frac{\Delta x^2}{\Delta t}.$$

*Deterministic types of the FHP model are possible using by using certain pseudo-random methods to determine the collision outcome. This means that time reversal can be possible in the FHP model also, but no differences in macroscopic behaviour are expected between deterministic and indeterministic types. [1]

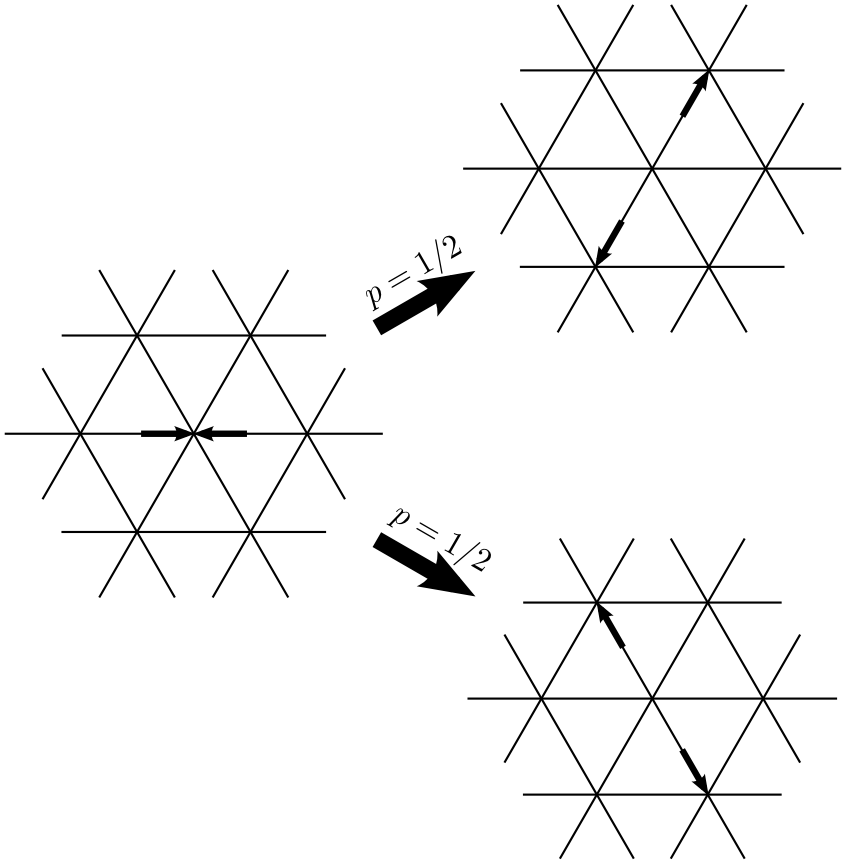


Figure 3.5: The head-on collision rule for the FHP model. The two possible resolutions have equal probability.

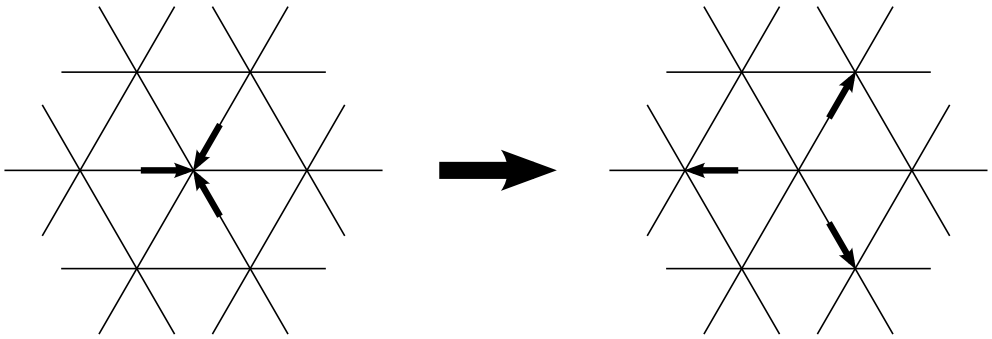


Figure 3.6: The triple collision rule for the FHP model.

Varieties exist of the FHP model, called the FHP-II and FHP-III models. They differ from the standard model in that rest particles and additional collisions are introduced. They have some differences in behaviour compared to the standard FHP model, for instance different viscosities, but have no new principles. We will therefore not look at them in any detail.

3.2.1 Advantages and disadvantages

The FHP model is essentially the HPP model with a change of the lattice and the collision rules. Thus it retains all the advantages of the HPP model mentioned in section 3.1.1, such as small storage space demands, exact dynamics and parallelism. It removes the most critical disadvantage, that of the inherent anisotropy of the HPP model, giving a behaviour consistent with the incompressible Navier-Stokes equations.

The problem of statistical noise remains. This is an general problem when trying to recover macroscopic quantities from microscopic-scope simulations, as the microscopic system is subject to random fluctuations that disappear in the continuum limit. On the other hand, if the subject of interest is studying such fluctuations in real systems, this statistical noise is a desired property. [26]

3.3 Lattice isotropy

Whether a lattice fulfils the condition of isotropy can be found from whether the set of lattice vectors fulfil the following conditions: [1, 23]

$$\sum_i c_{i\alpha} = 0, \tag{3.5a}$$

$$\sum_i c_{i\alpha} c_{i\beta} = C_2 \delta_{\alpha\beta}, \tag{3.5b}$$

$$\sum_i c_{i\alpha} c_{i\beta} c_{i\gamma} = 0, \tag{3.5c}$$

$$\sum_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} = C_4 \left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma} \right). \tag{3.5d}$$

Here, C_2 and C_4 are lattice constants specific to each lattice. $\delta_{\alpha\beta}$ is the Kronecker delta.

We will not go into why conditions 3.5 are necessary for lattice isotropy. An explanation can be found in section 3.3 of *Lattice-Gas Cellular Automata and Lattice Boltzmann Models*. [3]

For the hexagonal lattice, these conditions are fulfilled with $C_2 = 3$ and $C_4 = 3/4$. Therefore, the hexagonal lattice gives an isotropic behaviour of the system. The square lattice fulfils the three first conditions with $C_2 = 2$, but condition 3.5d is not fulfilled. This is the reason for the HPP model's anisotropic behaviour.

One strange property of these is that there is no simple three-dimensional set of lattice vectors that fulfils condition 3.5d. [1, 2] This seemed to be a show-stopper for 3D LGA simulations, but a way out was found by taking the 3D projection of a 4D lattice that fulfilled the condition. [27] The vectors for this 4D lattice consists of all spatial permutations of $\vec{c_i} = [\pm 1, \pm 1, 0, 0]$. This results in 24 different vectors in a 3D lattice that we call the *face-centered hypercubic (FCHC)* lattice.

In the 3D projection, the fourth spatial dimension in these vectors is removed, while the number of vectors remains 24. 12 of these are all spatial permutations of $\vec{c_i} =$

$[\pm 1, \pm 1, 0]$. The remaining 12 come from all six permutations of $\vec{c_i} = [\pm 1, 0, 0]$, with each permutation appearing twice.* This FCHC lattice fulfils conditions 3.5, with $C_2 = 12$ and $C_4 = 4$.

That the lattices discussed in this section fulfil the conditions of isotropy can be verified with the MATLAB script in Appendix C.1.

3.4 Macroscopic behaviour

It can be shown that the FHP method gives a macroscopic behaviour consistent with the Navier-Stokes equation, but we will not show this here as the derivation is quite arduous, and LGAs are not the focus of this thesis. Instead, we will merely show that LGAs give a behaviour consistent with the equation of continuity, equation 2.6. This will be a useful introduction to later, tougher derivations that will use some of the same principles. More information on these derivations for LGAs can be found for instance in Chopard's and Droz's book *Cellular Automata Modeling of Physical Systems*. [1]

With no collisions in the system, the evolution equation for an LGA can be written as

$$n_i\left(\vec{x} + \vec{c_i}, t + 1\right) = n_i\left(\vec{x}, t\right). \tag{3.6}$$

This equation says that for every time step, a particle with a speed $\vec{c_i}$ will move to the node where it is headed, retaining its speed when it arrives. Note that the time step can be written in the equation by increasing t by 1, since t is here in lattice units.

Of course, this equation does not describe the behaviour of an LGA, because we have no collisions. We add collisions to the equation with the collision operator $\Omega_i\left(\vec{x}, t\right)$, which is dependent on $n_i\left(\vec{x}, t\right)$. $\Omega_i\left(\vec{x}, t\right)$ can have the three values, -1, 0 and 1, for each i. These values indicate that a particle has moved away from direction $\vec{c_i}$ due to a collision, stayed on its path, or moved into direction $\vec{c_i}$ due to a collision, respectively. The evolution equation becomes

$$\boxed{n_i\left(\vec{x} + \vec{c_i}, t + 1\right) - n_i\left(\vec{x}, t\right) = \Omega_i\left(\vec{x}, t\right).} \tag{3.7}$$

For the rest of this section, we'll drop the parenthesis $(\vec{x}, t)$ when possible to save space and avoid messy notation. This parenthesis should be considered implied.

As an example of the behaviour of Ω_i, let's take the head-on collision behaviour of the HPP model as shown in Figure 3.3. The case when particles enter the same node with the velocities $\vec{c_1}$ and $\vec{c_3}$ is shown in Table 3.1.

Table 3.1: The head-on collision behaviour for the HPP model as shown in Figure 3.3, with the formalism from equation 3.7.

Quantity	i			
	1	2	3	4
$n_i\left(\vec{x}, t\right)$	1	0	1	0
$\Omega_i\left(\vec{x}, t\right)$	-1	1	-1	1
$n_i\left(\vec{x} + \vec{c_i}, t + 1\right)$	0	1	0	1

*The reason for this is that the fourth vector component in the 4D lattice, which is removed in the projection to 3D, can have two different values: 1 and -1.

It is possible to write the behaviour of Ω_i explicitly as a function of n_i, [1, 23] but we will not do that here as it is not useful in this derivation.

On the other hand, it is useful to look at some general properties of Ω_i. From conservation of mass and equation 3.2 it is obvious that

$$\sum_i n_i\left(\vec{x}+\vec{c_i}, t+1\right) = \sum_i n_i\left(\vec{x}, t\right). \tag{3.8}$$

Similarly, it is obvious from conservation of momentum and equation 3.3 that

$$\sum_i \vec{c_i} n_i\left(\vec{x}+\vec{c_i}, t+1\right) = \sum_i \vec{c_i} n_i\left(\vec{x}, t\right). \tag{3.9}$$

We can now find from equations 3.7, 3.8, and 3.9 that

$$\sum_i \Omega_i = 0, \tag{3.10a}$$

$$\sum_i \vec{c_i}\Omega_i = 0. \tag{3.10b}$$

Equation 3.10a comes from the sum over i in equation 3.7 combined with equation 3.8. Equation 3.10b similarly comes from the sum over i in equation 3.7 multiplied with $\vec{c_i}$ and combined with equation 3.9. We will not use equation 3.10b in this derivation, beyond showing here that it holds.

Now, imagine several similar LGA systems, all with the same geometry. Each of these systems has a different distribution of particles, but the particles are still distributed in such a way that the macroscopic quantities of mass and momentum are the same over all systems. We call this collection of systems an *ensemble*, and we find the mean of equation 3.7 over all systems in this ensemble. This can also be called the *ensemble average* of the equation.* The equation becomes

$$N_i\left(\vec{x}+\vec{c_i}, t+1\right) - N_i\left(\vec{x}, t\right) = \left\langle \Omega_i\left(\vec{x}, t\right)\right\rangle. \tag{3.11}$$

The ensemble average $N_i\left(\vec{x}, t\right)$ is a number between 0 and 1 which can be interpreted as the probability of $n_i\left(\vec{x}, t\right)$ being 1. Equations 3.2, 3.3, and 3.10 naturally still apply for ensemble averages, but the two first equations immediately give the correct macroscopic quantities.

Now, we can Taylor expand $N_i\left(\vec{x}+\vec{c_i}, t+1\right)$ to the first order, which gives

$$N_i\left(\vec{x}+\vec{c_i}, t+1\right) = N_i + \partial_t N_i + \overrightarrow{\nabla}\cdot\vec{c_i}N_i. \tag{3.12}$$

Inserting this into equation 3.11, we get

$$\langle\Omega_i\rangle = \partial_t N_i + \overrightarrow{\nabla}\cdot\vec{c_i}N_i. \tag{3.13}$$

We can sum this equation over all i to get

$$\partial_t \sum_i N_i + \overrightarrow{\nabla}\cdot\sum_i \vec{c_i}N_i = 0. \tag{3.14}$$

*In the language of statistical mechanics, this can be formulated more briefly by saying that we take the average of all micro-states that share the same macro-state.

This reduces through equations 3.2 and 3.3 to

$$\boxed{\partial_t \rho + \overrightarrow{\nabla} \cdot (\rho \vec{u}) = 0,} \tag{3.15}$$

which is equal to equation 2.6, the equation of continuity.

We have now shown a simple example of how one can derive macroscopic behaviour from the dynamics of a lattice gas automaton. Note also how we have not made any assumptions in this derivation about the lattice of the LGA. The only assumption about the LGA made and used during the derivation was conservation of mass in all collisions. This means that the continuity equation is fulfilled for any LGA that conserves mass in collisions, which makes sense since the continuity equation is essentially a statement of the conservation of mass in a macroscopic scope.

CHAPTER 4

The lattice Boltzmann method

In 1988, McNamara and Zanetti proposed a fix to the LGAs' problem of statistical noise. [28] With the FHP-III model as their basis, they replaced the Boolean occupation number n_i with its ensemble average,

$$f_i = \langle n_i \rangle ,$$

so that f_i is a number between 0 and 1. The evolution equation for a lattice gas, equation 3.7, then becomes the *lattice Boltzmann evolution equation*,

$$\boxed{f_i \left(\vec{x} + \vec{c_i}, t+1 \right) - f_i \left(\vec{x}, t \right) = \Omega_i \left(\vec{x}, t \right) .} \tag{4.1}$$

This cleverly removed the need for the averaging used by the LGA to find macroscopic quantities, along with its inherent statistical noise.* Equation 4.1 is the cornerstone of the lattice Boltzmann method (*LBM*).

This is similar to what we did in section 3.4, when we used the statistical average of the occupation number to prove analytically that the dynamics of LGAs satisfy the continuity equation. The difference is that McNamara and Zanetti actually simulated the ensemble average directly in the numerical method instead of it being a theoretical quantity in an analytic derivation to prove the method's conformance to the Navier-Stokes equation.

The macroscopic quantities of density and momentum can now easily be derived from f_i by

$$\rho \left(\vec{x}, t \right) = \sum_i f_i \left(\vec{x}, t \right) , \tag{4.2}$$

$$\rho \left(\vec{x}, t \right) \vec{u} \left(\vec{x}, t \right) = \sum_i \vec{c_i} f_i \left(\vec{x}, t \right) . \tag{4.3}$$

Since we are no longer keeping track of single particles, we are no longer operating on a microscopic scale. Instead we have moved to a mesoscopic scale, treating averages over many single particles.

*The tradeoffs of this are related to the fact that f_i is now a real number instead of a boolean. Real numbers take much more memory storage space and are subject to round-off errors.

Note how we do not need to restrict f_i to a number between 0 and 1 — it can be scaled with an arbitrary positive constant. Still, it is common to scale f_i in such a way that $\rho(\vec{x}, t) = 1$ in an undisturbed fluid at equilibrium.

From here on in this chapter, we drop the notation $(\vec{x}, t)$ unless it aids clarity.

4.1 The Boltzmann equation

As a digression, we can note that this new method bears remarkable similarity to the Boltzmann equation familiar from statistical mechanics. It represents the particle density in the position range $\vec{x}+d\vec{x}$ and momentum range $\vec{p}+d\vec{p}$ * at time t as the continuous function [4]

$$f\left(\vec{x}, \vec{p}, t\right) d\vec{x}d\vec{p}.$$

If collisions are neglected, the particles will simply stream in the direction they are headed. Knowing the particle distribution at time t enables us to know the particle distribution at time $t + dt$,

$$f\left(\vec{x} + \frac{\vec{p}}{m}dt, \vec{p} + d\vec{p}, t + dt\right) d\vec{x}d\vec{p} = f\left(\vec{x}, \vec{p}, t\right) d\vec{x}d\vec{p}.$$

If we do not neglect collisions, this equation does not hold. Some particles at $(\vec{x}, \vec{p}, t)$ will not arrive at $(\vec{x} + \frac{\vec{p}}{m}dt, \vec{p} + d\vec{p}, t + dt)$ because they have been collided out of this path. Other particles will arrive at $(\vec{x} + \frac{\vec{p}}{m}dt, \vec{p} + d\vec{p}, t + dt)$ because they have been collided *into* this path. We represent this difference in particles as $\Omega d\vec{x}d\vec{p}dt$, so that

$$\boxed{f\left(\vec{x} + \vec{c}dt, \vec{p} + \vec{F}dt, t + dt\right) d\vec{x}d\vec{p} - f\left(\vec{x}, \vec{p}, t\right) d\vec{x}d\vec{p} = \Omega d\vec{x}d\vec{p}dt.} \tag{4.4}$$

Here we have used that $\vec{p}/m = \vec{c}$ and $d\vec{p} = \vec{F}dt$, $\vec{c}$ being a velocity and $\vec{F}$ being an external force at $\vec{x}$.

If we take the first-order Taylor expansion of the left side of equation 4.4 we get what is commonly called the Boltzmann equation, [2, 4]

$$\left[\vec{c} \cdot \overrightarrow{\nabla}_x + \vec{F} \cdot \overrightarrow{\nabla}_p + \partial_t\right] f\left(\vec{x}, \vec{p}, t\right) = \Omega.$$

Here, $\overrightarrow{\nabla}_x$ is $(\frac{\partial}{\partial x_\alpha}, \frac{\partial}{\partial x_\beta}, \ldots)$ and $\overrightarrow{\nabla}_p$ is $(\frac{\partial}{\partial p_\alpha}, \frac{\partial}{\partial p_\beta}, \ldots)$.

The lattice Boltzmann evolution equation (4.1) can also be seen as a discretisation of equation 4.4. We neglect external forces and normalize the mass, m, to 1 so that $\vec{p} = \vec{c}$. Then we limit $\vec{c}$ to a set of vectors $\vec{c_i}$ that can be used to tile the entire space, such as the hexagonal tiling shown in figures 3.5 and 3.6. With discrete velocities, we can compress the notation from $f(\vec{x}, \vec{c_i}, t)$ to $f_i(\vec{x}, t)$ with no loss of information. Discretising time, we can let $\Omega dt \to \Omega_i(\vec{x}, t)$, where $\Omega_i(\vec{x}, t)$ is the net number of particles that was collided into or away from direction $\vec{c_i}$ at position $\vec{x}$ and time t. We are left with

$$f_i\left(\vec{x} + \vec{c_i}\Delta t, t + \Delta t\right) - f_i\left(\vec{x}, t\right) = \Omega_i\left(\vec{x}, t\right),$$

which in normalized lattice units ($|\vec{c_i}| \sim 1$, $\Delta t = 1$) becomes equal to equation 4.1, which was derived in a very different manner.

We have now shown that this equation can be seen not only as an evolution from lattice gas automata, but also as a discretisation of the Boltzmann equation; hence the name *lattice Boltzmann method*.

*In statistical mechanics, $(\vec{x}, \vec{p})$ are said to be coordinates in *phase space*.

4.2 The BGK operator

So far we have not discussed the nature of the collision operator Ω_i, other than mention that it was based on the FHP-III model's Boolean collision operator. We will not go into a general explanation of collision operators,* but rather talk about the one which has proved the most popular.

In the first years of the 1990s, several authors suggested simplified collision operators for the lattice Boltzmann model. In 1992, Qian, d'Humieres, and Lallemand proposed [29] to use a simplified collision operator similar to the one proposed for the Boltzmann equation by Bhatnagar, Gross, and Krook in 1954. [30] The BGK collision operator is given by

$$\boxed{\Omega_i = -\frac{1}{\tau}\left[f_i - f_i^{(0)}\right],} \tag{4.5}$$

where τ is a free parameter known as the *relaxation time* and $f_i^{(0)}$ is the equilibrium distribution of particles. The operator itself represents a relaxation of the distribution function f_i towards the equilibrium value $f_i^{(0)}$.

Since the collision operator must preserve both mass and momentum,

$$\sum_i \Omega_i = 0, \tag{4.6a}$$

$$\sum_i \vec{c}_i \Omega_i = 0. \tag{4.6b}$$

This can be found from the same arguments that lead to conditions 3.10.

From conditions 4.6, it is clear that the equilibrium distribution must preserve the mass and momentum, i.e.

$$\rho = \sum_i f_i^{(0)} = \sum_i f_i, \tag{4.7}$$

$$\rho\vec{u} = \sum_i \vec{c}_i f_i^{(0)} = \sum_i \vec{c}_i f_i. \tag{4.8}$$

The equilibrium distribution for a node is constructed from the node's macroscopic quantities ρ and $\vec{u}$, which are found from f_i.

If we insert the BGK collision operator (4.5) into the lattice Boltzmann evolution equation (4.1), we get the lattice Boltzmann evolution equation for the BGK operator,

$$\boxed{f_i\left(\vec{x} + \vec{c}_i, t+1\right) = \left(1 - \frac{1}{\tau}\right) f_i\left(\vec{x}, t\right) + \frac{1}{\tau} f_i^{(0)}\left(\vec{x}, t\right).} \tag{4.9}$$

We see that when $\tau = 1$, the right side of this equation becomes $f_i^{(0)}$. This is the case of complete relaxation, when all traces of $f_i(\vec{x}, t)$ are removed and only $f_i^{(0)}$ is propagated. When $\tau > 1$, it is called subrelaxation, as the particle distribution is not completely relaxed to equilibrium. $\tau < 1$ is called overrelaxation, since the particle distribution is moved beyond equilibrium. τ cannot be arbitrarily low, since numerical instabilities appear when $\tau \to 0.5$. [4, 23, 29]

We have not discussed yet how the equilibrium distribution is constructed. We come back to this in section 4.4.

*This can be found in *The Lattice Boltzmann Equation.* [2]

4.3 Lattice isotropy

Not every lattice is appropriate for use in the lattice Boltzmann method. Like for lattice gas automata, there are several conditions that a set of lattice vectors must meet to give sufficiently isotropic behaviour to recover the Navier-Stokes equation. With the BGK collision operator, these are [11]

$$\sum_i t_i = 1, \tag{4.10a}$$

$$\sum_i t_i c_{i\alpha} = 0, \tag{4.10b}$$

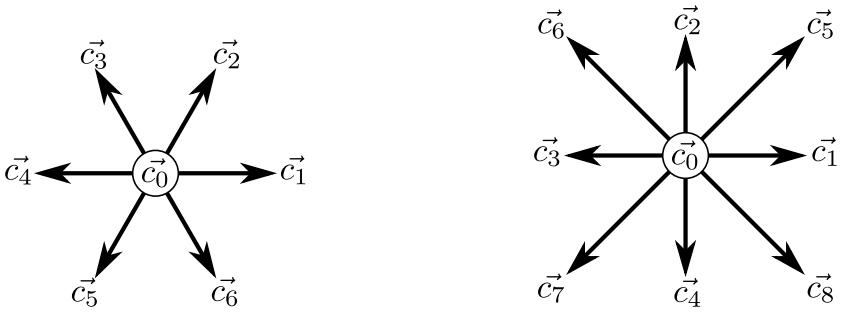
$$\sum_i t_i c_{i\alpha} c_{i\beta} = c_s^2 \delta_{\alpha\beta}, \tag{4.10c}$$

$$\sum_i t_i c_{i\alpha} c_{i\beta} c_{i\gamma} = 0, \tag{4.10d}$$

$$\sum_i t_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} = c_s^4 \left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right), \tag{4.10e}$$

$$\sum_i t_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} c_{i\epsilon} = 0. \tag{4.10f}$$

t_i is a set of lattice vector weights that must be chosen for each lattice to fulfil these conditions.



(a) D2Q7 (b) D2Q9

Figure 4.1: Two common sets of two-dimensional lattice vectors for lattice Boltzmann, with the most common vector numbering.

In two dimensions, there are several lattices that fulfil these conditions. It is common to use a special notation to denote different lattice types. A d-dimensional lattice with q lattice vectors is identified as a DdQq lattice. The FHP lattice with a rest particle vector is known as D2Q7, while the HPP lattice with four additional diagonal vectors $\vec{c_i} = (\pm 1, \pm 1)$ and a rest particle vector is known as D2Q9. Both of these are shown in Figure 4.1. As the D2Q9 lattice is most commonly used and the simplest to implement,* we will focus on that.

*The implementation advantage of the D2Q9 lattice over the D2Q7 lattice is that the D2Q9 lattice has a square grid of nodes while D2Q7 has a hexagonal grid, which is more difficult to implement on a computer. It is possible to map a hexagonal grid onto a square grid, [4] but far trickier than using a purely square grid.

In the D2Q9 lattice, we use three different t_i weights,

$$t_i = \begin{cases} t_0 & \text{for } i = 0, \text{ the rest vector,} \\ t_s & \text{for } i = 1, 2, 3, 4, \text{ the short vectors,} \\ t_l & \text{for } i = 5, 6, 7, 8, \text{ the long vectors.} \end{cases}$$

From conditions a, c and e in 4.10, we have that

$$t_0 + 4t_s + 4t_l = 1, \tag{4.11a}$$
$$2t_s + 4t_l = c_s^2, \tag{4.11b}$$
$$2t_s + 4t_l = 3c_s^4, \tag{4.11c}$$
$$4t_l = c_s^4. \tag{4.11d}$$

This set of equations admits only one solution for all four variables, shown in table 4.1(b).

Similarly, it can be easily shown that the D2Q7 lattice fulfils conditions 4.10. We will state for completeness that when $\vec{c_0}$ is weighted by t_0 and the other lattice vectors by t_s, the lattice weights and speed of sound are those shown in table 4.1(a).

Table 4.1: Lattice constants for two common 2D lattices.

(a) D2Q7 lattice

Constant	Value
t_0	$1/2$
t_s	$1/12$
c_s	$1/2$

(b) D2Q9 lattice

Constant	Value
t_0	$4/9$
t_s	$1/9$
t_l	$1/36$
c_s	$1/\sqrt{3}$

Due to the lattice Boltzmann method's advantage over lattice gas automata of being able to weigh lattice vectors, it is simpler to find appropriate lattices in 1D and 3D. The smallest set of lattice vectors which fulfil the isotropy conditions in 1D and 3D are the D1Q3 and D3Q15 lattices, respectively.

In the D1Q3 lattice, nodes are regularly spaced on a line, with lattice vectors to neighbouring nodes. In the D3Q15 lattice, the nodes are cubically spaced, with lattice vectors to nearest neighbours and third nearest neighbours. Both lattices have a speed of sound $c_s = 1/\sqrt{3}$. The lattice vectors and their weights for the D1Q3 and D3Q15 lattices are shown in Table 4.2. The lattice vectors themselves are shown in Figures 4.2 and 4.3.

It is interesting to note that the D1Q3 lattice is the projection of the D2Q9 lattice on one dimension, while the D2Q9 lattice itself is the projection of the D3Q15 lattice on two dimensions.

That these lattices and lattice weights fulfil the conditions of isotropy can be shown directly by calculating conditions 4.10 for a set of lattice vectors and lattice weights. This can be done easily with the MATLAB script in Appendix C.2.

For any lattice that fulfils these conditions, the basic lattice Boltzmann model with the BGK operator gives a behaviour in accordance with the compressible Navier-Stokes

Table 4.2: Lattice vector bases and their weights for the D1Q3 and D3Q15 lattices. The complete set of vectors is all spatial permutations of the vectors given here.

(a) D1Q3 lattice

Vector	**Weighting**
(0)	$t_0 = 2/3$
(± 1)	$t_s = 1/6$

(b) D3Q15 lattice

Vector	**Weighting**
$(0,0,0)$	$t_0 = 2/9$
$(\pm 1,0,0)$	$t_s = 1/9$
$(\pm 1,\pm 1,\pm 1)$	$t_l = 1/72$

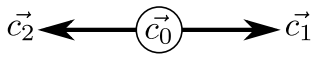


Figure 4.2: The vectors of the D1Q3 lattice.

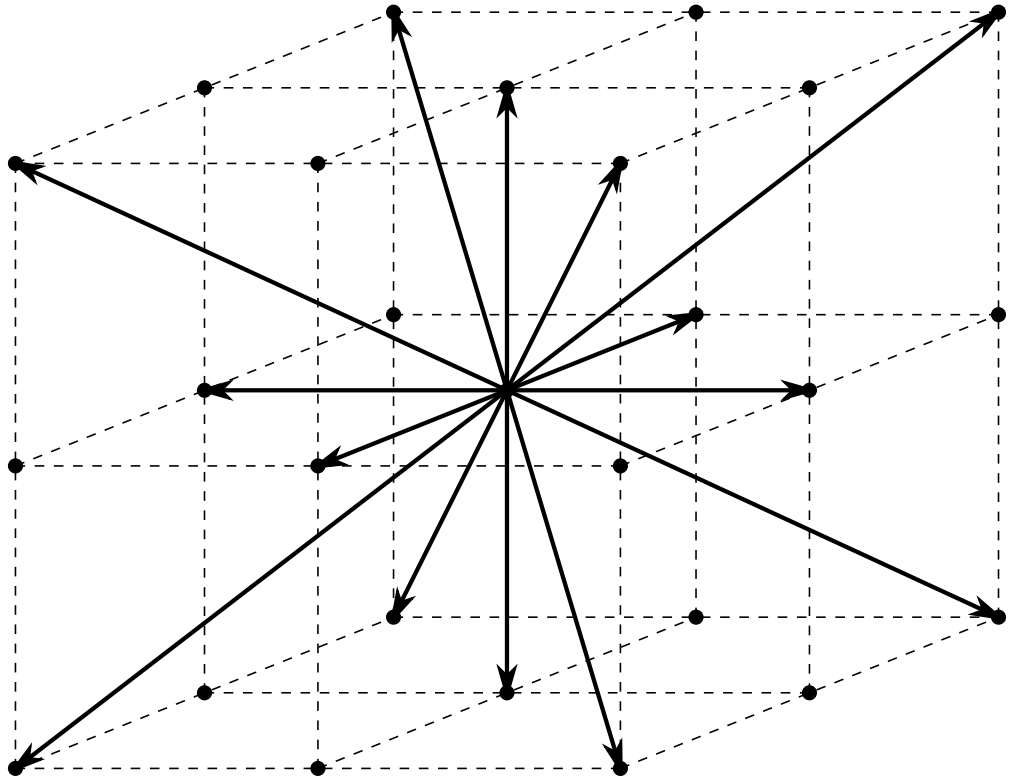


Figure 4.3: The vectors of the D3Q15 lattice.

equation (2.7) in the limit of low Mach number,* with kinematic shear and bulk viscosities given by

$$\nu = c_s^2 \left(\tau - \frac{1}{2} \right), \tag{4.12}$$

$$\nu' = \frac{2}{3}\nu. \tag{4.13}$$

This is shown in Appendix A.

4.4 The equilibrium distribution

The equilibrium distribution can be derived from the Maxwell-Boltzmann velocity distribution from statistical mechanics. This is the probability distribution of particles in a gas in equilibrium. In two dimensions, this distribution is given by [3]

$$w_B\left(\vec{v}\right) = \left(\frac{m}{2\pi k_B T} \right) \exp\left[-\frac{m\vec{v}^2}{2k_B T} \right], \tag{4.14}$$

where k_B is the Boltzmann constant and T the gas temperature. We rewrite $\vec{v}$ as $\vec{c_i} - \vec{u}$, where $\vec{u}$ is the mean velocity and $\vec{c_i}$ is the deviation from it.

Using the ideal gas law, [31]

$$p = \frac{N k_B T}{V}$$

and the isothermal ideal gas pressure relation (2.5), we have that

$$c_s^2 = \frac{k_B T}{m},$$

giving us that

$$w_B\left(\vec{c_i}\right) \propto \exp\left[-\frac{\left(\vec{c_i} - \vec{u}\right)^2}{2c_s^2} \right] = \exp\left[-\frac{\vec{c_i}^2}{2c_s^2} \right] \exp\left[-\frac{\vec{u}^2 - 2\vec{u}\cdot\vec{c_i}}{2c_s^2} \right].$$

Using the Taylor series expansion $e^x = 1 + x^1/1! + x^2/2! + \ldots$, we get that

$$\exp\left[-\frac{\vec{u}^2 - 2\vec{u}\cdot\vec{c_i}}{2c_s^2} \right] \approx 1 - \frac{\vec{u}^2 - 2\vec{u}\cdot\vec{c_i}}{2c_s^2} + \frac{\left(\vec{u}^2 - 2\vec{u}\cdot\vec{c_i}\right)^2}{8c_s^4},$$

where

$$\left(\vec{u}^2 - 2\vec{u}\cdot\vec{c_i}\right)^2 = \underbrace{u^4}_{\mathcal{O}(u^4)} - \underbrace{4u^2\left(\vec{u}\cdot\vec{c_i}\right)}_{\mathcal{O}(u^3)} + 4\left(\vec{u}\cdot\vec{c_i}\right)^2 = 4\left(\vec{u}\cdot\vec{c_i}\right)^2 + \mathcal{O}\left(u^3\right),$$

giving that

$$\exp\left[-\frac{\vec{u}^2 - 2\vec{u}\cdot\vec{c_i}}{2c_s^2} \right] \approx 1 + \frac{\vec{u}\cdot\vec{c_i}}{c_s^2} + \frac{\left(\vec{u}\cdot\vec{c_i}\right)^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2} + \mathcal{O}\left(u^3\right).$$

We will now show that

$$f_i^{(0)} = K t_i \left[1 + \frac{\vec{u}\cdot\vec{c_i}}{c_s^2} + \frac{\left(\vec{u}\cdot\vec{c_i}\right)^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2} \right]$$

*The Mach number is a dimensionless number defined as $\mathrm{Ma} = |\vec{u}|/c_s$.

is an appropriate choice for the equilibrium distribution of particles in the lattice Boltzmann method, and what the proportionality constant K is.

From equation 4.7, we know that

$$\rho = \sum_i f_i^{(0)} = K \sum_i t_i \left[1 + \frac{\vec{u} \cdot \vec{c_i}}{c_s^2} + \frac{(\vec{u} \cdot \vec{c_i})^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2} \right].$$

Rewriting the dot products with index notation, we get

$$\begin{aligned} \rho &= K \sum_i t_i \left[1 + \frac{u_\alpha c_{i\alpha}}{c_s^2} + \frac{u_\alpha u_\beta c_{i\alpha} c_{i\beta}}{2c_s^4} - \frac{u_\alpha u_\alpha}{2c_s^2} \right] \\ &= K \sum_i t_i + \frac{K u_\alpha}{c_s^2} \sum_i t_i c_{i\alpha} + \frac{K u_\alpha u_\beta}{2c_s^4} \sum_i t_i c_{i\alpha} c_{i\beta} - \frac{K u_\alpha u_\alpha}{2c_s^2} \sum_i t_i. \end{aligned}$$

Using conditions 4.10, this becomes

$$\rho = K + \frac{K u_\alpha u_\alpha}{2c_s^2} - \frac{K u_\alpha u_\alpha}{2c_s^2} = K,$$

giving us that the equilibrium distribution is

$$\boxed{f_i^{(0)} = \rho t_i \left[1 + \frac{\vec{u} \cdot \vec{c_i}}{c_s^2} + \frac{(\vec{u} \cdot \vec{c_i})^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2} \right].} \tag{4.15}$$

We should also prove that condition 4.8 is met with this distribution. We have similarly that

$$\sum_i c_{i\alpha} f_i^{(0)} = \rho \sum_i t_i c_{i\alpha} + \frac{\rho u_\beta}{c_s^2} \sum_i t_i c_{i\alpha} c_{i\beta} + \frac{\rho u_\beta u_\gamma}{2c_s^4} \sum_i t_i c_{i\alpha} c_{i\beta} c_{i\gamma} - \frac{\rho u_\beta u_\beta}{2c_s^2} \sum_i t_i c_{i\alpha}.$$

All but the second term on the right side disappear due to conditions 4.10, leaving us with

$$\sum_i c_{i\alpha} f_i^{(0)} = \rho u_\alpha \qquad \Leftrightarrow \qquad \sum_i \vec{c_i} f_i^{(0)} = \rho \vec{u}.$$

We have now shown that the equilibrium distribution of equation 4.15 fulfils all the demands placed on it. This equilibrium distribution is the most frequently used in the field, but not the only one that satisfies all the necessary conditions. For instance, the two last terms in equation 4.15 are not necessary to preserve the correct mass and momentum, and lattice Boltzmann models with these terms removed have been used. [1, 23, 32] In the Chopard-Luthi wave model described in section 7.1, these two terms are removed from the equilibrium distribution. Still, these terms are necessary to recover the Navier-Stokes equation from the dynamics of the lattice Boltzmann method.

It is worth noting that this equilibrium distribution can also be found by finding the function that maximizes entropy while preserving mass and momentum. [3, 33]

4.5 Simple boundaries

The simplest lattice Boltzmann boundary is the periodic boundary. Periodic edges act as if they are connected to the opposite edge of the system. If the left and right edges

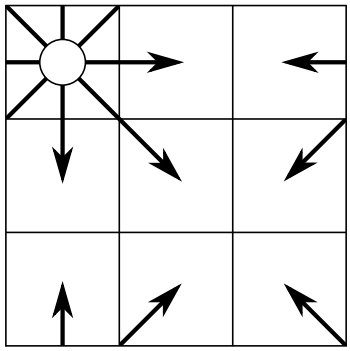


Figure 4.4: Particle streaming paths from a corner node in a periodic 2D system with a D2Q9 lattice and 3 x 3 nodes.

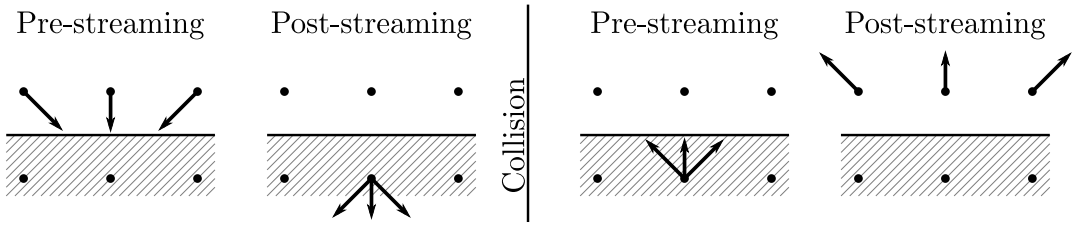


Figure 4.5: The on-grid bounce-back method. Particles which reach the wall node in one time step are reflected back in the next.

of a system are periodic, a particle distribution which streams left at the left edge of the system will reappear at the right edge, heading left. In a 2D system, the left-right boundaries and/or the top-bottom boundaries can be periodic. Figure 4.4 shows neighbouring nodes across periodic boundaries.

A periodic boundary naturally implies that the system is periodic. This can be desired in cases where a periodic behaviour is studied, such as the propagation of a plane sound wave in a duct, where the length of the system equals the wavelength. [17]

The other elementary lattice Boltzmann boundary is the hard wall, which reflects particles back and guarantees a non-slip condition with zero velocity at the wall. There are two variations on this method, the *on-grid* or *full-way* bounce-back method and the *mid-grid* or *half-way* bounce back method. In these two methods, certain nodes in the grid are marked as walls, and are thus not a part of the fluid. The particle distributions in wall nodes are not relaxed towards equilibrium, and do not act in accordance with the lattice Boltzmann evolution equation (4.9).

The on-grid method is the simplest of the two methods. In the collision step of each iteration, where the fluid nodes are relaxed, all particles' directions are reversed so that the particles are sent back in the direction they came from, as shown in Figure 4.5. The collision step is modified for the wall nodes but the streaming step is unchanged.

In the mid-grid method, particles which are set to stream into a wall node are reflected instead of streamed, as shown in Figure 4.6. Here, the streaming step is modified while the collision step remains the same.

In both bounce-back methods, the effective position of the wall is actually between the wall nodes and the fluid nodes adjacent to it. This can be seen for instance by

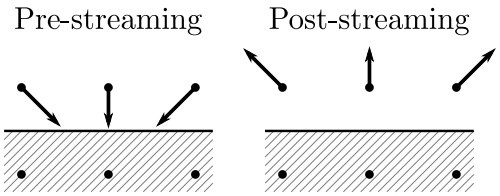


Figure 4.6: The mid-grid bounce-back method. Particles set to stream into a wall node are instead reflected.

comparing a LBM simulation of a forced channel flow (a *Poiseuille* flow) with an analytic solution. [34]

The mid-grid method is somewhat more difficult to implement than the on-grid method, but the mid-grid method gives a better accuracy. [6] We will use mid-grid walls exclusively in simulations later in the text.

4.6 Summary

Here we will sum up the basics of the lattice Boltzmann method.

The lattice Boltzmann method is based on a regular grid. Each node in the grid has several different variables associated with it, $f_i(\vec{x}, t)$. These variables represent the density of particles travelling in direction $\vec{c_i}$ at the node at $\vec{x}$ and time t. In the most commonly used two-dimensional lattice, there are nine different velocity vectors $\vec{c_i}$, shown in Figure 4.7.

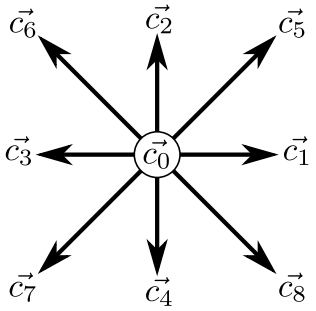


Figure 4.7: The D2Q9 lattice, which is the most commonly used two-dimensional lattice in lattice Boltzmann. The vectors are $\vec{c_0} = (0,0)$, $\vec{c_1} = (1,0)$, $\vec{c_2} = (0,1)$, $\vec{c_3} = (-1,0)$, $\vec{c_4} = (0,-1)$, $\vec{c_5} = (1,1)$, $\vec{c_6} = (-1,1)$, $\vec{c_7} = (-1,-1)$, and $\vec{c_8} = (1,-1)$.

The macroscopic quantities of particle density and velocity at the node can be recovered from $f_i(\vec{x}, t)$ by

$$\rho\left(\vec{x}, t\right) = \sum_i f_i\left(\vec{x}, t\right), \tag{4.16}$$

$$\rho\left(\vec{x}, t\right) \vec{u}\left(\vec{x}, t\right) = \sum_i \vec{c_i} f_i\left(\vec{x}, t\right), \tag{4.17}$$

where ρ is particle density and $\vec{u}$ the flow velocity.

In the fluid nodes, the particles' collisions and propagations is handled according to

$$f_i\left(\vec{x}+\vec{c_i},t+1\right)=\left(1-\frac{1}{\tau}\right)f_i\left(\vec{x},t\right)+\frac{1}{\tau}f_i^{(0)}\left(\vec{x},t\right), \tag{4.18}$$

where τ is a time relaxation constant and $f_i^{(0)}$ is the equilibrium distribution for the particles. τ is related to the kinematic shear viscosity through

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

where c_s is the speed of sound in the fluid, which is $1/\sqrt{3}$ in lattice units for the D2Q9 lattice. It is important to note that the lattice Boltzmann method becomes numerically unstable in the limit of very low viscosity, when $\tau \to 0.5$. The kinematic bulk viscosity is related to the shear viscosity through

$$\nu'=\frac{2}{3}\nu. \tag{4.19}$$

Each node's equilibrium distribution of particles, $f_i^{(0)}$, is constructed from the macroscopic quantites of density and velocity at that node, and is given by

$$f_i^{(0)}=\rho t_i\left[1+\frac{\vec{u}\cdot\vec{c_i}}{c_s^2}+\frac{\left(\vec{u}\cdot\vec{c_i}\right)^2}{2c_s^4}-\frac{\vec{u}^2}{2c_s^2}\right], \tag{4.20}$$

where t_i is a set of weights, one for each velocity vector. In the D2Q9 lattice, $t_0=4/9$, $t_i=1/9$ for $i=1,\ldots,4$, and $t_i=1/36$ for $i=5,\ldots,8$.

The algorithm itself consists of the following steps:

1. **Equilibrium:** Calculate the equilibrium distribution $f_i^{(0)}$ for all fluid nodes from the node's macroscopic variables ρ and $\vec{u}$ using equation 4.20.

2. **Collision:** Calculate the post-collision distribution of particles for all fluid nodes. This is given by the right side of equation 4.18.

3. **Streaming:** The particle distributions are updated by propagating the post-collision distribution of particles one lattice step in the direction they are headed. This completes equation 4.18.

4. **Macroscopic quantities:** From the current distribution of particles, calculate the macroscopic quantities ρ and $\vec{u}$ for each node from equations 4.16 and 4.17.

Each step relies on the previous step only, so that the steps can be cyclically permuted with no problems.

Periodic boundaries and no-slip walls are handled as described in section 4.5. In short, periodic boundaries are handled by letting particles that stream out of the system on one side stream into the system on the opposite side. On-grid walls are created by modifying the collision step to only reverse the particle directions on wall nodes. Mid-grid walls are created by modifying the streaming step to reverse the direction of particles that otherwise would stream into a wall node. Mid-grid walls give better accuracy.

An important property the lattice Boltzmann method retains from its lattice gas ancestors is locality. In each step given above, each node is updated separately from the

others. This means that each step can be parallelised very simply, and the algorithm is able to use as many processors as there are nodes in the grid.

Nothing has been said yet about the problem of initial conditions. The simplest way to initialize the system is to specify an initial distribution of the macroscopic variables $\rho(\vec{x}, 0)$ and $\vec{u}(\vec{x}, 0)$ and set all particle distributions to equilibrium based on these using equation 4.20.*

4.6.1 Example simulation

As an example of what we can do with what we've introduced so far, we'll look at the results of a simulation where a 101×101 D2Q9 system with rigid mid-grid walls is excited by a delta function at a single node. The delta function is implemented by letting a system at rest ($\rho = \rho_0 = 1$ everywhere) start with a density $\rho = 1.1$ at the coordinate $(30, 30)$.

Figure 4.8 shows snapshots in time of the results of this simulation at $\tau = 1$. Figure 4.9 shows the same results for a simulation run at $\tau = 0.6$.

We see from these figures how the pulse turn into a ripple which is reflected at the walls. The difference in τ between the two simulations is manifested in two properties. First, the mentioned instability for low τ is manifested by noise in the ripple at low t, as shown in subfigures 4.9(b), (c) and (d), but not in the corresponding subfigures in Figure 4.8.

The second property is the low viscosity that comes from a low τ which is manifested by a sharper ripple. The peaks and valleys in Figure 4.9 are broader and stronger than the ones in Figure 4.8.

Wall reflection

From the figures, we can see that the ripple is reflected in the manner expected from the acoustic method of images. [13] Since the mid-grid walls are placed at $y = 1.5$ as explained in section 4.5, there is a distance of 28.5 nodes between the source and the wall. Thus, we would expect to see a distance of 57 nodes between the original and reflected peak.

Figure 4.10 shows the amplitude along the line $x = 40$ in Figure 4.9(f). The peaks are positioned at $y = 31$ and $y = 88$ and the valleys at $y = 25$ and $y = 82$. In both cases, this equals a distance of 57 nodes, which is *exactly* what we expected from the method of images.

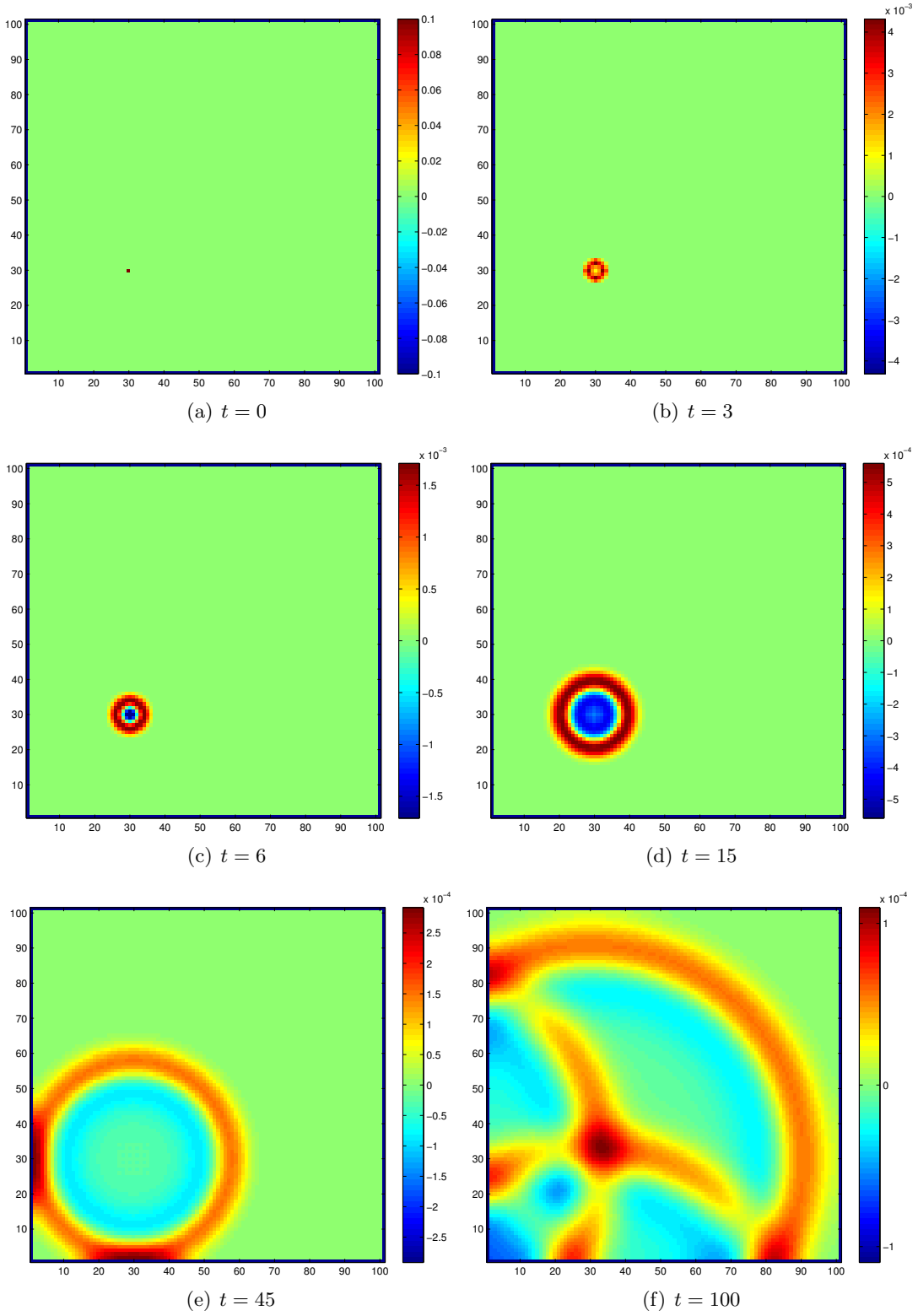
Another interesting feature of the figure is that we cannot really see any difference in amplitude or shape between the two peaks. This shows that the walls at the edge of the system can be treated acoustically as non-absorbing hard walls.

Speed of sound

Figure 4.11 shows the cross-sections at $x = 30$ for both simulations at $t = 100$. From these figures, we can numerically estimate the speed of sound in the lattice Boltzmann method and compare it to the earlier prediction of $c_s = 1/\sqrt{3} \approx 0.577$.

In Figures 4.11(a) and (b), second-order polynomial interpolation gives that the final peaks are positioned at $y = 90.7$ and $y = 88.6$, respectively. Since the sound has used

*According to Skordos, [35] this method gives a small error in the initial particle distribution. In his paper, he suggests an improved method for creating a particle distribution from an initial distribution of macroscopic variables.

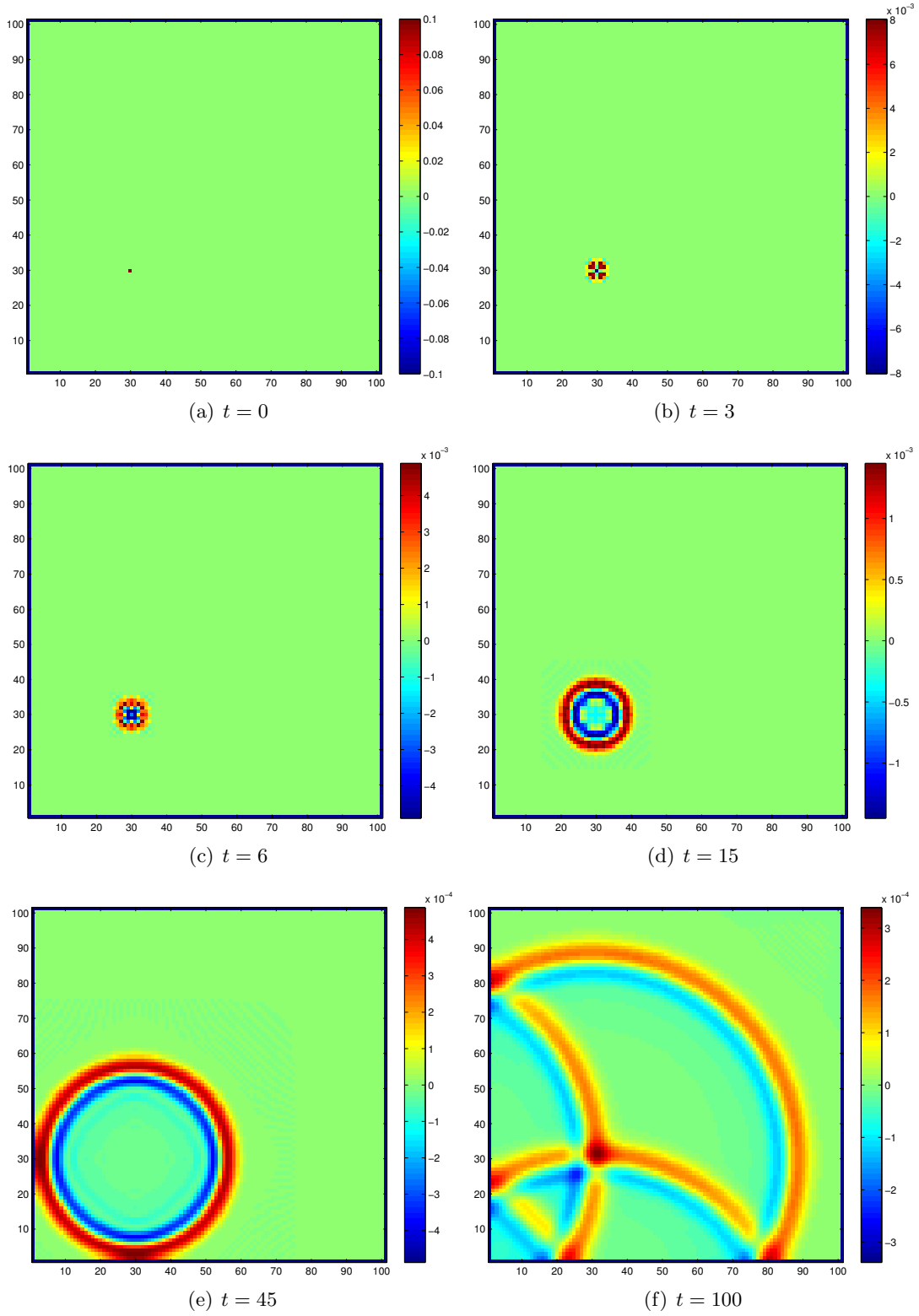


(a) $t = 0$ (b) $t = 3$

(c) $t = 6$ (d) $t = 15$

(e) $t = 45$ (f) $t = 100$

Figure 4.8: Six snapshots of off-equilibrium density in a walled system excited by a delta pulse at $(30, 30)$. Parameters are $N = 101$, $\tau = 1$.



(a) $t = 0$ (b) $t = 3$

(c) $t = 6$ (d) $t = 15$

(e) $t = 45$ (f) $t = 100$

Figure 4.9: Six snapshots of off-equilibrium density in a walled system excited by a delta pulse at $(30, 30)$. Parameters are $N = 101$, $\tau = 0.6$.

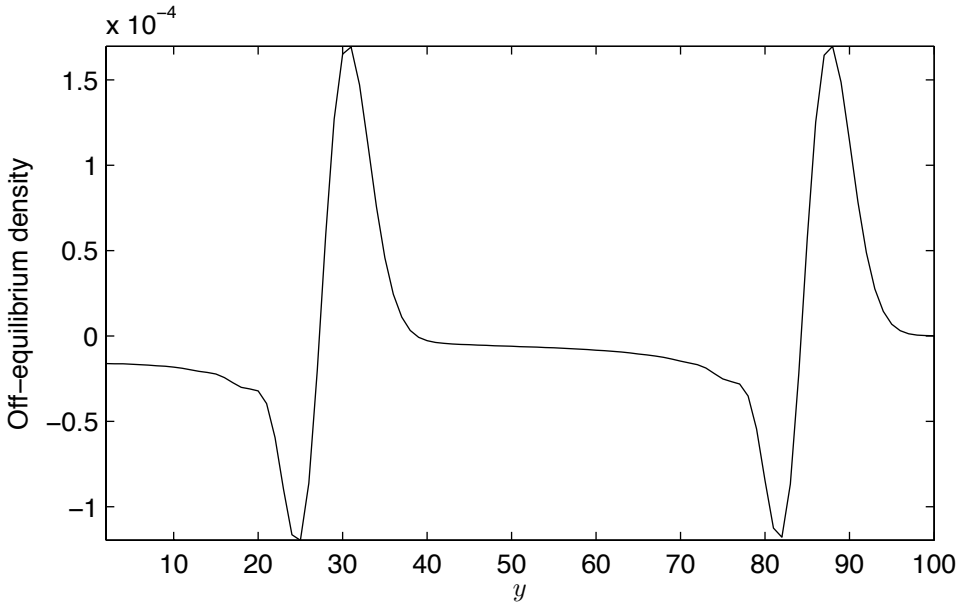


Figure 4.10: The cross-section along $x = 40$ in Figure 4.9(f).

100 time steps to propagate from $y = 30$, the two simulations give two sound speed estimates, $c_s = 0.607$ and $c_s = 0.586$, respectively. This shows that we cannot entirely trust the predicted value of c_s to be correct.

4.7 Unit conversion

There are two methods that are widely used to convert between lattice units and physical units. The simplest method is to convert directly between the two sets of units. The second method is to perform the conversion via a dimensionless system.

4.7.1 Direct conversion

In the first approach, taken by for instance ref. [4], the lattice units are related to physical units through the time step Δt and the node spacing Δx. For instance, the physical flow velocity at a node is given by

$$\vec{u}_{\mathrm{p}} = \vec{u}_{\mathrm{l}} \frac{\Delta x}{\Delta t}. \tag{4.21}$$

where $\vec{u}_{\mathrm{p}}$ is the velocity in physical units while $\vec{u}_{\mathrm{l}}$ is the velocity in lattice units. Similarly, the physical speed of sound is

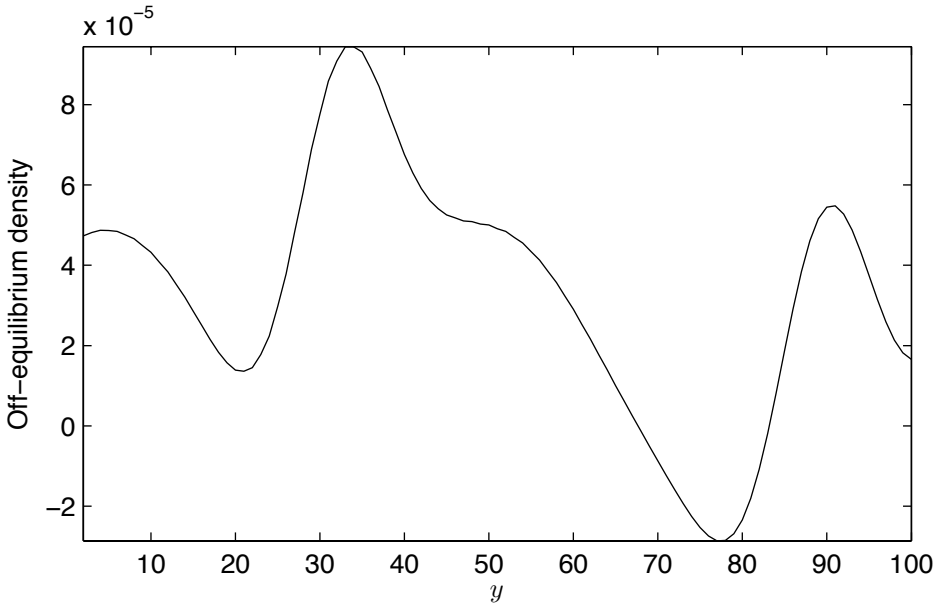
$$c_{s,\mathrm{p}} = c_{s,\mathrm{l}} \frac{\Delta x}{\Delta t}, \tag{4.22}$$

The kinematic viscosity in physical units is [4]

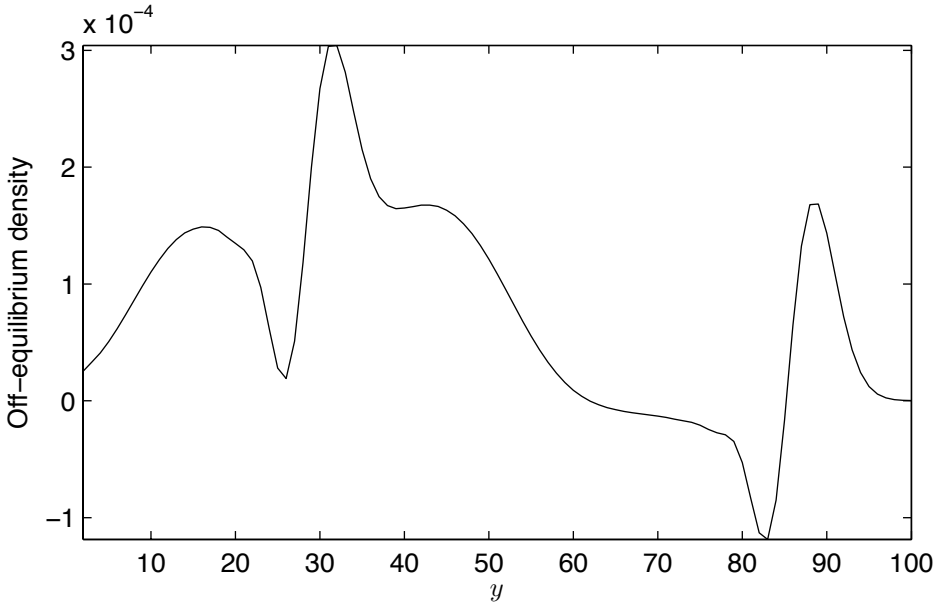
$$\nu_{\mathrm{p}} = \nu_{\mathrm{l}} \frac{\Delta x^2}{\Delta t} = c_{s,\mathrm{l}}^2 \left(\tau - \frac{1}{2}\right) \frac{\Delta x^2}{\Delta t} = c_{s,\mathrm{p}}^2 \left(\tau - \frac{1}{2}\right) \Delta t. \tag{4.23}$$

This means that the time step is given as

$$\Delta t = \frac{\nu_{\mathrm{p}}}{c_{s,\mathrm{p}}^2 \left(\tau - 1/2\right)}, \tag{4.24}$$



(a) Cross-section from Figure 4.8(f).



(b) Cross-section from Figure 4.9(f).

Figure 4.11: The cross-sections along $x = 30$ in Figures 4.8(f) and 4.9(f).

where only τ is a free variable, as ν_p and $c_{s,\mathrm{p}}$ are given by the physical system to be simulated. Similarly, using equation 4.22, we can show that the space step is given by

$$\Delta x = \frac{\nu_\mathrm{p}}{c_{s,\mathrm{l}} c_{s,\mathrm{p}} (\tau - 1/2)}. \tag{4.25}$$

Since we have assumed that we are dealing with an isothermal ideal gas, the pressure is proportional to the density through equation 2.5. This means that when ρ_0 is the equilibrium (or atmospheric) density and p_0 is the corresponding pressure,

$$\frac{p}{p_0} = \frac{\rho}{\rho_0}. \tag{4.26}$$

Both sides of this equation hold in both physical and lattice units. Because of this, we can find the physical pressure at any point as

$$p_\mathrm{p} = p_{0,\mathrm{p}} \frac{\rho_\mathrm{l}}{\rho_{0,\mathrm{l}}}. \tag{4.27}$$

4.7.2 Dimensionless formulation

The dimensionless approach is described for instance by ref. [18], which this section will largely follow. A physical system is converted to a continuous dimensionless system as shown in section 2.3, and then converted to a discrete lattice system. Going the dimensionless path is required for instance when analysing lattice Boltzmann accuracy, as described in section 5.1.

In the dimensionless system, the characteristic length and time of the system are both normalized to 1.* We divide this dimensionless system into a grid with N_ch nodes used to resolve its characteristic length. T_ch time steps are used to resolve the system's characteristic time. Space and time are then divided into intervals

$$\delta_x = 1/N_\mathrm{ch}, \tag{4.28}$$

$$\delta_t = 1/T_\mathrm{ch}. \tag{4.29}$$

These are used to convert dimensionless quantities to lattice quantities through dimensional analysis for the dimensionless system, getting

$$u_\mathrm{d} = \frac{\delta_x}{\delta_t} u_\mathrm{l}, \tag{4.30}$$

$$\nu_\mathrm{d} = \frac{1}{\mathrm{Re}} = \frac{\delta_x^2}{\delta_t} \nu_\mathrm{l}. \tag{4.31}$$

The choice of lattice size is generally speaking related to the desired precision of the model, as shown in section 5.1.

From an analysis of errors in the lattice Boltzmann method, it can be determined that to keep the error as low as possible, the relation between δ_x and δ_t must be [18]

$$\delta_t \propto \delta_x^2. \tag{4.32}$$

This is usually simple to fulfil in fluid simulations, as lattice velocities can be scaled to fit dimensionless velocities while keeping this relation. For instance, if $\delta_t = K\delta_x^2$, we have from equation 4.30 that u_l can be scaled with the system size as

$$u_\mathrm{l} = K \delta_x u_\mathrm{d} = \frac{K}{N_\mathrm{ch}} u_\mathrm{d}. \tag{4.33}$$

*Note that this means that the Reynolds number is given in the dimensionless system by $\mathrm{Re} = 1/\nu_\mathrm{d}$.

In acoustic simulations, on the other hand, we need to make heavy use of a lattice velocity which *cannot* be scaled — the speed of sound $c_{s,\mathrm{l}}$. This is not a problem in simulations of incompressible fluids, as the model's speed of sound does not play a significant role there, but it becomes problematic for lattice Boltzmann acoustics.

The dimensionless speed of sound, period, and wavelength becomes

$$c_{s,\mathrm{d}} = \frac{\delta_x}{\delta_t} c_{s,\mathrm{l}}, \tag{4.34}$$

$$T_{\mathrm{d}} = \delta_t T_{\mathrm{l}}, \tag{4.35}$$

$$\lambda_{\mathrm{d}} = \delta_x \lambda_{\mathrm{l}}. \tag{4.36}$$

This shows us that the only way to keep the dimensionless speed of sound constant without changing the system is to have $\delta_x \propto \delta_t$.

There is a way to get around this if we have system with a characteristic length and a characteristic time related to waves' wavelength and period. Let's say that we have a system with

$$\lambda_{\mathrm{d}} C_{\mathrm{length}} = l_{0,\mathrm{d}} = 1, \qquad T_{\mathrm{d}} C_{\mathrm{time}} = t_{0,d} = 1,$$

where C_{length} and C_{time} relate wavelength to characteristic length and wave period to characteristic time, respectively. If we demand that $\delta_t \propto \delta_x^2$, we can keep $c_{s,\mathrm{d}}$ constant by scaling one of these. This can be done either by

$$\frac{\delta_x^2}{\delta_t} = \frac{\lambda_{\mathrm{d}}}{\lambda_{\mathrm{l}}} \frac{c_{s,\mathrm{d}}}{c_{s,\mathrm{l}}} = \frac{1}{\lambda_{\mathrm{l}} C_{\mathrm{length}}} \frac{c_{s,\mathrm{d}}}{c_{s,\mathrm{l}}} \qquad \Leftrightarrow \qquad \lambda_{\mathrm{l}} C_{\mathrm{length}} = 1, \tag{4.37}$$

or by

$$\frac{\delta_x^2}{\delta_t} = \frac{T_{\mathrm{d}}}{T_{\mathrm{l}}} \frac{c_{s,\mathrm{d}}^2}{c_{s,\mathrm{l}}^2} = \frac{1}{T_{\mathrm{l}} C_{\mathrm{length}}} \frac{c_{s,\mathrm{d}}^2}{c_{s,\mathrm{l}}^2} \qquad \Leftrightarrow \qquad T_{\mathrm{l}} C_{\mathrm{length}} = 1. \tag{4.38}$$

This means that if $\delta_t \propto \delta_x^2$, the characteristic length of the system can have a constant relation to the wavelength of a sound wave, or the characteristic time can have a constant relation to the period. It is *impossible* to have both at the same time, as that would relate the two through the unscalable lattice velocity.

CHAPTER 5

Lattice Boltzmann boundary conditions

So far we have only talked about very simple boundary conditions (*BCs*) for the lattice Boltzmann method — solid walls which give no-slip behaviour. This is only sufficient for simulations with no external input to the system, where an initial distribution of density and velocity is left to dissipate. For more useful simulations, ways of introducing energy to the system must be discussed.

5.1 Numerical accuracy

First we need to discuss how the quality of a boundary condition is objectively determined. If a flow with a known analytic solution, for instance a Poiseuille flow, is simulated using a numerical method, the analytic solution $\vec{u}_a(\vec{r})$ can be compared with the numerical solution $\vec{u}(\vec{r})$.

For a stationary flow, the numerical method's *space accuracy* is of order k when the relation

$$\epsilon = \sqrt{\frac{\sum_{\vec{r}} |q_n(\vec{r}) - q_a(\vec{r})|^2}{N^d}} < \frac{\kappa}{N^k} \tag{5.1}$$

holds. Here, q_n is a numerically determined quantity (for instance ρ or $\vec{u}$), q_a is the analytic solution of the same quantity, N is the spatial resolution in each dimension, d the number of dimensions, and κ a constant.

For a time periodic flow with period T, the *time accuracy* is of order k when the relation

$$\epsilon = \frac{1}{T} \sum_t \sqrt{\frac{\sum_{\vec{r}} |\vec{u}(\vec{r},t) - \vec{v}(\vec{r},t)|^2}{N^d}} < \frac{\kappa'}{N^k}, \tag{5.2}$$

where κ' is a constant, holds.

From equation 5.1, k can be determined by taking the logarithm of both sides, giving

$$\log \epsilon < \log \kappa - k \log N. \tag{5.3}$$

From a plot of $\log \epsilon$ against $\log N$, k can be determined from the slope of different measurements of ϵ for different N. It is important when using different N that all

quantities are scaled so that the same dimensionless system is used, as discussed in section 4.7.2.

It has been determined that if boundaries are disregarded, the lattice Boltzmann method is of second order accuracy, meaning $k \approx 2$. [6] The space accuracy of the on-grid and mid-grid bounce-back methods described in section 4.5 are of first and second order, respectively. [6]

5.2 Zou-He pressure and density boundaries

There have been proposed many different ways of implementing pressure and density boundary conditions throughout the literature. [6, 11, 35–37] These conditions differ for instance in their method of implementation, their accuracy, and their locality.*

In this section, we will describe the Zou-He boundary condition [37] exclusively, as it has many desired properties of a pressure or density boundary condition — easy implementation, locality, and very good accuracy of second order, with a low κ. [38]

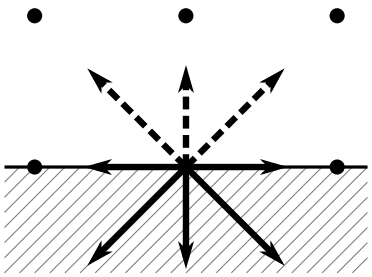


Figure 5.1: The hatched area represents the outside of the system. The known particle populations from inside the system (f_1, f_3, f_4, f_7, and f_8) are represented by solid lines, while the unknown populations from outside the system (f_2, f_5, and f_6) are dashed.

With the Zou-He BC,† particle populations f_i at a boundary which come from outside the system are set in the collision step to values which give the boundary node the desired density and velocity. Details of how the Zou-He BC is derived can be found in the paper where the boundary condition was proposed. [37]

Velocity boundary

A Zou-He velocity boundary for the D2Q9 model where a bottom node, like the one in Figure 5.1, is set to the velocity $\vec{u}$ is implemented by setting f_2, f_5, and f_6 to [37]

$$f_2 = f_4 + \frac{2}{3}\rho u_y, \tag{5.4}$$

$$f_5 = f_7 - \frac{1}{2}(f_1 - f_3) + \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y, \tag{5.5}$$

$$f_6 = f_8 + \frac{1}{2}(f_1 - f_3) - \frac{1}{2}\rho u_x + \frac{1}{6}\rho u_y, \tag{5.6}$$

where ρ is given by

$$\rho = \frac{1}{1 - u_y}\left[f_0 + f_1 + f_3 + 2(f_4 + f_7 + f_8)\right]. \tag{5.7}$$

*Locality is, as mentioned earlier, the property of being limited to single nodes.

†This also applies for the Inamuro boundary condition. [36]

The Zou-He velocity boundaries for the three other directions can be found by rotation of the lattice vector indices and velocities in equations 5.4–5.7.

Density boundary

Similarly, a Zou-He density boundary at a bottom node where the tangential velocity u_x equals 0 and ρ is specified is implemented by setting f_2, f_5, and f_6 to [37]

$$f_2 = f_4 + \frac{2}{3}\rho u_y, \tag{5.8}$$

$$f_5 = f_7 - \frac{1}{2}(f_1 - f_3) + \frac{1}{6}\rho u_y, \tag{5.9}$$

$$f_6 = f_8 + \frac{1}{2}(f_1 - f_3) + \frac{1}{6}\rho u_y, \tag{5.10}$$

where u_y is given by

$$u_y = 1 - \frac{[f_0 + f_1 + f_3 + 2(f_4 + f_7 + f_8)]}{\rho}. \tag{5.11}$$

As before, boundaries for other directions can be found by rotation of indices.

Boundary at corner node

It is necessary to pay special attention to the boundaries at their corner node ends, like shown in Figure 5.2.

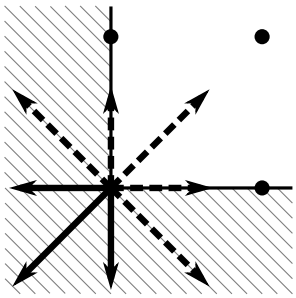


Figure 5.2: The hatched area represents the outside of the system. The known particle populations from inside the system (f_1, f_2, and f_5) are represented by solid lines, while the unknown populations from outside the system (f_3, f_4, f_6, f_7, and f_8) are dashed.

At a corner node at the inlet or outlet of a channel, it is natural to have $\vec{u} = 0$ due to no-slip conditions. When ρ is specified, the unknown particle distributions f_1, f_2, f_5, f_6, and f_8 are given by [37]

$$f_1 = f_3, \tag{5.12}$$

$$f_2 = f_4, \tag{5.13}$$

$$f_5 = f_7, \tag{5.14}$$

$$f_6 = f_8 = \frac{1}{2}[\rho - (f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_7)]. \tag{5.15}$$

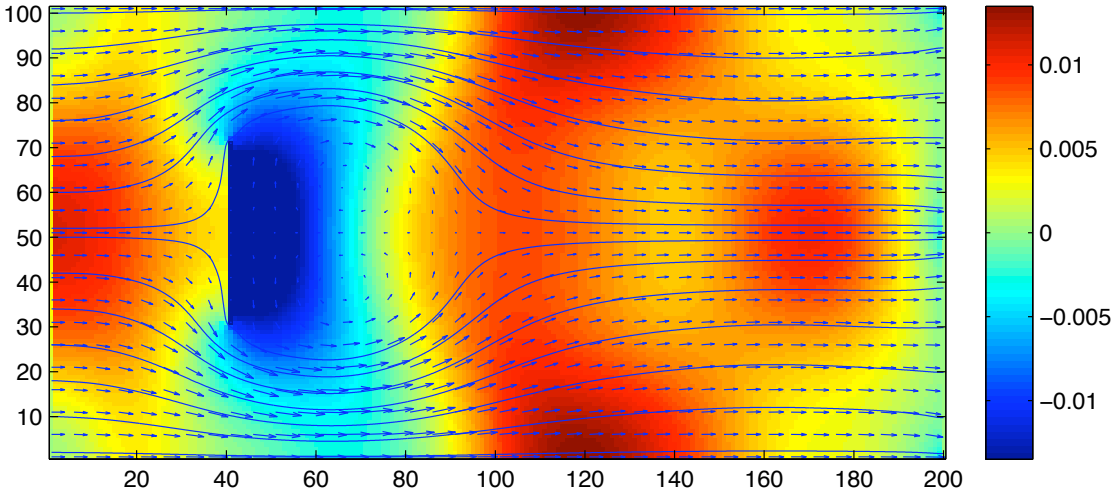


Figure 5.3: Streaming at $\tau = 0.6$ around a vertically periodic plate system with two Zou-He boundaries: $u_x = 0.025$ on the left and $\rho = 1$ on the right. The background color shows off-equilibrium density, the arrows show velocity, and the blue lines are streamlines.

Example

As a simple example of Zou-He density boundaries, a simulation has been performed where a fluid at $\tau = 0.6$ is streamed through a vertically periodic plate system of dimensions 200×101. The boundaries used are a Zou-He velocity boundary of $u_x = 0.025$ on the left, a Zou-He density boundary of $\rho = 1$ on the right, and periodic boundaries on top and bottom. The plate is implemented as a mid-grid wall from coordinate $(41, 31)$ to coordinate $(41, 71)$. The initial state of the fluid is equilibrium density at a constant velocity $u_x = 0.025$ everywhere. The result of this simulation after almost 2000 time steps is shown in Figure 5.3. Note how the fluid curves around the plate, creating vortices on its back side.

This simulation is actually unstable, as there occurs an instability at the right boundary if the simulation is left to run for long enough, causing rapid divergence of the density. This illustrates a problem with the Zou-He (and Inamuro) boundary condition — it tends to be unstable, particularly at high Reynolds numbers. [38]

5.3 Regularized boundaries

The regularized boundary is a more recent development, introduced in 2007. [11] It is based around the same concepts as the regularized lattice Boltzmann model discussed in section 7.5. Its most important advantage over the traditional boundary conditions such as Zou-He [37] and Inamuro [36] is that it remains numerically stable at far higher Reynolds numbers. [38] Its disadvantage is that while it also has second order accuracy ($k \approx 2$), it tends to be less accurate (higher κ value) than the traditional boundary conditions. An analysis by Lätt et al. [38] found that it is equally accurate in the best case, but an order of magnitude less accurate in the worst case.

For a velocity boundary at a bottom node, the density of the node is given by equation 5.7. For a similar density node, the normal velocity is given by equation 5.11.

With a free or specified tangential velocity, one now knows what the density ρ and velocity $\vec{u}$ should be at the boundary node.

The point of the regularized boundary condition is to recreate all particle distributions at the wall based on the density, the velocity and the tensor $\mathbf{\Pi}^{(1)}$, given by [11]

$$\mathbf{\Pi}^{(1)} = \sum_i \mathbf{Q}_i f_i^{(1)}, \tag{5.16}$$

where $\mathbf{Q}_i$ is defined as

$$\mathbf{Q}_i = \vec{c}_i\vec{c}_i - c_s^2\mathbf{I}, \qquad \text{or} \qquad Q_{i\alpha\beta} = c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta}. \tag{5.17}$$

Assuming that $f_i^{(1)} \approx f_i - f_i^{(0)}$ and using equation A.26, equation 5.16 becomes

$$\begin{aligned}\mathbf{\Pi}^{(1)} &\approx \sum_i \mathbf{Q}_i \left(f_i - f_i^{(0)}\right) = \sum_i \left(\vec{c}_i\vec{c}_i - c_s^2\mathbf{I}\right) f_i - \rho\vec{u}\vec{u} \\ &= \sum_i \vec{c}_i\vec{c}_i f_i - c_s^2\rho\mathbf{I} - \rho\vec{u}\vec{u}.\end{aligned} \tag{5.18}$$

This equation requires use of all f_i values, some of which are unknown at this point. The regularized boundary solves this problem in the same way as the Zou-He boundary condition. Bounceback of off-equilibrium parts, meaning $f_i = f_i^{(0)} + (f_{i'} - f_{i'}^{(0)})$ where $c_{i'} = -c_i$, is assumed to give an expression for the unknown values of f_i.

Note that it is possible to find explicit expressions for the tensor elements $\Pi_{\alpha\beta}^{(1)}$. These are given in ref. [11] for the D2Q9 and D3Q19 lattices.

With ρ, $\vec{u}$, and $\mathbf{\Pi}^{(1)}$ known, all the distribution functions f_i at the boundary node can be replaced through [11]

$$f_i = f_i^{(0)} + \frac{(1-1/\tau)t_i}{2c_s^4}\mathbf{Q}_i : \mathbf{\Pi}^{(1)}. \tag{5.19}$$

These particle distributions can then be propagated in the streaming step.

5.4 Acoustic point source

So far, the only acoustic-like behaviour we've been looking at is the propagation of the delta function source in Figures 4.8 and 4.9. These are not generally useful, as they have no familiar analytic solution and vary with the system's viscosity through τ, as seen in Figures 4.8 and 4.9.

In earlier papers on lattice Boltzmann acoustics, sound waves have been generated by setting up an initial density and velocity distribution corresponding to a wave. [17, 39–41] These papers have studied "infinite" waves by letting this initial wave distribution propagate through a periodic boundary.

For useful acoustic simulations of non-infinite waves, we need to be able to model a point source which sends out a sinusoidal signal. The simplest way to implement this is to lock a node's density to a sine function around the equilibrium density,

$$\rho\left(\vec{x}, t\right) = \rho_0 + \rho_s \sin\left(\frac{2\pi}{T}t\right), \tag{5.20}$$

where ρ_s is the point source amplitude, and T the period of the oscillation in lattice units.

Since wave propagation is a linear phenomenon and the Navier-Stokes equation can only be assumed to be linear for small disturbances, it is important that $\rho_s \ll \rho_0$, so that non-linear wave effects are avoided.

The macroscopic velocity $\vec{u}$ at the point source can be handled in two different ways. The first way is to lock it to a constant source velocity, typically $\vec{u} = 0$. The other way is to let the particles streaming into the node decide the macroscopic velocity at the point source, making the source emit particles with a velocity distribution consistent with its environment. The appropriate method to use depends on the circumstances.

The equilibrium distribution is calculated according to these values of ρ and $\vec{u}$. A similar method has also been used earlier, but in a different context. [32]

When using this boundary condition, it is important to keep in mind the workings of the BGK operator. We wish to have the density of the point source node locked to equation 5.20 at all times, so that the node always streams out the correct number of particles. With this boundary condition, the constraints of the equilibrium distribution (i.e. $\sum_i f_i^{(0)} = \sum_i f_i$) are not upheld, as ρ is constantly changed. It is therefore important to ensure that the point source node is always set to equilibrium in the collision step. This ensures that the correct number of particles is always present in the point source before streaming.

To demonstrate the behaviour of the point source, Figure 5.4 shows the state of two systems at $\tau = 1$ and $\tau = 0.6$ respectively, excited by a point source of period $T = 20$ and amplitude $\rho_s = 0.01$ in the middle of the system. It is clear that a circular wave pattern spreads from the point source. It is also apparent that the wave is more damped in the higher viscosity of the $\tau = 1$ system, as the outer part of the wave is hardly visible in Figure 5.4(e).

For the point source to be useful, it is important to ensure that it gives a behaviour which matches the analytic solution of the simulated system. In the two-dimensional LBM simulations we are performing, this point source is equivalent to a three-dimensional infinite line source in z direction, giving results independent of z. The stationary analytic solution for z-independent, cylindrically symmetric, viscously damped cylindrical waves is given in equation 2.21.

5.4.1 Single point source

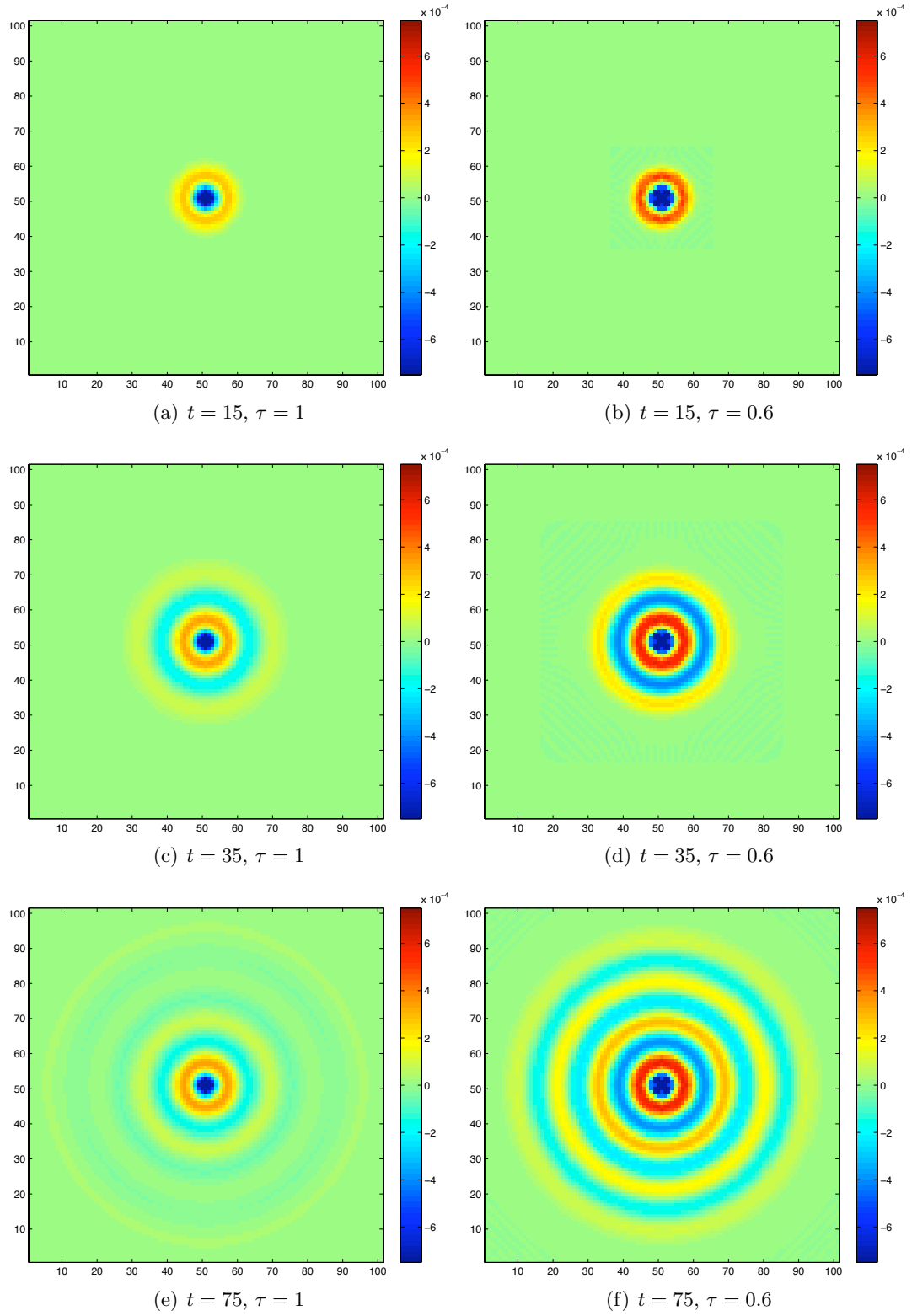
Through a comparison between the analytic solution and the numerical result, the two are shown to match very well for $\tau = 1$ and $T = 20$ if the constant A in equation 2.21 is given by

$$A = 0.135\rho_s \, \mathrm{e}^{-j2\pi/3}, \tag{5.21}$$

and the speed of sound in the expression for k (2.22) is $c_s = 0.607$, which is the value we found for $\tau = 1$ in section 4.6.1. A comparison of numerical result and analytic solution can be found in Figure 5.5.

The comparison shows that the numerical result, which is obviously transient, approaches the analytic stationary solution close to the point source at high t. At $t = 35$, there is a reasonable match between the numerical and analytic solution at the first peak, while at $t = 75$, there is a very good match between the two everywhere up to $x = 80$. It is worth noting that the match is never very good right next to the point source. This appears to be a limitation of the point source method.

What makes this result all the more impressive is that the viscous absorption coefficient α_s in k is calculated using the sum $4\nu/3+\nu'$, viscosities which have been calculated

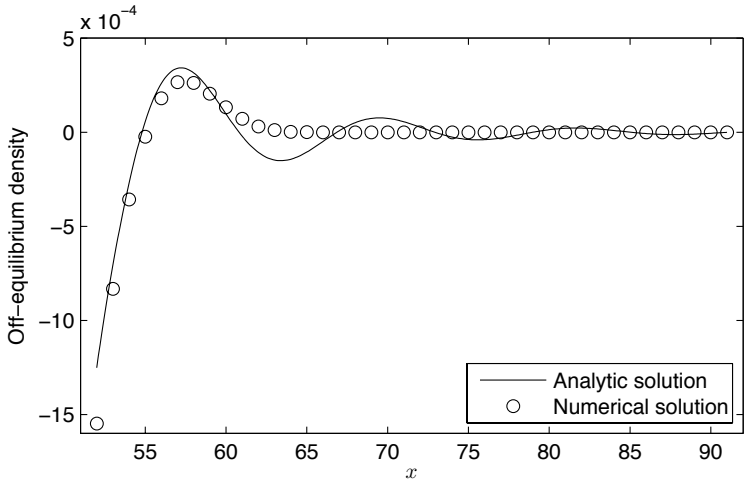


(a) $t = 15$, $\tau = 1$ (b) $t = 15$, $\tau = 0.6$

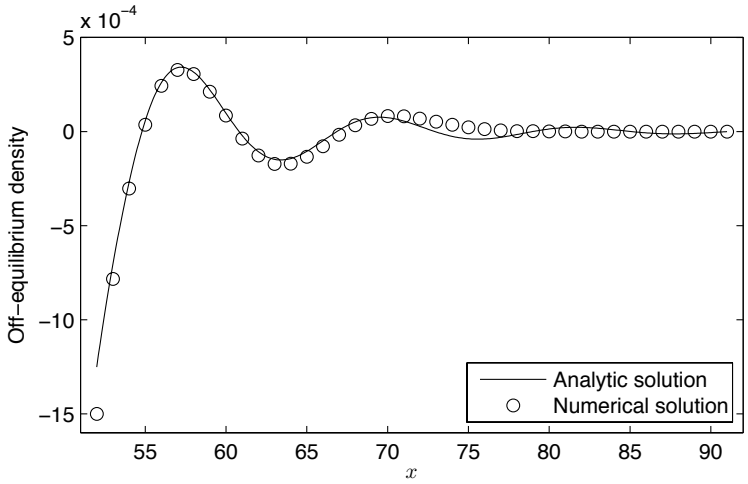
(c) $t = 35$, $\tau = 1$ (d) $t = 35$, $\tau = 0.6$

(e) $t = 75$, $\tau = 1$ (f) $t = 75$, $\tau = 0.6$

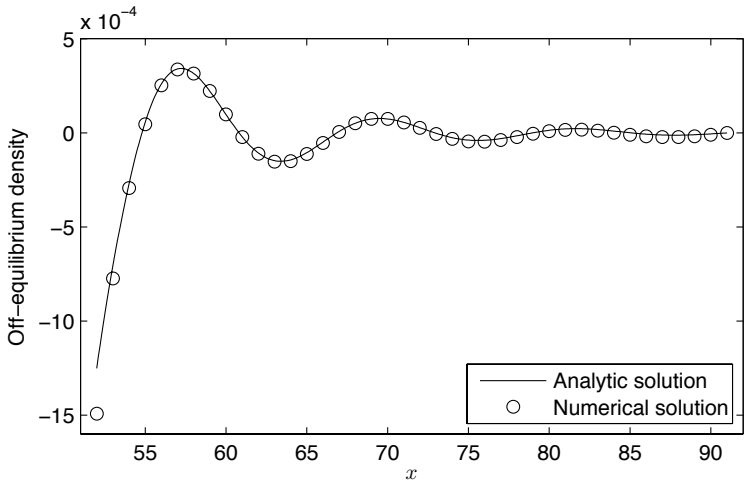
Figure 5.4: Two sets of three snapshots of off-equilibrium density in two different periodic systems excited by a point source at $(51, 51)$, with a period $T = 20$. The two systems have $\tau = 1$ and $\tau = 0.6$, while both have $N = 101$. All snapshots have been taken at the time of the point source's minimum pressure for easier comparison.



(a) $t = 15$

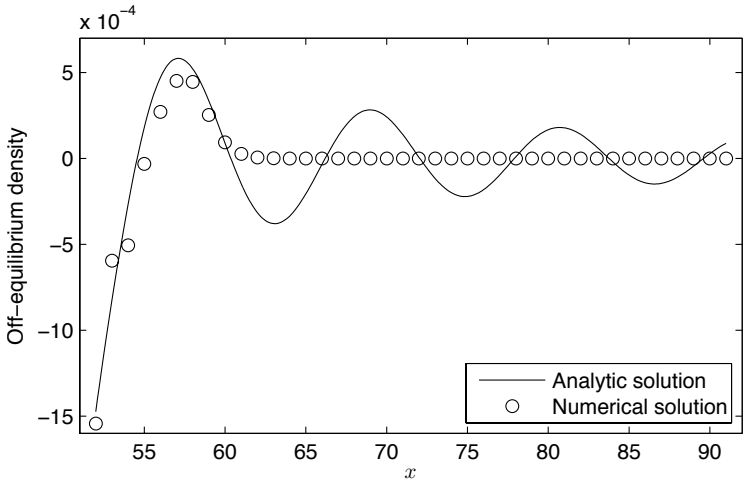


(b) $t = 35$

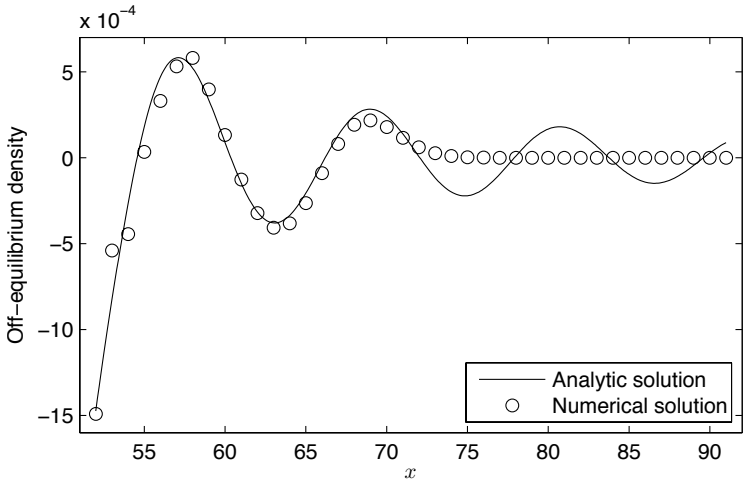


(c) $t = 75$

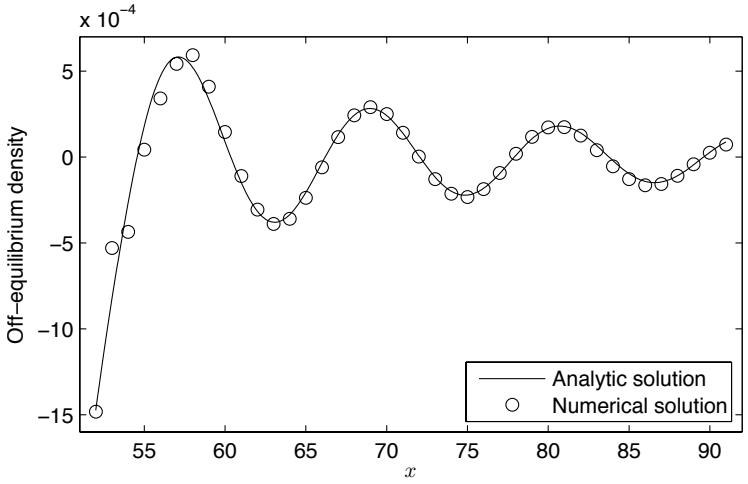
Figure 5.5: Cross-sections at $y = 51$ of the $\tau = 1$ system in Figure 5.4. The transient numerical solution is compared with the analytic stationary solution.



(a) $t = 15$



(b) $t = 35$



(c) $t = 75$

Figure 5.6: Cross-sections at $y = 51$ of the $\tau = 0.6$ system in Figure 5.4. The transient numerical solution is compared with the analytic stationary solution.

from equations 4.12 and 4.13. The result is near-perfect agreement between the attenuation of the analytic stationary solution and the numerical solution, which shows that the equations for lattice viscosity are correct.

It is also interesting to look at the same comparison for the $\tau = 0.6$ system. The calculation of the analytic solution is the same as for $\tau = 1$, except that $|A|$ must now be $0.15\rho_s$ to ensure a good match, while $c_s = 0.586$, as found in section 4.6.1. This comparison can be found in Figure 5.6.

The match between analytic and numerical solution is quite good for $\tau = 0.6$ also, but there are errors in a larger area around the source than for $\tau = 1$. This agrees well with findings in section 4.6.1, where it is found that there is more noise around a delta function source for $\tau = 0.6$ than for $\tau = 1$. It also looks like the speed of sound measured earlier for $\tau = 0.6$ is no longer entirely correct, as the analytic and numerical wavelengths are slightly different. This indicates that the speed of sound depends on more than the viscosity.

As we have seen, the proportionality constant $|A|/\rho_s$ changes with τ while $\angle A$ seems to be constant. It can also be shown that $|A|/\rho_s$ changes with the source's period T. We will not attempt to find a general expression for $|A|/\rho_s$ here.

Unfortunately, finding the numerical accuracy of the point source has not been attempted, due to time constraints.

5.4.2 Line of point sources

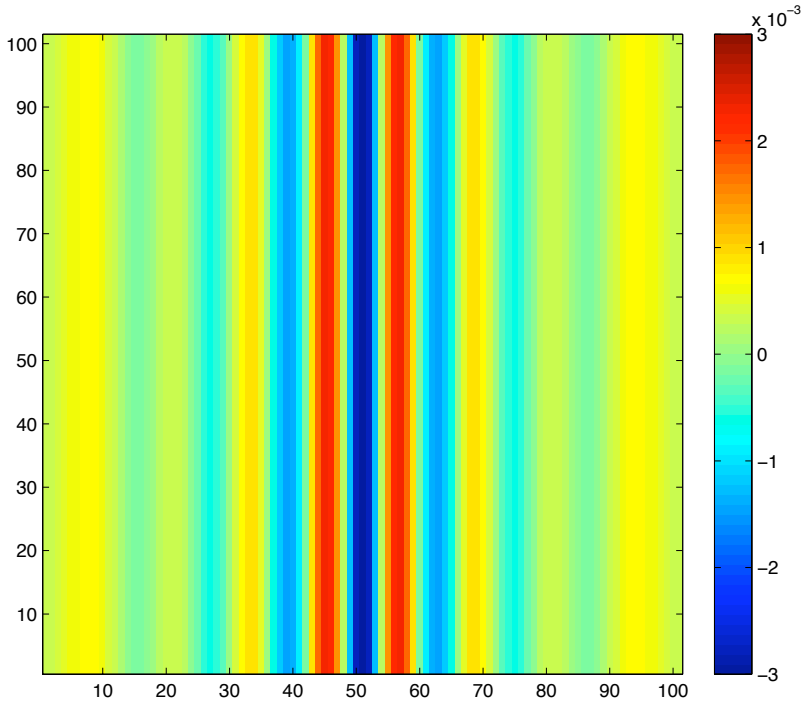
So far, we have used a single point source to simulate an infinite line source in z direction, giving a damped cylindrical wave. Placing a continuous line of point sources along $x = 51$ in a periodic system should let us simulate a plate source which is infinite in both y and z direction, giving two plane waves propagating symmetrically in x and $-x$ direction on both sides of the plate. These waves should propagate according to equation 2.18.

It turns out that this is the case. For a system and source where $\tau = 1$ and $T = 20$, we get matching behaviour for $|A|/\rho_s = 0.38$ and $\angle A = -1.67$. This is shown in Figure 5.7. There is a very good match except at the end, where the transient behaviour has not quite reached the stationary solution.

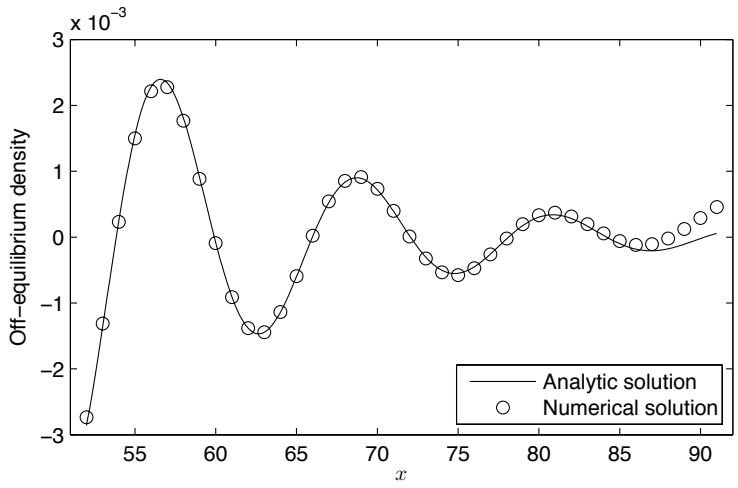
With the point sources proposed here, all particles that are streamed *into* a point source are "lost" when the point source updates its particle distribution according to its density, which is given by equation 5.20. We can use this to our advantage to create a one-sided plate source, by putting a line of point sources next to a wall. If $\vec{u}$ is locked to 0 at all point source nodes, all effects of the reflected particles will be removed. Such a system is shown in Figure 5.8.

In Figures 5.7 and 5.8, a very tall wavefront is visible as the furthest disturbance from the source. This is the first wavefront to be sent out from the source, but it is not absorbed in the same manner as the other wavefront for some unknown reason. It is in any case part of the transient, which should not be expected to match the analytic solution.

Note that since the systems in Figures 5.7 and 5.8 are periodic and completely y-invariant, they would give an identical behaviour even if $N_y = 1$. The behaviour would even be identical in the D1Q3 lattice, since it is a one-dimensional projection of the D2Q9 lattice.

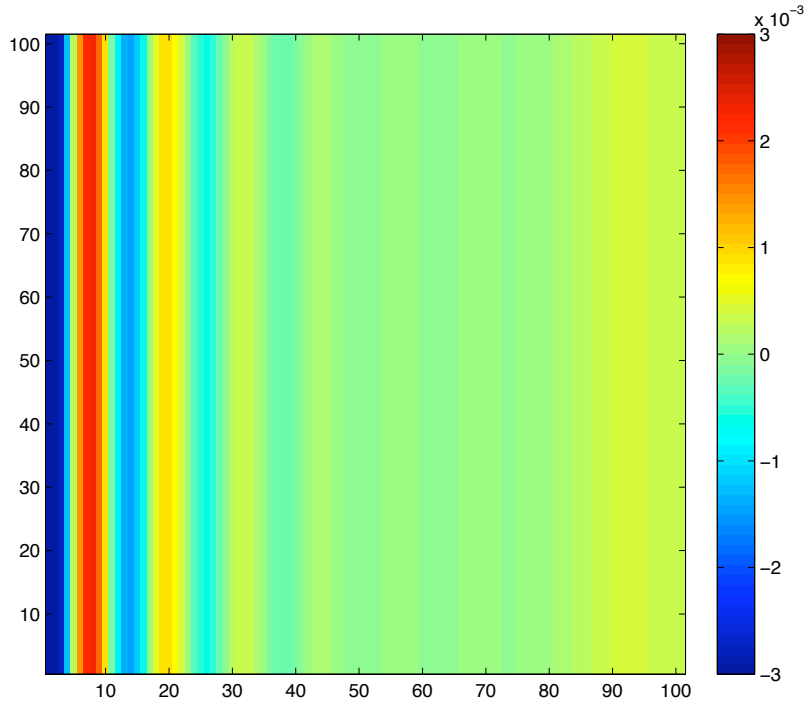


(a) View of the entire system.

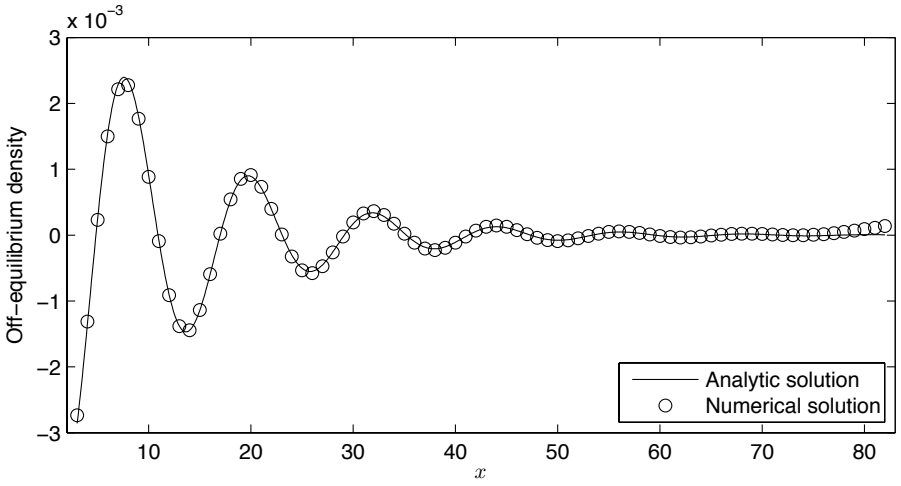


(b) Cross-section.

Figure 5.7: The state at $t = 75$ of a system with $\tau = 1$, $N = 101$, and a continuous line of point sources along $y = 51$.



(a) View of the entire system.



(b) Cross-section.

Figure 5.8: The state at $t = 155$ of a system with $\tau = 1$, $N = 101$, walls at $y = 1$, and a continuous line of point sources along $y = 2$.

CHAPTER 6

Simulations

Up until now, our simulations have been fairly theoretical, and we have only demonstrated the capabilities of the lattice Boltzmann method in very artificial cases. We will now perform somewhat more complex acoustic simulations, to see whether the lattice Boltzmann method gives us correct results in these cases.

6.1 Doppler effect

From the principle of Galilean invariance, we know that a stationary acoustic point source in a steady fluid flow of velocity $\vec{u}$ gives the same behaviour as if the point source were moving at the same velocity $\vec{u}$ in a stationary fluid. The only difference between the two cases is the frame of reference.

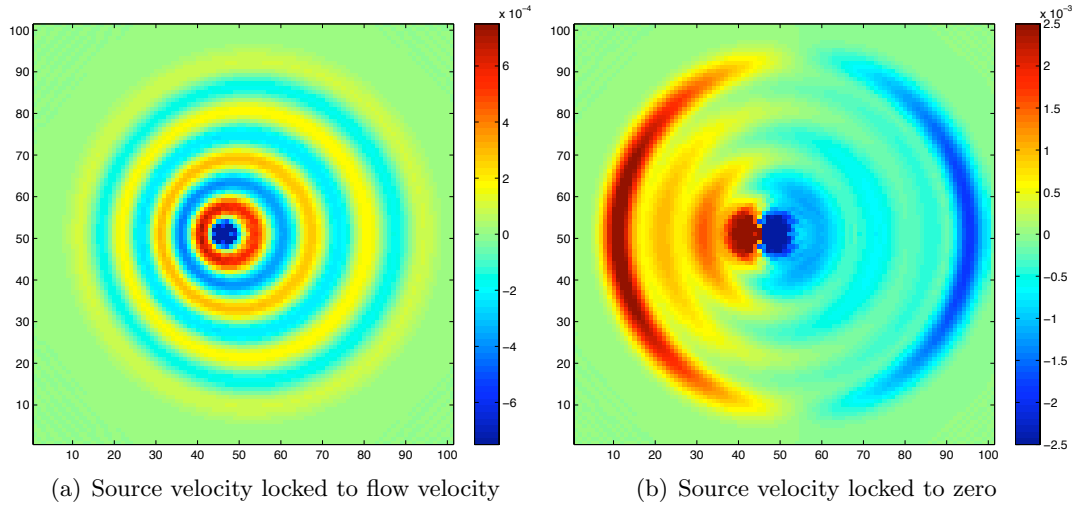
For this reason, we should get a Doppler effect if we put a point source in a steady flow. If the flow is towards the right, we should get the same behaviour as if the source was moving towards the left in an environment with no flow. This means that compared to the emitted signal, the waves from the source should have a longer wavelength on the right and a shorter wavelength on the left, essentially giving a wavefront pattern as a series of non-concentric circles. The wavefronts' centres should move with the flow velocity.

To see whether such a behaviour can be reproduced with the lattice Boltzmann model, we take a vertically periodic $\tau = 0.6$ system of dimension $N = 101$, and initialize it with a speed $u_x = 0.1$, $u_y = 0$ and Zou-He velocity boundaries on the left and right sides set to this speed. We place a point source at $(46, 51)$, emitting at $\rho_s = 0.01$ and $T = 20$.

The macroscopic flow velocity $\vec{u}$ at the node can be handled in different ways, as discussed in section 5.4. It can be locked to a constant (the two possible useful constants are $\vec{u} = (0, 0)$ and $\vec{u} = (0.1, 0)$), or it can be free, i.e. calculated from the macroscopic velocity of the particles entering into it.

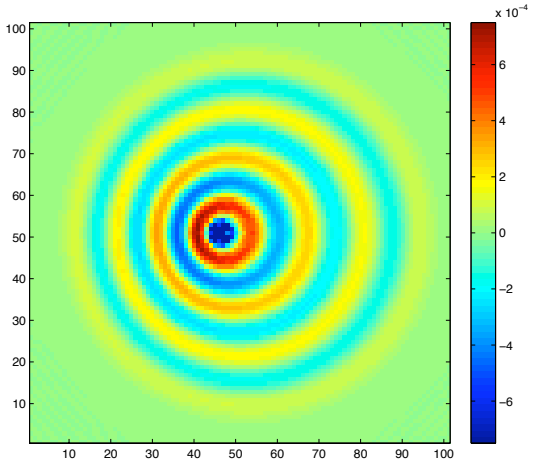
The results of this simulation, for each of the three possible velocity methods, are shown in Figure 6.1. For the source with velocity locked to the flow's velocity and the free-velocity source, the waves around the source show the expected pattern of non-concentric circles. The differences between the two methods are only barely discernible.

The result when the source velocity is locked to zero shows a different pattern of



(a) Source velocity locked to flow velocity

(b) Source velocity locked to zero



(c) Free source velocity

Figure 6.1: System state at $t = 75$ of a point source at $(46, 51)$ in an even flow of $\vec{u} = (0.1, 0)$.

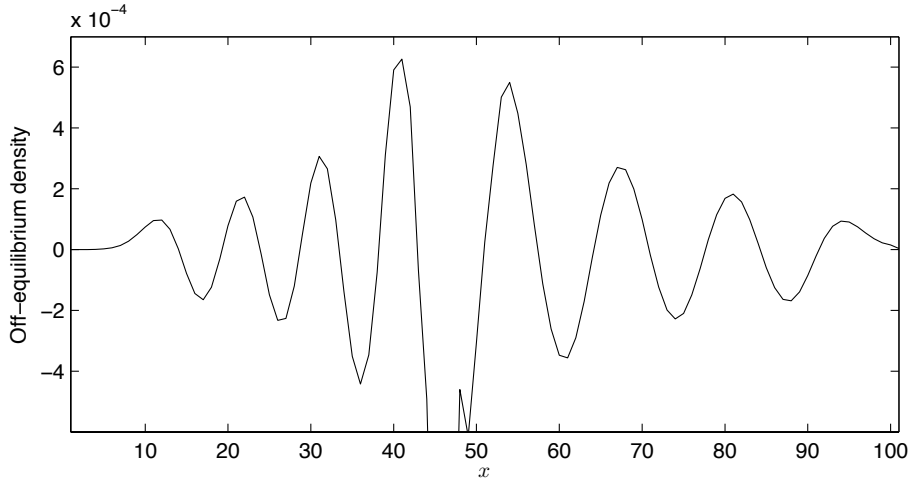


Figure 6.2: Cross-section at $y = 51$ of Figure 6.1(a).

an unfamiliar nature. The reason for the strange pattern is that the source now always sends out particles of a different velocity than its surroundings. This shows that it is important to ensure that the flow velocity in the point source is consistent with the source's environment.

The observed frequency of the signal around the point source should be [42]

$$f' = \left(\frac{c_s}{c_s - u_s} \right) f, \tag{6.1}$$

where f and f' is the emitted and observed frequency, respectively, and u_s is the source's speed towards an observer moving with the flow. Since $T = 20$, $c_s = 0.586$, and $u_{x,\,\text{flow}} = 0.1$, the wavelength observed directly in front of the point source should be

$$\lambda'_{\text{front}} = (c_s - u_{x,\,\text{flow}})\, T = 9.7, \tag{6.2}$$

while the wavelength directly behind the point source should be

$$\lambda'_{\text{back}} = (c_s + u_{x,\,\text{flow}})\, T = 13.7. \tag{6.3}$$

We can compare this to the results of our simulation. Figure 6.2 shows the cross-section at $y = 51$ and $t = 75$ of the system in Figure 6.1(a). From second-order polynomial interpolation, we find that the distinct peaks are positioned at $x = 21.6$, 31.1, 40.7, 54.0, 67.4, and 80.9. The distinct valleys are positioned at $x = 16.8$, 26.4, 35.9, 60.6, 74.1, and 87.6.

Since Figure 5.6 shows that the emitted wave from the point source does not stablilise at the first peak for $\tau = 0.6$, we should estimate the wavelength from the positions of the first and last valley on either side of the source. The distance between the two is two wavelengths. The measured value for λ'_{front} is 9.6, while the estimated value for λ'_{back} is 13.5. This is a reasonably good fit with the predicted values. The deviations might be caused by errors in the lattice Boltzmann method due to the high value of the Mach number and/or a wrong assumption about the value of the speed of sound in the system.

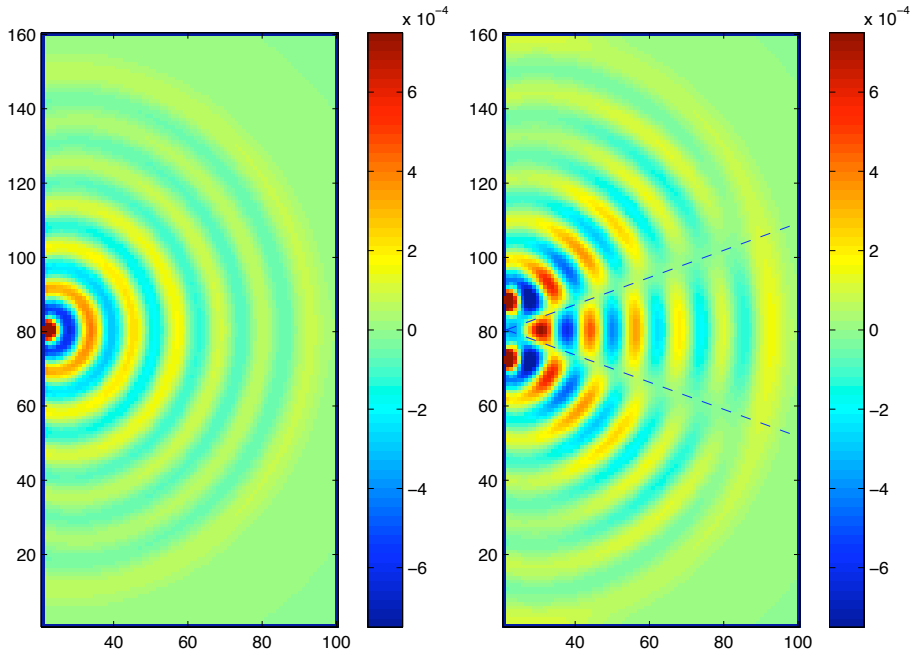
This result illustrates the power of the lattice Boltzmann method to couple acoustics and flows. There would be no particular problems with placing the point source in a more complex flow, as long as the free-velocity method is used for the point source.

6.2 Diffraction

Since the lattice Boltzmann method gives behaviour consistent with the wave equation, we should be able to simulate diffraction. This can be done by sending in a plane wave to a wall with one or more slits in it. These slits will act as point sources for the other side of the wall. If we have several slits, we should be able to see the wavefronts behind the wall interfere with each other.

To see whether diffraction can be simulated using the lattice Boltzmann method, we generate a walled 100×160 node system with a wall at $x = 21$. Slits are created as two-node openings in this wall. We place a line of point sources at $x = 2$, as described in section 5.4.2, to create a plane wave hitting the wall. The point sources emit at $\rho_s = 0.01$ and $T = 20$. To reduce viscous damping, we set $\tau = 0.6$.

Figure 6.3 shows the results of this simulation for single-slit and double-slit diffraction. For the single slit, we see that waves are radiated as if the slit were a point source.



(a) Single-slit diffraction

(b) Double-slit diffraction, with theoretical node lines

Figure 6.3: Diffraction in a $\tau = 0.6$ system excited by a line source at $x = 2$ (not shown), sending plane waves towards a wall with slits in it.

For the double slit, we see that the waves from the two slits interfere constructively at their centre line, while the interference is destructive at each side of this central maximum.

Since diffraction a behaviour general to wave mechanics, we can describe it through optical theory. The criterion for *Fraunhofer diffraction* is [43]

$$L \gg \frac{b^2}{\lambda}, \tag{6.4}$$

where b is the slit width, λ is the wavelength and L is the distance from the slits. Since this applies everywhere in our case except very close to the wall, we can use Fraunhofer diffraction theory to describe the interference pattern.

With Fraunhofer diffraction from a double slit, we find minima in the diffraction pattern at [43]

$$\sin\theta = \frac{cT(m + 1/2)}{a} \tag{6.5}$$

where m is any integer, a is the distance between the slit centers, and θ is the off-normal angle from the slit wall.

Since $a = 16$ in the system shown in Figure 6.3(b), the two first minima should be found at $\sin\theta = \pm 0.366$. Lines at this angle are shown in the figure, and we see that they match well with the node lines in the wave pattern.

6.3 Standing waves

A plane wave travelling in positive x direction is given by [13]

$$p_{a,i} = A\,\mathrm{e}^{j(\omega t - kx)}. \tag{6.6}$$

If this wave hits a hard wall at $x = 0$, a plane wave given by [13]

$$p_{a,r} = A\,\mathrm{e}^{j(\omega t + kx)} \tag{6.7}$$

is reflected at the wall, travelling in $-x$ direction. This means that the total pressure left of the wall is given by

$$p_{a,tot} = p_{a,i} + p_{a,r} = 2A\cos(kx)\,\mathrm{e}^{j\omega t}. \tag{6.8}$$

We can attempt to simulate this behaviour using the lattice Boltzmann method. Since a plane wave propagates in one dimension, we only need one dimension to simulate it.* With a periodic $N \times 1$ system with a point source at $x = 1$ and a wall node at $x = N$, plane waves will be sent from the source and reflected back at the wall. Note that we have now shifted the geometry from the equations above, as the wall is no longer at $x = 0$, but effectively at $N - 1/2$. The point source will work as a one-sided line source as described in section 5.4.2, since it has a wall to its left (the wall at $x = N$). To diminish the effects of viscous absorption, we set $\tau = 0.6$.

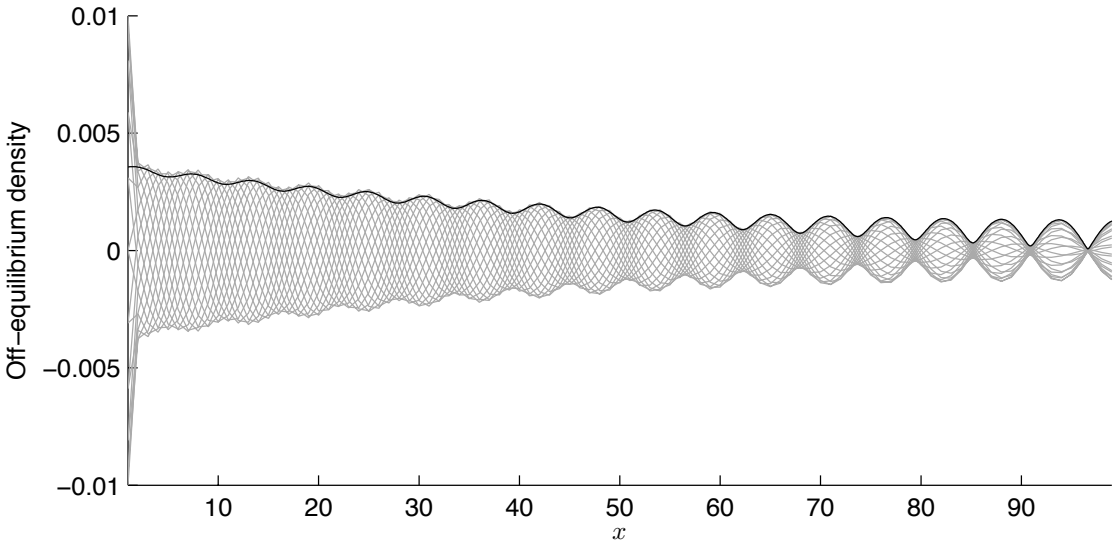


Figure 6.4: Standing wave pattern for $T = 20$. For each of the T last states of the system, one grey line is drawn. The black line on top is the analytically expected envelope.

We first perform a simulation with $N = 100$ and $T = 20$ at the point source. The system is left to run for sufficiently long to reach an equilibrium, and the results at this equilibrium are shown in Figure 6.4. We see that we have a good standing wave pattern

*In these simulations, a D2Q9 lattice is still used for simplicity, but it is here completely equivalent with using a D1Q3 lattice, due to the earlier mentioned fact that the D1Q3 lattice is a one-dimensional projection of the D2Q9 lattice.

in accordance with equation 6.8 close to the wall, while the pattern becomes less and less clear the closer we get to the source. The reason for the discrepancy between theory and simulation is that we have neglected viscous absorption. In the simulated system, the incoming and reflected waves are damped by roughly the same amount near the wall, and we have a fairly good standing wave pattern. Further away, though, the incoming waves have been damped less, and the reflected waves have been damped more. For this reason, the standing wave pattern deteriorates when we go further from the wall.

If we include absorption and shift the geometry so that the wall is at $x = L$, we can see from equation 2.18 that the incoming and reflected waves become

$$p_{a,i} = A\,\mathrm{e}^{-\alpha_s x}\,\mathrm{e}^{j(\omega t - k(x-L))}, \qquad p_{a,r} = A\,\mathrm{e}^{-\alpha_s (2L-x)}\,\mathrm{e}^{j(\omega t + k(x-L))}. \tag{6.9}$$

The total wave left of the wall becomes

$$p_{a,tot} = p_{a,i} + p_{a,r} = A\left[\mathrm{e}^{-\alpha_s x}\mathrm{e}^{j(\omega t - k(x-L))} + \mathrm{e}^{-\alpha_s (2L-x)}\mathrm{e}^{j(\omega t + k(x-L))}\right]. \tag{6.10}$$

Taking the real part of this, we find the actual, physical wave to be

$$\mathrm{Re}\left\{p_{a,tot}\right\} = A\left[\cos(\omega t - k[x-L])\,\mathrm{e}^{-\alpha_s x} + \cos(\omega t + k[x-L])\,\mathrm{e}^{-\alpha_s (2L-x)}\right]. \tag{6.11}$$

This expression can be used to calculate an envelope for the viscously absorbed standing wave pattern. It is difficult to do this analytically, but it can be done numerically by calculating equation 6.11 for a range of values of ωt, where $0 \leq \omega t \leq 2\pi$, and keeping the largest value. This can be done for all values x of interest.

Such an envelope is shown in Figure 6.4. Since the position of a wall is between the wall node and the fluid node next to it, the envelope was calculated for $L = 99.5$. Also, since A is an undefined constant, it was set to a value which gives a good match between the envelope and the wave pattern at the node closest to the wall.

Another constant which had to be tweaked for the envelope to match the experiment was the speed of sound, c_s. It can be calculated from the figure that the wavelength for these standing waves is 11.48 nodes. This means that $c_s = \lambda/T = 0.574$, which differs from the earlier measurement of $c_s = 0.586$ at $\tau = 0.6$ in section 4.6.1. Compared to the theoretical prediction of $c_s = 0.577$, this measured value is actually *smaller*. This tells us that the speed of sound in the material is not only dependent on τ.

Figure 6.4 shows a very good match between the theoretical and numerical envelope close to the wall, but we see that the match worsens closer to the point source. The reason for this may be the effect of the point source on the reflected wave. When the reflected wave meets the point source, it can no longer propagate unhindered, which might cause another reflection.

Equations 2.19 and 2.20 show that α_s is proportional to ω^2 when $\omega\tau_s \ll 1$, which is the case here.* The absorption exponent over a wavelength then becomes

$$\alpha_s \lambda \propto \omega^2 \frac{1}{\omega} \propto \frac{1}{T}.$$

This tells us that the absorption per wavelength decreases as $1/T$ in the limit of low $\omega\tau_s$, meaning that we can achieve a better standing wave pattern by scaling the system.

We double the scale of the system, setting $N = 200$ and $T = 40$. The speed of sound in the model was now measured to $c_s = 0.5758$. The results and the analytic envelope

*In this simulation, $\omega\tau_s = 4\pi/T(\tau - 0.5) = 0.063$.

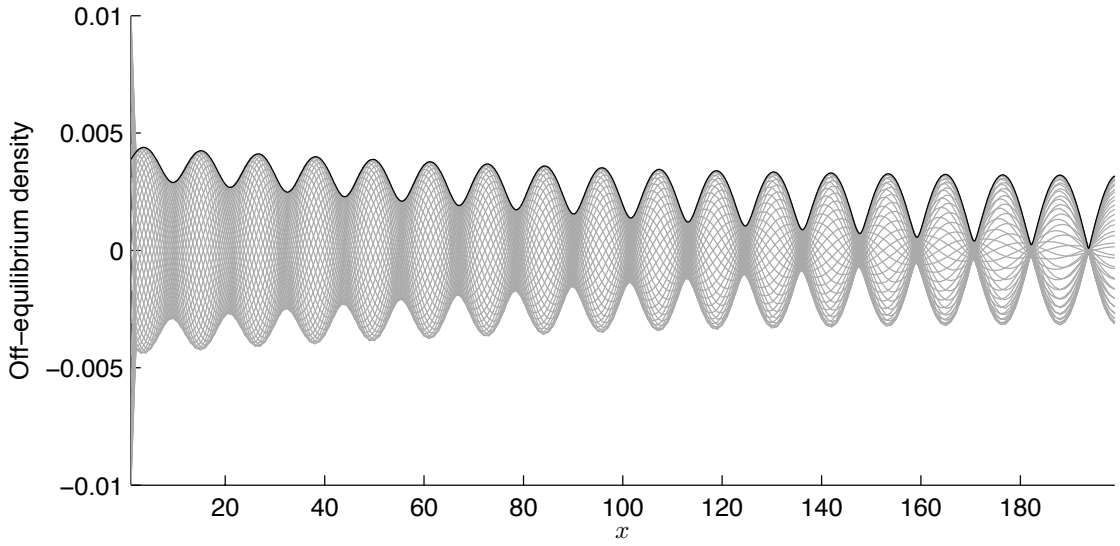


Figure 6.5: Standing wave pattern for $T = 40$. For each of the T last states of the system, one grey line is drawn. The black line on top is the analytically expected envelope.

are shown in Figure 6.5. We see that we now have a cleaner standing wave pattern than in Figure 6.4, and there is a very good match between the analytic and numerical envelope everywhere.

Note that in both Figure 6.4 and Figure 6.5, the first node has a density amplitude which does not match the rest of the wave. This is likely due to the fact that of the particles which stream out of this node, half of these stream to the right into the system, while the other half stream to the left, are reflected by the wall node at $x = N$, return and are overwritten. In other words, quite a few of the particles that contribute to the source node's density do not enter the system.

It might be possible to rectify this by initializing the source node with the correct speed. This means that one would need to use the specific acoustic impedance of the system, which is given by the complex number $z = p/u$. (z is complex because the phase difference between p and u.) Unfortunately, there has not been time to attempt this.

CHAPTER 7

Alternative lattice Boltzmann models

In this chapter we will give an overview of different variations of the lattice Boltzmann method. These variations range from fundamental changes to the the method itself to correction terms which extend its possibilities.

7.1 Chopard-Luthi wave model

Previously, we have used a lattice Boltzmann method which simulates the compressible Navier-Stokes equation. We have used this method at low excursions from equilibrium to simulate acoustic wave behaviour. It is not strictly necessary to go via the Navier-Stokes equation to simulate waves, as lattice Boltzmann models which simulate wave behaviour directly exist.

We will first go through the Chopard-Luthi wave model, first presented in Pascal Luthi's 1998 PhD thesis, supervised by Bastien Chopard. [32] Chopard later released several publications with different co-authors, describing this model. [1, 23, 44, 45]

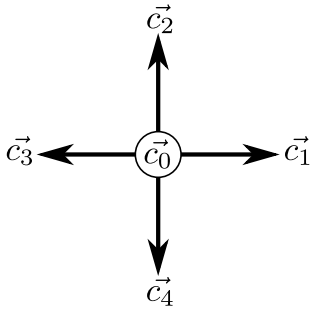


Figure 7.1: The D2Q5 lattice used in the Chopard-Luthi wave model.

This wave model uses a lattice which is uncommon in lattice Boltzmann: The D2Q5 lattice, which is essentially the HPP lattice with a rest particle vector. It does not fulfil the conditions of isotropy, but as we will see later, this will not matter.

The wave model is a variation of the BGK method, and a relaxation to equilibrium

is still given by the BGK collision operator. The evolution equation remains

$$f_i\left(\vec{x}+\vec{c_i},t+1\right)=\left(1-\frac{1}{\tau}\right)f_i\left(\vec{x},t\right)+\frac{1}{\tau}f_i^{(0)}\left(\vec{x},t\right). \tag{7.1}$$

As noted in section 4.4, the two last terms in the BGK equilibrium distribution function (4.15) are not needed to preserve the correct mass and momentum in the node. Indeed, since wave propagation is a linear phenomenon, it makes sense with a linear equilibrium distribution, [44]

$$f_i^{(0)}=\rho\left[a_i+b\frac{\vec{u}\cdot\vec{c_i}}{2}\right]. \tag{7.2}$$

Here, a_i and b are constants. For $i=1,2,3,4$, a_i is a single constant a. Since $\sum_i \vec{c_i}=0$ and $\sum_i c_{i\alpha}c_{i\beta}=2\delta_{\alpha\beta}$, conservation of ρ and $\rho\vec{u}$ requires that [44]

$$a_0+4a=1, \qquad b=1. \tag{7.3}$$

From a multi-scale Chapman-Enskog expansion like the one performed in Chapter A, it can be shown that the wave model gives a behaviour [1]

$$\partial_t\rho+\partial_\alpha\rho u_\alpha=0, \tag{7.4}$$

$$\partial_t\rho u_\alpha+2a\partial_\alpha\rho+(2\tau-1)\left[a\partial_\alpha\overrightarrow{\nabla}\cdot(\rho\vec{u})-\frac{1}{4}T_{\alpha\beta\gamma\delta}\partial_\beta\partial_\gamma\rho u_\delta\right]=0, \tag{7.5}$$

where $T_{\alpha\beta\gamma\delta}=\sum_i c_{i\alpha}c_{i\beta}c_{i\gamma}c_{i\delta}$. This tensor is not isotropic, but we can remove this problem by choosing $\tau=1/2$, turning equation 7.5 into

$$\partial_t\rho u_\alpha+2a\partial_\alpha\rho=0. \tag{7.6}$$

We have said earlier that $\tau=1/2$ is unstable. According to ref. [45], this is only the case for the D2Q9 lattice with this wave model, while the D2Q5 and D2Q7 lattices are numerically stable for $\tau=1/2$.

Subtracting the x_α derivative of 7.6 from the t derivative of equation 7.4, we get

$$\partial_t^2\rho-2a\nabla^2\rho=0, \tag{7.7}$$

which is a wave equation for ρ, with a propagation speed $c_s=\sqrt{2a}$. This shows that the Chopard-Luthi wave model gives wave equation behaviour when $\tau=1/2$.

From this and equation 7.3, we see that the velocity in the model is adjustable. The maximum possible velocity in the model, $c_{s,0}=1/\sqrt{2}$, is obtained when $a_0=0$ and $a=1/4$. An index of refraction n for the model can be defined as

$$n=\frac{c_{s,0}}{c_s}=\frac{1}{2\sqrt{a}}. \tag{7.8}$$

By inserting equations 7.8 and 7.3 into equation 7.2, and inserting that into equation 7.1, it can be shown that the evolution equation for the Chopard-Luthi wave model becomes

$$f_0\left(\vec{x}+\vec{c_i},t+1\right)=2\frac{n^2-1}{n^2}\rho-f_0\left(\vec{x},t\right), \tag{7.9}$$

$$f_i\left(\vec{x}+\vec{c_i},t+1\right)=\frac{1}{2n^2}\rho-f_{i'}\left(\vec{x},t\right) \qquad \text{for } i\neq 0. \tag{7.10}$$

i' is here the opposite index to i, so that $\vec{c_{i'}} = -\vec{c_i}$.

If n equals 1 and f_0 equals 0 everywhere, this is equivalent with the transmission line matrix (TLM) method, which is a discrete Huygens model which has been used to simulate the propagation of acoustic waves. [46] This is quite interesting, as the TLM method is derived from entirely different concepts than this lattice Boltzmann wave model.

While this wave model is interesting, it does not play to all the strengths of the lattice Boltzmann method, in particular its strength of being able to couple flows and waves. This is a general drawback of lattice Boltzmann wave models. Also, there unfortunately does not exist a great deal of literature on this model, so using the more well-tried TLM model might be a better choice for massively parallel wave propagation simulations.

7.2 Yan wave model

Another wave model for lattice Boltzmann was presented by Yan Guangwu in 2000. [47] This model is quite different from both the Chopard-Luthi wave model and the basic lattice Boltzmann method.

The basic premise of the model is that it defines the excursion $u(\vec{x}, t)$, where

$$\frac{\partial u(\vec{x}, t)}{\partial t} = \rho(\vec{x}, t). \tag{7.11}$$

u is then updated for each time step as

$$u(\vec{x}, t+1) = u(\vec{x}, t) + \frac{\partial u(\vec{x}, t)}{\partial t}. \tag{7.12}$$

The evolution equation in the article is given by

$$f_i\left(\vec{x} + c_i, t+1\right) - f_i\left(\vec{x}, t\right) = -\frac{1}{\tau}\left[f_i\left(\vec{x}, t\right) - f_i^{(0)}\left(\vec{x}, t\right)\right] + \phi\left(\vec{x}, t\right), \tag{7.13}$$

where $\phi\left(\vec{x}, t\right)$ is related to the source function and will be explained later.

A derivation in the article shows that with these conditions, the method gives behaviour according to the inhomogeneous wave equation,

$$\frac{\partial^2}{\partial t^2} u(\vec{x}, t) = c_s^2 \nabla^2 u(\vec{x}, t) + \psi(\vec{x}, t), \tag{7.14}$$

if the equilibrium distribution is given by

$$f_0^{(0)} = \rho - \lambda u d, \tag{7.15}$$

$$f_i^{(0)} = \frac{\lambda u d}{b}, \qquad \text{for } i \neq 0. \tag{7.16}$$

Here, $\psi(\vec{x}, t) = (b+1)\phi(\vec{x}, t)$ is a source term, $\lambda = c_s^2/(\tau - 1/2)$, d is the number of dimensions, and b is the number of non-zero particle velocities.*

While this model is also interesting, it is less successful than the Chopard-Luthi wave model at playing to the lattice Boltzmann method's strengths. It lacks the conceptual simplicity of lattice Boltzmann, since the density functions f_i no longer represent particle densities, but instead movements of the time derivative of the excursion. It is far more difficult to visualise what goes on in this model.

*For instance, in the D2Q9 lattice, $b = 8$.

7.3 External forces

In certain simulations, it is important to be able to include external forces. Gravity, for instance, plays an important role in fluid mechanics, and is necessary to incorporate in certain models to achieve the correct behaviour. Certain subfields of fluid mechanics, for instance magnetohydrodynamics, are totally dependent on external forces. This is also the case for certain other lattice Boltzmann models, such as the one described in section 7.6.

There are several ways of incorporating external forces on a node-per-node basis in the lattice Boltzmann method. For instance, the book *Lattice Boltzmann Modeling* [4] presents a method based on tweaking the velocity used to calculate the equilibrium density.

Another method of incorporating force was presented by Guo, Zheng and Shi in 2002, and was shown to be superior to the other commonly used force methods of that time. [48] It works by adding a force term F_i to the evolution equation, so that it is given by

$$f_i\left(\vec{x}+\vec{c_i}, t+1\right) - f_i\left(\vec{x}, t\right) = \Omega_i\left(\vec{x}, t\right) + F_i. \tag{7.17}$$

The force term is given by [48]

$$F_i = \left(1 - \frac{1}{2\tau}\right) t_i \left(\frac{\vec{c_i} - \vec{u}}{c_s^2} + \frac{\vec{c_i}\cdot\vec{u}}{c_s^4}\vec{c_i}\right)\cdot\vec{F}. \tag{7.18}$$

7.4 Adjustable bulk viscosity

In the derivation in Appendix A, we show that the basic lattice Boltzmann method with no correction terms give a behaviour consistent with the compressible Navier-Stokes equations when the kinematic shear and bulk viscosities are given by

$$\begin{aligned}\nu &= c_s^2\left(\tau - \frac{1}{2}\right),\\ \nu' &= \frac{2}{3}\nu.\end{aligned}$$

One unfortunate problem with this is that the bulk viscosity is locked to the shear viscosity. As physical fluids have different bulk viscosities, it is necessary for proper simulation of compressible flow that the bulk viscosity can be adjusted.

A method of setting the bulk viscosity of the system to a given value was first presented by Paul Dellar in 2001. [12] He showed that by adding a correction term to the evolution equation similar to the force term given in section 7.3, the bulk viscosity in the model could be adjusted. In Jonas Lätt's 2007 PhD thesis, he showed how this could be implemented more effectively, and also gave a correction for the interference between the bulk viscosity term and the force correction term given in equation 7.18. [11]

Lätt's bulk viscosity correction term is given by

$$B_i = \frac{ct_i}{2dc_s^4}\left(|\vec{c_i}|^2 - c_s^2 d\right)\left(-\frac{2c_s^2}{t_0}f_0^{(1)} + k\vec{F}\cdot\vec{u}\right), \tag{7.19}$$

where d is the number of spatial dimensions, and c and k are constants. k is given by

$$k = \frac{1}{1+c\tau}. \tag{7.20}$$

The viscosity is included through the constant c. When corrected to use the definition of bulk viscosity used in this text, the constant is given by

$$c = \frac{1}{\tau - \frac{2\tau^2 c_s^2}{d(\nu'+2/3)}}. \tag{7.21}$$

$f_0^{(1)}$ is given by

$$f_0^{(1)} = f_0 - \rho t_0. \tag{7.22}$$

7.5 Regularized model

In the derivation of the Navier-Stokes equation from lattice Boltzmann dynamics, given in Appendix A, we find $f_i^{(1)}$ by neglecting certain $\mathcal{O}(\mathrm{Ma}^3)$ terms. We also show that only one of the terms in the expression for $f_i^{(1)}$ is necessary to find the Navier-Stokes equation.

The regularized model, described in refs. [11, 49, 50], is a way to suppress these unnecessary terms by replacing the BGK collision operator with an operator that neglects them. This results in an evolution equation given by [50]

$$f_i\left(\vec{x}+\vec{c_i}, t+1\right) = f_i^{(0)} - \frac{\left(1-1/\tau\right) t_i}{2c_s^4} \sum_j \mathbf{Q}_i : \left[\vec{c_j}\vec{c_j}\left(f_i - f_i^{(0)}\right)\right], \tag{7.23}$$

where $\mathbf{Q}_i$ is defined as

$$\mathbf{Q}_i = \vec{c_i}\vec{c_i} - c_s^2\mathbf{I}, \qquad \text{or} \qquad Q_{i\alpha\beta} = c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta}. \tag{7.24}$$

This regularized model has been shown to give a better accuracy of near-third order and numerical stability at higher Reynolds numbers than the BGK model. [11]

7.6 Multiphase and multicomponent models

An often-quoted strength of the lattice Boltzmann method is the relative ease with which fluids with several components and/or several phases can be simulated.

The most popular method* of doing this is the Shan-Chen model, introduced in 1993 and further detailed by the authors in 1994. [51, 52] In this method, different particle distribution functions f_i^σ co-exist, where σ denotes one of S different components.

Forces between different components can be incorporated through nearest-neighbour interactions between nodes. The force on component σ in the node at $\vec{x}$ is given by

$$\vec{F}^\sigma(\vec{x}) = -\psi^\sigma(\vec{x}) \sum_{\bar{\sigma}} \mathcal{G}_{\sigma\bar{\sigma}} \sum_i \vec{c_i}\psi^{\bar{\sigma}}(\vec{x}+\vec{c_i}), \tag{7.25}$$

where $\psi^\sigma(\vec{x})$, which is a function of ρ^σ, is the effective number density of particles of component σ at $\vec{x}$, and $\mathcal{G}_{\sigma\bar{\sigma}}$ controls the interaction strength between component σ and component $\bar{\sigma}$. ψ^σ can be given by different functions of ρ^σ according to which physics one wishes to simulate. $\mathcal{G}_{\sigma\bar{\sigma}}$ is a $S \times S$ matrix which has to be symmetric to conserve net momentum. Depending on the value of $\mathcal{G}_{\sigma\sigma}$, phase transitions may also occur in the model. The force $\vec{F}^\sigma(\vec{x})$ can be applied through the force term described in section 7.3.

For more information on the Shan-Chen model, the reader is referred to the articles. [51, 52] Other multiphase/multicomponent models also exist. [53–55]

*It is the most-cited method of those mentioned in this section, as determined through ISI Web of Knowledge.

7.7 Thermal models

Another strength of the lattice Boltzmann method is the ability to use thermal models to simulate thermohydrodynamics. This is done by expanding the lattice Boltzmann collision operator Ω_i to conserve not only mass and momentum, but also the internal energy E, given by [56]

$$\rho E = \sum_i \frac{(\vec{c_i} - \vec{u})^2}{2} f_i. \tag{7.26}$$

The conservation requires that [56]

$$\sum_i \frac{\vec{c_i}^2}{2} \Omega_i = 0. \tag{7.27}$$

Achieving such a conservation requires an expanded lattice Boltzmann model in several ways. Although the details vary between implementations, the lattice commonly has an extended set of lattice vectors, the equilibrium distribution goes to the third order in $\vec{u}$, and the multi-scale Chapman-Enskog expansion of f_i, as seen in equation A.4, goes one order further in order to regain the energy equation.

Several different papers exist which describe methods to achieve this. Some of these are given in refs. [56–58]. Note that a general problem of these thermal models is that they tend to be more numerically unstable than the basic lattice Boltzmann method. [6]

CHAPTER 8

Discussion

So far, we've looked at the theory behind the lattice Boltzmann metod, including alternative models. We've taken a look at a method of implementing a point source which sends out cylindrical waves in 2D, and which sends out plane waves in 1D.

The lattice Boltzmann method, when used for acoustics, can be seen as a discrete Huygens method. This is a set of methods based on Huygens' principle, which states that any point on a wavefront is, in itself, a source of new waves. A discrete Huygens method is a numerical method, working on a grid, which reproduces the characteristics of Huygens' principle.

At a lattice Boltzmann node in equilibrium with its neighbours, an equal particle exchange is performed between the node and its neighbours, resulting in the particle distributions at that node being unchanged from one time step to the next.

When a wavefront reaches this node, more particles than usual will stream into it from the node behind. In the collision step, this particle distribution will be relaxed towards an equilibrium, resulting in the node streaming more particles than usual in the direction of the wavefront, and fewer particles than usual in the opposite direction. In this respect, the lattice Boltzmann method is similar to other discrete Huygens methods such as the TLM method described in ref. [46].

The most important difference between the lattice Boltzmann method and the TLM method is that the TLM method is a wave equation solver, while the lattice Boltzmann method is a full Navier-Stokes equation solver. This has both advantages and disadvantages for LB acoustics. One advantage is the fact that the LBM is ideal to simulate acoustics in complex flows, or to simulate viscously damped acoustics. A disadvantage is that viscosity is inescapable in the standard lattice Boltzmann model. It can be made arbitrarily low by letting $\tau \to 0.5$, but at the price of stability and accuracy.

8.1 Units

This is the elephant in the room, the big problem that we have not touched on yet. We described how to go between lattice units and physical units in section 4.7.1, but we have done no calculations on this yet.

From equations 4.24 and 4.25, we see that the time step Δt and space step Δx are uniquely determined from the physical characteristics of sound speed $c_{s,\mathrm{p}}$ and kinematic

shear viscosity ν_p, together with the BGK operator's relaxation time τ. We see that these steps are largest for materials with a high kinematic shear viscosity and a low speed of sound. Table 8.1 gives an overview of the viscosities and speeds of sound of different materials, together with their ratios from equations 4.24 and 4.25.

Table 8.1: Physical viscosities and speeds of sound in different fluids, together with relevant ratios. Numbers are taken from ref. [13].

Material	ν_p [Pa · s]	$c_{s,\mathrm{p}}$ [m/s]	$\nu_\mathrm{p}/c_{s,\mathrm{p}}$ [Pa · s^2/m]	$\nu_\mathrm{p}/c_{s,\mathrm{p}}^2$ [Pa · s^3/m^2]
Air (20°C)	$1.53 \cdot 10^{-5}$	343	$4.46 \cdot 10^{-8}$	$1.30 \cdot 10^{-10}$
O_2 (0°C)	$1.40 \cdot 10^{-5}$	317	$4.41 \cdot 10^{-8}$	$1.39 \cdot 10^{-10}$
CO_2 (0°C)	$7.32 \cdot 10^{-6}$	258	$2.83 \cdot 10^{-8}$	$1.10 \cdot 10^{-10}$
H_2 (0°C)	$9.78 \cdot 10^{-5}$	1270	$7.70 \cdot 10^{-8}$	$6.06 \cdot 10^{-11}$
Freshwater (20°C)	$1.00 \cdot 10^{-6}$	1481	$6.75 \cdot 10^{-10}$	$4.56 \cdot 10^{-13}$
Castor oil (20°C)	$1.01 \cdot 10^{-3}$	1540	$6.56 \cdot 10^{-7}$	$4.26 \cdot 10^{-10}$

The materials that are most suitable for lattice Boltzmann simulation are those with high values of $\nu_\mathrm{p}/c_{s,\mathrm{p}}$ and $\nu_\mathrm{p}/c_{s,\mathrm{p}}^2$. Gases tend to have a fairly high kinematic shear viscosity and a low speed of sound, giving reasonably high values for these ratios. Liquids generally have a higher speed of sound, which means that only highly viscous liquids are suitable. As we can see from the table, water is quite unsuitable.

Now, let's calculate the lattice Boltzmann time and space steps for air at $\tau = 0.6$. Equations 4.24 and 4.25 give

$$\Delta t = 1.30 \cdot 10^{-9}\,\mathrm{s},$$
$$\Delta x = 7.72 \cdot 10^{-7}\,\mathrm{m}.$$

This is an important result — it means that to simulate one period of a 20 kHz wave would require almost 40 000 time steps, and to simulate an area of one square millimeter would require a lattice of roughly 1300×1300 nodes.

The number of time steps that can be performed per node per second is known as one site update per second (su/s). A benchmark on the Ranger supercomputer at the Texas Advanced Computing Center, a supercomputer with 4096 64-bit AMD Opteron processor cores has given a speed of 3.5 Gsu/s for the well-optimized OpenLB C++ library. At this speed, a simulation with the number of time steps and system size given in the previous paragraph would take almost 20 seconds to perform. And this is a tiny two-dimensional system, simulated on a powerful supercomputer!

This shows us that it is *completely* unfeasible to perform lattice Boltzmann simulations in air for anything but ultrasound at very small spatial scales, at least with today's computers.

The problem can be alleviated in a few ways. First, if simulations are done on a highly viscous fluid with a relatively low speed of sound, the time and space resolutions Δt and Δx will increase. Perhaps large hydrocarbons or crude oil could be feasible and useful to simulate, but this would probably not increase the time and space steps by many orders of magnitude.

If a simulation is done in a system which is largely wall nodes, for instance porous stone, the problem is also alleviated somewhat, as site updates are not performed for these nodes.

Another way to alleviate the problem is to reduce the relaxation time τ, but this has a serious drawback: the reduction of stability and accuracy. Increasing the time and space steps by one order of magnitude would for instance mean changing τ from 0.6 to 0.51.

8.2 Potential and problems

In section 5.4 and Chapter 6, we have shown that it is quite possible to achieve an impressive match between lattice Boltzmann simulations and analytic solutions of acoustic behaviour. This proves that acoustic simulations with the lattice Boltzmann method are possible.

We have seen that wall nodes give the same acoustic behaviour as a hard wall right between the wall node and the fluid node beside it. We have seen that a simple point source in a two-dimensional system can give behaviour consistent with a three-dimensional infinite line source propagating cylindrical sound waves into a viscously absorbing environment, with a very good match between the predicted and measured viscosity, as seen by the viscous damping of the sound waves. We have seen that this point source can be placed in a line in a two-dimensional system, giving behaviour consistent with a three-dimensional infinite plate source propagating plane waves. We have seen that this point source can be used to model more complex acoustic behaviour such as the Doppler effect, diffraction and standing waves.

The question is, what kind of physical systems can be simulated? As we saw in the previous section, it is not feasible to use the lattice Boltzmann method for acoustics simulations of lower frequencies than ultrasound, or in large systems. If the lattice Boltzmann method is to be used in acoustics, it is in ultrasound simulations in small geometries, particularly geometries which are mostly wall nodes. One such use is simulation of transmitted and received sound from an ultrasound microbot in a blood vein. Another is simulation of ultrasound propagated through porous stone.

Direct problems

We have also seen direct problems with simulations from the lattice Boltzmann acoustic point source. First and foremost, we have the varying speed of sound in the model. While theory indicates that the speed of sound should be $c_s = 1/\sqrt{3} \approx 0.577$, measured values from simulations in this text have been as high as $c_s = 0.607$ and as low as $c_s = 0.574$. The sound speed changing from case to case is a significant problem, as one can never be sure what the speed of sound in the model will be without measuring it in a finished simulation.

It is possible that this variation in sound speed can be fixed through some correction to the lattice Boltzmann model, but it is not clear to the author what kind of correction would keep the speed of sound constant.

Another problem is the way in which we have matched our analytic and numerical solutions in section 5.4 and Chapter 6. The matching has been done through comparing amplitude and phase of the analytic and numerical solution, and adapting the magnitude and phase of the analytic solution to match the two. Without a general expression for this magnitude and phase, it is impossible to know exactly what a point source is simulating before analysing the result.

It should not be impossible to find a match for this this. For instance, plane waves should match the amplitude from the source if the source is initialized with the correct

particle velocity. This requires using the specific acoustic impedance, $z = p/u$,* for a viscous medium. For the cylindrical waves from a simulated line source, the match might be possible to find through the relation of the analytic solution near the source and the value which the point source is set to. Perhaps it is something as simple as the integral of the Hankel function in equation 2.21 in the area of the source node.

8.3 Looking forward

If the point source is to be used, it needs to be benchmarked to find its order of accuracy. A series of similar simulations must be performed at different resolutions, and the results must be compared with analytic solutions.

Which simulations should be performed to accomplish this is another question. One possibility might be to put one point source in the middle of a two-dimensional system and set the boundaries of the system to their analytic solution using similar point sources along the entire boundary. The system should be left to evolve to an equilibrium and the result should then be compared with the analytic solution.

A less demanding method could be to do this in one dimension only, with point sources at both sides of the system, propagating right-moving plane waves. These sources should be set to different densities, velocities, and phases, according to the analytic solution of the system.

The biggest problem with lattice Boltzmann acoustics is that low simulated viscosity, stability, and a large simulated space are incompatible. This is a general problem with viscosity in computational fluid dynamics — when the viscosity is lowered, finer details in the flow become apparent, and a finer grid is needed to faithfully simulate these details. Therefore, it is unrealistic to hope that a vastly improved lattice Boltzmann variant will appear to simulate low viscosity more accurately on the same grid, although the regularized lattice Boltzmann model might represent an improvement.

What *can* be hoped for is a more efficient version of the lattice Boltzmann method, perhaps tailor-made for acoustics, so that larger systems can be simulated more quickly than before. Also, it is certain that processor power will continue to increase. A variant of Moore's law, which has shown to be fairly accurate so far, predicts that processor performance per cost will double every 24 months. This means that processor performance will be increased by a factor of ~ 32 in ten years, and a factor of ~ 1024 in twenty years. While this is an impressive increase, it is still too small for us to expect to be able to feasibly perform acoustic lattice Boltzmann simulations of tangible acoustic systems for decades to come.

*Note that the specific acoustic impedance is a complex number, since phase is included in p and u.

CHAPTER 9

Conclusion

We have seen that the lattice Boltzmann method gives a behaviour consistent with the Navier-Stokes equation, and by extension, the lossy wave equation. A simple point source has been proposed, and has been shown able to give behaviour consistent with an infinite line source producing viscously damped cylindrical waves. It can also be used to simulate an infinite plate source, producing viscously damped plane waves.

A very good match between these simulated cylindrical and plane waves and their analytic counterparts has been found. The predicted shear and bulk viscosity from the lattice Boltzmann model has been put into these analytic solutions to predict viscous absorption, which has been shown to match the viscous absorption in the simulations with very good accuracy.

The point source has been used to perform lattice Boltzmann simulations of wave behaviour such as the Doppler effect, diffraction, and standing waves. There has unfortunately not been time to perform accuracy benchmarks on the point source to find its order of accuracy.

When used to perform simulations of acoustics, the lattice Boltzmann method can be seen as a discrete Huygens method, meaning that it treats each point at a wavefront as a source of new wavefronts. This places it in the same category as the TLM method, which has been used to perform simulations of pure wave propagation, both electromagnetic and acoustic.

The lattice Boltzmann method has been shown capable of simulating acoustic behaviour, but this is only feasible at ultrasound frequencies and at very small spatial scales. At lower frequencies or larger scales, calculations will take prohibitively long to perform.

APPENDIX A

Derivation of Navier-Stokes from LBM dynamics

As mentioned in earlier chapters, it is possible to show from the dynamics of the lattice Boltzmann method that it gives a behaviour consistent with the compressible Navier-Stokes equation (2.7). Unfortunately, few sources go into detail on this. Thorough derivations can be found for instance in refs. [1, 3, 11, 23], but the derivations are invariably complicated and/or terse. Therefore, we shall here try to give a derivation of this which is as simple as possible. This derivation will largely follow ref. [11], but will be more explicit. It will also make a simplification by assuming no external force, $\vec{f} = 0$. This derivation is quite long, but this is unfortunately inescapable.

As in section 3.4, where we derived the equation of continuity from LGA dynamics, we start by Taylor expanding the lattice Boltzmann evolution equation (4.1),

$$\begin{aligned}\Omega_i(\vec{x},t) &= f_i(\vec{x}+\vec{c_i},t+1) - f_i(\vec{x},t) \\ &\approx \underbrace{\left(\partial_t + \overrightarrow{\nabla}\cdot\vec{c_i}\right) f_i}_{\text{First order of expansion}} + \underbrace{\frac{1}{2}\left(\partial_t^2 + 2\partial_t\overrightarrow{\nabla}\cdot\vec{c_i} + \overrightarrow{\nabla}\overrightarrow{\nabla} : \vec{c_i}\vec{c_i}\right) f_i}_{\text{Second order of expansion}}. \end{aligned} \tag{A.1}$$

Here we have used equation 2.3 for the spatial derivatives and equation 2.1 to rewrite $(\overrightarrow{\nabla}\cdot\vec{c_i})^2$ as $\overrightarrow{\nabla}\overrightarrow{\nabla} : \vec{c_i}\vec{c_i}$, as well as the definition of a Taylor series for a function of two variables.*

A.1 Multi-scale Chapman-Enskog expansion

Now we will use a technique familiar from non-linear dynamics, and separate the time scale t into two different time scales, t_1 and t_2, so that $f_i(\vec{x},t) \to f_i(\vec{x},t_1,t_2)$. We define t_1 to be the time scale for fast phenomena such as advection, while t_2 is the time scale for slow phenomena such as diffusion. [6]

Using perturbation analysis, we can now expand the time derivative ∂_t using a smallness parameter ϵ, as

$$\partial_t = \epsilon\partial_{t_1} + \epsilon^2\partial_{t_2} + \mathcal{O}\left(\epsilon^3\right). \tag{A.2}$$

*This can for instance be found at `http://mathworld.wolfram.com/TaylorSeries.html`.

ϵ is here a smallness factor that ensures that the time derivative of the slow phenomena is smaller than the time derivative of the fast ones, while ∂_{t_1} and ∂_{t_2} themselves are of the same order of magnitude. We similarly expand the space derivative, but only for one term,

$$\overrightarrow{\nabla} = \epsilon \overrightarrow{\nabla}_1 + \mathcal{O}\left(\epsilon^2\right). \tag{A.3}$$

Now we do a similar perturbative expansion on f_i around the equilibrium distribution $f_i^{(0)}$, and get

$$f_i = f_i^{(0)} + \epsilon f_i^{(1)} + \mathcal{O}\left(\epsilon^2\right). \tag{A.4}$$

Finally, the expansion is done on the collision operator Ω_i, giving

$$\Omega_i = \Omega_i^{(0)} + \epsilon \Omega_i^{(1)} + \epsilon^2 \Omega_i^{(2)} + \mathcal{O}\left(\epsilon^3\right). \tag{A.5}$$

In the field of lattice Boltzmann, the expansion we have done here is commonly known as the Chapman-Enskog expansion, and the analysis we will be doing is called Chapman-Enskog analysis.*

Inserting equations A.2, A.3, and A.4 into equation A.1, we get

$$\begin{aligned}\Omega_i = &\left(\epsilon\partial_{t_1} + \epsilon^2\partial_{t_2} + \epsilon\overrightarrow{\nabla}_1\cdot\vec{c_i} + \frac{\epsilon^2}{2}\partial_{t_1}^2 + \epsilon^2\partial_{t_1}\overrightarrow{\nabla}_1\cdot\vec{c_i} + \frac{\epsilon^2}{2}\overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \vec{c_i}\vec{c_i}\right)\left(f_i^{(0)} + \epsilon f_i^{(0)}\right)\\ &+ \mathcal{O}\left(\epsilon^3\right).\end{aligned}$$

Gathering the terms on the right side according the order of their ϵ prefactor, we get

$$\begin{aligned}\Omega_i &= \epsilon\left[\left(\partial_{t_1} + \overrightarrow{\nabla}_1\cdot\vec{c_i}\right)f_i^{(0)}\right]\\ &\quad + \epsilon^2\left[\left(\partial_{t_1} + \overrightarrow{\nabla}_1\cdot\vec{c_i}\right)f_i^{(1)} + \left(\partial_{t_2} + \frac{1}{2}\partial_{t_1}^2 + \partial_{t_1}\overrightarrow{\nabla}_1\cdot\vec{c_i} + \frac{1}{2}\overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \vec{c_i}\vec{c_i}\right)f_i^{(0)}\right]\\ &\quad + \mathcal{O}\left(\epsilon^3\right)\\ &= \Omega_i^{(0)} + \epsilon\Omega_i^{(1)} + \epsilon^2\Omega_i^{(2)} + \mathcal{O}\left(\epsilon^3\right).\end{aligned}$$

Now, since the terms of different ϵ order are of entirely different orders of magnitude, we can consider them independent of each other. Therefore, we can separate the terms of different orders of magnitude into

$$\Omega_i^{(1)} = \left(\partial_{t_1} + \overrightarrow{\nabla}_1\cdot\vec{c_i}\right)f_i^{(0)}, \tag{A.6}$$

$$\Omega_i^{(2)} = \left(\partial_{t_1} + \overrightarrow{\nabla}_1\cdot\vec{c_i}\right)f_i^{(1)} + \left(\partial_{t_2} + \frac{1}{2}\partial_{t_1}^2 + \partial_{t_1}\overrightarrow{\nabla}_1\cdot\vec{c_i} + \frac{1}{2}\overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \vec{c_i}\vec{c_i}\right)f_i^{(0)}. \tag{A.7}$$

As there are no terms of order ϵ^0, it is clear that $\Omega_i^{(0)} = 0$. We will see that going to the second order in Ω_i, as we have done here, will be sufficient to recover the Navier-Stokes equation.

*This is named after the two physicists Sydney Chapman and David Enskog, who in the period of 1916-1917 independently used such an expansion to find a solution to the Boltzmann equation. [59]

A.2 Applying conservation properties

Let us now take another look at the conservation laws we have defined earlier in equations 4.7 and 4.8:

$$\rho = \sum_i f_i = \sum_i f_i^{(0)}, \tag{A.8}$$

$$\rho\vec{u} = \sum_i \vec{c}_i f_i = \sum_i \vec{c}_i f_i^{(0)}. \tag{A.9}$$

We also define the second order moment tensor* $\mathbf{\Pi}$ as

$$\mathbf{\Pi} = \sum_i \mathbf{Q}_i f_i, \tag{A.10}$$

where $\mathbf{Q}_i$ is defined as

$$\mathbf{Q}_i = \vec{c}_i\vec{c}_i - c_s^2\mathbf{I}, \qquad \text{or} \qquad Q_{i\alpha\beta} = c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta}. \tag{A.11}$$

BGK operator

Let's also take another look at the BGK collision operator (4.5). Having done the expansion of f_i around $f_i^{(0)}$ in equation A.4, the collision operator can be rewritten as

$$\Omega_i = -\frac{1}{\tau}\left[f_i - f_i^{(0)}\right] = -\frac{1}{\tau}\left[\epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} + \ldots\right]. \tag{A.12}$$

Equations 4.6a and 4.6b state that $\sum_i \Omega_i = \sum_i \vec{c}_i\Omega_i = 0$. From these equations and equation A.12 it is clear that

$$\sum_i \Omega_i^{(k)} = -\frac{1}{\tau}\sum_i f_i^{(k)} = 0 \qquad \text{for } k > 0, \tag{A.13}$$

$$\sum_i \vec{c}_i\Omega_i^{(k)} = -\frac{1}{\tau}\sum_i \vec{c}_i f_i^{(k)} = 0 \qquad \text{for } k > 0. \tag{A.14}$$

In other words, the BGK operator gives that the zeroth and first order moments of the off-equilibrium distribution must disappear.

Zeroth order moments

Now, we'll start using these properties. Taking the zeroth order moment of equation A.6, we get

$$\sum_i \Omega_i^{(1)} = \partial_{t_1}\sum_i f_i^{(0)} + \overrightarrow{\nabla}_1 \cdot \sum_i \vec{c}_i f_i^{(0)} = 0$$

$$\Rightarrow \boxed{\partial_{t_1}\rho + \overrightarrow{\nabla}_1 \cdot \rho\vec{u} = 0.} \tag{A.15}$$

This shows us that the LBM dynamics give us a continuity equation (2.6) for the time scale t_1.

*$\sum_i f_i$ is also known as the zeroth order moment of f_i, while $\sum_i \vec{c}_i f_i$ is known as the first order moment of f_i.

Now we take the zeroth order moment of equation A.7, and get

$$\sum_i \Omega_i^{(2)} = \partial_{t_1} \sum_i f_i^{(1)} + \overrightarrow{\nabla}_1 \cdot \sum_i \vec{c}_i f_i^{(1)} + \partial_{t_2} \sum_i f_i^{(0)} + \frac{1}{2}\partial_{t_1}^2 \sum_i f_i^{(0)} + \partial_{t_1} \overrightarrow{\nabla}_1 \cdot \sum_i \vec{c}_i f_i^{(0)} + \frac{1}{2}\overrightarrow{\nabla}_1 \overrightarrow{\nabla}_1 : \sum_i \vec{c}_i \vec{c}_i f_i^{(0)} = 0.$$

The two first terms on the right side disappear due to A.13 and A.14. The last term can be rewritten using the tensor $\mathbf{Q}_i$. Conservation laws A.8 and A.9 are used for the other terms. This gives us that

$$\partial_{t_2}\rho + \frac{1}{2}\partial_{t_1}^2\rho + \partial_{t_1}\overrightarrow{\nabla}_1 \cdot \rho\vec{u} + \frac{1}{2}\overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \sum_i \left(\mathbf{Q}_i + c_s^2\mathbf{I}\right) f_i^{(0)} = 0.$$

From equation 2.4, we have that $\overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \mathbf{I} = \overrightarrow{\nabla}_1^2$, giving us that

$$\boxed{\partial_{t_2}\rho + \partial_{t_1}\overrightarrow{\nabla}_1 \cdot \rho\vec{u} + \frac{1}{2}\left(\partial_{t_1}^2\rho + \overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \mathbf{\Pi}^{(0)} + c_s^2\overrightarrow{\nabla}_1^2\rho\right) = 0.} \tag{A.16}$$

First order moments

We take the first order moment of equation A.6. Using equation 2.2 and the same arguments as above, this gives

$$\sum_i \vec{c}_i\Omega_i^{(1)} = \partial_{t_1}\sum_i \vec{c}_i f_i^{(0)} + \overrightarrow{\nabla}_1 \cdot \sum \vec{c}_i\vec{c}_i f_i^{(0)} = 0$$

$$\Rightarrow \boxed{\partial_{t_1}\rho\vec{u} + \overrightarrow{\nabla}_1 \cdot \mathbf{\Pi}^{(0)} + c_s^2\overrightarrow{\nabla}_1\rho = 0.} \tag{A.17}$$

The last moment of Ω_i which must be calculated is the first order moment of equation A.7, which becomes

$$\sum_i \vec{c}_i\Omega_i^{(2)} = \partial_{t_1}\sum_i \vec{c}_i f_i^{(1)} + \overrightarrow{\nabla}_1 \cdot \sum_i \vec{c}_i\vec{c}_i f_i^{(1)} + \partial_{t_2}\sum_i \vec{c}_i f_i^{(0)} + \frac{1}{2}\partial_{t_1}^2\sum_i \vec{c}_i f_i^{(0)} + \partial_{t_1}\overrightarrow{\nabla}_1 \cdot \sum \vec{c}_i\vec{c}_i f_i^{(0)} + \frac{1}{2}\overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \sum_i \vec{c}_i\vec{c}_i\vec{c}_i f_i^{(0)} = 0$$

$$\Rightarrow \boxed{\overrightarrow{\nabla}_1 \cdot \mathbf{\Pi}^{(1)} + \partial_{t_2}\rho\vec{u} + \frac{1}{2}\partial_{t_1}^2\rho\vec{u} + \partial_{t_1}\overrightarrow{\nabla}_1 \cdot \mathbf{\Pi}^{(0)} + c_s^2\partial_{t_1}\overrightarrow{\nabla}_1\rho + \frac{1}{2}\overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \mathbf{R}^{(0)} = 0.} \tag{A.18}$$

$\mathbf{R}^{(0)}$ is a 3D tensor with components $R^{(0)}_{\alpha\beta\gamma} = \sum_i c_{i\alpha}c_{i\beta}c_{i\gamma}f_i^{(0)}$.

Finding the continuity equation

With the zeroth and first order moments found, we can start combining these results. Subtracting one half of the t_1 derivative of equation A.15 from equation A.16 gives

$$\partial_{t_2}\rho + \frac{1}{2}\left(\partial_{t_1}\overrightarrow{\nabla}_1 \cdot \rho\vec{u} + \overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \mathbf{\Pi}^{(0)} + c_s^2\overrightarrow{\nabla}_1^2\rho\right) = 0.$$

Since the contents of the parenthesis is the divergence of equation A.17, we can simply combine the two to get that

$$\boxed{\partial_{t_2}\rho = 0.} \tag{A.19}$$

Let's take a reality check on that result for a moment. We've said earlier that t_2 is the time scale for slow phenomena such as diffusion, while t_1 is the time scale for fast phenomena, such as advection. It makes intuitive sense that the redistribution of mass is a fast phenomenon, since mass is carried along with the velocity of the medium. Diffusive processes, on the other hand, merely exchange particles between different parts of the fluid, without a net transportation of mass, which is what equation A.19 also states.

Now, since this equation states that $\partial_t\rho = \epsilon\partial_{t_1}\rho$, equation A.15 shows us that the continuity equation

$$\boxed{\partial_t\rho + \overrightarrow{\nabla}\cdot\rho\vec{u} = 0} \tag{A.20}$$

is given by the dynamics of lattice Boltzmann. This is really a natural result of the fact that the continuity equation is an expression of the conservation of mass, which is guaranteed by the dynamics of the LBM.

Towards the momentum conservation equation

We'll get rid of the second derivative of t_1 in equation A.18 by subtracting from it one half the t_1 derivative of equation A.17, resulting in

$$\overrightarrow{\nabla}_1\cdot\mathbf{\Pi}^{(1)} + \partial_{t_2}\rho\vec{u} + \frac{1}{2}\left(\partial_{t_1}\overrightarrow{\nabla}_1\cdot\mathbf{\Pi}^{(0)} + c_s^2\partial_{t_1}\overrightarrow{\nabla}_1\rho + \overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \mathbf{R}^{(0)}\right) = 0.$$

We'll now recombine this equation with equation A.17 to get the momentum conservation equation. From the terms in the two equations, we can see that we must multiply them with ϵ^2 and ϵ respectively to retrieve the correct orders of magnitude before adding them. The result is

$$\begin{aligned}\left(\epsilon\partial_{t_1} + \epsilon^2\partial_{t_2}\right)\rho\vec{u} + \epsilon\overrightarrow{\nabla}_1\cdot\left(\mathbf{\Pi}^{(0)} + \epsilon\mathbf{\Pi}^{(1)}\right) + \left(\epsilon + \frac{\epsilon^2}{2}\partial_{t_1}\right)c_s^2\overrightarrow{\nabla}_1\rho \\ + \frac{\epsilon^2}{2}\partial_{t_1}\overrightarrow{\nabla}_1\cdot\mathbf{\Pi}^{(0)} + \frac{\epsilon^2}{2}\overrightarrow{\nabla}_1\overrightarrow{\nabla}_1 : \mathbf{R}^{(0)} = 0\end{aligned}$$

Using the multi-scale expansion in reverse and rewriting several of the terms as divergences, we can rewrite this as

$$\boxed{\partial_t\rho\vec{u} + \overrightarrow{\nabla}\cdot\left[\mathbf{\Pi}^{(0)} + \epsilon\mathbf{\Pi}^{(1)} + \left(1 + \frac{1}{2}\partial_t\right)c_s^2\rho\mathbf{I} + \frac{\epsilon}{2}\partial_{t_1}\mathbf{\Pi}^{(0)} + \frac{1}{2}\overrightarrow{\nabla}\cdot\mathbf{R}^{(0)}\right] = 0.} \tag{A.21}$$

This equation is actually equivalent with the Navier-Stokes equation, although it may not look very similar. Before we can show this, we must resolve the unknown moments in the equation.

A.3 Resolving 2nd and 3rd order moments

To resolve equation A.21, we must find out how $\mathbf{\Pi}^{(0)}$, $\mathbf{\Pi}^{(1)}$, and $\overrightarrow{\nabla} \cdot \mathbf{R}^{(0)}$ are related to the macroscopic variables ρ and $\vec{u}$. This can be done through calculation of the sums that they are defined by:

$$\mathbf{\Pi}^{(0)} = \sum_i \mathbf{Q}_i f_i^{(0)} = \sum_i \left(\vec{c}_i\vec{c}_i - c_s^2\mathbf{I}\right) f_i^{(0)}, \tag{A.22}$$

$$\mathbf{\Pi}^{(1)} = \sum_i \mathbf{Q}_i f_i^{(1)} = \sum_i \left(\vec{c}_i\vec{c}_i - c_s^2\mathbf{I}\right) f_i^{(1)}, \tag{A.23}$$

$$\overrightarrow{\nabla} \cdot \mathbf{R}^{(0)} = \overrightarrow{\nabla} \cdot \sum_i \vec{c}_i\vec{c}_i\vec{c}_i f_i^{(0)}. \tag{A.24}$$

$\mathbf{\Pi}^{(0)}$ and $\mathbf{R}^{(0)}$ can be calculated directly, since we have an explicit expression for the equilibrium distribution $f_i^{(0)}$ in equation 4.15. To calculate $\mathbf{\Pi}^{(1)}$, we first need to find an explicit expression for $f_i^{(1)}$.

In this section, it will be necessary to neglect terms which scale rapidly with the Mach number, which is given by $\mathrm{Ma} = |\vec{u}|/c_s$. To retrieve the Navier-Stokes equation, terms of $\mathcal{O}(\mathrm{Ma}^3)$ must be neglected. [11] This means that the lattice Boltzmann method will be a valid Navier-Stokes solver only for flows of low Mach number.

In the isothermal limit which we are operating in, $\partial_t\rho = \mathcal{O}(\mathrm{Ma})$. [11]. Also, the velocity $\vec{u}$ is of $\mathcal{O}(\mathrm{Ma})$, which is obvious from the definition of the Mach number.

Resolving $\mathbf{\Pi}^{(0)}$

In index notation, equation A.22 becomes

$$\Pi_{\alpha\beta}^{(0)} = \sum_i c_{i\alpha}c_{i\beta}f_i^{(0)} - c_s^2\delta_{\alpha\beta}\sum_i f_i^{(0)}. \tag{A.25}$$

The second sum becomes ρ due to conservation property A.8. Inserting the expression for the equilibrium distribution (4.15), the first term divided by ρ becomes

$$\frac{1}{\rho}\sum_i c_{i\alpha}c_{i\beta}f_i^{(0)} = \sum_i t_i c_{i\alpha}c_{i\beta} + \frac{u_\gamma}{c_s^2}\overbrace{\sum_i t_i c_{i\alpha}c_{i\beta}c_{i\gamma}}^{=0} + \frac{u_\gamma u_\delta}{2c_s^4}\sum_i t_i c_{i\alpha}c_{i\beta}c_{i\gamma}c_{i\delta} - \frac{u_\gamma u_\gamma}{2c_s^2}\sum_i t_i c_{i\alpha}c_{i\beta}.$$

From lattice isotropy conditions 4.10, the second term on the right side disappears and the others become

$$\frac{1}{\rho}\sum_i c_{i\alpha}c_{i\beta}f_i^{(0)} = c_s^2\delta_{\alpha\beta} + \frac{u_\gamma u_\delta}{2}\left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right) - \frac{u_\gamma u_\gamma}{2}\delta_{\alpha\beta}.$$

Since $u_\gamma u_\delta \delta_{\gamma\delta} = u_\gamma u_\gamma$, this becomes

$$\frac{1}{\rho}\sum_i c_{i\alpha}c_{i\beta}f_i^{(0)} = c_s^2\delta_{\alpha\beta} + \frac{u_\gamma u_\delta}{2}\left(\delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right).$$

By inserting this equation into equation A.25, we get an expression for $\Pi_{\alpha\beta}^{(0)}$,

$$\Pi_{\alpha\beta}^{(0)} = \frac{\rho}{2}\left[u_\gamma u_\delta\left(\delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right)\right] = \frac{\rho}{2}\left(u_\alpha u_\beta + u_\alpha u_\beta\right) = \rho u_\alpha u_\beta.$$

Going back to tensor notation, this becomes

$$\boxed{\mathbf{\Pi}^{(0)} = \rho\vec{u}\vec{u}.} \tag{A.26}$$

Resolving $\mathbf{R}^{(0)}$

In index notation, equation A.24 becomes

$$\left\{\overrightarrow{\nabla}\cdot R^{(0)}\right\}_{\alpha\beta} = \partial_\gamma \sum_i c_{i\alpha}c_{i\beta}c_{i\gamma}f_i^{(0)} \tag{A.27}$$

Inserting for $f_i^{(0)}$, this becomes

$$\begin{aligned}\left\{\overrightarrow{\nabla}\cdot R^{(0)}\right\}_{\alpha\beta} = \partial_\gamma\rho\overbrace{\sum_i t_i c_{i\alpha}c_{i\beta}c_{i\gamma}}^{=0} &+\partial_\gamma\frac{\rho u_\delta}{c_s^2}\sum_i t_i c_{i\alpha}c_{i\beta}c_{i\gamma}c_{i\delta} \\ &+\partial_\gamma\frac{\rho u_\delta u_\epsilon}{2c_s^4}\underbrace{\sum_i t_i c_{i\alpha}c_{i\beta}c_{i\gamma}c_{i\delta}c_{i\epsilon}}_{=0} - \partial_\gamma\frac{\rho u_\delta u_\delta}{2c_s^2}\underbrace{\sum_i t_i c_{i\alpha}c_{i\beta}c_{i\gamma}}_{=0}.\end{aligned}$$

All but the second term on the right side disappear from lattice conditions 4.10. The second term becomes

$$\begin{aligned}\left\{\overrightarrow{\nabla}\cdot R^{(0)}\right\}_{\alpha\beta} &= c_s^2\partial_\gamma\rho u_\delta\left(\delta_{\alpha\beta}\delta_{\gamma\delta}+\delta_{\alpha\gamma}\delta_{\beta\delta}+\delta_{\alpha\delta}\delta_{\beta\gamma}\right) \\ &= c_s^2\left(\delta_{\alpha\beta}\partial_\gamma\rho u_\gamma+\partial_\alpha\rho u_\beta+\partial_\beta\rho u_\alpha\right).\end{aligned}$$

In tensor notation, this becomes

$$\boxed{\overrightarrow{\nabla}\cdot\mathbf{R}^{(0)} = c_s^2\left[\left(\overrightarrow{\nabla}\cdot\rho\vec{u}\right)\mathbf{I}+\overrightarrow{\nabla}\rho\vec{u}+\left(\overrightarrow{\nabla}\rho\vec{u}\right)^{\mathrm{T}}\right].} \tag{A.28}$$

Finding $f_i^{(1)}$

So far we have resolved $\mathbf{\Pi}^{(0)}$ and $\mathbf{R}^{(0)}$, which both depend on $f_i^{(0)}$, which is a known expression. To resolve $\mathbf{\Pi}^{(1)}$, we first need to find the expression for $f_i^{(1)}$. From equations A.12 and A.6 we have that

$$-\frac{1}{\tau}f_i^{(1)} = \Omega_i^{(1)} = \left(\partial_{t_1}+\overrightarrow{\nabla}_1\cdot\vec{c_i}\right)f_i^{(0)}.$$

The expression of $f_i^{(0)}$ (4.15) can be rewritten using equations 2.1, 2.4 and A.11, giving

$$f_i^{(0)} = \rho t_i\left[1+\frac{\vec{u}\cdot\vec{c_i}}{c_s^2}+\frac{\vec{u}\vec{u}:\vec{c_i}\vec{c_i}}{2c_s^4}-\frac{\vec{u}\vec{u}:\mathbf{I}}{2c_s^2}\right] = \rho t_i\left[1+\frac{\vec{u}\cdot\vec{c_i}}{c_s^2}+\frac{\mathbf{Q}_i:\vec{u}\vec{u}}{2c_s^4}\right].$$

Inserting this into equation A.3, we find that

$$\begin{aligned} f_i^{(1)} &= -\tau t_i \left(\partial_{t_1} + \overrightarrow{\nabla}_1 \cdot \vec{c_i}\right) \left[\rho + \frac{\rho\vec{u}\cdot\vec{c_i}}{c_s^2} + \frac{\mathbf{Q}_i : \rho\vec{u}\vec{u}}{2c_s^4}\right] \\ &= -\tau t_i \Bigg[\partial_{t_1}\rho + \frac{\partial_{t_1}(\rho\vec{u}\cdot\vec{c_i})}{c_s^2} + \frac{\partial_{t_1}(\mathbf{Q}_i : \rho\vec{u}\vec{u})}{2c_s^4} \\ &\qquad + \vec{c_i}\cdot\overrightarrow{\nabla}_1\rho + \frac{\vec{c_i}\vec{c_i} : \overrightarrow{\nabla}_1(\rho\vec{u})}{c_s^2} + \frac{(\vec{c_i}\cdot\overrightarrow{\nabla}_1)(\mathbf{Q}_i : \rho\vec{u}\vec{u})}{2c_s^4}\Bigg]. \end{aligned} \tag{A.29}$$

The time derivatives in equation A.29 can be replaced with space derivatives. The first term can be replaced using equation A.15, which states that

$$\partial_{t_1}\rho = -\overrightarrow{\nabla}_1 \cdot \rho\vec{u}. \tag{A.30}$$

Taking the dot product of equation A.17 with $\vec{c_i}/c_s^2$ and using A.26, we get that

$$\frac{\partial_{t_1}(\rho\vec{u}\cdot\vec{c_i})}{c_s^2} = -\frac{\vec{c_i}\overrightarrow{\nabla}_1 : (\rho\vec{u}\vec{u})}{c_s^2} - \vec{c_i}\cdot\overrightarrow{\nabla}_1\rho. \tag{A.31}$$

This can be used to replace the second term in A.29.

The numerator in the third term in equation A.29 can be written as $\mathbf{Q}_i : \partial_{t_1}\rho\vec{u}\vec{u}$. The right matrix in this tensor contraction can be written in index notation as

$$\begin{aligned} \{\partial_{t_1}\rho\vec{u}\vec{u}\}_{\alpha\beta} = \partial_{t_1}\rho u_\alpha u_\beta &= u_\beta\partial_{t_1}\rho u_\alpha + \rho u_\alpha\partial_{t_1}u_\beta \\ &= u_\alpha\partial_{t_1}\rho u_\beta + \rho u_\beta\partial_{t_1}u_\alpha \\ &= u_\alpha u_\beta\partial_{t_1}\rho + \rho u_\beta\partial_{t_1}u_\alpha + \rho u_\alpha\partial_{t_1}u_\beta, \end{aligned}$$

where we have applied the chain rule in three different ways. Adding the first two and subtracting the third, we are left with

$$\partial_{t_1}\rho u_\alpha u_\beta = u_\beta\partial_{t_1}\rho u_\alpha + u_\alpha\partial_{t_1}\rho u_\beta - u_\alpha u_\beta\partial_{t_1}\rho.$$

Moving back to tensor notation, we can write this as

$$\begin{aligned} \partial_{t_1}\rho\vec{u}\vec{u} &= (\partial_{t_1}\rho\vec{u})\,\vec{u} + \vec{u}\,(\partial_{t_1}\rho\vec{u}) - (\partial_{t_1}\rho)\,\vec{u}\vec{u} \\ &= (\partial_{t_1}\rho\vec{u})\,\vec{u} + [(\partial_{t_1}\rho\vec{u})\,\vec{u}]^{\mathrm{T}} - (\partial_{t_1}\rho)\,\vec{u}\vec{u}. \end{aligned} \tag{A.32}$$

Using this, we can now rewrite the numerator in the third term as

$$\mathbf{Q}_i : \partial_{t_1}\rho\vec{u}\vec{u} = \mathbf{Q}_i : \big[(\partial_{t_1}\rho\vec{u})\,\vec{u} + [(\partial_{t_1}\rho\vec{u})\,\vec{u}]^{\mathrm{T}} - \overbrace{(\partial_{t_1}\rho)\,\vec{u}\vec{u}}^{\mathcal{O}(\mathrm{Ma}^3)}\big].$$

Since the third term is $\mathcal{O}(\mathrm{Ma}^3)$, it should be neglected, as mentioned at the start of this section. Also, since $\mathbf{Q}_i$ is symmetric, contracting it with a matrix gives the same result as contracting it with that matrix's transpose. We therefore have that

$$\mathbf{Q}_i : \partial_{t_1}\rho\vec{u}\vec{u} = 2\mathbf{Q}_i : (\partial_{t_1}\rho\vec{u})\,\vec{u}.$$

Using equation A.17, this becomes

$$\begin{aligned} \mathbf{Q}_i : \partial_{t_1}\rho\vec{u}\vec{u} &= 2\mathbf{Q}_i : \Big[-\overbrace{\overrightarrow{\nabla}_1\cdot(\rho\vec{u}\vec{u})}^{\mathcal{O}(\mathrm{Ma}^3)} - c_s^2\overrightarrow{\nabla}_1\rho\Big]\vec{u} \\ &= -2c_s^2\mathbf{Q}_i : \left(\overrightarrow{\nabla}_1\rho\right)\vec{u}. \end{aligned} \tag{A.33}$$

We insert equations A.30, A.31, and A.33 into equation A.29 to get

$$f_i^{(1)} = -\frac{\tau t_i}{c_s^2}\Big[- c_s^2 \overrightarrow{\nabla}_1 \cdot \rho\vec{u} - \vec{c_i}\overrightarrow{\nabla}_1 : (\rho\vec{u}\vec{u}) - \mathbf{Q}_i : \left(\overrightarrow{\nabla}_1 \rho\right)\vec{u} + \vec{c_i}\vec{c_i} : \overrightarrow{\nabla}_1 (\rho\vec{u}) + \frac{1}{2c_s^2}\left(\vec{c_i}\cdot\overrightarrow{\nabla}_1\right)(\mathbf{Q}_i : \rho\vec{u}\vec{u})\Big]. \tag{A.34}$$

We can simplify three of these terms by applying the chain rule to them in index notation,

$$\begin{aligned}
&- \mathbf{Q}_i : \left(\overrightarrow{\nabla}\rho\right)\vec{u} - c_s^2\overrightarrow{\nabla}\cdot\rho\vec{u} + \vec{c_i}\vec{c_i} : \overrightarrow{\nabla}(\rho\vec{u}) \\
&\quad = -c_{i\alpha}c_{i\beta}(\partial_\alpha\rho)u_\beta + c_s^2\delta_{\alpha\beta}(\partial_\alpha\rho)u_\beta - c_s^2\partial_\alpha\rho u_\alpha + c_{i\alpha}c_{i\beta}\partial_\alpha\rho u_\beta \\
&\quad = -c_{i\alpha}c_{i\beta}u_\beta\partial_\alpha\rho + c_s^2\left(u_\alpha\partial_\alpha\rho - u_\alpha\partial_\alpha\rho - \rho\partial_\alpha u_\alpha\right) + c_{i\alpha}c_{i\beta}u_\beta\partial_\alpha\rho + c_{i\alpha}c_{i\beta}\rho\partial_\alpha u_\beta \\
&\quad = c_{i\alpha}c_{i\beta}\rho\partial_\alpha u_\beta + c_s^2\delta_{\alpha\beta}\rho\partial_\alpha u_\beta \\
&\quad = \mathbf{Q}_i : \rho\left(\overrightarrow{\nabla}\vec{u}\right).
\end{aligned}$$

This gives us our final expression for $f_i^{(1)}$, namely

$$\boxed{f_i^{(1)} = -\frac{\tau t_i}{c_s^2}\left[\mathbf{Q}_i : \rho\left(\overrightarrow{\nabla}_1\vec{u}\right) - \vec{c_i}\overrightarrow{\nabla}_1 : \rho\vec{u}\vec{u} + \frac{1}{2c_s^2}\left(\vec{c_i}\cdot\overrightarrow{\nabla}_1\right)(\mathbf{Q}_i : \rho\vec{u}\vec{u})\right].} \tag{A.35}$$

Resolving $\mathbf{\Pi}^{(1)}$

Now that we have $f_i^{(1)}$, we can finally resolve $\mathbf{\Pi}^{(1)}$. Equation A.23 states that

$$\mathbf{\Pi}^{(1)} = \sum_i \mathbf{Q}_i f_i^{(1)} = \sum_i \vec{c_i}\vec{c_i} f_i^{(1)} - c_s^2\mathbf{I}\sum_i f_i^{(1)}. \tag{A.36}$$

The second term disappears due to equation A.13. We are left with the first term, which must be calculated directly by inserting equation A.35. This becomes

$$\mathbf{\Pi}^{(1)} = -\frac{\tau}{c_s^2}\left[\sum_i t_i\vec{c_i}\vec{c_i}\left[\mathbf{Q}_i : \rho\left(\overrightarrow{\nabla}_1\vec{u}\right)\right] - \sum_i t_i\vec{c_i}\vec{c_i}\left[\vec{c_i}\overrightarrow{\nabla}_1 : \rho\vec{u}\vec{u}\right] + \frac{1}{2c_s^2}\sum_i t_i\vec{c_i}\vec{c_i}\left[\left(\vec{c_i}\cdot\overrightarrow{\nabla}_1\right)(\mathbf{Q}_i : \rho\vec{u}\vec{u})\right]\right]. \tag{A.37}$$

These sums must be resolved using index notation and lattice isotropy conditions 4.10. The first sum in equation A.37 becomes

$$\begin{aligned}
&\sum_i t_i\vec{c_i}\vec{c_i}\left[\mathbf{Q}_i : \rho\left(\overrightarrow{\nabla}\vec{u}\right)\right] \\
&\quad = \rho\partial_\gamma u_\delta\sum_i t_i c_{i\alpha}c_{i\beta}c_{i\gamma}c_{i\delta} - c_s^2\delta_{\gamma\delta}\rho\partial_\gamma u_\delta\sum_i t_i c_{i\alpha}c_{i\beta} \\
&\quad = c_s^4\rho\partial_\gamma u_\delta(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) - c_s^4\rho(\partial_\gamma u_\delta)\delta_{\alpha\beta}\delta_{\gamma\delta} \\
&\quad = c_s^4\rho\partial_\gamma u_\delta(\delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \\
&\quad = c_s^4\rho(\partial_\alpha u_\beta + \partial_\beta u_\alpha) \\
&\quad = c_s^4\rho\left(\overrightarrow{\nabla}\vec{u} + [\overrightarrow{\nabla}\vec{u}]^{\mathrm{T}}\right).
\end{aligned} \tag{A.38}$$

The second sum in equation A.37 becomes

$$\sum_i t_i \vec{c}_i \vec{c}_i \left[\vec{c}_i \overrightarrow{\nabla} : \rho \vec{u}\vec{u} \right] = \partial_\delta \rho u_\gamma u_\delta \overbrace{\sum_i t_i c_{i\alpha} c_{i\beta} c_{i\gamma}}^{=0} = 0. \tag{A.39}$$

The third sum in A.37 becomes

$$\begin{aligned} &\sum_i t_i \vec{c}_i \vec{c}_i \left[\left(\vec{c}_i \cdot \overrightarrow{\nabla} \right) \left(\mathbf{Q}_i : \rho \vec{u}\vec{u} \right) \right] \\ &\quad = \partial_\gamma \rho u_\delta u_\epsilon \underbrace{\sum_i t_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} c_{i\epsilon}}_{=0} + c_s^2 \delta_{\delta\epsilon} \partial_\gamma \rho u_\delta u_\epsilon \underbrace{\sum_i t_i c_{i\alpha} c_{i\beta} c_{i\gamma}}_{=0} = 0. \end{aligned} \tag{A.40}$$

Inserting equations A.38, A.39, and A.40 into equation A.37, we get that

$$\boxed{\boldsymbol{\Pi}^{(1)} = -\tau \rho c_s^2 \left[\overrightarrow{\nabla}_1 \vec{u} + \left(\overrightarrow{\nabla}_1 \vec{u} \right)^{\mathrm{T}} \right].} \tag{A.41}$$

Resolving $\partial_{t_1} \boldsymbol{\Pi}^{(0)}$

The last unknown term in equation A.21 is $\partial_{t_1} \boldsymbol{\Pi}^{(0)}$. Using equation A.32, it becomes

$$\begin{aligned} \partial_{t_1} \boldsymbol{\Pi}^{(0)} &= \partial_{t_1} \rho \vec{u}\vec{u} = \left(\partial_{t_1} \rho \vec{u} \right) \vec{u} + \vec{u} \left(\partial_{t_1} \rho \vec{u} \right) - \overbrace{\left(\partial_{t_1} \rho \right) \vec{u}\vec{u}}^{\mathcal{O}(\mathrm{Ma}^3)} \\ &= \left(\partial_{t_1} \rho \vec{u} \right) \vec{u} + \vec{u} \left(\partial_{t_1} \rho \vec{u} \right). \end{aligned}$$

By inserting equation A.17, this becomes

$$\begin{aligned} \partial_{t_1} \boldsymbol{\Pi}^{(0)} &= -\Big(\overbrace{\overrightarrow{\nabla}_1 \cdot \rho \vec{u}\vec{u}}^{\mathcal{O}(\mathrm{Ma}^3)} + c_s^2 \overrightarrow{\nabla}_1 \rho \Big) \vec{u} - \vec{u} \Big(\overbrace{\overrightarrow{\nabla}_1 \cdot \rho \vec{u}\vec{u}}^{\mathcal{O}(\mathrm{Ma}^3)} + c_s^2 \overrightarrow{\nabla}_1 \rho \Big) \\ &\Rightarrow \boxed{\partial_{t_1} \boldsymbol{\Pi}^{(0)} = -c_s^2 \left[\left(\overrightarrow{\nabla}_1 \rho \right) \vec{u} + \vec{u} \left(\overrightarrow{\nabla}_1 \rho \right) \right]} \end{aligned} \tag{A.42}$$

A.4 Finding Navier-Stokes

We can now insert all the unknown terms into equation A.21 from equations A.26, A.28, A.41, and A.42. This becomes

$$\begin{aligned} \partial_t \rho \vec{u} + \overrightarrow{\nabla} \cdot \Bigg[&\rho \vec{u}\vec{u} - \tau \rho c_s^2 \left(\overrightarrow{\nabla} \vec{u} + \left[\overrightarrow{\nabla} \vec{u} \right]^{\mathrm{T}} \right) + p \mathbf{I} + \frac{c_s^2}{2} \partial_t \rho \mathbf{I} - \frac{c_s^2}{2} \left(\left[\overrightarrow{\nabla} \rho \right] \vec{u} + \vec{u} \left[\overrightarrow{\nabla} \rho \right] \right) \\ &+ \frac{c_s^2}{2} \left(\overrightarrow{\nabla} \cdot [\rho \vec{u}] \, \mathbf{I} + \overrightarrow{\nabla} \rho \vec{u} + \left[\overrightarrow{\nabla} \rho \vec{u} \right]^{\mathrm{T}} \right) \Bigg] = 0. \end{aligned} \tag{A.43}$$

Here we have used equation A.3 to change all $\epsilon \overrightarrow{\nabla}_1$ back to $\overrightarrow{\nabla}$, and equation 2.5 to move from density to pressure. This equation might seem like a confused jumble, but we will now show that it is actually the compressible Navier-Stokes equation (2.7).

First, we can use equation A.20 to rewrite a term so that it cancels another,

$$\frac{c_s^2}{2}\partial_t \rho \mathbf{I} = \frac{c_s^2}{2}\overrightarrow{\nabla}\cdot(\rho\vec{u})\,\mathbf{I}. \tag{A.44}$$

Then we can merge some terms using index notation and the chain rule,

$$\begin{aligned}
&\left\{\overrightarrow{\nabla}(\rho\vec{u}) + \left[\overrightarrow{\nabla}(\rho\vec{u})\right]^{\mathrm{T}} - \left(\overrightarrow{\nabla}\rho\right)\vec{u} - \vec{u}\left(\overrightarrow{\nabla}\rho\right)\right\}_{\alpha\beta} \\
&\quad = \partial_\alpha \rho u_\beta + \partial_\beta \rho u_\alpha - u_\beta \partial_\alpha \rho - u_\alpha \partial_\beta \rho \\
&\quad = \rho\partial_\alpha u_\beta + \rho\partial_\beta u_\alpha \\
&\quad = \left\{\rho\left(\overrightarrow{\nabla}\vec{u} + \left[\overrightarrow{\nabla}\vec{u}\right]^{\mathrm{T}}\right)\right\}_{\alpha\beta}
\end{aligned} \tag{A.45}$$

Using equations A.44 and A.45 with equation A.43, it is simplified to

$$\boxed{\partial_t \rho\vec{u} + \overrightarrow{\nabla}\cdot\left[\rho\vec{u}\vec{u} + p\mathbf{I} - \rho c_s^2\left(\tau - \frac{1}{2}\right)\left(\overrightarrow{\nabla}\vec{u} + \left[\overrightarrow{\nabla}\vec{u}\right]^{\mathrm{T}}\right)\right] = 0,} \tag{A.46}$$

which is a perfectly valid formulation of Navier-Stokes, used in for instance refs. [11, 12]. In Appendix B.1, we show that this is equivalent to equation 2.7's formulation of the compressible Navier-Stokes equation with a kinematic shear viscosity

$$\nu = c_s^2\left(\tau - \frac{1}{2}\right) \tag{A.47}$$

and a kinematic bulk viscosity

$$\nu' = \frac{2}{3}\nu. \tag{A.48}$$

A.5 Final notes

Equations A.20 and A.46 are very interesting results. They show us that the continuity equation and the Navier-Stokes equation, which are derived macroscopically, can be derived from mesoscopic quantities. Not only that, but from a *severely* simplified mesoscopic model, where we have restricted the particles' possible positions and velocities and modeled particle collisions by relaxation to an equilibrium.

Since these results were reached using lattice isotropy conditions 4.10 and equation 4.15's expression for the equilibrium distribution, the use of these isotropy conditions and that particular equilibrium distribution is justified. While it is still possible that other equilibrium distributions might give behaviour according to Navier-Stokes, we have at least shown that this equilibrium distribution fits the bill.

It is important to note that we have not shown that the lattice Boltzmann method gives a perfect match with the Navier-Stokes equation. In the derivation, we have made approximations and taken limits. For instance, we have neglected $\mathcal{O}(\mathrm{Ma}^3)$ terms, meaning that the lattice Boltzmann method has an error in its Navier-Stokes behaviour which scales with Ma^3.

We have also neglected any terms in the expansion of f_i of order ϵ^2 and higher, i.e. $f_i^{(k)}$ for $k \geq 2$. Although these terms are small, we must still assume that the terms give an undesired contribution to the lattice Boltzmann method's behaviour. Minimizing

this contribution is a key point in the regularized lattice Boltzmann mode described in section 7.5.

It is also an unfortunate aspect of the basic lattice Boltzmann method that while the shear viscosity ν can be changed through the relaxation time τ as shown in equation A.47, the bulk viscosity ν' is fixed to the value of ν as shown in equation A.48, denying the possibility of adjusting the two viscosities independently. Fortunately, there exists a method to adjust the bulk viscosity in lattice Boltzmann simulations, which is discussed in section 7.4.

In the literature, it is commonly stated that the lattice Boltzmann method is used to model incompressible flows. It is important to note that this is only a special case in the limit of low compression. Its full capability is to model compressible flows, as we have shown in the derivation in this chapter.

APPENDIX B

Miscellaneous derivations

This appendix is where shorter derivations are placed. This is done so that they will not break the flow of the main text.

B.1 From one Navier-Stokes formulation to another

In several sources [11, 12], the compressible Navier-Stokes equation is formulated as

$$\partial_t \rho\vec{u} + \overrightarrow{\nabla} \cdot \left[\rho\vec{u}\vec{u} + p\mathbf{I} - \rho\nu_1 \left(\overrightarrow{\nabla}\vec{u} + \left[\overrightarrow{\nabla}\vec{u}\right]^{\mathrm{T}}\right) + \rho\nu_2 \left(\overrightarrow{\nabla} \cdot \vec{u}\right)\mathbf{I}\right] = \vec{f}, \tag{B.1}$$

where ν_1 and ν_2 are two kinematic viscosities. Rewriting these to dynamic viscosities, the equation becomes

$$\partial_t \rho\vec{u} + \overrightarrow{\nabla} \cdot \left[\rho\vec{u}\vec{u} + p\mathbf{I} - \mu_1 \left(\overrightarrow{\nabla}\vec{u} + \left[\overrightarrow{\nabla}\vec{u}\right]^{\mathrm{T}}\right) + \mu_2 \left(\overrightarrow{\nabla} \cdot \vec{u}\right)\mathbf{I}\right] = \vec{f}, \tag{B.2}$$

Since this is the formulation that is conveniently reached when deriving Navier-Stokes from the lattice Boltzmann method as done in Appendix A, it is important to show that this formulation is equivalent with the formulation given in equation 2.7. This can be done by rewriting the equation term by term.

Term 1:

$$\partial_t \rho\vec{u} = \rho\partial_t\vec{u} + \vec{u}\partial_t\rho \overset{\text{Eq.2.6}}{=} \rho\partial_t\vec{u} - \vec{u}\left(\overrightarrow{\nabla} \cdot \rho\vec{u}\right)$$

Term 2:

$$\left\{\overrightarrow{\nabla} \cdot \rho\vec{u}\vec{u}\right\}_\alpha = \partial_\beta \rho u_\alpha u_\beta = \rho u_\beta \partial_\beta u_\alpha + u_\alpha \partial_\beta \rho u_\beta = \left\{\rho\left(\vec{u} \cdot \overrightarrow{\nabla}\right)\vec{u} + \vec{u}\left(\overrightarrow{\nabla} \cdot \rho\vec{u}\right)\right\}_\alpha$$
$$\Rightarrow \overrightarrow{\nabla} \cdot \rho\vec{u}\vec{u} = \rho\left(\vec{u} \cdot \overrightarrow{\nabla}\right)\vec{u} + \vec{u}\left(\overrightarrow{\nabla} \cdot \rho\vec{u}\right)$$

Term 3:

$$\left\{\overrightarrow{\nabla} \cdot p\mathbf{I}\right\}_\alpha = \partial_\beta p\delta_{\alpha\beta} = \partial_\alpha p = \left\{\overrightarrow{\nabla} p\right\}_\alpha$$
$$\Rightarrow \overrightarrow{\nabla} \cdot p\mathbf{I} = \overrightarrow{\nabla} p$$

Term 4:

$$\left\{\overrightarrow{\nabla}\cdot\left(\overrightarrow{\nabla}\vec{u}+\left[\overrightarrow{\nabla}\vec{u}\right]^{\mathrm{T}}\right)\right\}_{\alpha}=\partial_{\beta}\partial_{\alpha}u_{\beta}+\partial_{\beta}\partial_{\beta}u_{\alpha}=\left\{\overrightarrow{\nabla}\left(\overrightarrow{\nabla}\cdot\vec{u}\right)+\nabla^{2}\vec{u}\right\}_{\alpha}$$

$$\Rightarrow\overrightarrow{\nabla}\cdot\left(\overrightarrow{\nabla}\vec{u}+\left[\overrightarrow{\nabla}\vec{u}\right]^{\mathrm{T}}\right)=\overrightarrow{\nabla}\left(\overrightarrow{\nabla}\cdot\vec{u}\right)+\nabla^{2}\vec{u}$$

Term 5:

$$\left\{\overrightarrow{\nabla}\cdot\left(\overrightarrow{\nabla}\cdot\vec{u}\right)\mathbf{I}\right\}_{\alpha}=\partial_{\beta}\partial_{\gamma}u_{\gamma}\delta_{\alpha\beta}=\partial_{\alpha}\partial_{\gamma}u_{\gamma}=\left\{\overrightarrow{\nabla}\left(\overrightarrow{\nabla}\cdot\vec{u}\right)\right\}_{\alpha}$$

$$\Rightarrow\overrightarrow{\nabla}\cdot\left(\overrightarrow{\nabla}\cdot\vec{u}\right)\mathbf{I}=\overrightarrow{\nabla}\left(\overrightarrow{\nabla}\cdot\vec{u}\right)$$

Inserting the terms

Inserting the rewritten terms into equation B.2, we get

$$\rho\partial_{t}\vec{u}+\rho\left(\vec{u}\cdot\overrightarrow{\nabla}\right)\vec{u}+\overrightarrow{\nabla}p+\mu_{1}\nabla^{2}\vec{u}+\left(\mu_{1}+\mu_{2}\right)\overrightarrow{\nabla}\left(\overrightarrow{\nabla}\cdot\vec{u}\right)=\vec{f}$$

This is an exact match with equation 2.7 if

$$\mu_{1}=\mu\qquad\text{and}\qquad\mu_{2}=\mu'-\frac{2}{3}\mu.$$

This means that equation B.2 is a perfect match with equation 2.7 when it is written as

$$\partial_{t}\rho\vec{u}+\overrightarrow{\nabla}\cdot\left[\rho\vec{u}\vec{u}+p\mathbf{I}-\mu\left(\overrightarrow{\nabla}\vec{u}+\left[\overrightarrow{\nabla}\vec{u}\right]^{\mathrm{T}}\right)+\left(\mu'-\frac{2}{3}\mu\right)\left(\overrightarrow{\nabla}\cdot\vec{u}\right)\mathbf{I}\right]=\vec{f}.\tag{B.3}$$

B.1.1 Viscosity errors in lattice Boltzmann

In the lattice Boltzmann method, it is incorrect to say that $\rho\nu$ equals a constant μ, as ν is a constant given by $c_s^2(\tau-1/2)$. We can therefore no longer assume μ_1 and μ_2 constant as we did above, and the rewriting of term 4 becomes incorrect. Term 5 is normally not a problem since it is zero in the basic lattice Boltzmann model.

$\rho\nu$ can be held constant by adapting τ node-for-node. If

$$\tau=\frac{\tau_0-1/2}{\rho/\rho_0}+\frac{1}{2},\tag{B.4}$$

where τ_0 is the desired constant value of τ and ρ_0 is the equilibrium density, $\rho\nu$ becomes

$$\rho\nu=\rho c_s^2\left(\tau-\frac{1}{2}\right)=\rho c_s^2\frac{\tau_0-1/2}{\rho/\rho_0}=\rho_0 c_s^2\left(\tau_0-\frac{1}{2}\right).\tag{B.5}$$

This results in a constant viscosity everywhere in the system.

B.2 Viscosity-damped waves from an infinite line source

In *Fundamentals of Acoustics*, [13] a derivation is performed to find an expression for cylindrical waves, and another is performed for viscous damping of an acoustic plane wave. No derivation is performed for viscous damping of cylindrical waves, so we will here couple the two derivations to derive an expression for a viscously damped cylindrical stationary wave. To keep this short, we will refer to the book when possible.

Regardless of geometry, the lossy wave equation (2.15) can be reduced to a lossy Helmholtz equation [13]

$$\nabla^2 p_a + k^2 p_a = 0 \tag{B.6}$$

for any monofrequency $p_a(\vec{x}, t)$ (i.e. varying in time with $\mathrm{e}^{j\omega t}$). The angular wavenumber k is now the complex number [13]

$$k = \frac{\omega}{c} - j\alpha_s, \tag{B.7}$$

where α_s is the spatial absorption coefficient given in equation 2.19.

In cylindrical coordinates, equation B.6 becomes

$$\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} + \frac{\partial^2}{\partial z^2}\right) p_a + k^2 p_a = 0.$$

Assuming that we have cylindrical symmetry ($\partial p_a/\partial\theta = 0$) and that the surce is an infinite line in z direction ($\partial p_a/\partial z = 0$) at $r = 0$, this can be written as

$$r^2 \frac{d^2 p_a}{dr^2} + r\frac{dp_a}{dr} + k^2 r^2 p_a = 0. \tag{B.8}$$

Equation B.8 is a transformed Bessel equation, which gives us the solution [60]

$$p(r, t) = [AJ_0(kr) + BY_0(kr)]\,\mathrm{e}^{j\omega t}, \tag{B.9}$$

where A and B are arbitrary constants, and J_0 and Y_0 are Bessel functions of the first and second kind. Since Y_0 diverges for $r \to 0$, equation B.9 does not apply at $r = 0$.

If p_a is an outgoing travelling wave, $B = -jA$, so that the Bessel function superposition in equation B.9 takes the form of the Hankel function [13]

$$H_0^{(2)}(kr) = J_0(kr) - jY_0(kr).$$

This means that

$$p(r, t) = AH_0^{(2)}(kr)\mathrm{e}^{j\omega t} \tag{B.10}$$

is the exact solution for $r > 0$ for a cylindrical wave travelling outwards from an infinitely long line source at $r = 0$. Viscous absorption comes from the fact that k is the complex number given by equation B.7.

APPENDIX C

CODE

This chapter holds a two MATLAB scripts which can be used to verify the lattice isotropy conditions for lattice gas automata and the lattice Boltzmann method. It does not hold any actual lattice Boltzmann code, as the code which was written for and used in this project has become too messy to include here.

Fortunately, it is simple to find lattice Boltzmann example code online. It is possible to find simple demonstration programs in many languages, spanning from C to Ruby, at `http://lbmethod.org/numerics:codes`. Also available is OpenLB, an open-source optimized C++ library for lattice Boltzmann calculations, which can be found at `http://openlb.org`.

C.1 Verification of LGA lattice isotropy

This MATLAB code is used to verify the lattice isotropy conditions for lattice gas automata (3.5). These conditions are discussed in section 3.3. The code calculates the sums in the conditions directly for a specified lattice and a specified set of dimensional indices.

The code is used by setting the **lattice** variable to the value that corresponds to the desired lattice, and setting the variables **a**, **b**, **c**, and **d** to the desired values of α, β, γ, and δ.

```
% ========================
% USER DEFINABLE CONSTANTS
% ========================

lattice =  6;   % 4 for square, 6 for hexagonal, 24 for FCHC

a = 1;          % alpha
b = 1;          % beta
c = 1;          % gamma
d = 1;          % delta

% ================================
% END OF USER DEFINABLE CONSTANTS
% ================================

clear v;
switch lattice
    case 4
        % Two-dimensional square lattice
        v(1, 1) =  1;   v(1, 2) =  0;
        v(2, 1) =  0;   v(2, 2) =  1;
        v(3, 1) = -1;   v(3, 2) =  0;
        v(4, 1) =  0;   v(4, 2) = -1;
    case 6
        % Two-dimensional hexagonal lattice
        v(1, 1) = cos(0*pi/3);  v(1, 2) = sin(0*pi/3);
        v(2, 1) = cos(1*pi/3);  v(2, 2) = sin(1*pi/3);
        v(3, 1) = cos(2*pi/3);  v(3, 2) = sin(2*pi/3);
        v(4, 1) = cos(3*pi/3);  v(4, 2) = sin(3*pi/3);
        v(5, 1) = cos(4*pi/3);  v(5, 2) = sin(4*pi/3);
        v(6, 1) = cos(5*pi/3);  v(6, 2) = sin(5*pi/3);
    case 24
        % Three-dimensional FCHC lattice
        v(1, 1) =  1;   v(1, 2) =  0;   v(1, 3) =  0;
        v(2, 1) =  1;   v(2, 2) =  0;   v(2, 3) =  0;
        v(3, 1) = -1;   v(3, 2) =  0;   v(3, 3) =  0;
        v(4, 1) = -1;   v(4, 2) =  0;   v(4, 3) =  0;
        v(5, 1) =  0;   v(5, 2) =  1;   v(5, 3) =  0;
        v(6, 1) =  0;   v(6, 2) =  1;   v(6, 3) =  0;
        v(7, 1) =  0;   v(7, 2) = -1;   v(7, 3) =  0;
        v(8, 1) =  0;   v(8, 2) = -1;   v(8, 3) =  0;
        v(9, 1) =  0;   v(9, 2) =  0;   v(9, 3) =  1;
        v(10,1) =  0;   v(10,2) =  0;   v(10,3) =  1;
        v(11,1) =  0;   v(11,2) =  0;   v(11,3) = -1;
        v(12,1) =  0;   v(12,2) =  0;   v(12,3) = -1;
```

```
        v(13,1) =  1;   v(13,2) =  1;   v(13,3) =  0;
        v(14,1) = -1;   v(14,2) =  1;   v(14,3) =  0;
        v(15,1) =  1;   v(15,2) = -1;   v(15,3) =  0;
        v(16,1) = -1;   v(16,2) = -1;   v(16,3) =  0;
        v(17,1) =  1;   v(17,2) =  0;   v(17,3) =  1;
        v(18,1) = -1;   v(18,2) =  0;   v(18,3) =  1;
        v(19,1) =  1;   v(19,2) =  0;   v(19,3) = -1;
        v(20,1) = -1;   v(20,2) =  0;   v(20,3) = -1;
        v(21,1) =  0;   v(21,2) =  1;   v(21,3) =  1;
        v(22,1) =  0;   v(22,2) = -1;   v(22,3) =  1;
        v(23,1) =  0;   v(23,2) =  1;   v(23,3) = -1;
        v(24,1) =  0;   v(24,2) = -1;   v(24,3) = -1;
    otherwise
        error('Invalid lattice.');
end

latsum = zeros(4, 1);

% Perform summation
for i = 1:length(e)
    latsum(1) = latsum(1) + v(i,a);
    latsum(2) = latsum(2) + v(i,a) * v(i,b);
    latsum(3) = latsum(3) + v(i,a) * v(i,b) * v(i,c);
    latsum(4) = latsum(4) + v(i,a) * v(i,b) * v(i,c) * v(i,d);
end

% Eliminate eventual round-off errors due to imprecise pi
for i = 1:length(latsum)
    if (latsum(i) < 1E-10)
        latsum(i) = 0;
    end
end

% Display results
disp(' ');
disp(['1st order: ' num2str(latsum(1))]);
disp(['2nd order: ' num2str(latsum(2))]);
disp(['3rd order: ' num2str(latsum(3))]);
disp(['4th order: ' num2str(latsum(4))]);
disp(' ');
```

C.2 Verification of LBM lattice isotropy

This MATLAB code is used to verify that a set of lattice vectors and lattice weights fulfil the lattice isotropy conditions for the lattice Boltzmann method (4.10). These conditions are discussed in section 4.3. The code calculates the sums in the conditions directly for a specified lattice, a specified set of weights, and a specified set of dimensional indices.

The code is used by setting the **lattice** variable to the value that corresponds to the desired lattice, setting the variables **a**, **b**, **c**, and **d** to the desired values of α, β, γ, and δ, and finally setting the desired weights.

```
% ========================
% USER DEFINABLE CONSTANTS
% ========================

lattice = 9;     % 3 for D1Q3, 7 for D2Q7, 9 for D2Q9, 15 for D3Q15

a = 1;           % alpha
b = 1;           % beta
c = 1;           % gamma
d = 1;           % delta
e = 1;           % epsilon

t0 = 4/9;        % t_0
ts = 1/9;        % t_i for short vectors
tl = 1/36;       % t_i for long vectors

% ================================
% END OF USER DEFINABLE CONSTANTS
% ================================

clear v;
clear t;
switch lattice
    case 3
        % D1Q3 line lattice
        v(1, 1) =  0;   t(1) = t0;
        v(2, 1) =  1;   t(2) = ts;
        v(3, 1) = -1;   t(3) = ts;
    case 7
        % D2Q7 hexagonal lattice
        v(1, 1) = 0;            v(1, 2) = 0;            t(1) = t0;
        v(2, 1) = cos(0*pi/3);  v(2, 2) = sin(0*pi/3);  t(2) = ts;
        v(3, 1) = cos(1*pi/3);  v(3, 2) = sin(1*pi/3);  t(3) = ts;
        v(4, 1) = cos(2*pi/3);  v(4, 2) = sin(2*pi/3);  t(4) = ts;
        v(5, 1) = cos(3*pi/3);  v(5, 2) = sin(3*pi/3);  t(5) = ts;
        v(6, 1) = cos(4*pi/3);  v(6, 2) = sin(4*pi/3);  t(6) = ts;
        v(7, 1) = cos(5*pi/3);  v(7, 2) = sin(5*pi/3);  t(7) = ts;
    case 9
        % D2Q9 square lattice
        v(1, 1) =  0;   v(1, 2) =  0;   t(1) = t0;
        v(2, 1) =  1;   v(2, 2) =  0;   t(2) = ts;
        v(3, 1) =  0;   v(3, 2) =  1;   t(3) = ts;
        v(4, 1) = -1;   v(4, 2) =  0;   t(4) = ts;
        v(5, 1) =  0;   v(5, 2) = -1;   t(5) = ts;
```

```
        v(6, 1) =  1;   v(6, 2) =  1;   t(6) = tl;
        v(7, 1) = -1;   v(7, 2) =  1;   t(7) = tl;
        v(8, 1) = -1;   v(8, 2) = -1;   t(8) = tl;
        v(9, 1) =  1;   v(9, 2) = -1;   t(9) = tl;
    case 15
        v(1, 1) =  0;   v(1, 2) =  0;   v(1, 3) =  0;   t(1) = t0;
        v(2, 1) =  1;   v(2, 2) =  0;   v(2, 3) =  0;   t(2) = ts;
        v(3, 1) = -1;   v(3, 2) =  0;   v(3, 3) =  0;   t(3) = ts;
        v(4, 1) =  0;   v(4, 2) =  1;   v(4, 3) =  0;   t(4) = ts;
        v(5, 1) =  0;   v(5, 2) = -1;   v(5, 3) =  0;   t(5) = ts;
        v(6, 1) =  0;   v(6, 2) =  0;   v(6, 3) =  1;   t(6) = ts;
        v(7, 1) =  0;   v(7, 2) =  0;   v(7, 3) = -1;   t(7) = ts;
        v(8, 1) =  1;   v(8, 2) =  1;   v(8, 3) =  1;   t(8) = tl;
        v(9, 1) =  1;   v(9, 2) =  1;   v(9, 3) = -1;   t(9) = tl;
        v(10,1) =  1;   v(10,2) = -1;   v(10,3) =  1;   t(10)= tl;
        v(11,1) =  1;   v(11,2) = -1;   v(11,3) = -1;   t(11)= tl;
        v(12,1) = -1;   v(12,2) =  1;   v(12,3) =  1;   t(12)= tl;
        v(13,1) = -1;   v(13,2) =  1;   v(13,3) = -1;   t(13)= tl;
        v(14,1) = -1;   v(14,2) = -1;   v(14,3) =  1;   t(14)= tl;
        v(15,1) = -1;   v(15,2) = -1;   v(15,3) = -1;   t(15)= tl;
    otherwise
        error('Invalid lattice.');
end

latsum = zeros(6, 1);

% Perform summation
for i = 1:length(v)
    latsum(1) = latsum(1) + t(i);
    latsum(2) = latsum(2) + t(i) * v(i,a);
    latsum(3) = latsum(3) + t(i) * v(i,a) * v(i,b);
    latsum(4) = latsum(4) + t(i) * v(i,a) * v(i,b) * v(i,c);
    latsum(5) = latsum(5) + t(i) * v(i,a) * v(i,b) * v(i,c) * v(i,d);
    latsum(6) = latsum(6) + t(i) * v(i,a) * v(i,b) * v(i,c) * v(i,d)
   * v(i,e);
end

% Eliminate eventual round-off errors due to imprecise pi
for i = 1:length(latsum)
    if (latsum(i) < 1E-10)
        latsum(i) = 0;
    end
end

% Display results
disp(' ');
disp(['0th order: ' num2str(latsum(1))]);
disp(['1st order: ' num2str(latsum(2))]);
disp(['2nd order: ' num2str(latsum(3))]);
disp(['3rd order: ' num2str(latsum(4))]);
disp(['4th order: ' num2str(latsum(5))]);
disp(['5th order: ' num2str(latsum(6))]);
disp(' ');
```

Bibliography

[1] B. Chopard and M. Droz, *Cellular Automata Modelling of Physical Systems*. Cambridge University Press, 1998.

[2] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Oxford University Press, 2001.

[3] D. Wolf-Gladrow, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models*. Springer, 2005.

[4] M. Sukop and D. Thorne, *Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers*. Springer, 2006.

[5] R. Benzi, S. Succi, and M. Vergassola, "The lattice Boltzmann equation: Theory and applications," *Physics Reports*, vol. 222, no. 3, pp. 145–197, 1992.

[6] S. Chen and G. D. Doolen, "Lattice Boltzmann method for fluid flows," *Annual Review of Fluid Mechanics*, vol. 30, pp. 329–364, 1998.

[7] G. Hazi, A. R. Imre, G. Mayer, and I. Farkas, "Lattice Boltzmann methods for two-phase flow modeling," *Annals of Nuclear Energy*, vol. 29, no. 12, pp. 1421–1453, 2002.

[8] R. R. Nourgaliev, T. N. Dinh, T. G. Theofanous, and D. Joseph, "The lattice Boltzmann equation method: theoretical interpretation, numerics and implications," *International Journal of Multiphase Flow*, vol. 29, no. 1, pp. 117–169, 2003.

[9] D. Yua, R. Mei, L.-S. Luob, and W. Shyy, "Viscous flow computations with the method of lattice Boltzmann equation," *Progress in Aerospace Sciences*, vol. 39, no. 5, pp. 329–367, 2003.

[10] C. Aidun, "Lattice Boltzmann method," *Annual Review of Fluid Mechanics*, vol. 42, 2010.

[11] J. Lätt, *Hydrodynamic Limit of Lattice Boltzmann Equations*. PhD thesis, University of Geneva, 2007. Freely available online at `http://pleiades.epfl.ch/~jlatt/`.

[12] P. Dellar, "Bulk and shear viscosities in lattice Boltzmann equations," *Physical Review E*, vol. 64, no. 3, p. 031203, 2001.

[13] L. Kinsler, A. Frey, A. Coppens, and J. Sanders, *Fundamentals of Acoustics*. John Wiley & Sons, 4th ed., 2000.

[14] F. Irgens, *Continuum Mechanics: Volume 1*. Department of Structural Engineering, Norwegian University of Science and Technology, 2005.

[15] L. Rosenhead, "The second coefficient of viscosity: A brief review of fundamentals," *Proceedings of the Royal Society of London, Series A*, vol. 226, no. 1164, pp. 1–6, 1954.

[16] Y. Qian and Y. Zhou, "Complete Galilean-invariant lattice BGK models for the Navier-Stokes equation," *Europhysics Letters*, vol. 42, no. 4, pp. 359–364, 1998.

[17] J. M. Buick, C. A. Greated, and D. M. Campbell, "Lattice BGK simulation of sound waves," *Europhysics Letters*, vol. 43, no. 3, pp. 235–240, 1998.

[18] J. Lätt, "Choice of units in lattice Boltzmann simulations." Freely available online at `http://lbmethod.org/_media/howtos:lbunits.pdf`.

[19] P. Morse and H. Feshbach, *Methods of Theoretical Physics*, vol. 1. McGraw-Hill, 1953.

[20] P. Morse and K. U. Ingard, *Theoretical acoustics*. McGraw-Hill, 1968.

[21] J. Hardy, Y. Pomeau, and O. de Pazzis, "Time Evolution of a Two-Dimensional Classical Lattice System," *Physical Review Letters*, vol. 31, no. 5, pp. 276–279, 1973.

[22] J. Hardy, Y. Pomeau, and O. de Pazzis, "Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions," *Journal of Mathematical Physics*, vol. 14, no. 12, pp. 1746–1759, 1973.

[23] B. Chopard, P. O. Luthi, and A. Masselot, "Cellular Automata and Lattice Boltzmann Techniques: An Approach to Model and Simulate Complex Systems," *Advances in Complex Systems*, vol. 5, no. 2/3, pp. 103–246, 2002. Freely available online at `http://cui.unige.ch/~chopard/CA/acsProof.pdf`.

[24] U. Frisch, B. Hasslacher, and Y. Pomeau, "Lattice-Gas Automata for the Navier-Stokes Equation," *Physical Review Letters*, vol. 56, no. 14, pp. 1505–1508, 1986.

[25] U. Frisch, D. d'Humieres, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet, "Lattice gas hydrodynamics in two and three dimensions," *Complex Systems*, vol. 1, no. 4, pp. 649–707, 1987.

[26] P. Grosfils, J.-P. Boon, and P. Lallemand, "Spontaneous Fluctuation Correlations in Thermal Lattice-Gas Automata," *Physical Review Letters*, vol. 68, no. 7, pp. 1077–1080, 1992.

[27] D. d'Humières, P. Lallemand, and U. Frisch, "Lattice Gas Models for 3D Hydrodynamics," *Europhysics Letters*, vol. 2, pp. 291–297, 1986.

[28] G. McNamara and G. Zanetti, "Use of the Boltzmann Equation to Simulate Lattice-Gas Automata," *Physical Review Letters*, vol. 61, no. 20, pp. 2332–2335, 1988.

[29] Y. Qian, D. d'Humières, and P. Lallemand, "Lattice BGK Models for Navier-Stokes Equation," *Europhysics Letters*, vol. 17, no. 6, pp. 479–484, 1992.

[30] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A Model for Collision Processes in Gases," *Physical Review*, vol. 94, no. 3, pp. 511–525, 1954.

[31] J. O. Andersen, *Introduction to Statistical Mechanics*. Department of Physics, NTNU, 2nd ed., 2008.

[32] P. O. Lüthi, *Lattice Wave Automata: From Radio Waves to Fractures Propagation*. PhD thesis, University of Geneva, 1998. Freely available online at `http://cuiwww.unige.ch/~luthi/a.pdf`.

[33] J. M. V. A. Koelman, "A Simple Lattice Boltzmann Scheme for Navier-Stokes Fluid Flow," *Europhysics Letters*, vol. 15, pp. 603–607, 1991.

[34] D. P. Ziegler, "Boundary Conditions for Lattice Boltzmann simulations," *Journal of Statistical Physics*, vol. 71, no. 5/6, pp. 1171–1177, 1993.

[35] P. A. Skordos, "Initial and boundary conditions for the lattice Boltzmann method," *Physical Review E*, vol. 48, no. 6, pp. 4823–4842, 1993.

[36] T. Inamuro, M. Yoshino, and F. Ogino, "A non-slip boundary condition for lattice Boltzmann simulations," *Physics of Fluids*, vol. 7, no. 12, pp. 2928–2930, 1995.

[37] Q. Zou and X. He, "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model," *Physics of Fluids*, vol. 9, no. 7, pp. 1591–1598, 1997.

[38] J. Lätt, B. Chopard, O. Malaspinas, M. Deville, and A. Michler, "Straight velocity boundaries in the lattice Boltzmann method," *Physical Review E*, vol. 77, no. 5, p. 056703, 2008.

[39] J. A. Cosgrove, J. M. Buick, D. M. Campbell, and C. A. Greated, "Numerical simulation of particle motion in an ultrasound field using the lattice Boltzmann model," *Ultrasonics*, vol. 43, pp. 21–25, 2004.

[40] J. M. Buick, C. L. Buckley, C. A. Greated, and J. Gilbert, "Lattice Boltzmann BGK simulation of nonlinear sound waves: the development of a shock front," *Journal of Physics A*, vol. 33, no. 21, pp. 3917–3928, 2000.

[41] D. Haydock and J. M. Yeomans, "Lattice Boltzmann simulations of acoustic streaming," *Journal of Physics A*, vol. 34, no. 25, pp. 5201–5213, 2001.

[42] E. W. Weisstein, "Doppler effect." From *ScienceWorld* – A Wolfram Web Resource. `http://scienceworld.wolfram.com/physics/DopplerEffect.html`.

[43] F. L. Pedrotti, L. M. Pedrotti, and L. S. Pedrotti, *Introduction to Optics*. Pearson Education, 3rd ed., 2007.

[44] B. Chopard and P. O. Luthi, "Lattice Boltzmann computations and applications to physics," *Theoretical Computer Science*, vol. 217, no. 1, pp. 115–130, 1999.

[45] B. Chopard, P. O. Luthi, and S. Marconi, "A Lattice Boltzmann Model for Wave and Fracture phenomena." Published at arXiv. Freely available at `http://arxiv.org/abs/cond-mat/9812220`.

[46] Y. Kagawa, T. Tsuchiya, B. Fujii, and K. Fujioka, "Discrete Huygens' model approach to sound wave propagation," *Journal of Sound and Vibration*, vol. 218, no. 3, pp. 419–444, 1998.

[47] G. Yan, "A Lattice Boltzmann Equation for Waves," *Journal of Computational Physics*, vol. 161, no. 1, pp. 61–69, 2000.

[48] Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the lattice Boltzmann method," *Physical Review E*, vol. 65, no. 4, p. 046308, 2002.

[49] J. Lätt and B. Chopard, "Lattice Boltzmann Method with regularized non-equilibrium distribution functions." Published at arXiv. Freely available at `http://arxiv.org/abs/physics/0506157v1`.

[50] J. Lätt and B. Chopard, "Lattice Boltzmann method with regularized pre-collision distribution functions," *Mathematics and Computers in Simulation*, vol. 72, no. 2–6, pp. 165–168, 2006.

[51] X. Shan and H. Chen, "Lattice Boltzmann model for simulating flows with multiple phases and components," *Physical Review E*, vol. 47, no. 3, pp. 1815–1819, 1993.

[52] X. Shan and H. Chen, "Simulation of nonideal gases and liquid-gas phase transitions by the lattice Boltzmann equation," *Physical Review E*, vol. 49, no. 4, pp. 2941–2948, 1994.

[53] A. K. Gunstensen, D. H. Rothman, S. Zaleski, and G. Zanetti, "Lattice Boltzmann model of immiscible fluids," *Physical Review A*, vol. 43, no. 8, pp. 4320–4327, 1991.

[54] M. R. Swift, W. R. Osborn, and J. M. Yeomans, "Lattice Boltzmann Simulation of Nonideal Fluids," *Physical Review Letters*, vol. 75, no. 5, pp. 830–833, 1995.

[55] E. G. Flekkøy, "Lattice Bhatnagar-Gross-Krook models for miscible fluids," *Physical Review E*, vol. 47, no. 6, pp. 4247–4257, 1993.

[56] F. J. Alexander, S. Chen, and J. D. Sterling, "Lattice Boltzmann thermohydrodynamics," *Physical Review E*, vol. 47, no. 4, pp. R2249–R2252, 1993.

[57] Y. H. Qian, "Simulating Thermohydrodynamics with Lattice BGK Models," *Journal of Scientific Computing*, vol. 8, no. 3, pp. 231–242, 1993.

[58] Y. Chen, H. Ohashi, and M. Akiyama, "Thermal lattice Bhatnagar-Gross-Krook model without nonlinear deviations in macrodynamic equations," *Physical Review E*, vol. 50, pp. 2776–2783, 1994.

[59] S. Chapman and T. G. Cowling, *The mathematical theory of non-uniform gases.* Cambridge University Press, 1939.

[60] E. W. Weisstein, "Bessel Differential Equation." From *MathWorld* – A Wolfram Web Resource. `http://mathworld.wolfram.com/BesselDifferentialEquation.html`.