Master Thesis

Hydrodynamical experiments with Lattice Boltzmann models

by
Sebastian Hinck
from Stade

Submitted in partial fulfilment of the requirements
for the degree of Master of Science in Physics,
University of Bremen

February 2015

Universität Bremen

ALFRED-WEGENER-INSTITUT
HELMHOLTZ-ZENTRUM FÜR POLAR- UND MEERESFORSCHUNG

Author:
Sebastian Hinck

Matriculation No.:
2134316

E-mail:
s.hinck@uni-bremen.de

Supervisor:
Prof. Dr. Gerrit Lohmann

Second Reviewer:
Prof. Dr. Stefan Bornholdt

Abstract

In this thesis different numerical models based on the lattice Boltzmann equation are presented and tested. Therefore the models are applied to two classical two-dimensional hydrodynamical problems. Parameter studies are performed with particular regard to changes in flow dynamics at hydrodynamical instabilities.

The iLBGK model [Z. Guo et al., 2000] with a D2Q9 lattice is used to study the 2D flow past a cylinder placed between two walls. The transition from a steady flow to a vortex shedding regime is analyzed by varying the Reynolds number and the distance of the cylinder to one wall. Due to interaction of the cylinder's wake with the wall vorticity, the transition is delayed as the cylinder approaches the wall. The results are compared with the findings of Zovatto & Pedrizzetti [2001].

For the simulation of thermal flows, the multi-distribution-function (MDF) approach [Z. Guo et al., 2002a; He et al., 1998] is used. This approach uses the Boussinesq approximation to separate the fluid and the thermal components of the flow, which are solved on separate lattices. Two implementations of this approach are carried out using the LBGK and MRT models [Ginzburg, 2005; Wang et al., 2013]. These thermal models are used to study the 2D Rayleigh-Bénard problem for a fixed Prandtl number $Pr = 0.71$. The transition from the solely conductive to the convective regime is found to be dependent on the wavenumber k of a perturbation. The usage of lateral periodic boundary conditions restricts the possible values for k, which depend on the aspect ratio of the numerical domain. Checked against theoretical results, the critical Rayleigh numbers obtained with the MRT model are found to be more accurate than those obtained with the LBGK model.

German abstract

In dieser Arbeit werden verschiedene auf der Lattice Boltzmann Gleichung basierende numerische Modelle vorgestellt und anhand von zwei klassischen zweidimensionalen Problemen der Fluiddynamik getestet. Instabilitäten in der Dynamik der Strömung werden durch Veränderung wichtiger Parameter des Systems untersucht.
Mithilfe des iLBGK Modells [Z. Guo et al., 2000] wird die Strömung über einen Zylinder in einem durch zwei Wände beschränkten Kanal analysiert. Der Übergang von stationärer Strömung hin zum Ablösen von Wirbeln wird in Abhängigkeit von der Reynoldszahl und dem Abstand des Zylinders von einer Wand untersucht. Wechselwirkungen des Nachlaufs mit der Wandvortizität bewirken bei Verringerung des Abstandes zwischen Zylinder und Wand eine Verzögerung dieses Übergangs. Die gefundenen Ergebnisse werden mit denen von Zovatto & Pedrizzetti [2001] verglichen.
Für die Simulation thermischer Strömungen wird der *multi-distribution-function* (MDF) [Z. Guo et al., 2002a; He et al., 1998] Ansatz benutzt. Dieser Ansatz bedient sich der Boussinesq-Näherung, um die kinematischen und thermischen Komponenten auf zwei separaten Gittern zu berechnen. Das LBGK und das MRT [Ginzburg, 2005; Wang et al., 2013] Modell werden jeweils unter Nutzung dieses Ansatzes auf das 2D Rayleigh-Bénard Problem angewendet, um dieses bei konstanter Prandtlzahl $Pr = 0.71$ zu studieren. Der Übergang vom rein stationären zum konvektiven Zustand erweist sich dabei als abhängig von der Wellenzahl k einer Störung. Die Verwendung seitlich periodischer Randbedingungen beschränkt die möglichen Werte für k, welche durch das Seitenverhältnis des numerischen Bereichs bestimmt sind. Verglichen mit theoretischen Werten ergeben sich aus dem MRT Modell deutlich genauere Werte für die kritische Rayleighzahl als aus dem LBGK Modell.

Acknowledgements

I would like to thank Prof. Gerrit Lohmann for providing me with the opportunity to study such an interesting topic in his working group, for all our helpful discussions and his kind support.

I am especially grateful to Dragos B. Chirila for always sharing so gladly his valuable time, advice and insight. I benefited greatly from his assistance when I first set out to familiarize myself with the subject-matter, and, all along the way, drew a lot of motivation from his encouragement, pointers and feedback.

I owe further thanks to Prof. Stefan Bornholdt for agreeing to act as my second reviewer.

Jeff and Lukas reviewed big parts of my work and provided helpful comments.

Last but not least, I would like to thank my family and friends, and anyone I may have forgotten.

Without all of you, this work would not have been possible.

Contents

1. Introduction

More than 70% of Earth's surface is covered with water, of which the largest portion is found in the oceans. Due to the high heat-capacity of water, the oceans constitute a big reservoir of energy. Therefore the oceans play a major role in the Earth's climate system. The energy is globally redistributed by large-scale currents which are mainly driven by differences in temperature and salinity of the water but also by interactions with the atmosphere [Marshall & Plumb, 2007; Olbers et al., 2012].

Currents and heat transports are also a major factor in many fields of engineering. Drag and lift forces induced by flows around objects must not be neglected: these forces can be volitional, as for example in aerodynamics, but they can also have negative effects as vibrations, which can have catastrophic effects on constructions like bridges and underwater pipes [Zovatto & Pedrizzetti, 2001], to name just a few. The transport of heat also plays a big role when constructing heat exchange devices, as for example in engines or power plants [Kaminski & Jensen, 2011].

What all these different examples have in common is that they all rely on the dynamics of fluids. Understanding these dynamics therefore is of great importance.

In a fundamental sense, a fluid can be described as a conglomerate of particles in interaction with each other. Given the vast number of particles in a real fluid this approach is not constructive, since it is impossible to track each and every particle. However, such a large number of particles bears the possibility of a statistical treatment. Macroscopic variables such as pressure, momentum and temperature can be described by statistical averages of the whole ensemble [Schwabl, 2006; Succi, 2001]. This microscopic point of view is coupled with the macroscopic point of view by the Boltzmann equation [Boltzmann, 1872], which describes the evolution of the probability density of finding a particle at a certain location with a certain momentum. The macroscopic variables are then defined as the moments of this probability distribution function. In the limit of a small Knudsen number the Boltzmann equation recovers the Navier-Stokes equation [Hänel, 2004], which is the classical description used in fluid mechanics, based on the continuum theory [Landau & Lifshitz, 1987].

The Navier-Stokes equation contains nonlinear terms, which may result in a high complexity of the dynamics. The solutions are found to be quite sensitive to the properties of the flow. Hydrodynamic instabilities are characterized by a sudden change in the

dynamics of a flow (for example, the sudden onset of convection or turbulence).
Due to the nonlinearity of the Navier-Stokes equation it can, in general, not be solved analytically; solutions have to be found by experiments [Tritton, 1988]. This can be done either in a laboratory or, thanks to the rapid developments in computer resources in recent decades, numerically. For obvious reasons it is not possible to study atmospheric and oceanic dynamics in laboratory experiments. Even experimental studies for less complex models are often not viable or simply too expensive. Computational fluid dynamics (CFD) offers a variety of different methods for studying fluid flows numerically. Most of those methods approximate solutions by solving algebraic systems obtained from the discretized Navier-Stokes equations [Anderson, 1995; Z. Guo & Shu, 2013].
A different and relatively new approach is the Lattice Boltzmann method (LBM) [Benzi et al., 1992; Succi, 2001]. In this method the lattice Boltzmann equation (LBE) is iterated, which is obtained by discretization of the Boltzmann equation in time and phase space [He & Luo, 1997]. The LBE can be split into two parts: a collision and a streaming step. The streaming step is only a simple copying operation of the distribution functions to the nearest neighbors, while in the collision step the distribution functions are relaxed towards the local Boltzmann-Maxwellian equilibrium, as required by Boltzmann's H-theorem [Wolf-Gladrow, 2000]. LBM models are relatively easy to implement due to their simple representation by collision and streaming steps. Furthermore, the algorithm is local except of the streaming step, which provides a simple possibility for parallelization. Simple and accurate boundary conditions allow for a simple realization of complex geometries.
In this thesis several numerical models based on the LBE are tested and validated by a study of classical fluid dynamical problems. The simulations are done with particular focus on locating hydrodynamical instabilities. The obtained numerical results are compared and validated using either analytical results, if available, or other numerical studies that were obtained using different numerical models.

In chapter 2 a short theoretical introduction of the Boltzmann equation is given and some numerical schemes based on the LBE are presented. Chapter 3 deals with the numerical implementation of the LBM. It provides some guidelines for the parametrization, implementation and parallelization of the algorithm and for the proper treatment of initial and boundary conditions. The presented models are tested in chapter 4 using the two-dimensional flow around a cylinder in an enclosed channel and the two-dimensional Rayleigh-Bénard setup. For both experiments parameter studies are performed with regard to the location and type of hydrodynamical instabilities. Chapter 5 presents the findings and gives a short overview about possible further investigations.

2. Theoretical Background

2.1. The Boltzmann equation

A fluid can be seen as a set of N atoms interacting with each other. Such a system can be described by the Hamiltonian equations [Schwabl, 2006; Wolf-Gladrow, 2000], which requires tracking the position and momentum of every particle (i.e. $2 \cdot DN$ unknowns for a D-dimensional setup). The problem with this approach becomes clear when one realizes the vast number of molecules of a real gas (N is of order of the Avogadro number $N_A \sim 10^{23}$). In the macroscopic world, information about every particle is not needed as the macroscopic properties of the fluid, such as density ρ, velocity of the flow $\mathbf{u}$, and temperature T, are given by statistical averages over the ensemble [Schwabl, 2006; Succi, 2001]. In statistical mechanics, one introduces a probability distribution that describes the probability of finding the system in a certain state at time t, but this still requires knowledge of the states of all particles. In 1872 Boltzmann derived an equation to describe the evolution of the probability density $P(\mathbf{x}, \mathbf{p}, t)$, where $P(\mathbf{x}, \mathbf{p}, t)\mathtt{dxdp}$ is the probability of finding a particle in an infinitesimal cubelet around $(\mathbf{x}, \mathbf{p})$ in the $(2 \cdot D)$-dimensional phase space $\mu = \{(\mathbf{x}, \mathbf{p}) : \mathbf{x}, \mathbf{p} \in \mathbb{R}^D\}$ (the momentum $\mathbf{p}$ is given by the mass m of a particle and its microscopic velocity $\boldsymbol{\xi}$: $\mathbf{p} = m\boldsymbol{\xi}$), by making three assumptions [Boltzmann, 1872; Succi, 2001; Wolf-Gladrow, 2000]:

- the particles interact only through a short-range two body potential (binary collisions)
- external forces do not influence the local collision process (because collisions are assumed to be very fast)
- velocities of the colliding particles are uncorrelated before the collision (molecular chaos hypothesis)

One can define a slightly different distribution function f, that is related to P:

$$f = NmP, \tag{2.1}$$

where N is the total number of particles in the system. f can be seen as the mass density in the phase space [Cercignani, 1988]. Thus f is not normalized to 1, as P was:

$$\int \mathrm{d}\mathbf{x} \int \mathrm{d}\mathbf{p} P = 1, \qquad \int \mathrm{d}\mathbf{x} \int \mathrm{d}\mathbf{p} f = Nm. \tag{2.2}$$

In the following, f will be used instead of P.
The Boltzmann equation then reads as follows:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} f + \mathbf{F} \cdot \nabla_{\mathbf{p}} f = \left.\frac{\partial f}{\partial t}\right|_{\mathrm{coll}} \tag{2.3}$$

Where $\nabla_{\mathbf{x}}$ and $\nabla_{\mathbf{p}}$ are the gradients in physical and momentum space, respectively, $\mathbf{F}$ is an external force acting on the particles and $\left.\frac{\partial f}{\partial t}\right|_{\mathrm{coll}}$ includes the contribution of the collisions. It can also be written in terms of gains and losses:

$$\left.\frac{\partial f}{\partial t}\right|_{\mathrm{coll}} = \Gamma_+ - \Gamma_- \tag{2.4}$$

Here $\Gamma_+ \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{p}\mathrm{d}t$ denotes the probability that one particle enters the infinitesimal cubelet around $(\mathbf{x}, \mathbf{p})$ during the time interval $[t, t + \mathbf{d}t]$ and $\Gamma_- \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{p}\mathrm{d}t$ is the probability that a particle leaves it due to collisions. This term can be rewritten as an integral over the velocity ξ and the solid angle Ω:

$$\Gamma_+ - \Gamma_- = \frac{1}{m} \int \mathrm{d}\boldsymbol{\xi}_1 \int \mathrm{d}\Omega \; \sigma(\Omega) \left|\boldsymbol{\xi} - \boldsymbol{\xi}_1\right| \left[f(\boldsymbol{\xi}') f(\boldsymbol{\xi}_1') - f(\boldsymbol{\xi}) f(\boldsymbol{\xi}_1)\right] \tag{2.5}$$

In this two-particle collision integral $\sigma(\Omega)$ is the differential collision cross section which transfers the velocities prior the collision (unprimed) into the velocities after the collision (primed) [Boltzmann, 1872; Wolf-Gladrow, 2000].

Using these results, Boltzmann showed that the quantity $H(t) = -\int \mathrm{d}\mathbf{x} \int \mathrm{d}\boldsymbol{\xi} f \ln f$, which is related to the entropy, monotonically increases with time ($\frac{\mathrm{d}H}{\mathrm{d}t} \geq 0$). This relationship is know as the *H-Theorem*. When the system is in local equilibrium ($f = f^{\mathrm{eq}}$) H reaches a maximum and the collision-induced gains and losses cancel each other out. One can show that $\ln f^{\mathrm{eq}}$ can be written as a linear combination of the collision invariants (1, ξ, ξ^2) that are proportional to mass, momentum and kinetic energy. Therefore the equilibrium distribution function f^{eq} is the Boltzmann-Maxwellian distribution [Boltzmann, 1872; Cercignani, 1988; Wolf-Gladrow, 2000]:

$$f^{\mathrm{eq}}(\mathbf{x}, \boldsymbol{\xi}) = \rho (2\pi RT)^{-\frac{D}{2}} \exp\left(-\frac{(\boldsymbol{\xi} - \mathbf{u})^2}{2RT}\right) \tag{2.6}$$

The macroscopic variables density ρ, velocity $\mathbf{u}$ and temperature T (which is given by the internal energy per unit mass m: $\varepsilon = \frac{D}{2}RT = \frac{D}{2}\frac{k_\mathrm{B}}{m}T$, k_B: Boltzmann constant) are given as the velocity moments of f^eq [Cercignani, 1988]:

$$\rho = \int \mathrm{d}\boldsymbol{\xi}\ f = \int \mathrm{d}\boldsymbol{\xi}\ f^\mathrm{eq} \tag{2.7a}$$

$$\rho\mathbf{u} = \int \mathrm{d}\boldsymbol{\xi}\ \boldsymbol{\xi} f = \int \mathrm{d}\boldsymbol{\xi}\ \boldsymbol{\xi} f^\mathrm{eq} \tag{2.7b}$$

$$\rho\varepsilon = \frac{1}{2}\int \mathrm{d}\boldsymbol{\xi}\ (\boldsymbol{\xi}-\mathbf{u})^2 f = \frac{1}{2}\int \mathrm{d}\boldsymbol{\xi}\ (\boldsymbol{\xi}-\mathbf{u})^2 f^\mathrm{eq} \tag{2.7c}$$

In general the considered systems are not in equilibrium and thus the collision term does not vanish. It is therefore appropriate to find simpler expression $J(f)$ for this complex non-linear term. This simplified collision operator has to satisfy two principal characteristics [Cercignani, 1990]:

- it has to retain the collisional invariants
- it has to keep the tendency of the system to a Boltzmann-Maxwellian distribution

It is found that the BGK-operator, as proposed by Bhatnagar, Gross, & Krook [1954], has exactly these properties.

$$J(f) = \lambda^{-1}\,(f^\mathrm{eq} - f) \tag{2.8}$$

This expression models the relaxation of f towards the Boltzmann-Maxwellian equilibrium distribution f^eq with a relaxation time λ.

It can be shown that from the Boltzmann equation, the Navier-Stokes equation can be obtained by the Chapman-Enskog expansion [Chapman, 1916, 1918; Enskog, 1917]. A short sketch of this derivation is included in appendix B.

2.2. Lattice Boltzmann

The lattice Boltzmann equation is derived by discretizing the continuous Boltzmann equation (2.3) in space and time. Phase-space is also discretized, i.e. we restrict the velocities of the particles to a set of b discrete velocities $\mathbf{c}_i$, with $0 \leq i \leq (b-1)$. This choice has to be made carefully, so that the hydrodynamical moments can be computed exactly and that symmetries of the equation one tries to solve are not violated. The underlying lattice, i.e. the discretization of space, is determined by the choice of discrete velocities: particles starting with a discretized velocity $\mathbf{c}_i$ from a lattice node at $\mathbf{x}$ arrive

after a time-step δ_t at $\mathbf{x} + \mathbf{c}_i\delta_t$, which should again be a lattice site. Depending on the dimensionality n of the problem and the discrete velocities $\mathbf{c}_i$ one gets different lattices which are referred to in the literature using the $\mathbf{D}n\mathbf{Q}b$ convention. A sketch of the D2Q9 lattice is shown in Fig 2.1.

There are several models based on the lattice Boltzmann equation that were developed to model certain problems. This section gives a short overview of the models used in this thesis. First (section 2.2.1) we deal with the lattice Boltzmann (LBGK) model which uses the BGK approximation to model the collision process. All the other models treated here (except for the MRT model, which uses a different collision operator) are based on the LBGK, which is why derivation of the LBGK is described in more detail. Unfortunately the LBGK is not capable of solving the incompressible Navier-Stokes equation exactly (due to compressibility errors), therefore we introduce an incompressible LBGK model in section 2.2.2. Section 2.2.3 shows a model to simulate thermal flows and section 2.2.4 presents a model, that uses a multi-relaxation-time (MRT) collision operator (the BGK approximation uses only a single relaxation time (SRT)).

2.2.1. LBGK

The lattice Boltzmann equation can be derived from the continuous Boltzmann equation (2.3) by discretization in time and of the phase space. For the LBGK model, which is one of the simplest and most widespread models using the lattice Boltzmann equation, the rather complex collision integral $\left.\frac{\partial f}{\partial t}\right|_{\text{coll}}$ in equation (2.3), that is given by eq. (2.5), is replaced by the simpler BGK model $J(f)$ from equation (2.8):

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} f = \lambda^{-1} \left(f^{\text{eq}} - f\right) \tag{2.9}$$

Since a system without an external force $\mathbf{F}$ is regarded, the force term in equation (2.3) is omitted.

At first the velocity space $\boldsymbol{\xi}$ is discretized into a set of discrete velocities $\mathbf{c}_i$ that are chosen in such a manner that the conservation constraints are preserved [Z. Guo & Shu, 2013]. The choice of these discrete velocities $\mathbf{c}_i$ also has to include certain symmetries that are needed by the formulation of the problem to be solved [He & Luo, 1997] (here the isothermal Navier-Stokes equations). The equilibrium distribution function f^{eq} - equation (2.6) - is expanded in a Taylor series for small velocities $\mathbf{u}$ of the flow. Under the assumption of a low Mach number ($Ma = \frac{|\mathbf{u}|}{c_{\text{s}}} \ll 1$, c_{s} is the speed of sound) terms of order $\mathcal{O}(\mathbf{u}^3)$ and higher are neglected.

$$f^{\mathrm{eq}} = \frac{\rho}{(2\pi RT)^{\frac{D}{2}}} \exp\left(-\frac{\boldsymbol{\xi}^2}{2RT}\right)\left[1 + \frac{\boldsymbol{\xi}\cdot\mathbf{u}}{RT} + \frac{(\boldsymbol{\xi}\cdot\mathbf{u})^2}{2(RT)^2} - \frac{\mathbf{u}^2}{2RT}\right] + \mathcal{O}(\mathbf{u}^3) \tag{2.10}$$

Using a Gaussian quadrature the discrete velocity set $\mathbf{c}_i$ is found that solves exactly

$$\int \mathrm{d}\boldsymbol{\xi}\boldsymbol{\xi}^k f^{\mathrm{eq}} = \sum_i w_i \mathbf{c}_i^k f^{\mathrm{eq}}(\mathbf{c}_i), \tag{2.11}$$

where the maximum value for k depends on the highest moment, which the quadrature should be able to compute exactly (for the isothermal Navier-Stokes: $0 \leq k \leq 3$). After finding an appropriate quadrature, a discrete distribution function $f_i = w_i f(\mathbf{x}, \mathbf{c_i}, t)$ [Z. Guo & Shu, 2013] is defined, where w_i are weight coefficients of the quadrature.

Now only equation (2.9) has to be discretized in time. This is done by integrating eq. (2.9) from t to $t + \delta_t$ along a characteristic line $\boldsymbol{\xi}$ [He & Luo, 1997]. The evolution equation of the LBGK model reads as follows:

$$f_i(\mathbf{x} + \mathbf{c}_i\delta_t, t + \delta_t) - f_i(\mathbf{x}, t) = \tau^{-1}\left(f_i^{\mathrm{eq}}(\mathbf{x}, t) - f_i(\mathbf{x}, t)\right) \tag{2.12}$$

The discretized equilibrium distribution function f_i^{eq} is

$$f_i^{\mathrm{eq}} = w_i\rho\left[1 + \frac{\mathbf{c}_i\cdot\mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i\cdot\mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2}\right], \tag{2.13}$$

where the speed of sound $c_s = \sqrt{RT}$ is coupled to the lattice speed c, which is, as the lattice weights w_i, lattice dependent.

According to equation (2.11) the hydrodynamic moments given by equation (2.7) can now be written as:

$$\rho = \sum_i f_i = \sum_i f_i^{\mathrm{eq}} \tag{2.14a}$$

$$\rho\mathbf{u} = \sum_i \mathbf{c}_i f_i = \sum_i \mathbf{c}_i f_i^{\mathrm{eq}} \tag{2.14b}$$

$$\rho\varepsilon = \frac{1}{2}\sum_i (\mathbf{c}_i - \mathbf{u})^2 f_i = \frac{1}{2}\sum_i (\mathbf{c}_i - \mathbf{u})^2 f_i^{\mathrm{eq}} \tag{2.14c}$$

i	w_i	$\mathbf{c}_i/c$
0	$\frac{4}{9}$	$(0,0)$
1	$\frac{1}{9}$	$(1,0)$
2		$(0,1)$
3		$(-1,0)$
4		$(0,-1)$
5	$\frac{1}{36}$	$(1,1)$
6		$(-1,1)$
7		$(-1,-1)$
8		$(1,-1)$

TABLE 2.1.: Lattice parameters for the D2Q9 model. The discrete velocity vectors are given in terms of the lattice speed $c = \sqrt{3}c_s = \sqrt{3RT}$

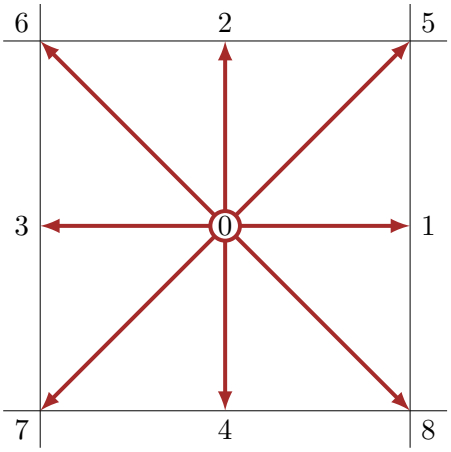

FIGURE 2.1.: Sketch of the D2Q9 lattice. The arrows represent the discrete velocities $\mathbf{c}_i$ and the numbers are the corresponding index i. Each lattice node contains nine populations of distribution functions.

By a Chapman-Enskog expansion[1] the hydrodynamic equations [Z. Guo & Shu, 2013; Watari, 2012] are obtained:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.15a}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla (\rho \mathbf{u} \cdot \mathbf{u}) = -\nabla p + \nu \left[\nabla^2 (\rho \mathbf{u}) + \nabla (\nabla \cdot (\rho \mathbf{u}))\right] \tag{2.15b}$$

Here p is the pressure and ν is the kinematic shear viscosity:

$$p = c_s^2 \rho \tag{2.16a}$$

$$\nu = c_s^2 \delta_t \left(\tau - \frac{1}{2}\right) \tag{2.16b}$$

It can be seen that, if the density variation is small, equations (2.15) are identical to the compressible Navier-Stokes equations [Z. Guo et al., 2000]. In the incompressible limit, when the density variation is negligible, the incompressible Navier-Stokes equations are obtained:

$$\nabla \cdot \mathbf{u} = 0 \tag{2.17a}$$

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} \tag{2.17b}$$

For solving the isothermal Navier-Stokes equations in two dimensions, one often uses

[1] In appendix B, the basic steps of the Chapman-Enskog expansion for the derivation of the Navier-Stokes equation from the continuous Boltzmann equation are sketched out. Watari [2012] provides a nice overview of the CE expansion of the discrete LBE.

the **D2Q9** lattice, which has a set of 9 discrete velocities. In this model the relation between the speed of sound c_s and the lattice speed c is $c_s^2 = \frac{c^2}{3}$. The lattice weights w_i and the discrete velocity vectors $\mathbf{c}_i$ are given in table 2.1. A sketch of the lattice can be found in Fig 2.1. The derivation of all important parameters of some lattices can be found in He & Luo [1997] and a table with the parameters of the most common lattices is given in Z. Guo & Shu [2013].

2.2.2. Incompressible LBGK

As mentioned in section 2.2.1, the LBGK model does not recover exactly the incompressible Navier-Stokes equations. Since it can only be used to model compressible flows in the incompressible limit, Z. Guo et al. [2000] have proposed an incompressible model based on the LBGK. Here the distribution function is not based on the density ρ but on the pressure p. In the paper cited above only the two dimensional iD2Q9 model is presented. A more general description for arbitrary lattices is given in Nan-Zhong et al. [2004].

The evolution equation for the iLBGK model is the same as the LBGK equation (2.12). For the pressure-based distribution function g_i is used to distinguish it from the density-based.

$$g_i(\mathbf{x} + \mathbf{c}_i\delta_t, t + \delta_t) - g_i(\mathbf{x}, t) = \frac{1}{\tau}\left[g_i^{\mathrm{eq}}(\mathbf{x}, t) - g_(\mathbf{x}, t)\right]. \tag{2.18}$$

The equilibrium distribution function is defined as

$$g_i^{\mathrm{eq}} = \lambda_i p + s_i(\mathbf{u}). \tag{2.19}$$

λ_i and $s_i(\mathbf{u})$ are defined as follows:

$$s_i(\mathbf{u}) = w_i\left[\frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2}\right] \tag{2.20a}$$

$$\lambda_0 = (w_0 - 1)/c_s^2 \tag{2.20b}$$

$$\lambda_i = w_i/c_s^2 \quad , \qquad i \neq 0. \tag{2.20c}$$

The velocity and the pressure can be obtained by

$$\mathbf{u} = \sum_{i \geq 0} \mathbf{c}_i g_i \tag{2.21a}$$

$$p = -\lambda_0^{-1}\left[\sum_{i>0} g_i + s_0(\mathbf{u})\right]. \tag{2.21b}$$

The viscosity ν is computed in the same way as in the LBGK model (see eq. (2.16b)).

In the appendix of Z. Guo et al. [2000], the Chapman-Enskog expansion for the iLBGK model is summarized and it is shown that the incompressible Navier-Stokes equations are recovered accurate to the order $\mathcal{O}(\delta_t^2)$ in the continuity equation and accurate to the order $\mathcal{O}(\delta_t^2 + \delta_t Ma^2)$ in the momentum equation.

2.2.3. Coupled LBGK

For the discussions in the previous sections only mass and momentum conservation were considered, since those models were restricted to isothermal problems. For a lot of applications,however, it is crucial to include thermal effects. One approach to do this is to recover the energy equation by choosing a more complex lattice with more discrete velocities and to include higher order velocity terms in the equilibrium distribution function. This approach is also called multi-speed (MS) due to the larger set of velocities $\mathbf{c}_i$. But this model lacks in numerical stability, while allowing only small variation in temperature [Z. Guo et al., 2002a]. Also, the Prandtl number Pr (the ratio between the kinematic viscosity ν and the thermal diffusivity κ) is fixed for this approach using the BGK model, which only uses a single relaxation time to relax the moments to equilibrium [Cercignani, 1988; Shi et al., 2004].

In the multi distribution function (MDF) approach, the temperature is regarded as a passive scalar that is passively advected by the fluid. Therefore one assumes the viscous heat dissipation and compression work to be negligible [Z. Guo et al., 2002a]. The temperature equation can be solved on an own lattice using the LBGK model with an own distribution function. The effect of the temperature field is added as an external force acting on the fluid that is proportional to the density variation due to temperature differences (Boussinesq approximation). This latter model will be described in the following.

The equations for the fluid can be solved as before using the LBGK or the incompressible iLBGK (as done in Z. Guo et al. [2002a]) model, described in the previous sections. The only difference is a force term in the evolution equation, to be treated in more detail later. When assuming that the viscous heat dissipation and the compression work are negligible, as noted above, and that the temperature field is advected by the fluid, the temperature field can be described by an advection-diffusion equation [Tritton, 1988]:

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \kappa \nabla^2 T \tag{2.22}$$

This equation is solved on a separate lattice, the nodes of which overlap those of the original lattice (for solving the incompressible fluid component). For the sake of simplicity, the space and time discretizations of the two lattices are often taken to be identical.

However, because the temperature field T is a scalar and its higher order derivatives are not of interest here, it is sufficient to choose a lattice with less discrete velocities to solve equation (2.22) (in two dimensions i.e. D2Q4, D2Q5). By a Chapman-Enskog expansion it can be shown that the LBGK equation for the temperature field, where T_i is the temperature distribution function,

$$T_i\left(\mathbf{x}+\mathbf{c}_i\delta_t, t+\delta_t\right) - T_i\left(\mathbf{x},t\right) = \frac{1}{\tau_T}\left[T_i^{\mathrm{eq}}\left(\mathbf{x},t\right) - T_i\left(\mathbf{x},t\right)\right] \tag{2.23}$$

solves equation (2.22) accurate to the order $\mathcal{O}(\delta_t^2)$. The thermal diffusivity κ is defined in a similar way as the viscosity ν was:

$$\kappa = c_s^2\delta_t\left(\tau_T - \frac{1}{2}\right) \tag{2.24}$$

It is important to note that the speed of sound c_s in this equation is, of course, given by the lattice on which the temperature is solved on, and that the relaxation time τ_T can be chosen independently of τ to adjust κ. In the approximation of the equilibrium distribution function only terms of order $\mathcal{O}(\mathbf{u})$ are kept:

$$T_i^{\mathrm{eq}} = w_i T\left[1 + \frac{\mathbf{c}_i\cdot\mathbf{u}}{c_s^2}\right]. \tag{2.25}$$

According to equation (2.14a), the temperature can be obtained by summing up all distribution functions.

$$T = \sum_i T_i \tag{2.26}$$

In Z. Guo et al. [2002a], the D2Q4 lattice is used. The discrete velocities are parallel to the x- and y-axis, the lattice weights w_i are all $\frac{1}{4}$ and the speed of sound is $c_s = \frac{c}{\sqrt{2}}$. For the implementations in the present thesis, a D2Q5 lattice was chosen. The structure is similar to the D2Q4, but it has an additional rest-velocity c_0. The weights are $w_0 = \frac{1}{3}$ and $w_i = \frac{1}{6}$ for $i = 1-4$ and the speed of sound is the same as for the D2Q9 lattice $c_s = \frac{c}{\sqrt{3}}$ [Latt, 2007].

The Boussinesq approximation considers the density to be constant, except in the body force term. The density can be written as a linear function of the temperature:

$$\rho = \rho_0\left(1 - \beta(T - T_0)\right) \tag{2.27}$$

The fluid at its average temperature T_0 has the average density ρ_0; β is the coefficient of thermal expansion. Using the Boussinesq approximation, the system can be described

by the following set of equations:

$$\nabla \cdot \mathbf{u} = 0 \tag{2.28a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla(\mathbf{u} \cdot \mathbf{u}) = -\nabla p + \nu \nabla^2 \mathbf{u} - \mathbf{g}\beta(T - T_0) \tag{2.28b}$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \kappa \nabla^2 T. \tag{2.28c}$$

The force term in equation (2.28b) is equation (2.27) multiplied by the gravitational acceleration $\mathbf{g}$, while its constant part is absorbed into the pressure term [Z. Guo et al., 2002a]. This force term has to be introduced into the evolution equation of the fluid. Z. Guo et al. [2002a] use quite a simple formulation. This approach gives proper results as long as the temperature gradients are small; if this is not the case, discrete lattice effects have to be considered, as discussed by Z. Guo et al. [2002b]. They give a different approach for the implementation of the force for the LBGK model, which includes these effects. The force term that has to be added to the right hand side of equation 2.12 reads as follows,

$$F_i = \left(1 - \frac{1}{2\tau}\right) \omega_i \left[\frac{\mathbf{c}_i - \mathbf{u}}{c_s^2} + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^4}\mathbf{c}_i\right] \cdot \mathbf{G}, \tag{2.29}$$

where $\mathbf{G} = \rho_0 \beta(T - T_0)\mathbf{g}$ is the gravitational force, which must also be considered when computing the velocity of the fluid:

$$\rho \mathbf{u} = \sum_i \mathbf{c}_i f_i + \frac{\delta_t}{2}\mathbf{G}. \tag{2.30}$$

2.2.4. Multi-relaxation-time

In the previous section some lattice Boltzmann models that are based on the BGK approximation of the collision term were presented. This simplifies the algorithm a lot, but it also has some disadvantages: It is relatively unstable numerically, and for some boundary conditions the effective position of the boundary depends non-trivially on the relaxation time τ [Wang et al., 2013]. These drawbacks are overcome by the multi-relaxation-time (MRT) LB model proposed in d'Humieres [1992]; d'Humieres et al. [2002]; Lallemand & Luo [2000]. The model also uses a linearized collision operator that looks similar to the one of the BGK model, but the collision process does take place in the momentum space and there are multiple relaxation rates defined for different moments. The distribution functions $\mathbf{f}$, where the bold $\mathbf{f} = (f_0, f_1, ..., f_N)^T$ denotes vector notation, can be mapped onto moments by multiplying by a transformation matrix $\mathbf{M}$.

$$\mathbf{m} = \mathbf{M} \cdot \mathbf{f}, \qquad \mathbf{f} = \mathbf{M}^{-1} \cdot \mathbf{m} \tag{2.31}$$

$\mathbf{m}$ is the vector containing the hydrodynamical moments ($\mathbf{m} = (m_0, m_1, ..., m_N)^T$) and $\mathbf{M}^{-1}$ is the inverse of $\mathbf{M}$. The evolution equation can be written as

$$\mathbf{f}(\mathbf{x} + \mathbf{c}\delta_t, t + \delta_t) = \mathbf{f}(\mathbf{x}, t) + \mathbf{Q}(\mathbf{x}, t) \tag{2.32a}$$

$$\mathbf{Q}(\mathbf{x}, t) = \mathbf{M}^{-1} \cdot \mathbf{S} \cdot [\mathbf{m}^{\mathrm{eq}}(\mathbf{x}, t) - \mathbf{m}(\mathbf{x}, t)] \tag{2.32b}$$

Here $\mathbf{m}^{\mathrm{eq}}$ is a vector of the equilibrium functions of the moments and $\mathbf{S}$ is a diagonal matrix containing the relaxation rates of the moments.

This model can also be used to model thermal flows using a similar approach as shown in section 2.2.4. Therefore the thermal component has to be solved by modeling an advection-diffusion equation using the MRT model [Ginzburg, 2005] and coupled to the fluid component via the Boussinesq approximation, as described in Wang et al. [2013]. As before, the fluid and thermal components are solved on a D2Q9 and D2Q5 lattice, respectively. This approach is used in section 4.2 and described in the following.

The two lattices are chosen such that the positions of their lattice nodes match. The convention for numbering the discrete velocities is the same as in table 2.1 (the D2Q5 lattice has, of course, only the first five velocities $\mathbf{c}_i$, with $i = 0, \ldots, 4$). To keep both models distinguishable, the notation for the fluid model is kept as given above, whereas for the thermal component the transformation matrix is called $\mathbf{N}$, the vector of the moments $\mathbf{n}$, the relaxation matrix $\mathbf{R}$ and the vector of distribution functions $\mathbf{g}$. The transformation matrices are defined as:

$$\mathbf{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{N} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -4 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 1 & -1 \end{pmatrix}.$$

The first three elements of the vector $\mathbf{m}$, which contains the moments of the fluid are the density ρ and the two components of the momentum $\rho\mathbf{u}$. Since only incompressible fluids are studied here, it is possible to do some approximations:

$$\begin{aligned} \rho &= \rho_0 + \delta\rho = \rho_0 + \sum_i f_i \\ \rho_0 \mathbf{u} &= \sum_i \mathbf{c}_i f_i, \end{aligned} \tag{2.33}$$

with $\rho_0 = 1$ being the reference density, which is scaled in such a way that it is unity here, and $\delta\rho$ as density fluctuations around ρ_0. The equilibrium moments $\mathbf{m}^{\mathrm{eq}}$ can be written as:

$$\begin{aligned} m_0^{\mathrm{eq}} &= \delta\rho, \qquad m_1^{\mathrm{eq}} = \rho_0 u_x, \qquad m_2^{\mathrm{eq}} = \rho_0 u_y \\ m_3^{\mathrm{eq}} &= -2\delta\rho + 3\rho_0(u_x^2 + u_y^2), \quad m_4^{\mathrm{eq}} = \rho_0(u_x^2 - u_y^2) \\ m_5^{\mathrm{eq}} &= \rho_0 u_x u_y, \quad m_6^{\mathrm{eq}} = -\rho_0 u_x, \quad m_7^{\mathrm{eq}} = -\rho_o u_y \\ m_8^{\mathrm{eq}} &= \delta\rho - 3\rho_0(u_x^2 + u_y^2). \end{aligned} \tag{2.34}$$

For the thermal field, the temperature and can be computed as

$$T = \sum_i g_i. \tag{2.35}$$

The equilibrium moments $\mathbf{n}^{\mathrm{eq}}$ are

$$n_0^{\mathrm{eq}} = T, \quad n_1^{\mathrm{eq}} = u_x T, \quad n_2^{\mathrm{eq}} = u_y T, \quad n_3^{\mathrm{eq}} = aT, \quad n_4^{\mathrm{eq}} = 0, \tag{2.36}$$

u_x and u_y are the velocity components derived from the fluid model, and a is a parameter connected to the thermal diffusivity κ and is treated later.
The relaxation matrices are in diagonal form and can be written as

$$\mathbf{S} = \mathrm{diag}(0, 1, 1, s_e, s_\nu, s_\nu, s_q, s_q, s_\epsilon) \tag{2.37}$$

$$\mathbf{R} = \mathrm{diag}(0, \sigma_\kappa, \sigma_\kappa, \sigma_e, \sigma_\nu). \tag{2.38}$$

The choice of these parameters influences the stability of the model. Wang et al. [2013] use the two-relaxation-times model (TRT), for which $s_\nu = s_e = s_\epsilon$ and $\sigma_e = \sigma_\nu$ is chosen. This choice is also made here. Similar to the relaxation parameter τ for the LBGK model, s_ν is related to the viscosity ν

$$\nu = \frac{1}{3}\left(\frac{1}{s_\nu} - \frac{1}{2}\right) \tag{2.39}$$

and s_q is related to s_ν:

$$s_q = 8\frac{2 - s_\nu}{8 - s_\nu}. \tag{2.40}$$

For physical reasons, both parameters must be within the range $[0, 2)$ [Wang et al., 2013]. The thermal diffusivity κ is related to σ_κ:

$$\kappa = \frac{4 + a}{10}\left(\frac{1}{\sigma_\kappa} - \frac{1}{2}\right). \tag{2.41}$$

These parameters are fixed for physical reasons [Wang et al., 2013]:

$$\sigma_\kappa = 3 - \sqrt{3} \qquad \text{and} \tag{2.42}$$

$$\sigma_\nu = 2(2\sqrt{3} - 3), \tag{2.43}$$

which results that κ is only dependent on the parameter a:

$$\kappa = \frac{\sqrt{3}}{60}(4 + a), \qquad \text{with } -4 < a < 1. \tag{2.44}$$

The implementation of the body force in MRT models is more intuitive than it is for the LBGK models because the momentum can be directly added to the corresponding moment. To achieve better accuracy, this is done by adding one half of the momentum before and after the collision, respectively [Dellar, 2003]. This procedure is also known as 'Strang-Splitting'. The Boussinesq approximation is used to write the force-term as

$$F_i = w_i \frac{\mathbf{c}_i}{c_s^2} \cdot \mathbf{G}. \tag{2.45}$$

Here w_i stands for the same lattice weights as for the LBGK models (see table 2.1) and $\mathbf{G} = \rho_0 \beta (T - T_0)\mathbf{g}$ is the gravitational force, as in section 2.2.3.

3. Numerical implementation

In the previous chapter the derivation of the lattice Boltzmann equation was described and a small overview of some models that are based on it was presented. In this chapter the numerical implementation of these models is treated. Therefore the system has to be made dimensionless and the parameters, to ensure they describe the given problem while keeping the model in a numerically stable regime and lead to results with the desired accuracy in an appropriate computation time, have to be found (see section 3.1). The implementation is done in *C++* (for more information see, for example, Stroustrup [2013]). The most important parts of the algorithm are shown with the help of snippets of the source code in section 3.2. After that, two crucial steps in computational fluid dynamics are discussed: the finding of appropriate initial conditions (see section 3.3) and the proper treatment of the boundaries (see section 3.4). Section 3.6 introduces the C++ libraries that were used. One big advantage of the (most basic) LBM over other methods in numerical fluid dynamics is the fact that the algorithm is local: for the computation of the state of a node at the next time-step, or rather the collision-step, no information from other nodes of the fluid domain is needed. Therefore, this algorithm can easily be parallelized for faster execution on modern multi-core computers. Section 3.5 describes how this is done.

For the sake of simplicity, and because the D2Q9 lattice is used for all numerical experiments in this thesis (except for solving the advection diffusion equation for the thermal flows), the discussion in this chapter is restricted to this specific lattice.

3.1. Parameter scaling

Computer-based fluid simulations are supposed to model the behavior of real physical systems. In physical descriptions of those systems, the variables are given in terms of physical units (such as the SI unit system), but the choice of those units is rather arbitrary as the variables could also be given in any other reference system. One could, for example, choose a set of characteristic quantities to scale all variables and transform the physical system to a dimensionless formulation. The theory of dynamic similarity

[Tritton, 1988] implies, that for certain physical systems, solutions can be found independent of their physical scales as long as certain ratios of the characteristic scales agree. These ratios are dimensionless numbers that characterize the behavior of the system. Some examples from fluid mechanics that will emerge again later are the Reynolds (Re), Grashof (Gr), Prandtl (Pr) and Rayleigh (Ra) numbers.

$$Re = \frac{\text{inertial force}}{\text{viscous force}} = \frac{u_0 L_0}{\nu} = \frac{L_0^2}{t_0 \nu} \tag{3.1a}$$

$$Gr = \frac{\text{buoyancy force}}{\text{viscous force}} = \frac{g\beta\Delta T L_0^3}{\nu^2} \tag{3.1b}$$

$$Pr = \frac{\text{viscous diff.}}{\text{thermal diff.}} = \frac{\nu}{\kappa} \tag{3.1c}$$

$$Ra = Gr \cdot Pr = \frac{g\beta\Delta T L_0^3}{\nu\kappa} \tag{3.1d}$$

In these equations L_0 and t_0 are the characteristic length and time scales. $u_0 = \frac{L_0}{t_0}$ is the characteristic velocity of the flow; ν is the viscosity, κ the thermal diffusivity, g the gravity, β the thermal expansion coefficient, and ΔT the characteristic temperature difference in the fluid.

In computational fluid dynamics, the setup of the problem is given in a dimensionless form because it is more general than a physical description and because it is easy to switch between both formulations.

For numerical implementations, the configuration space is discretized, and it is appropriate to formulate the problem in terms of this fragmentation. Therefore, the spacing between two sites and the length of a time step are used as characteristic quantities.

In this section we provide a short overview of these transformations. However, the formulation and discretization depend strongly on the problem and the knowledge about the flow, and are therefore described in more detail for each experiment in the chapter 4.

Dimensionless formulation

Identifying the relevant scales of the flow makes it possible to transfer the system into a dimensionless reference frame. The typical length and time scales are denoted here L_0 and t_0, respectively. ρ_0 is the mean density. The conversion between the physical and

dimensionless systems is described by the set of equations in (3.2).

$$x^{(\mathrm{p})} = x^{(\mathrm{dl})} L_0^{(\mathrm{p})} \rightarrow \quad \partial_x^{(\mathrm{p})} = {L_0^{(\mathrm{p})}}^{-1} \partial_x^{(\mathrm{dl})} \tag{3.2a}$$

$$t^{(\mathrm{p})} = t^{(\mathrm{dl})} t_0^{(\mathrm{p})} \rightarrow \quad \partial_t^{(\mathrm{p})} = {t_0^{(\mathrm{p})}}^{-1} \partial_t^{(\mathrm{dl})} \tag{3.2b}$$

$$u^{(\mathrm{p})} = u^{(\mathrm{dl})} u_0^{(\mathrm{p})} \quad , \quad u_0^{(\mathrm{p})} = \frac{L_0^{(\mathrm{p})}}{t_0^{(\mathrm{p})}} \tag{3.2c}$$

$$\nu^{(\mathrm{p})} = \nu^{(\mathrm{dl})} \nu_0^{(\mathrm{p})} \quad , \quad \nu_0^{(\mathrm{p})} = \frac{{L_0^{(\mathrm{p})}}^2}{t_0^{(\mathrm{p})}} \tag{3.2d}$$

$$\rho^{(\mathrm{p})} = \rho^{(\mathrm{dl})} \rho_0^{(\mathrm{p})} \tag{3.2e}$$

$$p^{(\mathrm{p})} = p^{(\mathrm{dl})} p_0^{(\mathrm{p})} \quad , \quad p_0^{(\mathrm{p})} = \rho_0^{(\mathrm{p})} \left(\frac{L_0^{(\mathrm{p})}}{t_0^{(\mathrm{p})}} \right)^2 , \tag{3.2f}$$

where superscripts 'p' and 'dl' are physical and dimensionless frames of reference, respectively. It can be noted that L_0, t_0, ρ_0, and therefore also u_0, are unity in this dimensionless formulation. As noted above, the dimensionless numbers (such as the Reynolds number Re) are independent of the reference frame. From the definition of the Reynolds number one obtains:

$$Re = \frac{u_0 L_0}{\nu} = \frac{{L_0}^2}{t_0 \nu} = \frac{{L_0^{(\mathrm{dl})}}^2}{t_0^{(\mathrm{dl})} \nu^{(\mathrm{dl})}} = \frac{1}{\nu^{(\mathrm{dl})}} \tag{3.3}$$

Formulation in lattice units

The representation of the variables in lattice units is derived in a similar manner as delineated above. The lattice spacing δ_x and a time step δ_t are chosen as characteristic length and time scales. To keep the mean density in lattice units $\rho_0^{(\mathrm{lb})}$ unity, $\rho_0^{\prime(\mathrm{dl})}$ is set to be 1. The prime ($'$) is used in the following equations to indicate that these variables are not the same variables (in another unit system) as the variables of the same name

in equations (3.2).

$$x^{(\mathrm{dl})} = x^{(\mathrm{lb})}\delta_x^{(\mathrm{dl})} \rightarrow \quad \partial_x^{(\mathrm{dl})} = {\delta_x^{(\mathrm{dl})}}^{-1}\partial_x^{(\mathrm{lb})} \tag{3.4a}$$

$$t^{(\mathrm{dl})} = t^{(\mathrm{lb})}\delta_t^{(\mathrm{dl})} \rightarrow \quad \partial_t^{(\mathrm{dl})} = {\delta_t^{(\mathrm{dl})}}^{-1}\partial_t^{(\mathrm{lb})} \tag{3.4b}$$

$$u^{(\mathrm{dl})} = u^{(\mathrm{lb})}c^{(\mathrm{dl})} \quad , \quad c^{(\mathrm{dl})} = \frac{\delta_x^{(\mathrm{dl})}}{\delta_t^{(\mathrm{dl})}} \tag{3.4c}$$

$$\nu^{(\mathrm{dl})} = \nu^{(\mathrm{lb})}\nu_0'^{(\mathrm{dl})} \quad , \quad \nu_0'^{(\mathrm{dl})} = \frac{{\delta_x^{(\mathrm{dl})}}^2}{\delta_t^{(\mathrm{dl})}} \tag{3.4d}$$

$$\rho^{(\mathrm{dl})} = \rho^{(\mathrm{lb})}\rho_0'^{(\mathrm{dl})} \tag{3.4e}$$

$$p^{(\mathrm{dl})} = p^{(\mathrm{lb})}p_0'^{(\mathrm{dl})} \quad , \quad p_0'^{(\mathrm{dl})} = \rho_0'^{(\mathrm{dl})}\left(\frac{\delta_x^{(\mathrm{dl})}}{\delta_t^{(\mathrm{dl})}}\right)^2, \tag{3.4f}$$

where 'lb' denotes that the variables are given in lattice units. It is obvious that δ_t and δ_t in lattice units are unity. For the lattice speed c and the speed of sound c_s one gets:

$$\begin{aligned} & \delta_x^{(\mathrm{lb})} = \Delta x = 1 \quad , \quad \delta_t^{(\mathrm{lb})} = \Delta t = 1 \\ \rightarrow \quad & c^{(\mathrm{lb})} = \frac{\Delta x}{\Delta t} = 1 \quad \rightarrow \quad c_s^{(\mathrm{lb})} \overset{\mathrm{D2Q9}}{=} \frac{c^{(\mathrm{lb})}}{\sqrt{3}} = \frac{1}{\sqrt{3}} \end{aligned} \tag{3.5}$$

The difficulty with the scaling of the parameters is the choice of δ_x and δ_t, which is not explicit. The lattice spacing δ_x can be set relatively freely. Therefore L_0 is split into N_{x0} parts of length $\delta_x = \frac{L_0}{N_{x0}}$. When doing this, one has to bear in mind that accuracy increases quadratically with increasing resolution (because the lattice Boltzmann methods introduced in the previous sections are second order accurate in space) but so does the computational effort (for a 2D setup; for a 3D setup the computational costs scale $\sim N_{x0}^3$). The choice of the size of a time step δ_t is coupled to δ_x. If chosen too small, the computational effort increases, and if it is chosen too large, the error terms associated with δ_t grow and ruin accuracy gained from the choice of δ_x. Therefore, the error terms associated with δ_x and δ_t should be of the same order.

$$\mathcal{O}\left(\epsilon(\delta_x)\right) = \mathcal{O}\left(\epsilon(\delta_t)\right) \tag{3.6}$$

As already mentioned above, the lattice Boltzmann models used here are second order accurate in space:

$$\epsilon(\delta_x) \sim \delta_x^2 \tag{3.7}$$

The numerical accuracy in time in terms of the Mach number

$$Ma = \frac{|u_0|}{c_s} = \frac{|u_0^{(\mathrm{lb})}|}{c_s^{(\mathrm{lb})}} \overset{\mathrm{D2Q9}}{=} \sqrt{3}\frac{\delta_t^{(\mathrm{dl})}}{\delta_x^{(\mathrm{dl})}} \tag{3.8}$$

for the LBGK model [Hänel, 2004] and the incompressible LBGK model [Z. Guo et al., 2000] are

$$\text{LBGK:} \quad \epsilon(\delta_t) \sim Ma^2 \sim \frac{\delta_t^2}{\delta_x^2} \tag{3.9a}$$

$$\text{iLBGK:} \quad \epsilon(\delta_t) \sim \delta_t^2 + \delta_t Ma^2 \sim \delta_t^2 + \frac{\delta_t^3}{\delta_x^2} \tag{3.9b}$$

With equation (3.6) constraints for δ_t are found:

$$\text{LBGK:} \quad \delta_t \sim \delta_x^2 \tag{3.10a}$$

$$\text{iLBGK:} \quad \delta_t \sim \delta_x^{4/3} \tag{3.10b}$$

3.2. Basic algorithm

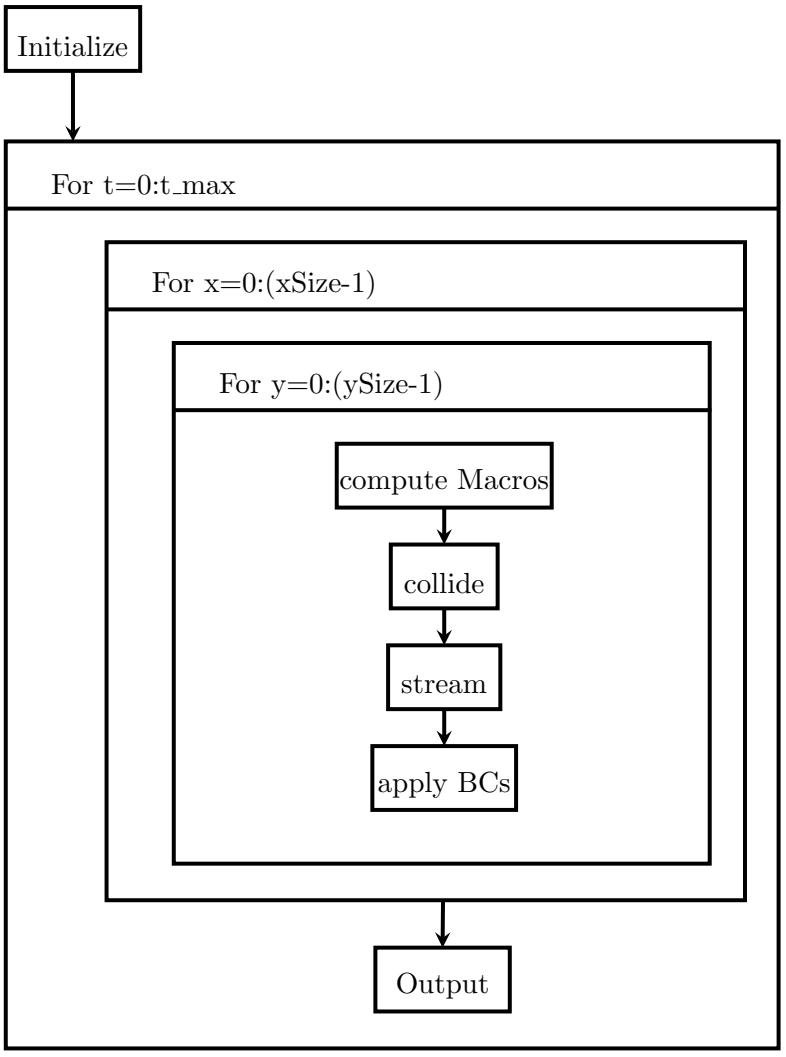

FIGURE 3.1.: Sketch of the lattice Boltzmann algorithm.

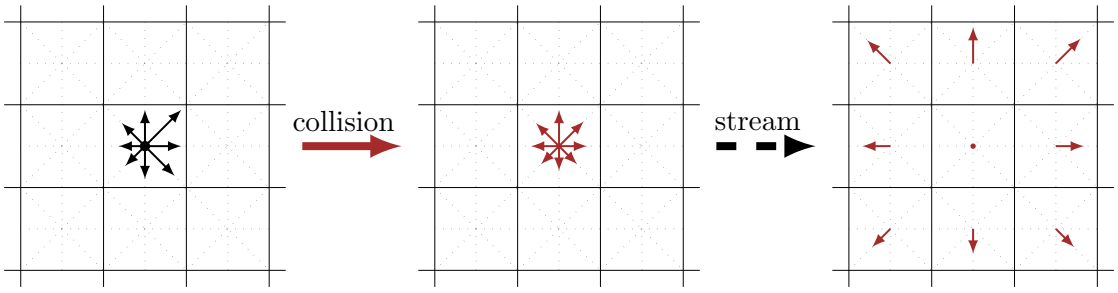

FIGURE 3.2.: Illustration of the collision and streaming processes.

The algorithm for iterating the lattice Boltzmann equation is quite simple and short. A rough sketch is shown in Fig. 3.1. During the initialization, information about the domain size and the properties of the flow is read from a configuration file, and the memory for the macroscopic variables and the distribution functions is allocated. Depending on the problem, the macroscopic variables and the distribution functions are set to initial conditions. This will be discussed in more detail in section 3.3.
After creating the environment for the simulation, the iterations begin. For every time-step (this loop is represented in Fig. 3.1 by the biggest box) each lattice node is updated (nested loops over the discretized domain - in Fig. 3.1, loops over x and y) by evaluating the lattice Boltzmann equation (eq. (2.12)). This equation can be rewritten in two parts (as illustrated in Fig. 3.2), namely a collision step and a streaming step:

$$f_i^*(\mathbf{x},t) = f_i(\mathbf{x},t) + \tau^{-1}\left(f_i^{\mathrm{eq}}(\mathbf{x},t) - f_i(\mathbf{x},t)\right) \qquad \text{collision} \qquad (3.11a)$$

$$f_i(\mathbf{x}+\mathbf{c}_i\delta_t, t+\delta_t) = f_i^*(\mathbf{x},t) \qquad \text{streaming} \qquad (3.11b)$$

In the collision step (3.11a), first the post-collision distributions $f_i^*(\mathbf{x},t)$ are computed from the distributions $f_i(\mathbf{x},t)$ and macroscopic variables at that node. In a second step, (3.11b) these post-collision distributions are streamed to the corresponding adjacent nodes, which is basically a simple copying process. Two arrays are used to store these distribution functions; one for the incoming distribution functions $f_i(\mathbf{x},t)$ and one which the post-collision distribution functions are "streamed" into. To evaluate eq. (3.11a) the equilibrium distribution function f_i^{eq} is needed. It depends on a number of macroscopic variables (compare eq. (2.13)) given by equations (2.14). Function `computeMacros` (see listing 3.1) computes the macroscopic variables for a node at position (`_x, _y`). The arrays of the macroscopic variables density `rho` and the velocity components in x and y direction `ux`, `uy` are computed from the 'incoming' distribution functions. The components of the discrete velocities $\mathbf{c}_i$ are stored in variables `cx` and `cy`. Function `ind2inx` returns the array index of the element, given by the lattice indices `_x` and `_y` (for addressing the distribution functions of a node, a third index for the discrete velocity needs to be specified). Once the macroscopic variables have been computed, the equilibrium

```
1  void computeMacros(int _x, int _y){
2   int i_count, xy_inx, xyi_inx;
3
4   xy_inx = ind2inx(_x, _y);
5
6   rho[xy_inx] = 0.0;
7   ux[xy_inx] = 0.0;
8   uy[xy_inx] = 0.0;
9
10  for (i_count = 0; i_count < 9; i_count++){
11    xyi_inx = ind2inx(_x, _y, i_count);
12    rho[xy_inx] += df_in[xyi_inx];
13    ux[xy_inx] += cx[i_count] * df_in[xyi_inx];
14    uy[xy_inx] += cy[i_count] * df_in[xyi_inx];
15  }
16
17  ux[xy_inx] = ux[xy_inx]/rho[xy_inx];
18  uy[xy_inx] = uy[xy_inx]/rho[xy_inx];
19
20 }
```

LISTING 3.1: Computation of the macroscopic variables ρ and $\mathbf{u}$

distribution function f_i^{eq} can be computed by function `compute_dfEq` (see listing 3.2). This function is called in the collision process in function `collide` (listing 3.3), which models equation (3.11a). Here, instead of the relaxation time τ, the collision frequency `Omega` is used, which is just another notation ($\omega = \tau^{-1}$). It should be noted that the collision process is only executed for lattice nodes that are inside the fluid domain. This is why line 8 checks if the variable `type` for the current node equals `'f'`, which stands for 'fluid'. The array `type` acts as a mask that informs about the position of each node (e.g. **f**luid or **w**all). The routine to be carried out instead of the collision in case of a boundary is discussed in section 3.4. An array containing the post-collision distribution functions of the current node is returned and the function `stream` (listing 3.4) is called. In the streaming step, the post-collision distributions f_i^* are copied to the adjacent node, which corresponds to their discrete velocity $\mathbf{c}_i$. To prevent segmentation faults, it is necessary to check if the adjacent node is inside the domain and find a solution if not (see section 3.4).

The next process in Fig. 3.1 is called *apply BCs*. This should indicate that proper boundary conditions must be defined. But there is no general rule where and when to

```
1  double compute_dfEq(int _x, int _y, int _i){
2   double dfEq, uv, u_sq;
3   int xy_inx = ind2inx(_x, _y);
4
5   uv = cx[_i] * ux[xy_inx] + cy[_i] * uy[xy_inx];
6   u_sq = ux[xy_inx] * ux[xy_inx] + uy[xy_inx] * uy[xy_inx];
7
8   dfEq = rho[xy_inx] * w[_i] * ( 1. + 3. * uv + 4.5 * uv * uv - 1.5 * u_sq );
9
10   return dfEq;
11 }
```

LISTING 3.2: Computation of the equilibrium distribution function f_i^{eq}

```
1  void collide(int _x, int _y, double *_df_xy){
2    int i_count;
3    int xy_idx;
4    double df_eq_i;
5
6    xy_idx = ind2inx(_x, _y);
7
8    if (type[xy_idx] == 'f'){
9      for (i_count = 0; i_count < 9; i_count++){
10       df_eq_i = compute_dfEq(_x, _y, i_count);
11       _df_xy[i_count] = (1. - Omega) * df_in[ind2inx(_x,_y,i_count)]  + Omega *
       df_eq_i;
12     }
13   }
14 }
```

LISTING 3.3: Collision process: relaxation of the distribution functions to their equilibrium.

apply the BC. It depends strongly on the boundary's type and design. Here again the reader is referred to section 3.4.

Once the collision and streaming processes for all lattice nodes have been completed, all relevant results can be written into an output file. In the implementations for this thesis the results are written into a netCDF file (for more information, see appendix C).

The last task before every next time-step is to swap the two lattices that contain the distribution functions. The 'outgoing' distribution functions become the 'incoming' ones of the next time-step. Therefore, the two pointers `df_in` and `df_out`, which pointed to the 'incoming' and 'outgoing' arrays, are changed (this is done by function `switchLattice`, called at the end of function `nextTimestep` in listing 3.6). This routine runs until the last time-step is completed and the simulation is finished.

```
1  void stream(int _x, int _y, double *_df_xy){
2    int i_count , x_new, y_new;
3
4    for (i_count = 0; i_count < 9; i_count++){
5      x_new = _x + cx[i_count];
6      y_new = _y + cy[i_count];
7
8      if (x_new < 0){
9        x_new = xSize - 1;
10     }
11     else if (x_new >= xSize){
12       x_new = 0;
13     }
14
15     if (y_new < 0){
16       y_new = ySize - 1;
17     }
18     else if (y_new >= ySize){
19       y_new = 0;
20     }
21
22     df_out[ind2inx(x_new, y_new, i_count)] = _df_xy[i_count];
23   }
24 }
```

LISTING 3.4: Streaming of the post-collision distributions to their neighbors

3.3. Initial Conditions

One of the most important tasks when setting up a simulation in computational fluid dynamics (CFD) is the proper choice of the initial conditions. It is given in terms of the macroscopic variables, which can be tricky to achieve, especially for more complex flows where no direct analytical solution is available. Most CFD solvers work directly with those variables, so that one only has to assign the variables with the initial conditions, but for lattice Boltzmann models initial values for the distribution functions f_i have to be found, and typically they are not given.
While the macroscopic variables can easily be computed as the moments of the distribution functions (see equation (2.14)), the inverse mapping is not that straightforward. Skordos [1993] discusses the initialization for the lattice Boltzmann and presents a method to determine the distribution functions f_i from the macroscopic variables and some of their derivatives.
It is quite common to use the equilibrium distributions (equation (2.13)) for approximating the distribution functions, especially when interested only in the long-term behavior of the system. The simulations for this thesis are initialized in this way.

3.4. Boundary Conditions

In the interior of the fluid domain, the evolution of the distribution functions is well defined by equations (2.12) - (2.14) once proper initial conditions have been found (see section 3.3). Due to the lack of neighboring nodes, problems arise at the boundaries. The distribution functions at the margins, with an associated discrete velocity pointing inside the fluid domain, remain undefined, and this blows up the whole simulation. Therefore it is of great importance to find good approximations for these values. The approximation must fit the physical requirements of the problem and guarantee the stability of the simulation. There are different kinds of boundary conditions (BC) that are used in the implementations for this thesis. They are described shortly in the following.

3.4.1. Periodic

The periodic BC is one of the simplest. During the streaming step it is determined whether the destination node of the distribution function is inside or outside the fluid domain. If it is outside, the distribution function is copied to the node on the opposite side of the domain. For a 2D domain with one periodic dimension, one can imagine the domain as being mapped onto a cylinder, as illustrated in Fig. 3.3. This is quite

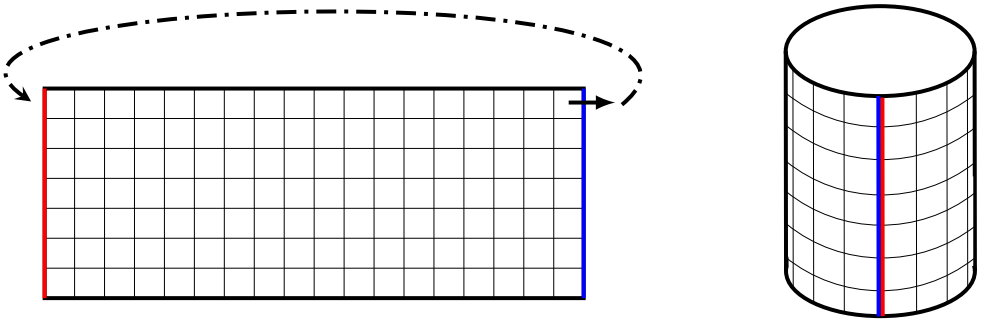

FIGURE 3.3.: Sketch of the periodic boundary condition.

an elegant way because no approximations have to be done, but it is not very realistic physically because it describes an infinite fluid domain.
This BC is implemented by default into the function `stream` (see listing 3.4) because it prevents the copying of the distribution functions outside the domain, which would result in a segmentation fault. Should there be a need to apply other BCs, this can be done once the streaming steps of the whole domain are complete.
This BC is used in x-direction for the Rayleigh-Bénard convection in section 4.2.

3.4.2. Bounce-Back

The bounce-back BC is used for rigid (no-slip) walls and adiabatic walls (in thermal flows for the thermal component). It is derived in a phenomenological way: particles that hit a wall are reflected back along the direction they came from (see Fig. 3.4). At these boundaries no collision process is effected, as all incoming distribution functions are simply flipped to the opposite direction. At a wall the function `applyBounceBackBoundaries` (see listing 3.5) is called instead of the function `collide`. The variable `opp` is an array containing the opposite indices.
Despite its simplicity, this formulation achieves astonishingly good results, which is why it is so popular and is used in all implementations of this thesis. Ziegler [1993] showed that this BC is second order accurate only when the wall is thought to be halfway between the boundary node and the last fluid node.

```
1 void applyBounceBackBoundaries(int _x, int _y, double *_df_xy){
2   int i_count;
3
4   for (i_count = 0; i_count < 9; i_count++){
5     _df_xy[i_count] = df_in[ind2inx(_x, _y, opp[i_count])];
6   }
7 }
```

LISTING 3.5: Implementation of the bounce-back BC.

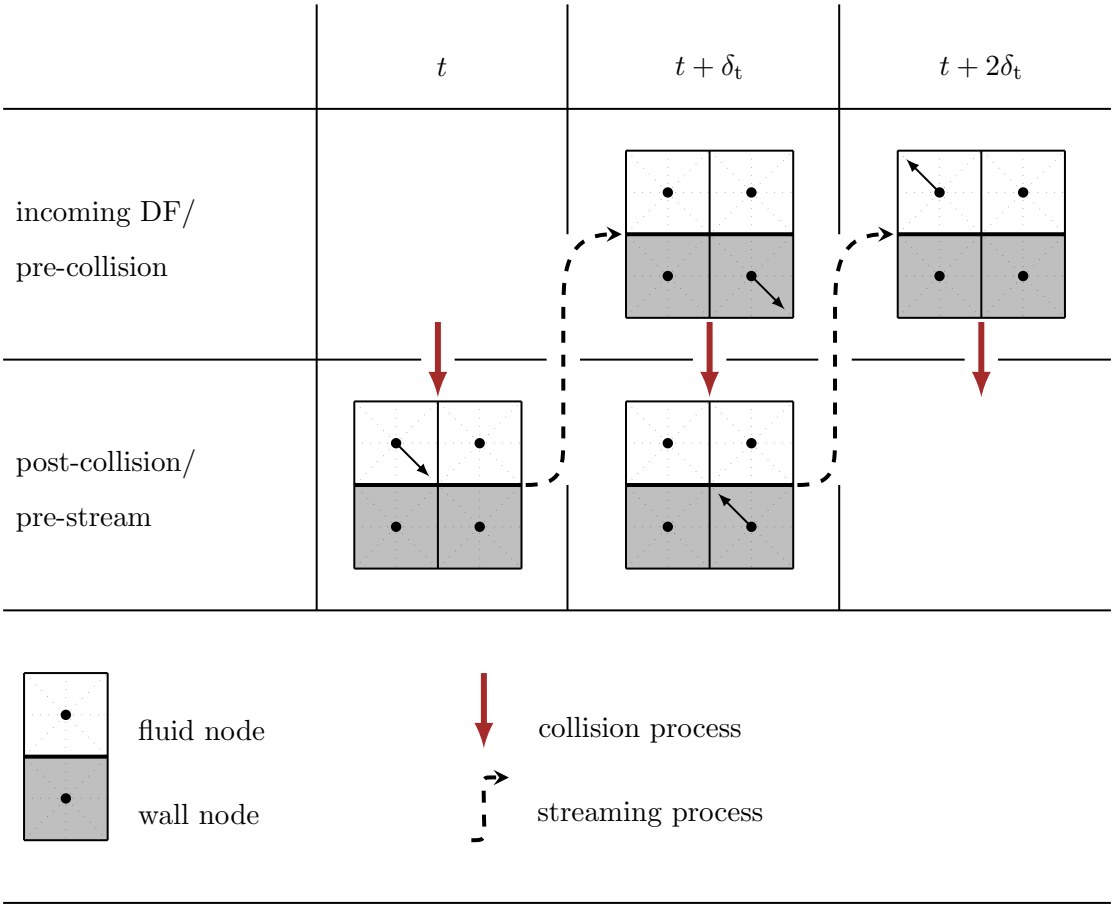

FIGURE 3.4.: Sketch of the bounce-back boundary condition. Instead of the collision process the directions of all distribution functions are inversed at a wall node.

3.4.3. Constant value

For the D2Q9 lattice there are three unknown distribution functions at a boundary node, while the other six are known. To solve equations (2.14a) and (2.14b) three more constraints are needed. By prescribing macroscopic variables as density ρ or velocity $\mathbf{u}$ at the boundary, one or two constraints are added, respectively, yet the system is still not completely described. As a further constraint, Zou & He [1997] assume the non-equilibrium parts (basically $f_i - f_i^{\mathrm{eq}}$) of the incoming and outgoing distribution functions to be equal. At a bottom boundary this can be formally written as:

$$f_2 - f_2^{\mathrm{eq}} = f_4 - f_4^{\mathrm{eq}} \tag{3.12}$$

With this constraint the system can be solved for a given velocity but not for known density. In such a case it is quite common to set the flow to be perpendicular to the boundary by setting a velocity component zero. This BC is only mentioned because it is relatively popular (due to its simplicity); however, it is not used in this form for

the implementations for this thesis. This boundary problem only occurs as a fixed temperature condition, which is even simpler to solve since only a D2Q5 lattice is used (see section 2.2.3) for the thermal component and thus only one distribution function is unknown. For a bottom boundary with temperature T_b from equation (2.26) the unknown distribution function is derived:

$$T_2 = T_b - T_0 - T_1 - T_3 - T_4 \tag{3.13}$$

These BCs are applied after all distribution functions have been updated (i.e. all streaming processes have been completed).

For the prescription of a constant temperature at a wall, Wang et al. [2013] present another method that can be used for the MRT model (see section 2.2.4). It is called 'anti-bounce-back' BC because it uses a principle similar to the bounce-back scheme.

$$g_i = -g_{\bar{i}} + \frac{4+a}{10} T_b \tag{3.14}$$

Here $g_{\bar{i}}$ is the distribution function associated with the discrete velocity $\mathbf{c}_{\bar{i}} = -\mathbf{c}_i$ and a is a model parameter (discussed in section 2.2.4).

3.4.4. Non-equilibrium extrapolation

S. Chen et al. [1996] propose a new BC; they argue that "the lattice Boltzmann method can (...) be viewed as a special finite difference scheme for the kinetic equation of the discrete-velocity distribution function", which can then be computed by a second order extrapolation of the known distribution functions inside the fluid domain. Z.-L. Guo et al. [2002] propose a similar BC by which only the non-equilibrium part of the distribution function is extrapolated. It is demonstrated that this approach is second order accurate and even more stable numerically (because only a first order extrapolation is used) than the BC of S. Chen et al. [1996]. This non-equilibrium extrapolation scheme is used for the inlet (constant velocity) and outlet (constant pressure) BC in the setup of the flow past a cylinder (see section 4.1).
It is assumed that the unknown distribution functions at the boundary $f_i(\mathbf{x}_b, t)$ can be divided up into an equilibrium and a non-equilibrium part:

$$f_i(\mathbf{x}_b, t) = f_i^{\mathrm{eq}}(\mathbf{x}_b, t) + f_i^{\mathrm{neq}}(\mathbf{x}_b, t) \tag{3.15}$$

The non-equilibrium part then is approximated by the non-equilibrium part of the corresponding distribution function of the neighboring fluid node $f_i(\mathbf{x}_f, t)$, with $\mathbf{x}_f = \mathbf{x}_b + \delta_t \mathbf{c}_i$.

$$f_i^{\text{neq}}(\mathbf{x}_b, t) \approx f_i^{\text{neq}}(\mathbf{x}_f, t) = f_i(\mathbf{x}_f, t) - f_i^{\text{eq}}(\mathbf{x}_f, t) \tag{3.16}$$

To compute the equilibrium distribution at the boundary, density ρ_b (or pressure p_b) and velocity $\mathbf{u}_b$ are needed, but depending on the BC only one variable is known. The unknown variable is approximated by using its value from the node at $\mathbf{x}_f$.

$$f_i^{\text{eq}}(\mathbf{x}_b, t) = f_i^{\text{eq}}(\rho_b, \mathbf{u}_f) \tag{3.17a}$$

$$f_i^{\text{eq}}(\mathbf{x}_b, t) = f_i^{\text{eq}}(\rho_f, \mathbf{u}_b) \tag{3.17b}$$

Equation (3.17a) is used when the density is known (analog for known pressure) and (3.17b) is used when the velocity at the boundary is known.

3.5. Parallelization

In recent years computing systems that allow for the parallel use of multiple CPU- and GPU-cores have become more widely accessible. In theory these machines can speedup the execution of an optimized code proportional to the number of cores used (although good speedup can be difficult to achieve in practice). This technique is capable of greatly increasing the efficiency of the lattice Boltzmann methods.

The lattice Boltzmann method as described in the previous chapters only acts locally at each lattice node. For updating the distribution functions and macroscopic variables, only the state of the current lattice node is used.[1] Due to this locality the order of processing the nodes is irrelevant and we can divide the fluid into an arbitrary number of sub-domains that are handled separately. This provides opportunity to optimize the algorithm for the execution on multi-core processors, as mentioned above. This is done by creating threads that can be executed in parallel, and by assigning one thread to each individual sub-domain.

In the implementations for this thesis, the handling of threads is included in the function `nextTimestep` by using the *Boost.Thread* library (see listing 3.6). The domain is divided into a certain number (defined by `numTreads`) of horizontal stripes that range from `y_min` to `y_max`. For every sub-domain a thread is created that executes the function `collideAndStream` for the given range. After having created all threads the algorithm pauses until all threads are done.

[1] Note, as outlined in section 3.4: Non-local boundary conditions can only be applied once all the update processes of the whole domain are completed.

```
1  void nextTimestep(){
2    int thread_count, y_min, y_max;
3    boost::thread_group threads;
4
5    for (thread_count = 0; thread_count < numThreads; thread_count++){
6      y_min = (int) ((double)(thread_count * ySize) / ((double) numThreads) );
7      y_max = (int) ((double)((thread_count+1) * ySize) / ((double) numThreads) );
8
9      threads.create_thread(boost::bind(&collideAndStream, this, y_min, y_max));
10   }
11
12   threads.join_all();
13
14   switchLattice();
15
16 }
```

LISTING 3.6: The function `nextTimestep` creates a group of threads that update the fluid domain in parallel.

3.6. Used libraries

The implementation of the numerical experiments in this thesis is done in *C++*. In addition to the standard C++ libraries, some other libraries are used. These are presented briefly below.

Libconfig

To set up a simulation, several parameters are needed that provide information about the problem to be studied. These parameters could also be hard-coded but that would make it necessary to recompile the entire source code every time the setup changes. It is easier to create simple text files that contain all the needed information and are passed to the program and read prior to the initialization of the model. The library *libconfig* provides a good C++ interface to evaluate such configuration files. These files have to follow a certain syntax. Listing 3.7 shows a short example of a configuration file. Listing 3.8 shows the C++ code to read it.
Libconfig and a more detailed documentation are available online:

`http://www.hyperrealm.com/libconfig/`

```
1 #Set max number of timesteps
2 Ntime = 50000;
3
4 #Set grid size
5 Size = {
6   Nx = 200;
7   Ny = 102;
8 };
```

LISTING 3.7: Example of a configuration file.

```
1 int Ntime, xSize, ySize;
2
3 libconfig::Config ExampleConfig;  //create object of type libconfig::Config
4
5 ExampleConfig.readFile('ExampleConfig.cfg');  //read file 'ExampleConfig.cfg'
6
7 Ntime = ExampleConfig.lookup("Ntime");     //read number of timesteps
8 xSize = ExampleConfig.lookup("Size.Nx");   //read grid size
9 ySize = ExampleConfig.lookup("Size.Ny");
```

LISTING 3.8: Example of reading a configuration file using *libconfig*.

Boost

Boost is a large free collection of reviewed C++ libraries. Three libraries from this project are used for the implementations for this thesis:

- The *Boost.filesystem* library, which provides, among many other features, an interface for manipulating file paths.
- The *Boost.thread* library, which is a handy tool for easily creating and managing threads to run parts of the code in parallel (more details in section 3.5). To perform this task, it was necessary to use
- the *Boost.bind* library, which serves to create an object from a function and its arguments, which can then be passed and executed in a thread.

A detailed documentation and the whole collection of libraries are available online:

`http://www.boost.org/`

netCDF

The Network Common Data Form (*netCDF*) developed by Unidata is a data format widely used in geo-sciences. It provides interfaces for various programming languages to read and write array-oriented data. It is easy to define dimensions and bind them to variables to save the results. NetCDF files are self-descriptive, which is why the data can be accessed by a long list of compatible software. A more detailed description of how the files are initialized and how the data is written into them using the C++ interface is given with the help of short examples in appendix C.

More information, documentations and the libraries are available online:

`http://www.unidata.ucar.edu/software/netcdf/`

4. Numerical experiments

4.1. Flow past a cylinder

In this section the flow about a circular cylinder placed inside a channel is analyzed. The channel is confined by two parallel walls, hence the velocity profile of the undisturbed flow is parabolic.

In the unbound case, where the cylinder is placed inside a flow without confinements, the flow only depends on the Reynolds number. If it is below approximately 4 the flow does not separate from the cylinder surface and is symmetrical up- and downstream the cylinder [Tritton, 1988], but this behavior is not studied here. As the Reynolds number is increased, the flow separates from the surface and a pair of counter-rotating vortices develop and stay attached to the cylinder on its downstream side. The length of this wake increases with Re until it exceeds a critical value and the symmetrical wake becomes unsteady. This happens when the cylinder-based Reynolds number is between 40 and 50 [Tritton, 1988; Zovatto & Pedrizzetti, 2001]. The flow in the cylinder's wake is characterized by two opposite-signed vortices that alternately separate from the cylinder and form a *von Kármán vortex street*. If the Reynolds number is increased further there is a transition to three-dimensional flow ($Re \approx 180$) [Williamson, 1996].
Unlike in the unbound case, the behavior of the flow around a cylinder placed between two walls does depend not only on the Reynolds number but also on the distance between the cylinder and the walls. For a constant diameter of the cylinder (i.e. constant blocking ratio) the dependence of the resulting flow is studied for several Reynolds numbers and distances to one wall. This study focuses on the transition from the steady flow, where two vortices are attached to the cylinder, to the unsteady flow, where the vortices shed from the cylinder. Besides the period lengths of this vortex shedding, some features of the flow are discussed. The results are compared to the findings of Zovatto & Pedrizzetti [2001] and J.-H. Chen et al. [1995], who also studied the two dimensional flow about a cylinder in a closed channel.

4.1.1. Mathematical formulation

The description of the setup is given in terms of dimensionless variables due to the independence of the flow on the choice of physical scales (compare section 3.1). The system is fully characterized by the Reynolds number Re, the diameter of the cylinder relative to the channel height, and the position of the cylinder relative to the walls. The problem is solved in two dimensions, which corresponds to a physical setup of a flow about an infinitely long cylinder; the channel extends along the x-direction and is confined in y-direction by two walls that are placed symmetrically around $y = 0$ with distance H. All length scales are measured in terms of the channel height H, that is why $H = 1$ in the dimensionless formulation (H is also called the characteristic length scale l_0 of the system). The direction of the undisturbed flow (far away from the cylinder) is along the x-axis, while the axis of the cylinder is perpendicular to it, parallel to the $z-axis$. Since only a slice in the (x, y)-plane is observed, the cylinder is represented by a circle of diameter d, which is also given relative to the channel height H and therefore $0 \leq d < 1$. It is centered around $x = 0$, while its position in y-direction depends on the size of the gap Δ to one wall (compare Fig. 4.1). In the following this gap is characterized by the gap parameter $\gamma = \frac{\Delta}{d}$, which is the minimum distance of the cylinder's mantle to one wall in terms of the cylinder diameter. It ranges from 0, when the cylinder touches one wall, to $\gamma = \frac{1-d}{2d}$, when the cylinder is centered in the channel. As the characteristic speed, the mean velocity in the channel of the undisturbed flow is taken:

$$u_0 = \bar{u} = \frac{1}{H} \int_{-H/2}^{H/2} u_x(y) dy = 1, \tag{4.1}$$

with $u_x(y) = u_{\max} \left[1 - (\frac{2}{H} y)^2\right]$. This parabolic, so-called Poiseuille profile, is the solution of the undisturbed flow in a channel. The maximum velocity in case of the mean velocity being unity is $u_{\max} = \frac{3}{2}$. Having defined the characteristic velocity and length scales, one obtains the characteristic time scale $t_0 = \frac{l_0}{u_0} = \frac{H}{\bar{u}} = 1$.
Once all the characteristic scales are defined, the Reynolds number is given by

$$Re = \frac{\bar{u} H}{\nu}. \tag{4.2}$$

In the unbound case, in the absence of the walls (and thus constant velocity $\mathbf{u}_\infty$ far away from the obstacle) one usually uses the cylinder-bound Reynolds number $Re_{\text{cyl}} = \frac{u_\infty d}{\nu}$ to describe the flow. To better compare the unbound case with the bounded one, which is studied here, one can convert Re to Re_{cyl}:

$$Re_{\text{cyl}} = Re \; v_{\text{cyl}} d, \tag{4.3}$$

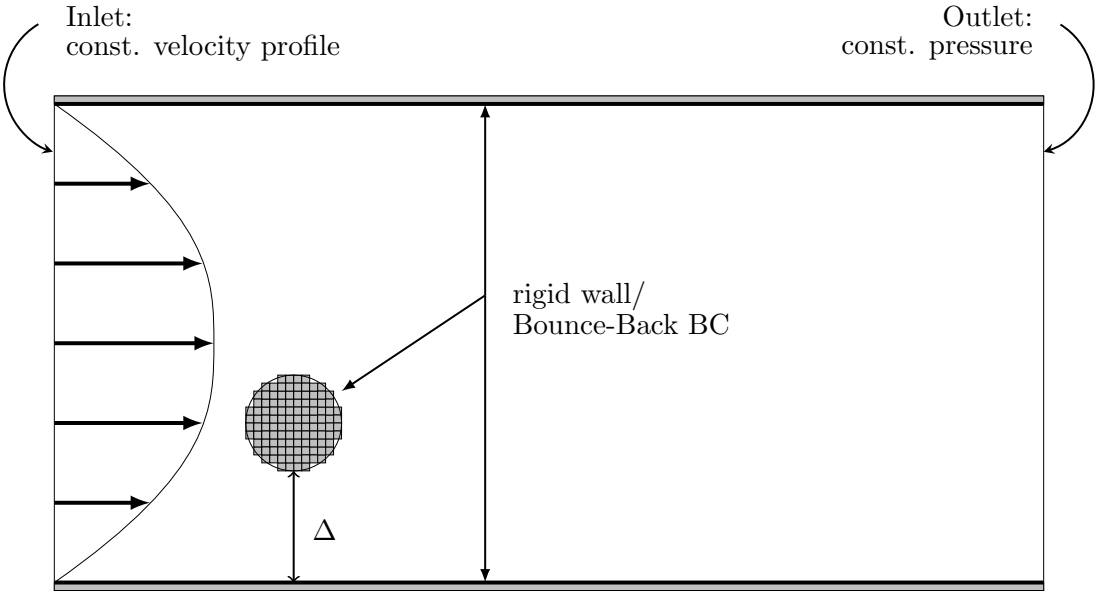

FIGURE 4.1.: Flow past a cylinder: Sketch of the setup. A cylinder of diameter d is placed inside a channel with distance Δ to one wall. The arrows on the left denote the parabolic Poiseuille velocity profile, applied at the inlet.

where v_{cyl} is the mean velocity in front of the cylinder

$$v_{\text{cyl}} = \frac{1}{d}\int_{\gamma d-\frac{1}{2}}^{\gamma d+d-\frac{1}{2}} u_x(y)\mathrm{d}y = -6d^2\gamma^2 + 6d(1-d)\gamma + d(3-2d). \tag{4.4}$$

For given Re, d and γ the system is completely described and its temporal evolution is given by the incompressible Navier-Stokes equations. As mentioned above, the dependence of the system on the diameter of the cylinder is not studied here. Instead the diameter is fixed, while the Reynolds number Re and the gap parameter γ are varied. Analogous to the studies of Zovatto & Pedrizzetti [2001], the diameter of the cylinder is set to $d = 0.2$, which gives a maximum gap parameter $\gamma = 2$ (in case of the centered cylinder).

4.1.2. Numerical setup

Now that the problem is formulated, a numerical model capable of simulating its physics has to be adapted. Since the system is described by the athermal incompressible Navier-Stokes equations, the iLBGK model (see section 2.2.2) is used. The size of the domain is already confined by the walls in y-direction, but still has to be restricted in x-direction for the numerical implementation. Especially downstream the cylinder, the domain needs to be sufficiently long to avoid interaction of the wake with the outlet boundary, and thus

alteration of the flow. This choice is adopted from Zovatto & Pedrizzetti [2001]. The computational domain is restricted to 3 upstream and 8 downstream the center of the cylinder (measured in dimensionless units). Therefore the total length of the numerical domain in x-direction is 55 cylinder diameters.
The discretization is described basically in section 3.1. The (dimensionless) viscosity is given by the Reynolds number. Temporal and spatial step sizes of the lattice are found by prescribing the Mach-number and by increasing the spatial resolution until the collision parameter τ is above a certain threshold (unless stated otherwise, $Ma = 0.1$ and $\tau \geq 0.55$ were used).
The implementation of solid vertical or horizontal walls can easily be done using the bounce-back boundary condition (see section 3.4.2). A curved wall can be approximated either by a combination of vertical and horizontal bounce-back boundaries or, if necessary, by a more improved treatment (a nice overview of several approaches is given by Z. Guo & Shu [2013]). For the sake of simplicity, the former approach is applied for the implementation of the cylinder. At the inlet and outlet of the channel, the non-equilibrium extrapolation schemes (section 3.4.4) for constant velocity and pressure are used: at the inlet, the Poiseuille velocity profile (as given above) is prescribed, and at the outlet the pressure is set to be zero (see Fig. 4.1).
As initial conditions, the equilibrium distributions computed from the velocity profile and the pressure, that is set everywhere to zero, are used (compare section 3.3).

4.1.3. Results & Discussion

The initialization of the simulations is far from equilibrium. It is as if the cylinder had been dropped into the flow instantaneously. This results in large perturbations in pressure, which spread across the whole domain starting from the cylinder. In a physical experiment those perturbations would disappear quickly, simply leaving the domain through the open end of the channel. In this numerical implementation, however, those waves are reflected by all boundaries and keep traveling through the domain for longer time. Depending on the Reynolds number and geometry of the system, but also on the temporal and spatial resolution of the lattice, those unphysical modes take some time to disappear or become negligible compared to the physical modes.
It should be noted here, that, for sake of clarity, some figures mentioned in this section can be found in appendix D.
To analyze the behavior of the flow, the vorticity at one point behind the cylinder is studied in more detail. Unless noted otherwise, this point is chosen one diameter away from the cylinder on its centerline: $x_\mathrm{p} = \frac{3}{2}d$, $y_\mathrm{p} = (\gamma + \frac{1}{2})d - \frac{1}{2}$. The vorticity is computed from the velocity field using finite differences. This time series is then decomposed into

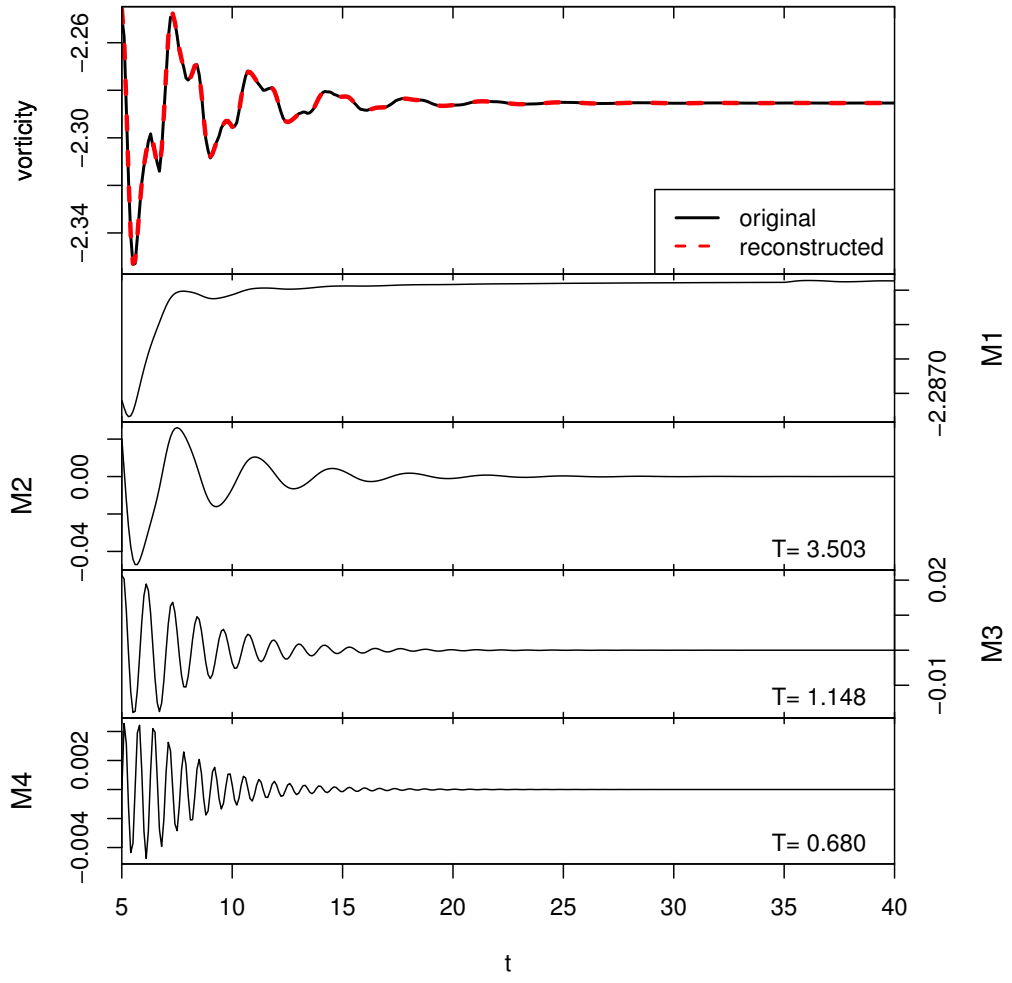

FIGURE 4.2.: In the top panel of this figure the vorticity at point $(x_\mathrm{p}, y_\mathrm{p})$ for $Re = 100$ and $\gamma = 0.25$ is plotted over time. The four other plots below show the results of the *SSA decomposition*. The time series was split into several modes, where M1 is the trend and M2-M4 are the first three most dominant oscillating modes. Their mean period length is written in the lower right corner of each panel. The sum of those first four modes is plotted in the top panel (reconstructed) to show that the original data set can be well described by these modes. It should be noted that the beginning of the simulation is omitted in the SSA because only the slowly decaying modes are of interest here.

different modes using the singular spectrum analysis[1] (SSA) for finding the dominant modes (compare Fig. 4.2). To get a complete overview of the spectrum and its temporal evolution, the time series is further analyzed by a wavelet transformation. This is done using the *R*-package *biwavelet*[2]. The (bias-corrected) wavelet power spectrum is shown in the lower panel of Fig. 4.3 for $Re = 100$ and $\gamma = 0.25$. The hatched area in these kinds of plots represents the so-called "cone of influence" (COI). This is an artifact of the method when applied to finite non-cyclic time series and a result of discontinuities

[1]The singular spectrum analysis is done with help of the *R*-package *RSSA*. The package and more information can be found online: `http://cran.r-project.org/web/packages/Rssa/index.html`

[2]The package and more detailed information can be found online: `http://cran.r-project.org/web/packages/biwavelet/index.html`

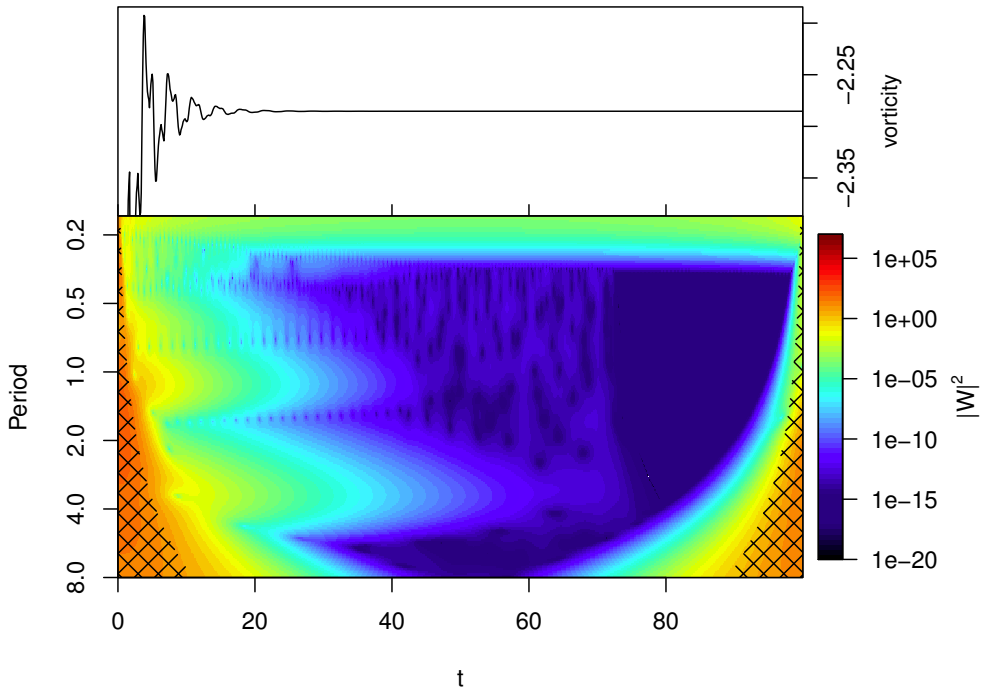

FIGURE 4.3.: Temporal evolution of the vorticity at (x_p, y_p) (top) and its wavelet power spectrum (bottom) for $Re = 100$ and $\gamma = 0.25$. The hatched areas mark the COI (see text for more information).

at the edges of the dataset. In the COI these edge effects are dominant and rapidly vanish outside the COI (see for example Torrence & Compo [1998]). However, the COI (and a little beyond) have to be ignored in the analysis the spectrum. The modes that were found with the SSA decomposition (Fig. 4.2) can easily be identified here. All modes decrease exponentially and thus the system is stable for this set of parameters. It becomes obvious when looking at several wavelet power spectra of different runs (see figures 4.4, 4.6-4.7 and D.5-D.9) that there are always modes of very similar period length that decrease slower than others (the two major modes have period lengths close to 3.4 and 1.15 and are often roughly multiples of each other). The closer the set of parameters is to a critical combination, the slower these modes decrease and shift to smaller period lengths, which might be a result of other modes getting more influence and overlapping with the former named ones. It is supposed that these modes originate from the model and its discretization. This gets clear when looking at the results for $Re = 1000$ and $\gamma = 0.25$. For the standard-used discretization with $Ma = 0.1$ and $\tau \geq 0.55$ these modes start increasing again very slowly after some time (see Fig. 4.4). In a second run for this set of parameters, the Mach number was set to 0.05 and $\tau \geq 0.52$. The results look very different from the results from the first run. The dominant modes have different period lengths and are decreasing slowly (compare Fig. 4.5). Despite this discrepancy between both model runs, this parameter combination is assumed to be stable because all other modes decrease rather quickly in both runs. It is supposed

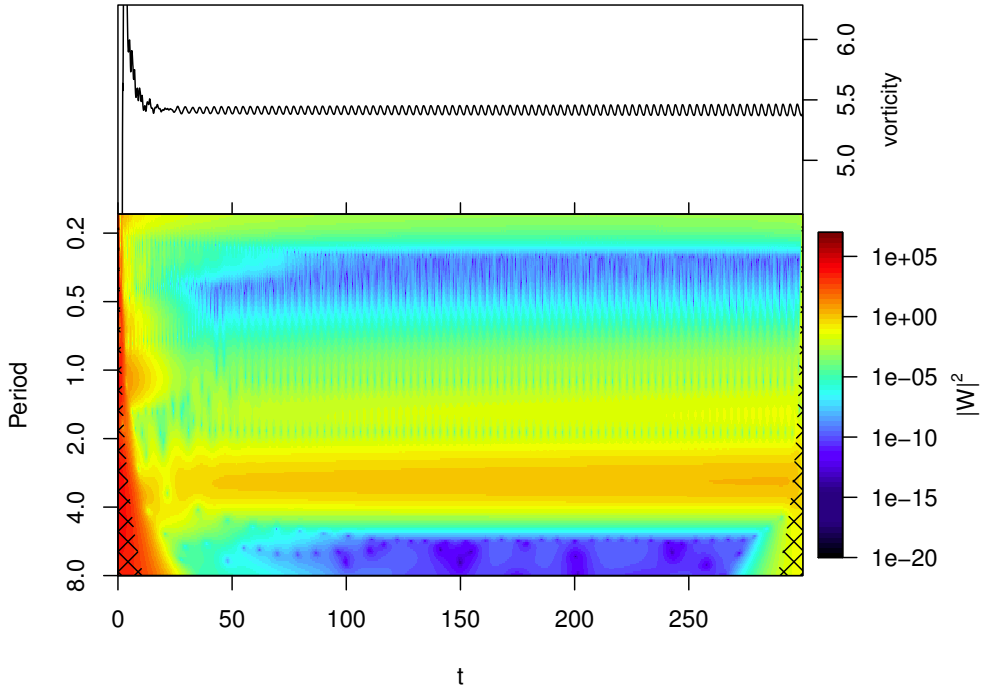

FIGURE 4.4.: Temporal evolution of the vorticity at $(x_\mathrm{p}, y_\mathrm{p})$ (top) and its wavelet power spectrum (bottom) for $Re = 1000$ and $\gamma = 0.25$. For discretization, $Ma = 0.1$ and $\tau \geq 0.55$ were used. The mode ~ 3.4 slowly increases. The hatched areas mark the COI (see text for more information).

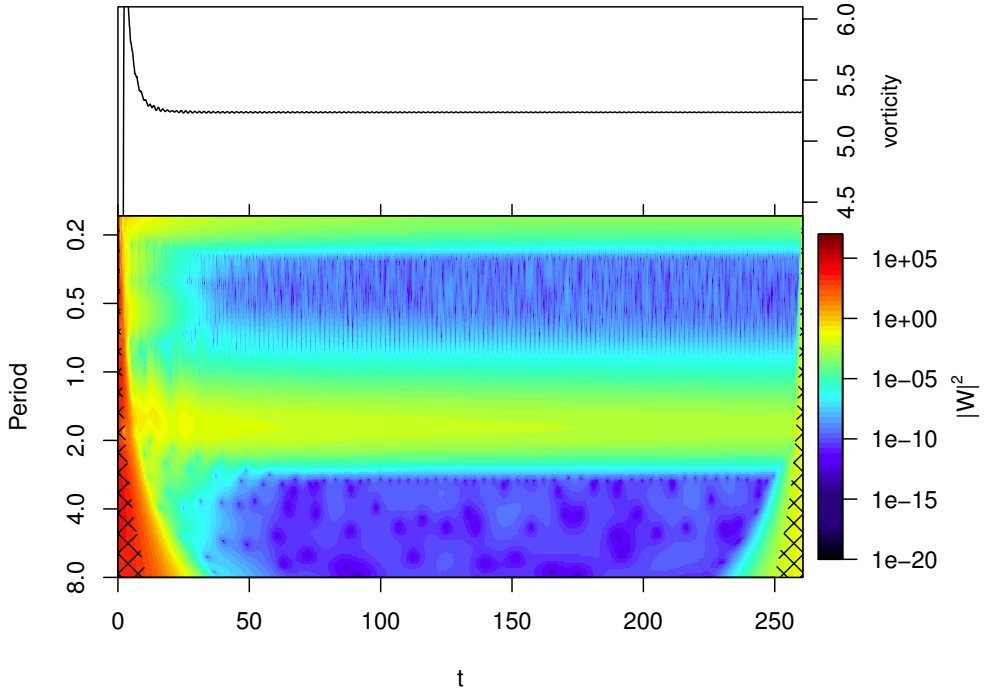

FIGURE 4.5.: Temporal evolution of the vorticity at $(x_\mathrm{p}, y_\mathrm{p})$ (top) and its wavelet power spectrum (bottom) for $Re = 1000$ and $\gamma = 0.25$. For discretization, $Ma = 0.05$ and $\tau \geq 0.52$ were used. The most persistent mode slowly decreases. The hatched areas mark the COI (see text for more information).

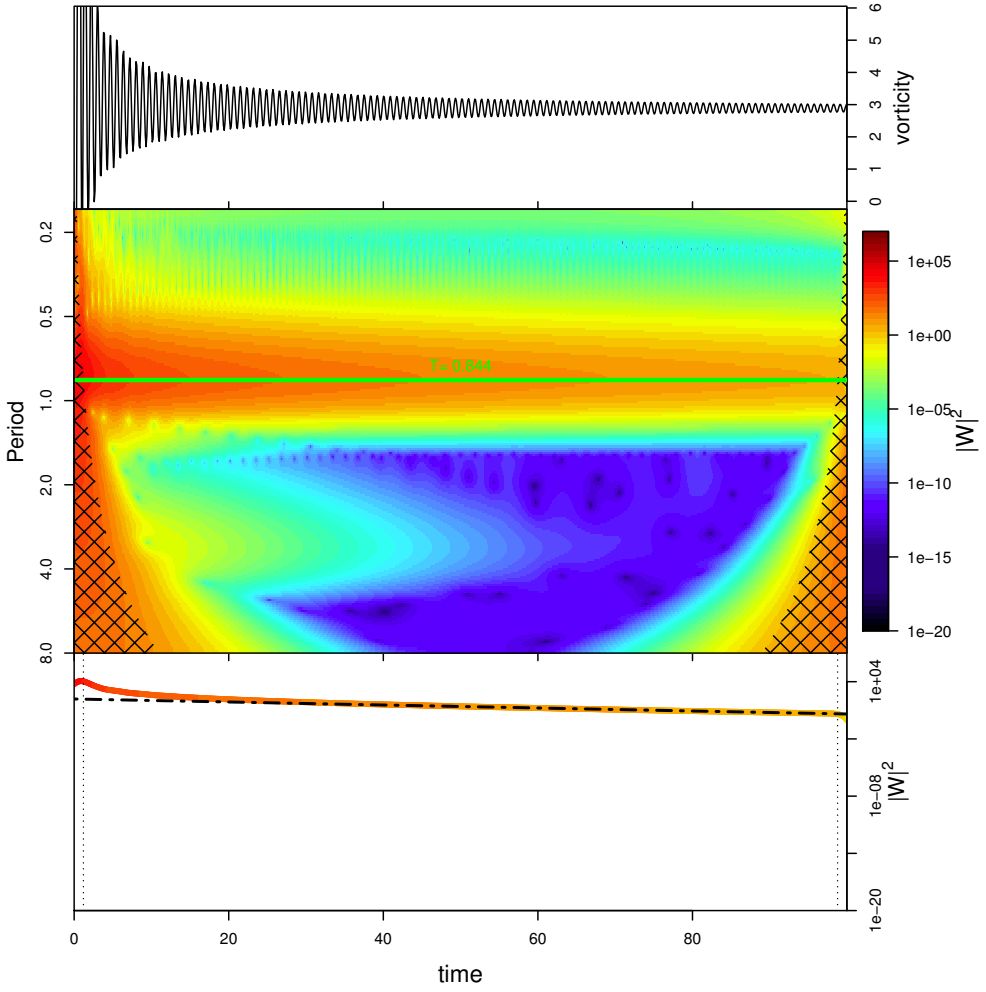

FIGURE 4.6.: Temporal evolution of the vorticity at (x_p, y_p) (top) and its wavelet power spectrum (middle) for $Re = 300$ and $\gamma = 1.25$. The amplitude of the most persistent mode ($T \approx 0.844$) decreases very slowly, hence this set of parameters is taken as being stable, but it is not sure whether the mode will ever completely vanish. The lower panel shows a transact of the power spectrum along the green line. The hatched areas (middle) and the thin dotted lines (bottom) mark the COI (see text for more information). To determine the exponential decay of the persistent mode, a line (thick dashed line in the lower panel) was fitted to the transect (in the interval $t \in [40, 95]$). Zovatto & Pedrizzetti [2001] characterize this set of parameters as unstable with a period $T = 0.84$.

that this issue can be solved by using a finer grid and smaller Mach number, but this is not explored in this study. Nevertheless, as can be seen later, the results are in good agreement with the results obtained by Zovatto & Pedrizzetti [2001]. As mentioned above, the closer the parameters are to a critical combination, the slower the decrease of the perturbations is. This can be seen very clearly in Fig. 4.6 (for $Re = 300, \gamma = 1.25$). Even after a quite long simulation time the system is still far from a steady state. For this set of parameters Zovatto & Pedrizzetti [2001] characterized the system as unstable with a period of $T = 0.84$. In this study the period of this persistent mode is also $T \approx 0.84$, but the amplitude is exponentially decreasing, therefore it is taken here as

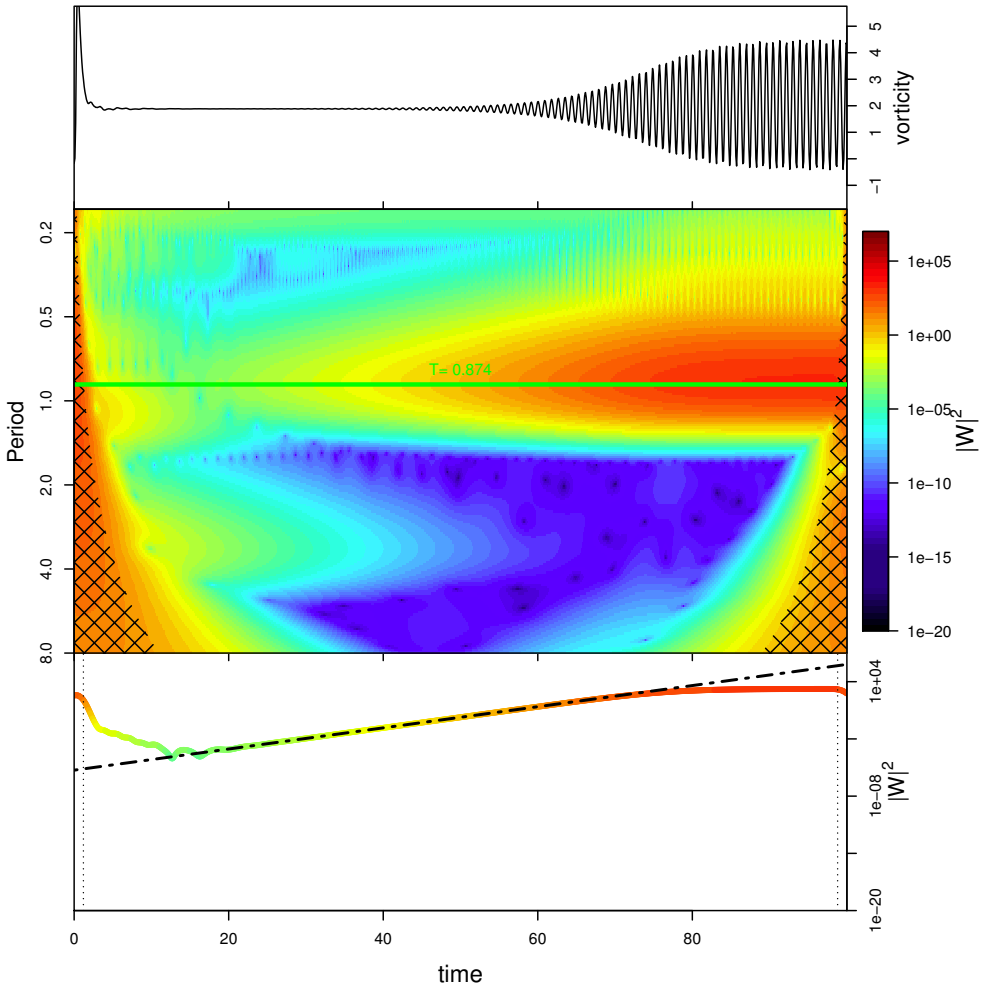

FIGURE 4.7.: Temporal evolution of the vorticity at $(x_\mathrm{p}, y_\mathrm{p})$ (top) and its wavelet power spectrum (middle) for $Re = 230$ and $\gamma = 2$. The combination of parameters is instable and vortex shedding occurs with a period length of $T = 0.874$. The transect of the power spectrum at this period (marked by the green line) is shown in the bottom panel. The hatched areas (middle) and the thin dotted lines (bottom) mark the COI (see text for more information). To determine the exponential growth rate k of the characteristic mode, a line (thick dashed line in the lower panel) is fitted to the transect (in the interval $t \in [25, 65]$).

stable, though it is not certain that the shedding will ever completely vanish. One would suggest that for an unstable set of parameters the modes that result from the vortex shedding increase exponentially. This would help in finding a good approximation of the critical point in parameter space by analyzing the growth rates. But in this study it can only be done for the symmetrical case with $\gamma = 2$ when the perturbations from the initialization act symmetrical and the system is in a numerically steady state. Were it not for a small perturbation that breaks this symmetry, the flow would remain stable. As an example, Fig. 4.7 shows the slow evolution of the system for $Re = 230$ and $\gamma = 2$. The plot in the lower panel shows a transact through the wavelet power spectrum $|W|^2$, which is proportional to the square of the amplitude of the particular mode. This means

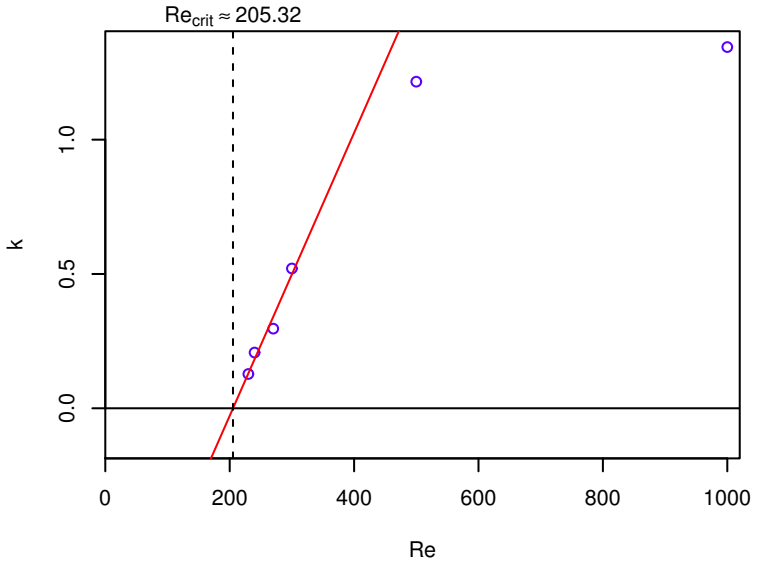

FIGURE 4.8.: Analysis of the growth rates k of the perturbations for $\gamma = 2$. Near the critical Reynolds number k is assumed to depend linearly on Re, thus only measurements for $Re < 400$ are considered. The critical value for Re, where $k = 0$, can be approximated as $Re_{\text{crit}} \approx 205$.

the exponential growth rate k of the unsteady mode is half the growth rate of $|W|^2$. This value can be found by linear regression of the logarithm of $|W|^2$, shown as the dashed line in the lower panel of Fig. 4.7. The growth rate k is determined for all unsteady cases for $\gamma = 2$ and plotted against the Reynolds number (see Fig. 4.8). Close to the critical Reynolds number Re_{crit} the dependence of k is supposed to be linear in Re, and thus the critical point with $k = 0$ can be found by linear regression (only the values of $Re < 400$ were recognized in the regression). The same case for a centered cylinder between two parallel walls was also studied by J.-H. Chen et al. [1995] who report a value of $Re_{\text{crit}} \approx 231$, while here a value of $Re_{\text{crit}} \approx 205$ is found.

When the cylinder is not placed in the center of the channel ($\gamma < 2$), the oscillatory flow sets in almost instantly due to the strong unsymmetrical perturbation (see Fig. D.7). For these cases the development of the vorticity can not be studied as above, and only the period of the main mode can be determined.

Table 4.1 contains the period lengths of all experiments from this study and also the results of Zovatto & Pedrizzetti [2001]. It can be seen that in most cases both results match very well. There are only two cases close to the critical point where the results are not the same in both studies. One is for $Re = 230$ and $\gamma = 2$, where Zovatto & Pedrizzetti [2001] report stable behavior and this study finds vortex shedding with a period of $T = 0.87$. The second case is for $Re = 300$ and $\gamma = 1.25$, which was already discussed earlier. The results shown in table 4.1 are also visualized in Fig. 4.9. For comparison with the results obtained by Zovatto & Pedrizzetti [2001] Fig. D.1 shows their plot. The blue symbols represent unstable and the black ones stable behavior. The

γ	2.00		1.75		1.25		0.75		0.50		0.25	
Re	CS	ZP	CS	ZP	CS	ZP	CS	ZP	CS	ZP	CS	ZP
100	∞	∞	∞	∞	∞	∞	∞	∞			∞	∞
200	∞	∞	∞	∞	∞	∞	∞	∞			∞	∞
230	0.87	∞	∞	∞								
240	0.86	0.85	∞	∞								
270	0.84	0.84	0.83	0.83	∞	∞	∞	∞			∞	∞
280		0.83		0.83	∞	∞		∞				∞
300	0.83	0.81	0.81	0.81	∞	0.84	∞	∞			∞	∞
400					0.80		∞	∞				
450					0.77		∞	∞				
500	0.73	0.73	0.72	0.74	0.76	0.76	0.84	0.86			∞	∞
600								0.84	∞			
650									∞			
700								0.82	∞			
800									0.87		∞	
1000	0.66	0.67	0.67	0.68	0.71	0.71	0.77	0.78	0.84		∞	∞

TABLE 4.1.: Period lengths for different combinations of Re and γ. The table shows the results from the current study (CS) next to the results of Zovatto & Pedrizzetti [2001] (ZP). It can be seen that the results are generally in pretty good agreement.

shaded area highlights the region between the last stable and the first unstable simulation in which the bifurcation occurs. The red dashed line is a very simple approximation of the critical parameter set. The critical Reynolds number is therefore assumed to be in the middle of two values for which the behavior changes, and fitted by a hyperbolic function. To show that this γ-dependence of Re_{crit} does not originate from the changes

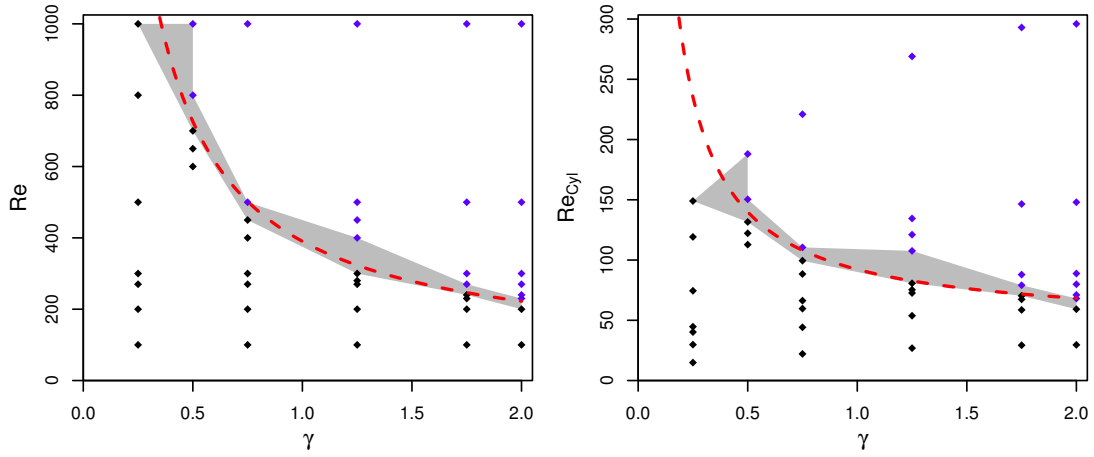

FIGURE 4.9.: Visualization of the results from table 4.1 to show the dependence of the critical Reynolds number on γ. On the left the channel-based Reynolds number Re, and on the right the cylinder-based Reynolds number Re_{cyl} are shown. Blue diamonds represent unsteady and black ones steady behavior. The areas shaded in gray denote the parameter space in which the bifurcation occurs. The critical set of parameters is approximated by the red dashed hyperbolic curves that are fitted through these areas.

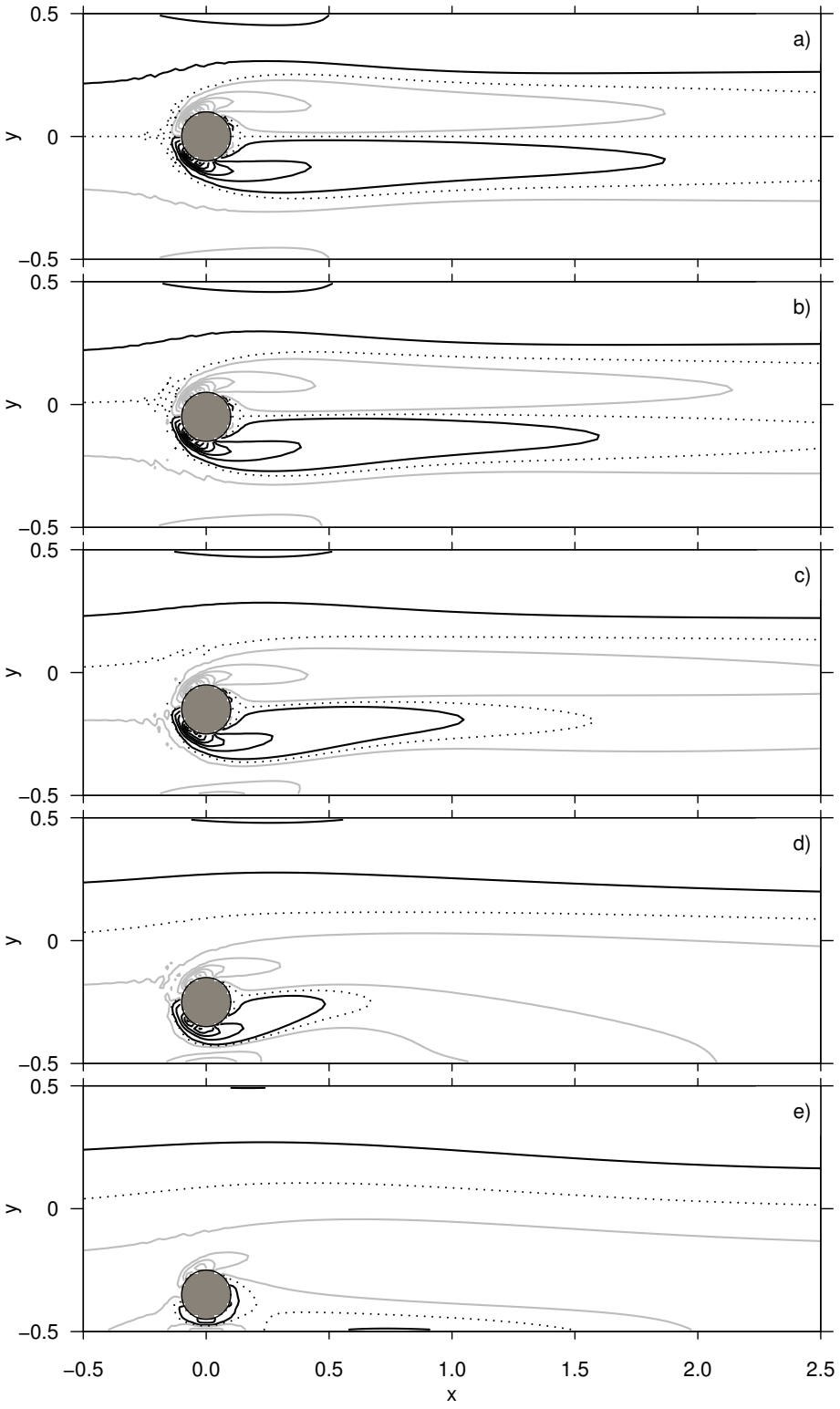

FIGURE 4.10.: Contours of vorticity for $Re = 200$, a) $\gamma = 2$, b) $\gamma = 1.75$, c) $\gamma = 1.25$, d) $\gamma = 0.75$, e) $\gamma = 0.25$. The dotted line stands for zero vorticity, while the black and gray contours denote positive and negative vorticity, respectively, starting at ± 2.5 with ± 10 spacing between successive lines.

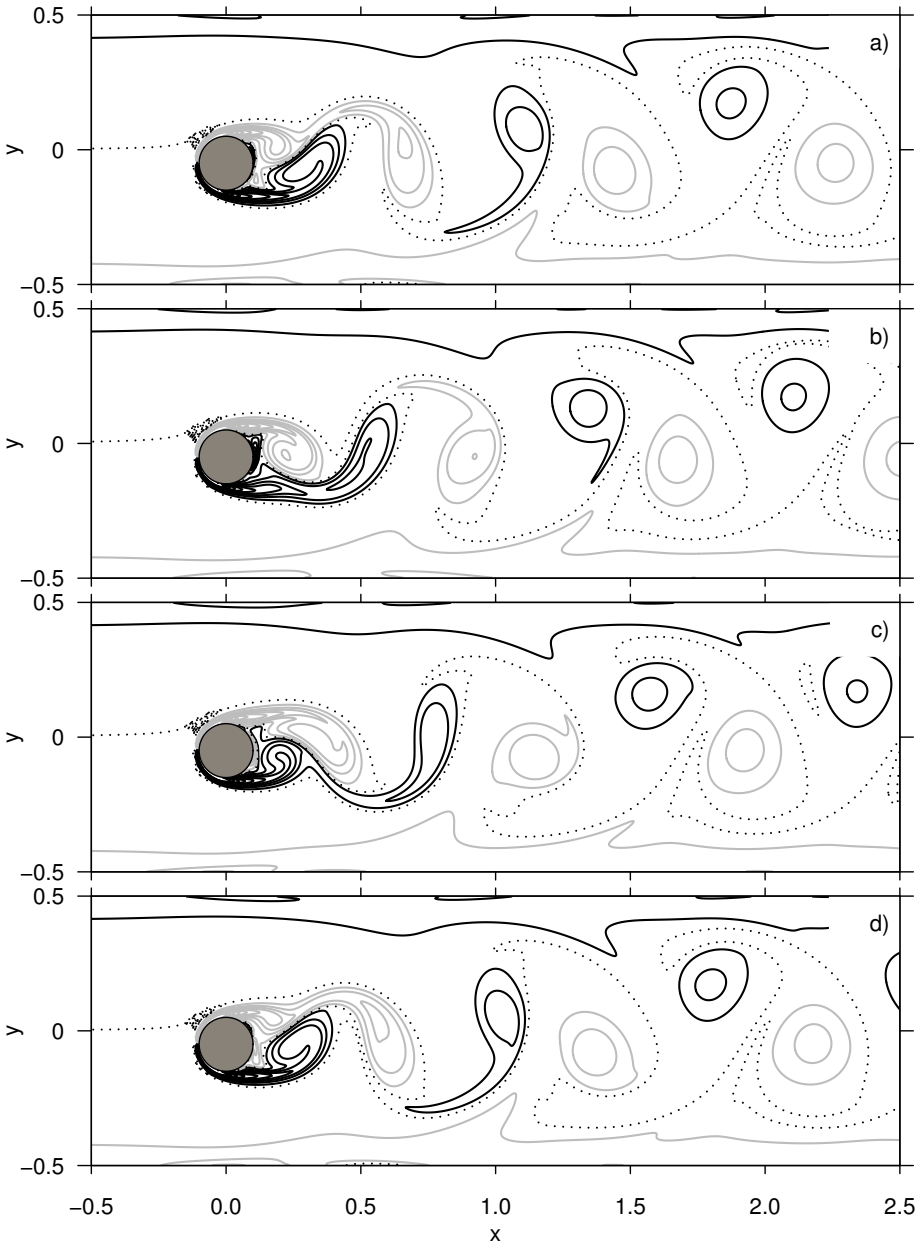

FIGURE 4.11.: $Re = 1000$, $\gamma = 1.75$, vorticity over one period ($T = 0.67$) (a-d) with step size $\Delta t = 0.2$ between successive plots. The dotted line stands for zero vorticity, while the black and gray contours denote positive and negative vorticity, respectively, starting at ± 5 with ± 10 spacing between successive lines.

in the velocity profile in the right plot, the cylinder based Reynolds number Re_{cyl} is also plotted against the gap parameter γ.

In the following, some features of the flow are discussed with the help of some snapshots of the vorticity, which is indicated by contour lines. The contour level of zero vorticity is depicted by a dotted line, while the black and gray lines denote positive (anti-clockwise) and negative (clockwise) vorticity, respectively. For better comparison of the results,

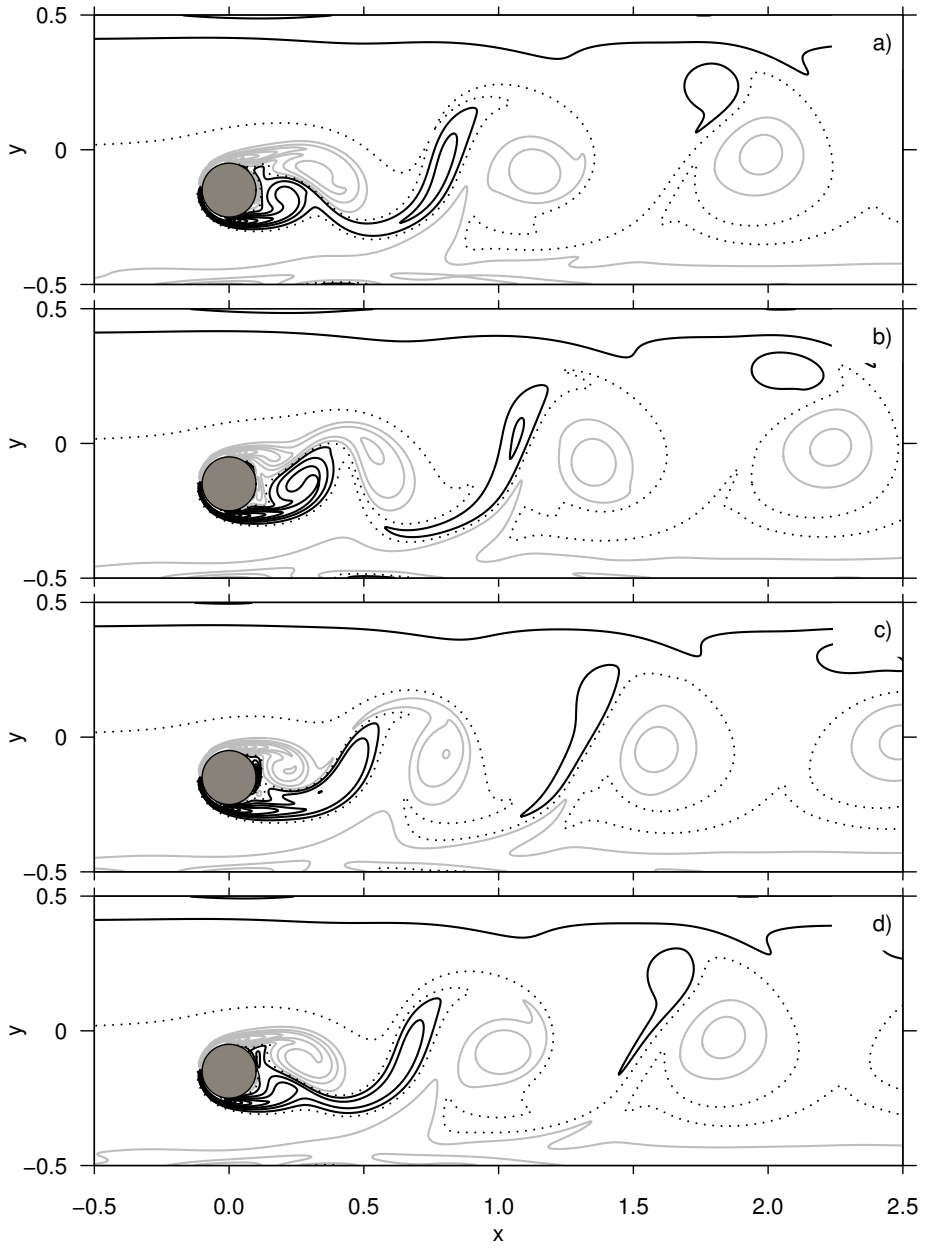

FIGURE 4.12.: $Re = 1000$, $\gamma = 1.25$, vorticity over one period ($T = 0.71$) (a-d) with step size $\Delta t = 0.2$ between successive plots. The values of the contours are the same as in Fig. 4.11.

the style of the figures and the values of the contour levels are borrowed from Zovatto & Pedrizzetti [2001] (some figures of that paper are also shown in appendix D). The contour levels start at ± 5 and are spaced ± 10, except of in Fig. 4.10, where the contours start at ± 2.5 with the same spacing as in the other plots[3]. Fig. 4.10 shows the dependence of the steady flow (in this example $Re = 200$) from the gap parameter γ. When the

[3]It is supposed that for Fig. 3 in Zovatto & Pedrizzetti [2001] these different values for the contour levels were used, though it was not separately noted (compare figures 4.10 and D.2).

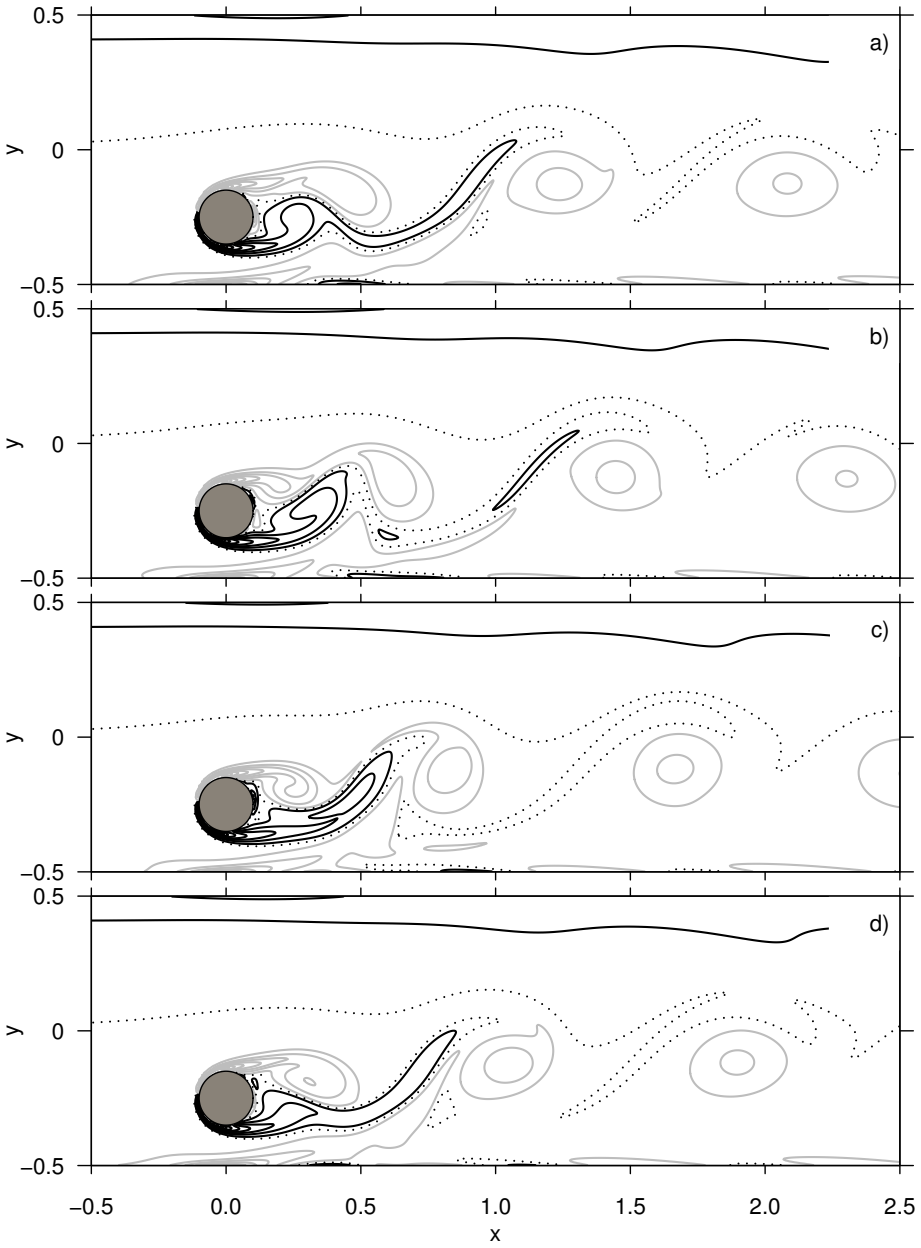

FIGURE 4.13.: $Re = 1000$, $\gamma = 0.75$, vorticity over one period ($T = 0.77$) (a-d) with step size $\Delta t = 0.2$ between successive plots. The values of the contours are the same as in Fig. 4.11.

cylinder is in the center of the channel ($\gamma = 2$), the flow is symmetrical and the wake consists of two counter-rotating vortices that are attached to the cylinder. Due to the acceleration of the flow around the cylinder, the vorticity in the boundary layer of the walls is strengthened in the vicinity of the cylinder. When the cylinder is shifted closer to one wall the flow becomes unsymmetrical. The vortex closer to the wall reduces its length, while the other elongates and soon dominates the wake. The wake also slightly bends away from the wall. When the cylinder is close enough to the wall, the vorticity

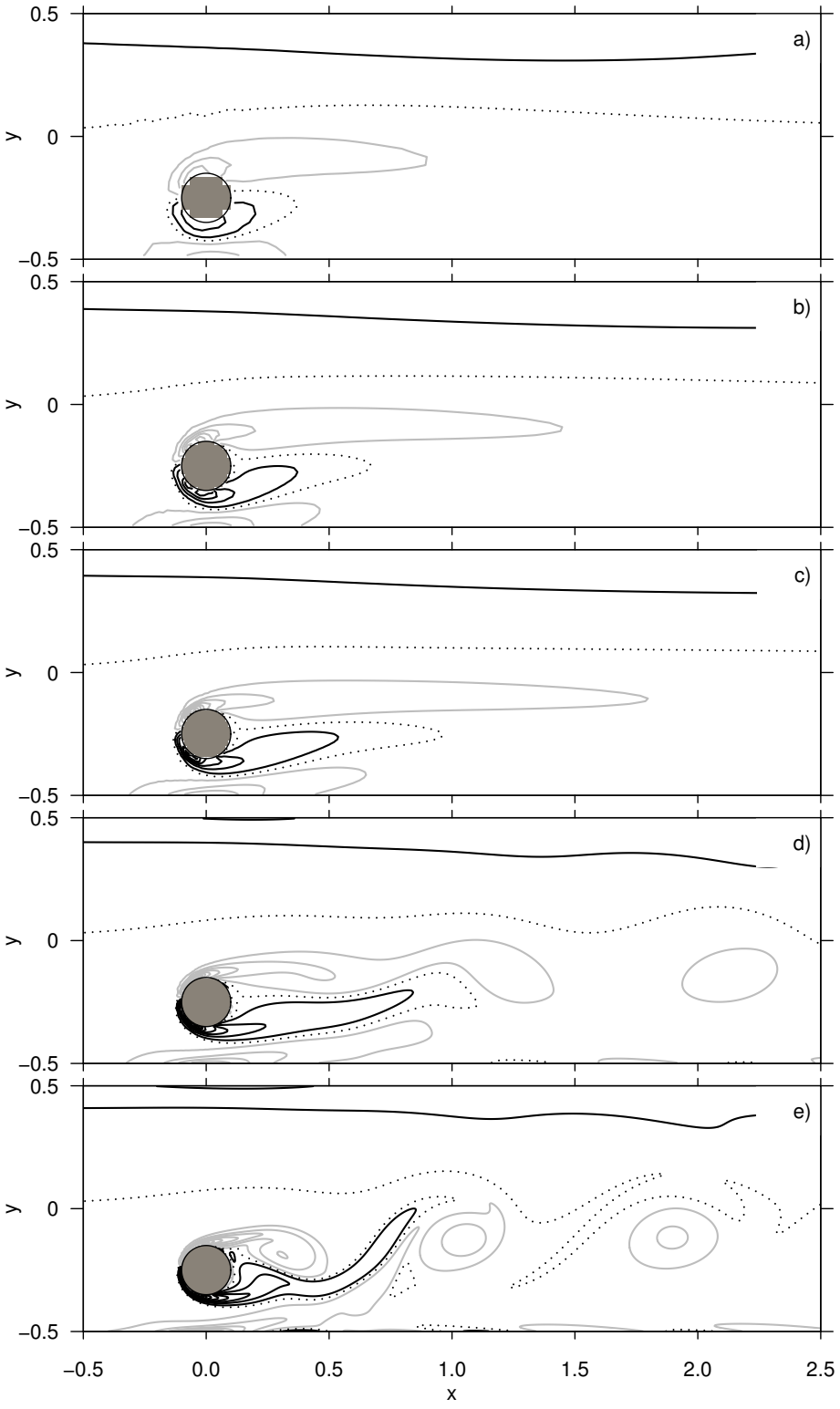

FIGURE 4.14.: Contours of the vorticity for $\gamma = 0.75$ and varying Reynolds number: a) $Re = 100$, b) $Re = 200$, c) $Re = 300$, d) $Re = 500$, e) $Re = 1000$. The values of the contours are the same as in Fig. 4.11.

at the side remote from the wall combines with the vorticity of the velocity profile. For $\gamma = 0.25$ the vortex close to the wall has almost disappeared and the flow resembles the solution of the flow around an object mounted on a surface.
The unsteady flow is discussed for $Re = 1000$. Fig. 4.11 shows a sequence of snapshots of the vorticity field over approximately one period for $\gamma = 1.75$. For this set of parameters, the wake has become unsteady and the two vortices alternately shed from the cylinder and travel downstream along the channel. This behavior is very similar to the von Kármán vortex street in the unbound case, with the difference that here the paths of both vortices cross and both change their positions. It can already be seen that the vortex from the side facing the wall is slightly weaker than its counterpart.
For further the wall is approached the more this becomes obvious. An analog time series for $\gamma = 1.25$ is shown in Fig. 4.12. Due to stronger interaction with the wall vorticity of opposite sign, the vortex becomes strongly stretched and thereby loses a lot of its strength.
For $\gamma = 0.75$ (see Fig. 4.13 and Fig. D.3 for the plot for this parameter-set in Zovatto & Pedrizzetti [2001]), the stretching process is so strong that the whole energy of the vortex is dissipated before it could take its place in the vortex street, which thus consists only of a single row of same-sign vortices.
For the same gap parameter γ the dependence on the Reynolds number Re is studied in Fig. 4.14. From the top panel to the bottom the Reynolds number is increased and the transition from a steady to an unsteady regime can be retraced. In the first three panels ($Re = 100,\ 200$ and 300) the wake is stable and remains stuck to the cylinder. It can be seen how the length of the wake grows with Re. At some length it becomes unstable and starts oscillating at the end, and vortices are shed (d). As Re is increased further, the oscillation grows and takes over more of the wake (e). A similar plot from Zovatto & Pedrizzetti [2001] for $\gamma = 0.75$ and varying Re is shown in the appendix (Fig. D.4).

4.2. Rayleigh-Bénard convection

The topic of this section is the free convection in the two-dimensional Rayleigh-Bénard configuration. The fluid is confined by two parallel walls at the top and at the bottom. Here, the evolution of the flow when heated at the lower and cooled at the upper boundary is studied. Hence, the system can not be described only by athermal equations as in the last experiment, but also the thermal component and the body force due to density variations have to be included.
The flow can be characterized by the Rayleigh number Ra, which depends on the temperature gradient, gravitational acceleration and some properties of the fluid (see eq.

(3.1d)). For low Ra, when the temperature gradient is not strong enough, the transport of thermal energy is done solely by diffusion and the fluid stays at rest. The temperature profile in this case is linear. When the Rayleigh number is increased, the flow does not change until a critical value Ra_{crit} is reached. For higher Rayleigh numbers the flow becomes unstable and convection cells are formed. It can be shown that the critical Rayleigh number from which a small perturbation gets amplified depends on the wave number $\mathbf{k}$ of the perturbation. The lowest critical Rayleigh number (for the Rayleigh-Bénard convection between two rigid walls) is $Ra_{\text{crit}} = 1707.762$ for a wavenumber $k = 3.117$. This can be derived by perturbation theory as shown by Chandrasekhar [1970] and Gershuni & Zhukhovitskii [1976]. The wave length of this mode is $\lambda = \frac{2\pi}{k} \approx 2.016$.

4.2.1. Mathematical formulation

The setup of the problem consists of a fluid between two parallel and horizontal walls of distance H. The lower wall is heated to T_{hot}, while the upper wall is kept at T_{cold}. The temperature difference between both walls is $\Delta T = T_{\text{hot}} - T_{\text{cold}}$. It is assumed that the density variation due to the thermal expansion of the fluid is linear,

$$\rho = \rho_0 + \delta\rho = \rho_0 \left(1 - \beta(T_0 - T)\right) \tag{4.5}$$

Here β is the thermal expansion coefficient and ρ_0 is the density of the fluid with temperature T_0. The choice of ρ_0 or rather T_0 is not important, because only density variations $\delta\rho$ drive the flow and the constant part can be absorbed into the pressure term in the incompressible Navier-Stokes-equation (compare section 2.2.3). The expression from equation (4.5) is only used when computing the gravitational force, otherwise the fluid is treated as being incompressible, which is also known as Boussinesq approximation (see for example Tritton [1988]).
Instead of defining the system through physical variables one can also characterize the system by the dimensionless quantities Rayleigh and Prandtl number (compare with section 3.1):

$$Ra = \frac{g\beta\Delta T L_0^3}{\nu\kappa} \tag{4.6}$$

$$Pr = \frac{\nu}{\kappa}, \tag{4.7}$$

with g, ν and κ being the gravitational acceleration, the viscosity and the thermal diffusivity. As characteristic length scale L_0 the height of the channel H is chosen, while the characteristic time scale is taken as $t_0 = \frac{L_0^2}{\kappa}$. The temperature is made dimensionless

in such a way that T_{hot} and T_{cold} are ± 0.5, respectively, and thus $\Delta T = 1$. In this study the Prandtl number is fixed to $Pr = 0.71$.

4.2.2. Numerical setup

In sections 2.2.3 and 2.2.4 two models capable of simulating thermal flows were presented. Both models solve the fluid and thermal component separately on a different lattice each, and are then coupled via the Boussinesq approximation. For this experiment both models are used. In the following the implementation of this experiment is described for both models, which differ only in a few details.

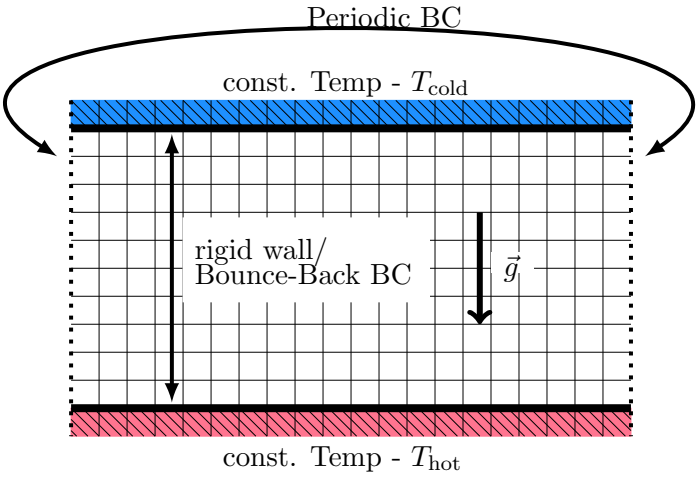

FIGURE 4.15.: Sketch of the numerical setup for the Rayleigh Bénard convection.

First the fluid domain has to be restricted. As already mentioned above, only a two-dimensional slice from the x-y-plane is simulated, which, from a physical point of view, would resemble a channel extended infinitely in z-direction (without any perturbations of the flow in z-direction). In y-direction the domain is confined by the two parallel walls. The x-direction should be unrestricted and is therefore realized by periodic boundary conditions (see section 3.4.1). Due to this artificially imposed symmetry due to the periodic BCs, only certain modes of the convection cells can occur. Therefore one has to assure that the length of the channel in x-direction is a multiple of the wavelength λ of the mode one is interested in. Thus, given the large influence of the length of the domain L_x on the observed flow, it must be well chosen. The rigid walls are realized using the Bounce-Back BCs (see section 3.4.2), while for the temperature a constant value at the wall is prescribed. For the LBGK model this is done using the BC given in section 3.4.3. The problem with these two very simple boundary conditions is that their effective positions do not exactly match! The wall for the Bounce-Back BC is located in between the last fluid and the wall node, while the constant value BC is applied directly

at the boundary. To reduce this effect, the temperature at the wall node is linearly extrapolated using the temperature close to the boundary and the desired temperature at the wall. This problem can be avoided with the MRT model by using the so-called 'anti-bounce-back' BC (see section 3.4.3).
A sketch of the numerical setup and all used boundary conditions is shown in Fig. 4.15. Once the system has been defined via Ra and Pr, a valid discretization has to be found and all variables have to be rewritten in lattice units (compare section 3.1). The domain should be divided into N_y grid cells, with a height $\delta_x = \frac{L_0}{N_y}$. In dimensionless formulation $L_0 = 1$ and thus $\delta_x = N_y^{-1}$. In section 4.2.1 the characteristic time was defined as $t_0 = \frac{L_0^2}{\kappa}$, which is unity also in the dimensionless formulation. Using the notation of equation (3.4), the temporal step size δ_t can be expressed as

$$\begin{aligned} t_0^{(\mathrm{dl})} = t_0^{(\mathrm{lb})}\delta_t^{(\mathrm{dl})} = \frac{{L_0^{(\mathrm{lb})}}^2}{\kappa^{(\mathrm{lb})}}\delta_t^{(\mathrm{dl})} = 1 \\ \rightarrow \quad \delta_t^{(\mathrm{dl})} = \kappa^{(\mathrm{lb})}{\delta_x^{(\mathrm{dl})}}^2, \end{aligned} \tag{4.8}$$

with $L_0^{(\mathrm{lb})} = N_y = {\delta_x^{(\mathrm{dl})}}^{-1}$. The thermal diffusivity κ in lattice units is given through equation (4.7) depending on the viscosity ν. A characteristic velocity of a thermal convective flow is $u_c = \sqrt{\beta g \Delta T L_0} = \sqrt{\frac{Ra}{Pr}}\frac{\nu}{L_0}$. Using this velocity in the definition of the Mach number Ma and considering that Ma needs to be small, an expression for the viscosity in lattice units can be found:

$$\begin{aligned} Ma = \frac{u_c}{c_s} = \sqrt{\frac{Ra}{Pr}}\frac{\nu}{L_0 c_s} = \sqrt{\frac{3Ra}{Pr}}\frac{\nu^{(\mathrm{lb})}}{N_y} \ll 1 \\ \rightarrow \quad \nu^{(\mathrm{lb})} = \frac{Ma^*}{\sqrt{3}}\sqrt{\frac{Pr}{Ra}}N_y \end{aligned} \tag{4.9}$$

Here the fact is used that the speed of sound in lattice units is known: $c_s = \frac{1}{\sqrt{3}}$. The star next to the Mach number denotes that it has to be chosen small (if not stated otherwise $Ma^* = 0.1$ is used). Knowing the viscosity ν means knowing κ and thereby via equation (4.8) also the temporal discretization δ_t.

As initial conditions the equilibrium distribution functions are used, as discussed in section 3.3, which are computed from prescribed macroscopic variables. The density is set to 1 for the LBGK model, and to zero for the MRT model (the MRT model computes density variations from ρ_0). The velocities in both models are set to zero in the whole domain and the temperature is set to a linear profile that matches the boundary conditions. It is necessary to add a slight perturbation to break the symmetry of the system, otherwise the fluid would stay in a state of artificial numerical stability. This is done by adding a small deviation from the temperature profile at one node.

4.2.3. Results & Discussion

As long as the Rayleigh number is below a certain threshold, all perturbations are dampened and any heat transport is solely due to conduction. This value depends on the wavelength λ, or rather its wave vector $k = \frac{2\pi}{\lambda}$. This critical Rayleigh number of marginal stability can be examined by linear stability analysis, as is well described in Chandrasekhar [1970]. For the Rayleigh Bénard convection between two rigid walls the solution can only be approximated by solving an eigenvalue problem for given k. The curve of marginal stability is shown in Fig. 4.19 (see Fig. D.10 for the graph from Chandrasekhar [1970]). It is found that it has a minimum for $k \approx 3.117$ with $Ra_{\text{crit}} \approx 1707.762$, which corresponds to a wavelength $\lambda \approx 2.016$. It is obvious that the symmetry due to the periodic boundary conditions in x-direction restricts the possible convection patterns to those where the length of the domain L_x is a multiple of the wavelength:

$$L_x = n\lambda_n = n\frac{2\pi}{k_n} \quad \text{, with } n = 1, 2, \ldots \tag{4.10}$$

Determining the minimal critical Rayleigh number require a domain of very high resolution, so that the aspect ratio $\Gamma = L_x/L_y = \lambda_1 = 2.016$. To approximate this value, the aspect ratio was set to 2, so that the mode that becomes unstable at first has a

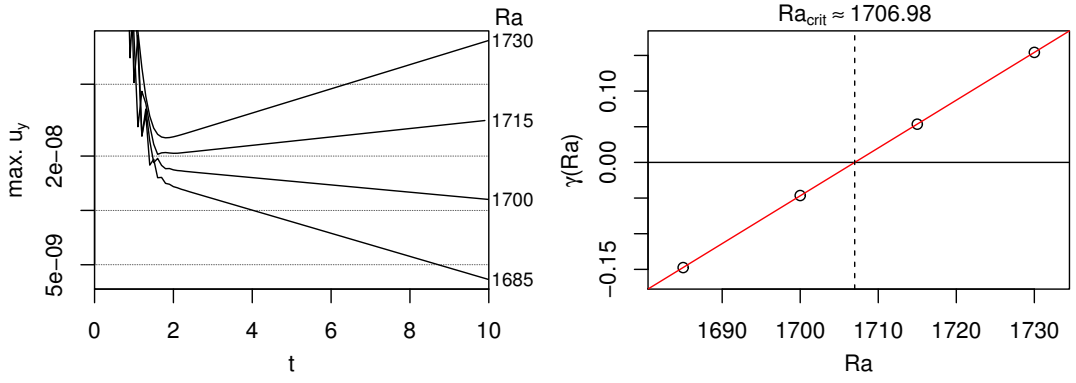

FIGURE 4.16.: Stability analysis of the Rayleigh-Bénard convection by analyzing the response of the system to a small initial perturbation depending on the Rayleigh number Ra. If the system is sub-critical, the maximum vertical velocity $u_{\text{y,max}}$ dampens; otherwise it will increase exponentially until a dynamical equilibrium is reached. After some initial rearrangement this behavior can be seen in the left panel for the LBGK model (spatial resolution: 322×161). By fitting lines to the logarithm of $u_{\text{y,max}}$ (in the interval $t \in [2.5, 10]$) the growth rates $\gamma(Ra)$ are found, which are shown in the right panel. By linear regression the critical Rayleigh number with $\gamma(Ra_{\text{crit}}) = 0$ is found. The results for both models (LBGK and MRT) at different spatial resolutions for an aspect ratio $\Gamma = 2$ are given in table 4.2.

N_y	Ra_{crit} LBGK	MRT
11	1822.909	1707.223
21	1727.703	1707.392
41	1710.626	1707.765
81	1707.547	1707.903
161	1706.983	1707.923

TABLE 4.2.: Critical Rayleigh numbers for $\lambda = 2$ obtained for different spatial resolutions N_y using the LBGK and MRT lattice Boltzmann models. From perturbation theory [Chandrasekhar, 1970] $Ra_{\text{crit}} \approx 1707.922$. This value is close, but not the same as the minimal Ra_{crit}.

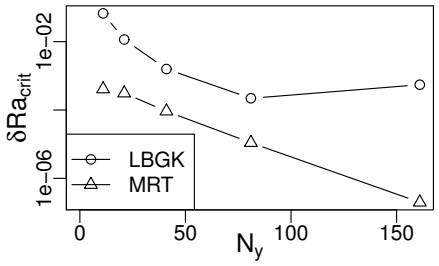

FIGURE 4.17.: Relative deviations δRa_{crit}, depending on N_y, from the theoretical value $Ra_{\text{crit}} \approx 1707.922$ of the results shown in table 4.2 for the LBGK and MRT model.

wave vector of length $k = \pi$, with $Ra_{\text{crit}} \approx 1707.922$[4]. The evolution of the system close to the point of marginal stability is quite slow, which is why the simulations for this experiment are not performed until the system reaches a steady state. The trend can already be seen earlier by analyzing the maximum vertical velocity $u_{\text{y,max}}$ that is induced by a small perturbation in the initialization. If the system is sub-critical this velocity is expected to exponentially decrease, and otherwise to exponentially increase. The growth rates $\gamma(Ra)$ are supposed to depend linearly on Ra in vicinity to Ra_{crit} (see the left panel in Fig. 4.16) and can be approximated by fitting a line to the logarithm of $u_{\text{y,max}}$. In the right panel of Fig. 4.16 the growth rates are plotted against Ra to obtain the critical Rayleigh number, where $\gamma(Ra_{\text{crit}}) = 0$. This point is approximated by a linear regression of the data points. The analysis to obtain the critical Rayleigh number for $\lambda = 2$ was done for several spatial resolutions using either the LBGK and the MRT model. The results are presented in table 4.2. The accuracy of the results depends strongly on the resolution, especially for the LBGK. The MRT gives, even for very coarse grids, very exact results. The relatively large differences from the theoretical value for the LBGK model at coarse resolution are most likely related to the displacement of the thermal boundary and the physical wall.

With these results, one point close to the minimum of the curve of marginal stability (Fig. 4.19) was confirmed. To determine the critical Rayleigh numbers for modes of different wavelengths, some simulations of different aspect ratios Γ are done. For moderate Ra the modes are preferred where the extent of the convection cells in x- and y-direction is similar. This becomes more obvious when looking at the curve of marginal stability in terms of the number of convection cells n and the horizontal length of the domain L_x,

[4]This value is very close to the minimum of the curve of marginal stability (see Fig. 4.19) for $k \approx 3.117$ or rather $\lambda \approx 2.016$. This should not be confused as in Wang et al. [2013]. They compare their results for $\lambda = 2$ with this minimum value and thus obtain larger deviations.

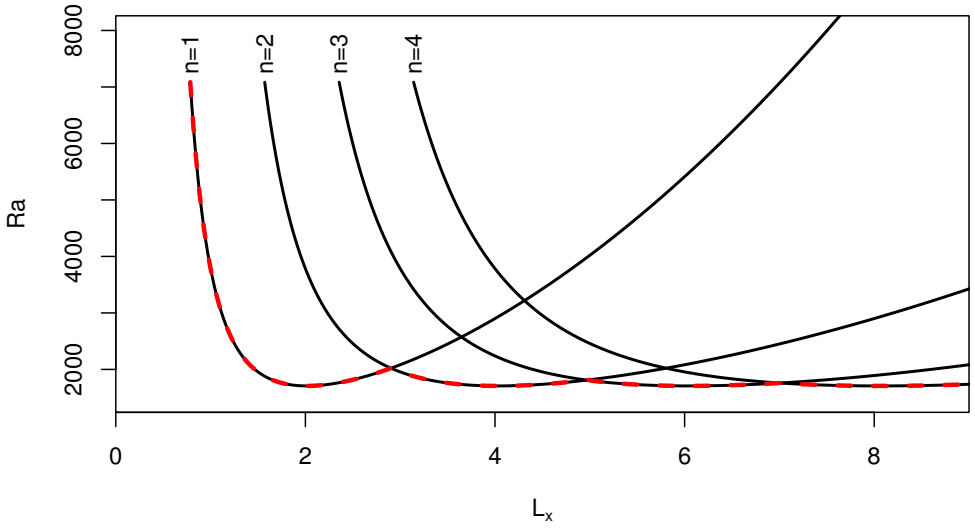

FIGURE 4.18.: Curve of marginal stability for different modes n depending of the domain width L_x. The red dashed line marks the minimal Rayleigh number for which the flow becomes unstable for a given L_x.

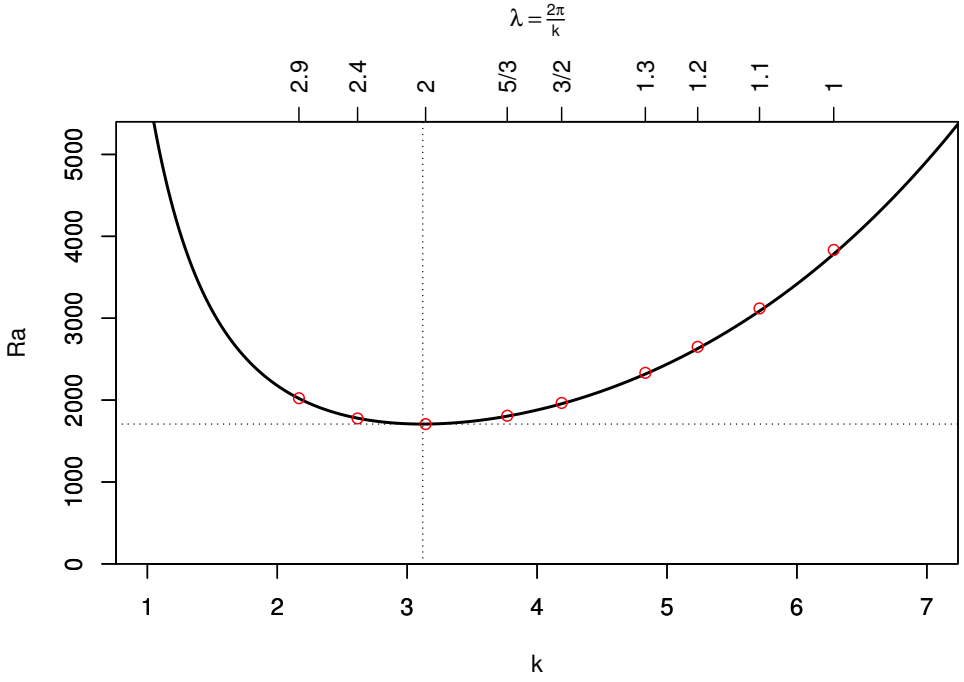

FIGURE 4.19.: The curve of marginal stability as it is predicted by perturbation theory (see for example Chandrasekhar [1970]). For parameter pairs below the curve the system is sub-critical and the heat transport is solely done via conduction. For points above the curve the system is super-critical and convection sets in. The convection cells occur in antisymmetric pairs (at least for moderate Ra) with wavelength λ. The dotted lines mark the minimum of the curve at $k \approx 3.117$ and $Ra \approx 1707.762$. The red circles represent the critical Rayleigh numbers for different k's, which was realized by varying the aspect ratio Γ of the fluid domain. The values of these points are given in table 4.3.

k	λ	exp. Ra_{crit}	theo. Ra_{crit}	L_x
2.17	2.90	2023.15	2017.52	2.9
2.62	2.40	1776.39	1779.09	2.4
3.14	2.00	1707.35	1707.90	2.0
3.77	1.67	1809.33	1804.18	5.0
4.19	1.50	1965.51	1956.34	3.0
4.83	1.30	2333.18	2316.63	1.3
5.24	1.20	2650.64	2632.97	1.2
5.71	1.10	3120.00	3085.92	1.1
6.28	1.00	3834.70	3779.98	1.0

TABLE 4.3.: This table presents the experimental obtained critical Rayleigh numbers for different realizations of k. The fourth column holds the theoretical values found by perturbation theory. The last column shows the horizontal length of the domain used to force a wavelength λ (see equation (4.10)). For this experiment the MRT model was used with a vertical resolution of 20 grid points.

which is depicted in Fig. 4.18. This figure was inspired by Fig. D.11 from Prat et al. [1998]. The connection between the critical Rayleigh number, which depends on k, and L_x is given by equation (4.10). The simulations for this experiment are solely done with the MRT model with a vertical resolution of 20 grid points. The analysis was done as before, with the results shown in table 4.3 and plotted in Fig. 4.19 next to the curve of marginal stability, as it was found in perturbation theory. It can be seen that especially for moderate Ra the results match the theoretical findings quite closely, although for reasons of computational workload only coarse grids were used. The following results were again all computed on a grid with aspect ratio $\Gamma = 2$ using the coupled LBGK model on a 322×161 grid. The simulations were carried out until the flow had stopped changing for a longer period of time (about 10-30 dimensionless time-steps). It should be mentioned that there is no guarantee that the flow stays like this forever. To study the behavior at a broader range of the Rayleigh number, experiments are done for $Ra = 2000$ to 82000 by 1000. Since the experiments here are only done using 2D models, the physical validity of the results is only guaranteed as long as no three-dimensional effects appear. Shan [1997] reports to find slight 3D-perturbations already for $Ra = 6000$. For higher values, it would be necessary to implement a 3D-version of this setup to recover a real physical image. However, this is beyond the scope of this thesis and thus not done here. It will be seen that the dynamics in this '2D-world' become quite complex as well. This fact should be kept in mind when reading the following part.

First the results for moderate Ra are studied. For $Ra < Ra_{\text{crit}}$, as mentioned before, the system is in a solely conducting state and the fluid is at rest. The temperature distribution is perfectly linear between the two walls and the pressure field is parabolic. It should be emphasized that, whenever the pressure is mentioned in the following, the pressure difference is meant. This is defined via equation (2.16a) through the density

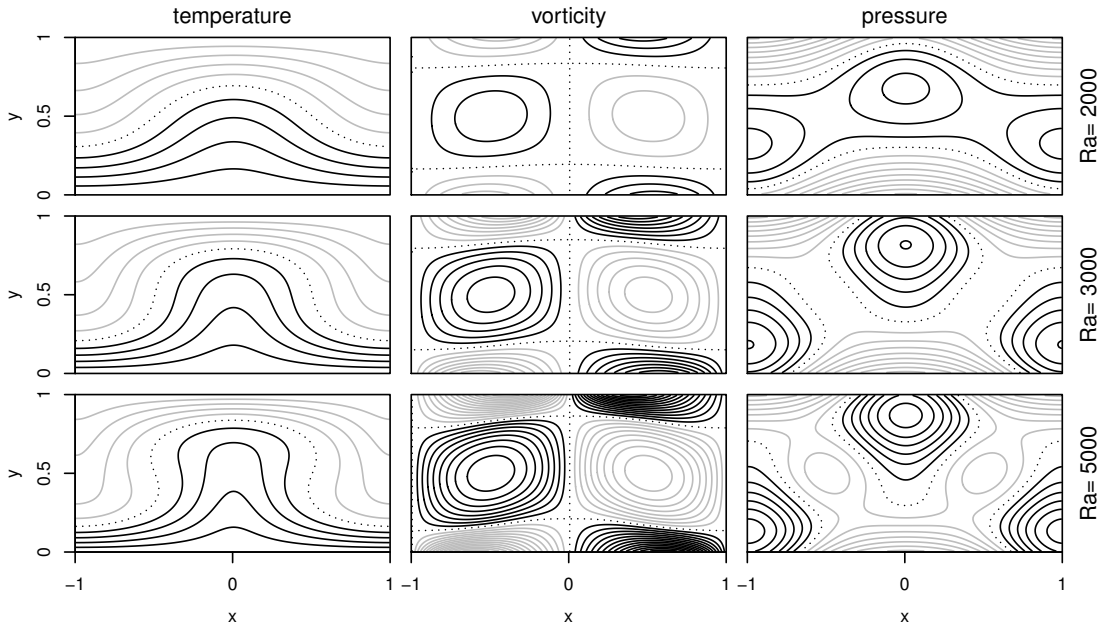

FIGURE 4.20.: Contour plots of the temperature, vorticity and pressure fields after reaching a dynamical equilibrium for different (moderate) Rayleigh numbers. One pair of convection cells (mode $n = 1$) with $\lambda = 2$ has been established. With increasing Ra the isothermal-temperature lines are deformed more strongly and also the vorticity and pressure fields develop. The dotted lines label a value of zero. Successive lines are evenly spaced by ± 0.1, ± 15 and $\pm 5 \cdot 10^{-5}$ for temperature, vorticity and pressure, respectively. Black lines denote positive, gray lines negative steps originating from the dotted lines.

difference, which was introduced when absorbing the isothermal part from the Boussinesq approximation into the pressure term of the Navier-Stokes equation (see section 4.2.1). When exceeding Ra_{crit}, convection sets in and the formerly parallel lines of constant temperature start bending. Fig. 4.20 shows the contour plots of the temperature, vorticity and pressure fields for $Ra = 2000, 3000, 5000$ after the dynamical equilibrium is reached. It can be seen that the fields become more and more marked with increasing Ra and that a pair of two antisymmetric vortices have developed, transporting heat from the hot to the cold wall. The kind of heat transport of a flow is described by the dimensionless Nusselt number Nu, which is defined as the total heat transport over the theoretical heat transport due to conduction. Therefore it is 1 for $Ra < Ra_{\mathrm{crit}}$ and above 1 otherwise. In the following only the volume-averaged Nusselt number (indicated by the bar) that describes the heat flow in vertical direction is studied:

$$Nu = \overline{Nu_y} = 1 + \frac{\overline{u_y T} H}{\kappa \Delta T}. \tag{4.11}$$

Fig. 4.21 shows the obtained results for Nu depending of Ra: as long as the system is sub-critical Nu is constant at 1; it immediately starts growing if super-critical. It is

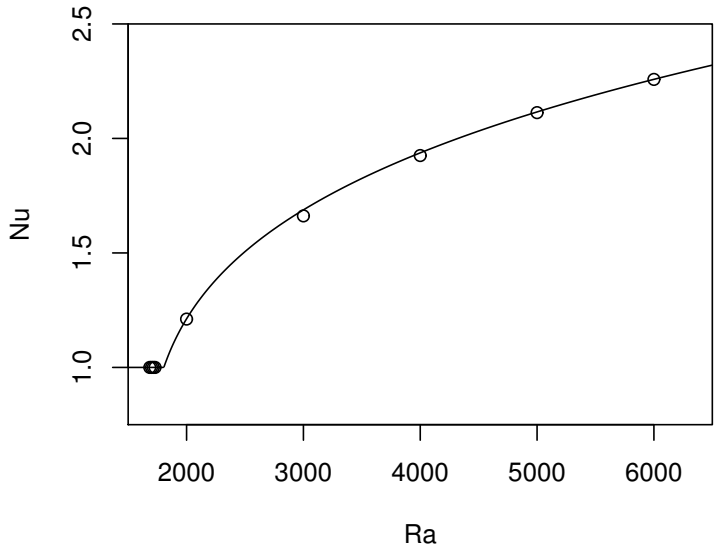

FIGURE 4.21.: Nusselt number Nu as a function of Ra. In the sub-critical regime no convection takes place and thus by definition $Nu = 1$. At the critical point for $Ra = Ra_{\text{crit}}$ the conductive solution becomes unstable and a basic convective mode (here $n = 1$ with $\lambda = 2$) becomes stable. In this super-critical section the Nusselt number scales approximately with $Ra^{0.25}$.

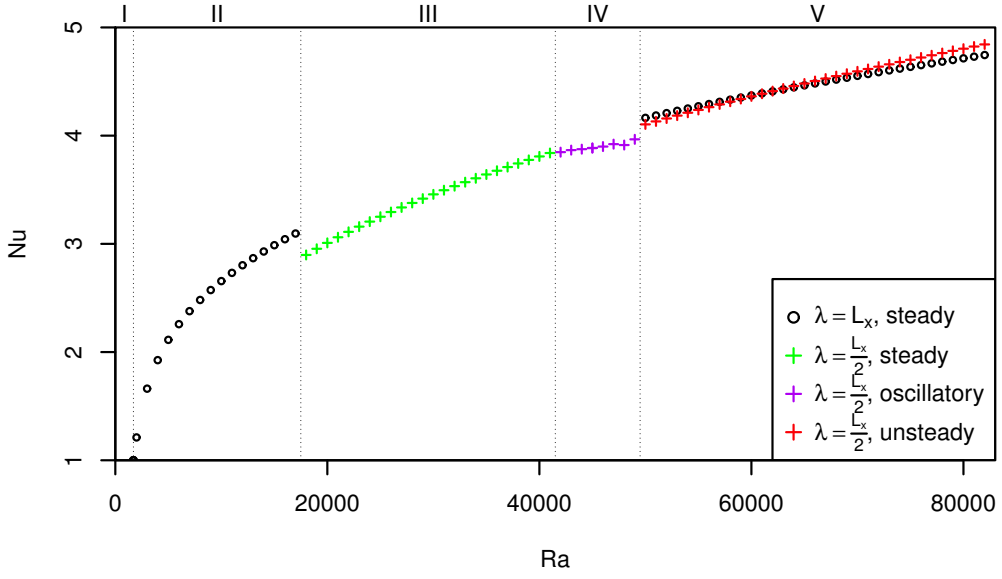

FIGURE 4.22.: Dependence of the Nusselt number Nu on Ra. The range of the Rayleigh number is larger than in Fig. 4.21. It can be seen that for larger Ra the flow passes through several bifurcations. The range shown here can roughly be divided into five sections (I-V), for which the flow shows different behavior. The dotted vertical lines are drawn in between two values of Ra for which the flow changes. These changes occur at approximately 1708, 17500, 41500 and 49500. The kind of flow for the results can be distinguished by color and symbol. The circles and crosses denote one and two pairs of convection cells, respectively.

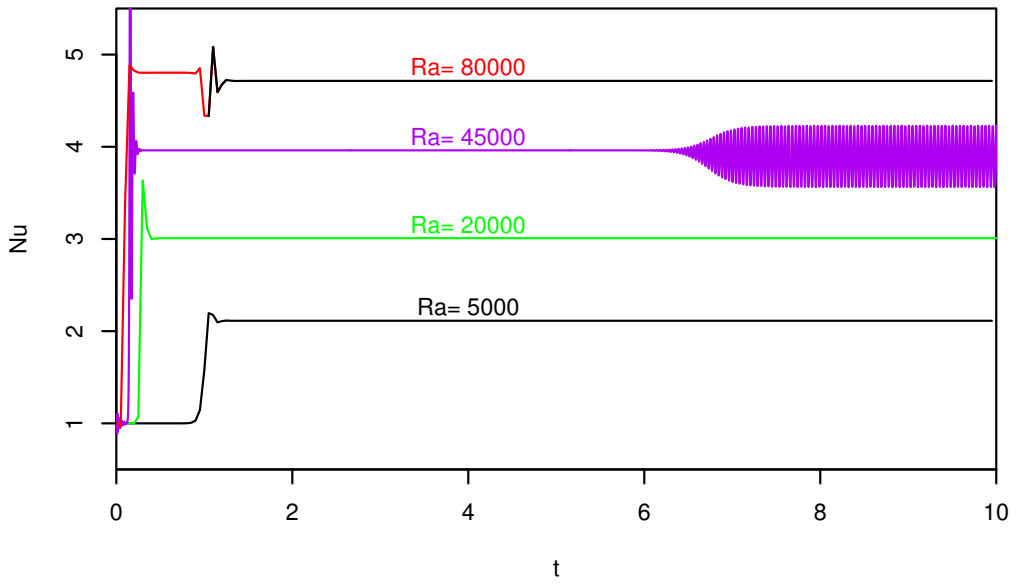

FIGURE 4.23.: Temporal evolution of the Nusselt number Nu for different Ra. The color code is analogous to Fig. 4.22 and distinguishes the behavior of the flow. The peak in the line for $Ra = 80000$ marks the moment when the modes switch, which is also illustrated by the change in color.

found that Nu approximately scales with $Ra^{0.25}$ for moderate Ra. The curve that was fitted to the results and illustrates this behavior is plotted in the same figure. For higher Rayleigh numbers non-linear effects arise, which can be seen in Fig. 4.22. At some Ra the smooth increase of the Nusselt number is interrupted by sudden steps, indicating a change in the flow. The range of Rayleigh numbers that is studied here can be subdivided into five sections (I-V) with different behavior of the flow. The dotted lines are plotted in between the values of Ra for which such a change happens. The occurring convection patterns can be described by the number of convection cells n (analogous to

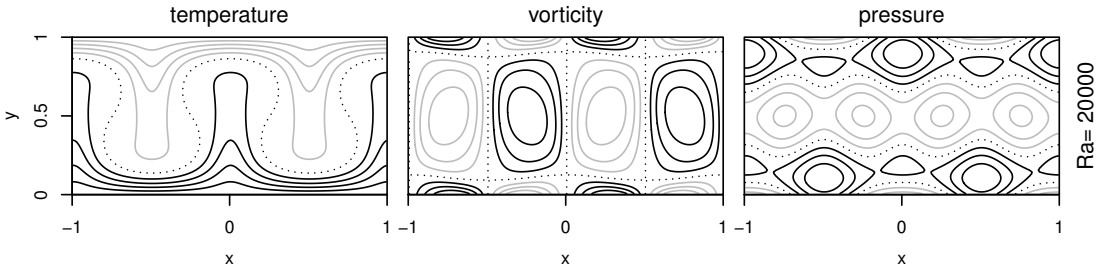

FIGURE 4.24.: Contour plots of the temperature, vorticity and pressure fields after reaching a dynamical equilibrium for $Ra = 20000$. Two pairs of convection cells (mode $n = 2$) with $\lambda = 1$ are established. The dotted lines label a value of zero. Successive lines are evenly spaced by ± 0.1, ± 100 and $\pm 5 \cdot 10^{-5}$ for temperature, vorticity and pressure, respectively. Black lines denote positive, gray lines negative steps originating from the dotted lines.

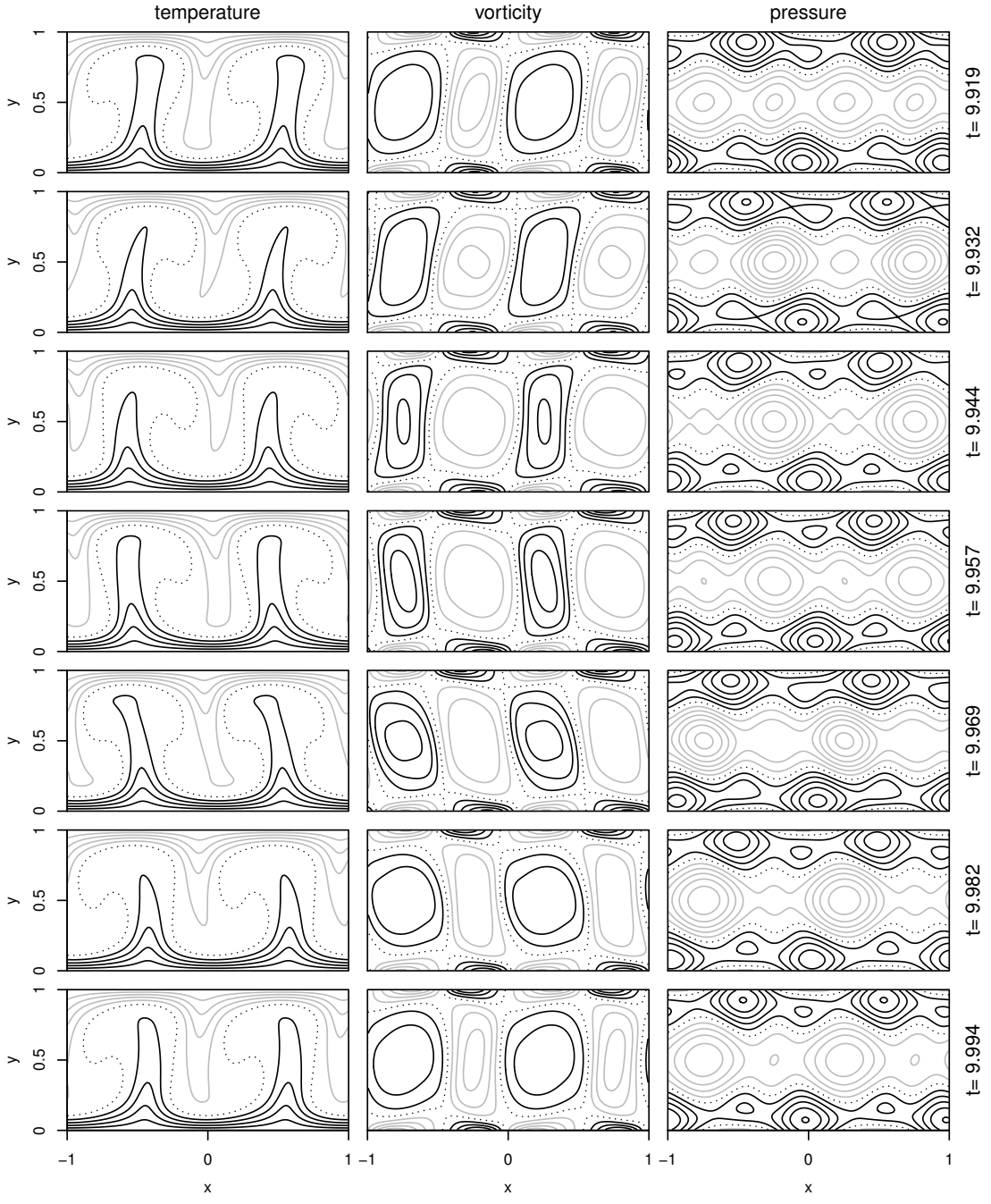

FIGURE 4.25.: Successive snapshots of the temperature, vorticity and pressure fields for the time-dependent solution for $Ra = 45000$. Two pairs of convection cells (mode $n = 2$) with $\lambda = 1$ are found, which are periodically fluctuating. The snapshots are evenly distributed over approximately one period. The dotted lines label a value of zero. Successive lines are evenly spaced by ± 0.1, ± 200 and $\pm 5 \cdot 10^{-5}$ for temperature, vorticity and pressure, respectively. Black lines denote positive, gray lines negative steps originating from the dotted lines.

eq. (4.10)) that fit into the domain of length $L_x = 2$. To better distinguish between the different states, the data points in Fig. 4.22 are depicted using different symbols and colors. The black circles denote the appearance of mode $n = 1$, which remained stable whenever emerged. The crosses mark mode $n = 2$, which is either stable (green), oscillating (purple) or unstable (red). The temporal behavior of the Nusselt number for a representative result for each section (except the first non-convective) is shown in Fig. 4.23. The color code is the same as in Fig. 4.22. The first two sections (I-II) were already discussed above. The third (III) ranges approximately from $Ra = 18000$ to 41000. In this range the mode with $n = 1$ is replaced by mode $n = 2$ of half wavelength. As before, the results are time-independent and shown as green crosses in Fig. 4.22. This behavior can be seen in Fig. 4.23 (for $Ra = 20000$ - green line): After a short phase of initialization the Nusselt number reaches its final value (same as for $Ra = 5000$ - black line). Contour plots for the steady flow with mode $n = 2$ are shown in Fig. 4.24. The two pairs of convection cells, compressed along the x-axis, can easily be identified. Note that the values of the contour lines for the vorticity have been rescaled, compared to Fig. 4.20, to properly represent the larger range of values.
The fourth section ($42000 \leq Ra \leq 49000$, section IV in Fig. 4.22) is a bit more cumbersome. As before, after initialization the mode with $n = 2$ shows up, but it starts oscillating after a short time (see purple line in Fig. 4.23 for $Ra = 45000$). This oscillating mode seems to be persistent: even after long simulation runs this behavior did not change. The values for Nu in this range in Fig. 4.22 are temporal averages of the Nusselt number over several periods and are plotted as purple crosses. A time series of contours of the fields spanning approximately one period is shown in Fig. 4.25. A detailed study of such phenomena in two-dimensional periodic domains for a similar setup was done by Prat et al. [1995, 1998]. They find that these time-dependent modes are 'mixed-modes' of modes that are in resonance. These effects are not studied in more detail here. This oscillatory mode was also reported by Shan [1997] for $Ra = 50000$. A time series of this oscillatory mode from this paper, similar to Fig. 4.25, is shown in Fig. D.12.

The last section (V), for $Ra \geq 50000$, again shows a different behavior of the flow. After initialization the mode with $n = 2$ occurs quickly, and after some time this mode becomes unstable and there is a transition to flow with mode $n = 1$. In Fig. 4.22 there are two symbols for these Rayleigh numbers: one for the unstable (red crosses) and one for the stable mode (black circles). This transition can also be seen in the temporal evolution of Nu in Fig. 4.23 for $Ra = 80000$ as a peak. The different colors of the line indicate the change of the mode. Snapshots of the temperature, vorticity and pressure fields at different moments of this transition are shown in Fig. 4.26. It is also worth noting that the Nusselt number of the flow for the unstable mode is at first below the one of the stable mode, while this is the opposite way for $Ra \geq 62000$.

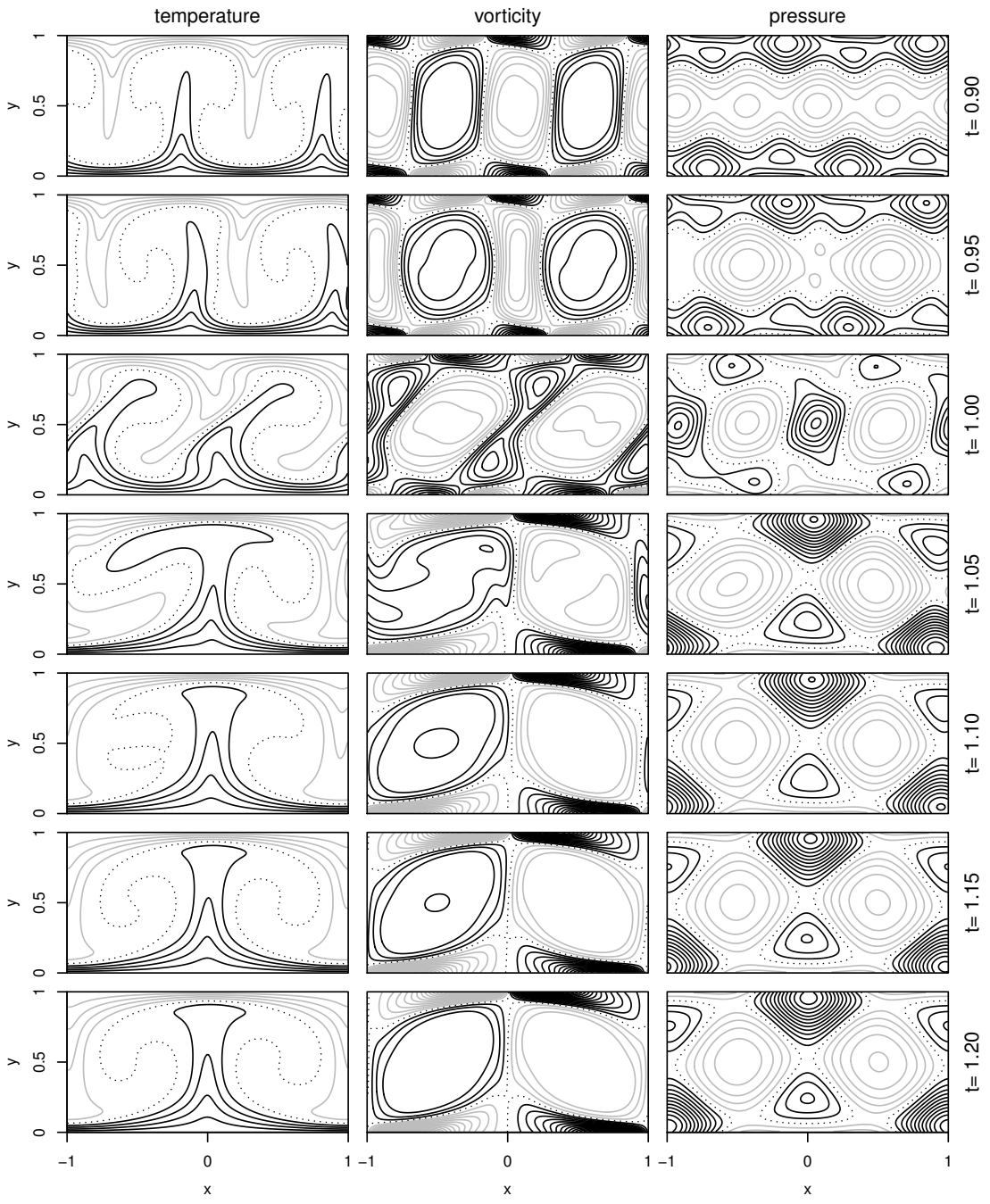

FIGURE 4.26.: Successive snapshots of the temperature, vorticity and pressure fields for $Ra = 80000$. After initialization convection starts with two pairs of convection cells, which quickly becomes unstable and switches to the mode with only one pair of convection cells. Except for this transition, which is shown here, the solution seems to be time-independent. The dotted lines label a value of zero. Successive lines are evenly spaced by ± 0.1, ± 150 and $\pm 5 \cdot 10^{-5}$ for temperature, vorticity and pressure, respectively. Black lines denote positive, gray lines negative steps originating from the dotted lines.

5. Conclusions & Outlook

In this thesis different numerical models based on the lattice Boltzmann equation are validated using two classical hydrodynamical problems.
In the first part, a short introduction to the theory behind the Lattice Boltzmann equation was given and different models for both athermal and thermal flows were presented. two single-relaxation-time models based on the BGK-approximation of the collision operator and one multi-relaxation-time model was introduced [Ginzburg, 2005; Wang et al., 2013]. Using relevant parts of the source-code, preliminary remarks for the implementation of the Lattice Boltzmann models were given, including the proper treatment of initial and boundary conditions, as well as possible ways to improve the computational efficiency of the algorithm by parallelization.
In the second part, the previously introduced models were applied to the two-dimensional flow past a cylinder in an enclosed channel and the two-dimensional Rayleigh-Bénard convection. Parameter studies were performed for both experiments. The aim of these studies was to locate hydrodynamical instabilities for which the characteristics of the flow change. The flow about a cylinder was also studied numerically by i.a. Zovatto & Pedrizzetti [2001]. The Rayleigh-Bénard problem was studied theoretically by i.a. Chandrasekhar [1970] and numerically by i.a. Wang et al. [2013].
For the two-dimensional flow past a cylinder between two parallel walls, the transition from steady laminar to time-dependent flow was studied depending on the Reynolds number Re and a gap parameter that describes the minimum gap between cylinder and wall. In the case of an unsteady flow, vortices are shed from the cylinder, forming patterns similar to a von Kármán vortex street. It was found, that the distance between cylinder and wall has a strong influence on the cylinder's wake, and thus on the dynamics of the flow. The results obtained from this study were found to be in very good agreement with the results found by Zovatto & Pedrizzetti [2001], who studied the same problem using a different numerical approach. Their model solved the hydrodynamical equations on a triangular mesh with grid refinement in vicinity to the cylinder and the walls using a finite element method.
With the Rayleigh Bénard setup, the natural convection in a horizontal layer was studied for a fixed Prandtl number $Pr = 0.71$. By varying the Rayleigh number Ra, the transition from the steady, solely conductive solution to the onset of convection was

analyzed. Theoretical considerations from linear perturbation theory (see for example Chandrasekhar [1970]) predict the critical Rayleigh number Ra_{crit}, above which a perturbation is amplified, to depend on the wavenumber k of the perturbation. The same dependence was found from the numerical study of this thesis. For higher Rayleigh numbers the flow passes through several bifurcations. Prat et al. [1995, 1998], who also studied the 2D Rayleigh-Bénard system, both numerically and analytically, report qualitatively similar results. However, the results in their papers were obtained for $Pr = 10$ and are thus not directly comparable.

One very important aspect found in this context is the influence of the horizontal width of the numerical domain when using periodic boundary conditions. Due to this periodicity an artificial symmetry is induced into the system, which excludes a majority of possible convection modes: only modes for which the length of the domain is a multiple of the wavelength are possible. This has to be kept in mind when studying this instability.

The instability for $\lambda = 2$, which is close to the minimal critical Rayleigh number, was analyzed using the LBGK and MRT models at different spatial resolutions. The results for the critical Rayleigh numbers are much more accurate for the MRT model than for the LBGK model. It should be noted that these considerable deviations might also be attributed in part to the different types of thermal boundary conditions used for the models. The values obtained by the MRT model are astonishingly accurate even on a very rough lattice, thereby resulting in a much better computational efficiency despite a slightly more complex algorithm.

These studies still have vast potential for further improvements. A crucial part are the boundary conditions, which could be further adapted to certain problems. An advanced implementation of rigid wall boundaries could, for example, be extended to allow for an exact placement of the wall and also for a proper handling of curved boundaries. Furthermore, the treatment of open boundaries that prevent the reflection of density waves back into the domain is a big challenge.

Two-dimensional approximations are only physically justified as long as there are no three-dimensional solutions of the flow, and are generally valid only within a small range in parameter space. To study the physical system for parameters beyond this range, which might be important especially for more applied experiments, the experiment needs to be implemented in three dimensions. Adding a new dimension would mean a rapid increase of the computational effort. This trend could be antagonized by a dynamical adaptation of the grid size, as well as by the use of sub-grid models that approximate the dissipative effects not resolved by a rough spatial resolution.

A. Conservation laws

The contribution of the collision integral $\left.\frac{\partial f}{\partial t}\right|_{\text{coll}}$ (equation (2.5)) vanishes by definition when multiplying the Boltzmann equation (2.3) with a collisional invariant $\chi(\mathbf{x}, \boldsymbol{\xi})$ and integrating it over the microscopic velocities $\boldsymbol{\xi}$ (in the following a system with no external force is assumed: $\mathbf{F} = 0$).

$$\int \mathrm{d}\boldsymbol{\xi}\chi \left[\frac{\partial}{\partial t} + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}}\right] f = \int \mathrm{d}\boldsymbol{\xi}\chi \left.\frac{\partial f}{\partial t}\right|_{\text{coll}} = 0 \tag{A.1}$$

By expanding this formulation and assuming summation over indices that occur twice one gets:

$$\frac{\partial}{\partial t}\int \mathrm{d}\boldsymbol{\xi}\chi f + \frac{\partial}{\partial x_\alpha}\int \mathrm{d}\boldsymbol{\xi}\chi\xi_\alpha f - \int \mathrm{d}\boldsymbol{\xi}\frac{\partial \chi}{\partial x_\alpha}\xi_\alpha f = 0 \tag{A.2}$$

Using the following notation

$$\langle A \rangle = \int \mathrm{d}\boldsymbol{\xi} A f \tag{A.3}$$

equation (A.2) becomes:

$$\frac{\partial}{\partial t}\langle \chi \rangle + \frac{\partial}{\partial x_\alpha}\langle \chi\xi_\alpha \rangle - \left\langle \frac{\partial \chi}{\partial x_\alpha}\xi_\alpha \right\rangle = 0 \tag{A.4}$$

Mass conservation

By setting $\chi = 1$ the continuity equation is derived. Equation (A.4) becomes

$$\begin{aligned} &\frac{\partial}{\partial t}\langle 1 \rangle + \frac{\partial}{\partial x_\alpha}\langle \xi_\alpha \rangle \\ &= \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_\alpha}(\rho u_\alpha) = 0 \end{aligned} \tag{A.5}$$

Here the relations of the moments (2.7a) and (2.7b) were used.

Momentum conservation

To get the equation for momentum conservation χ is set to be a component of the microscopic velocity $\boldsymbol{\xi}$:

$$\chi = \xi_\alpha \tag{A.6}$$

From equation (A.4) one gets

$$\begin{aligned} &\frac{\partial}{\partial t}\langle \xi_\alpha \rangle + \frac{\partial}{\partial x_\beta}\langle \xi_\alpha \xi_\beta \rangle \\ &= \frac{\partial}{\partial t}(\rho u_\alpha) + \frac{\partial}{\partial x_\beta}\left[\rho u_\alpha u_\beta + \langle (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta)\rangle\right] \\ &= 0 \end{aligned} \tag{A.7}$$

In the second line the identity

$$\begin{aligned} \langle \xi_\alpha \xi_\beta \rangle &= \langle (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta)\rangle + \langle \xi_\alpha \rangle u_\beta + u_\alpha \langle \xi_\beta \rangle - u_\alpha u_\beta \langle 1 \rangle \\ &= \langle (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta)\rangle + \rho u_\alpha u_\beta \end{aligned} \tag{A.8}$$

was used. By using the continuity equation (eq. (A.5)) and identifying the pressure tensor

$$\hat{P}_{\alpha\beta} = \langle (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta)\rangle \tag{A.9}$$

we get

$$\rho\left(\frac{\partial u_\alpha}{\partial t} + u_\beta \frac{\partial u_\alpha}{\partial x_\beta}\right) = -\frac{\partial}{\partial x_\beta}\hat{P}_{\alpha\beta} \tag{A.10}$$

B. Chapman-Enskog expansion

The Chapman-Enskog expansion was developed by Chapman [1916, 1918] and Enskog [1917]. With this multi-scale expansion it is possible to derive macroscopic descriptions (for example the Navier-Stokes equation) of a system from the Boltzmann equation. For the sake of simplicity, the collision integral of the Boltzmann equation is approximated here by the BGK-collision operator. The basic steps of this expansion are shown here and are kept close to the derivation of Wolf-Gladrow [2000].

For this expansion the distribution function and the space and time variables are expanded in a series:

$$f = f^{(0)} + \varepsilon f^{(1)} + \varepsilon^2 f^{(2)} + \dots \tag{B.1a}$$

$$\partial_t = \varepsilon \partial_t^{(1)} + \varepsilon^2 \partial_t^{(2)} + \dots \tag{B.1b}$$

$$\partial_{x_\alpha} = \varepsilon \partial_{x_\alpha}^{(1)} + \dots \tag{B.1c}$$

The idea of the expansion is to represent the variables by their different relevant scales. The expansion is just a formal expansion to keep track of the different orders of magnitude of the terms. During the transformations the parameter ε is supposed to be small, why smaller terms are denoted by larger exponents. Later ε can simply set to 1 to get rid of it again.
The zeroth order approximation of the distribution function $f^{(0)}$ is given by the Boltzmann-Maxwellian distribution function given in equation (2.6).

Introducing the expansions (B.1a) - (B.1c) in the continuity and momentum equation (eq. (A.5) and (A.10)) and dropping all terms of higher order than two, one gets:

$$(\varepsilon \partial_t^{(1)} + \varepsilon^2 \partial_t^{(2)})\rho = -\varepsilon \partial_{x_\alpha}^{(1)} (\rho u_\alpha) \tag{B.2}$$

$$\rho \left(\varepsilon \partial_t^{(1)} + \varepsilon^2 \partial_t^{(2)} + u_\beta \varepsilon \partial_{x_\beta}^{(1)} \right) u_\alpha = -\varepsilon \partial_{x_\beta}^{(1)} \hat{P}_{\alpha\beta}^{(0)} - \varepsilon^2 \partial_{x_\beta}^{(1)} \hat{P}_{\alpha\beta}^{(1)} \tag{B.3}$$

Accordingly to equation (A.9) $P_{\alpha\beta}^{(n)}$ is:

$$P_{\alpha\beta}^{(n)} = \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) f^{(n)} \tag{B.4}$$

Regarding the terms of different orders in ε separately yields to the following set of equations:

$$\varepsilon^1: \qquad \partial_t^{(1)}\rho = -\partial_{x_\alpha}^{(1)}(\rho u_\alpha) \tag{B.5a}$$

$$\varepsilon^2: \qquad \partial_t^{(2)}\rho = 0 \tag{B.5b}$$

$$\varepsilon^1: \qquad \left(\partial_t^{(1)} + u_\beta \partial_{x_\beta}^{(1)}\right) u_\alpha = -\rho^{-1}\partial_{x_\beta}^{(1)}\hat{P}_{\alpha\beta}^{(0)} \tag{B.5c}$$

$$\varepsilon^2: \qquad \partial_t^{(2)} u_\alpha = -\rho^{-1}\partial_{x_\beta}^{(1)}\hat{P}_{\alpha\beta}^{(1)} \tag{B.5d}$$

First order approximation

If only terms of maximum order ε^1 are considered (equations (B.5a) and (B.5c)), the continuity and the Euler equations are obtained from equations (B.2) and (B.3):

$$\partial_t \rho = -\nabla(\rho \mathbf{u}) \tag{B.6}$$

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\rho^{-1}\nabla p, \tag{B.7}$$

where p is the pressure that is given by the zeroth order approximation of the pressure tensor (from eq. (B.4))

$$\hat{P}_{\alpha\beta}^{(0)} = \int \mathrm{d}\boldsymbol{\xi}(\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) f^{(0)} = \delta_{\alpha\beta}\rho RT = \delta_{\alpha\beta} p \tag{B.8}$$

Second order approximation

For a second order approximation one has to retain the terms of order ε^2. This does not change the continuity equation because there is no higher order contribution from the mass conservation equation (see eq. (B.5b)). For the second order approximation of the momentum conservation equation an expression for $\hat{P}_{\alpha\beta}^{(1)}$ has to be found (compare eq. (B.5d)).

The BGK collision operator $J(f)$ can be rewritten with the expansion of f (eq. (B.1a))

$$J(f) = -\omega(f - f^{(0)}) \tag{B.9}$$

$$= -\omega(\varepsilon f^{(1)} + \varepsilon^2 f^{(2)} + \dots) \tag{B.10}$$

$$= J^{(0)} + \varepsilon J^{(1)} + \varepsilon^2 J^{(2)} + \dots \tag{B.11}$$

With this expression and the expansions the Boltzmann equation has the following form:

$$\left(\varepsilon\partial_t^{(1)} + \varepsilon^2\partial_t^{(2)} + \xi_\alpha\varepsilon\partial_{x_\alpha}^{(1)}\right)\left(f^{(0)} + \varepsilon f^{(1)} + \varepsilon^2 f^{(2)}\right) = J^{(0)} + \varepsilon J^{(1)} + \varepsilon^2 J^{(2)} \tag{B.12}$$

By regarding the terms of different orders in ε again separately one gets for the zeroth and first order:

$$J^{(0)} = 0 \tag{B.13a}$$

$$\partial_t^{(1)} f^{(0)} + \xi_\alpha \partial_{x_\alpha}^{(1)} f^{(0)} = J^{(1)} = -\omega f^{(1)} \tag{B.13b}$$

The first term reflects the fact, that the collision term vanishes for the Boltzmann-Maxwellian function (compare section 2.1) and the second term gives an expression for $f^{(1)}$ in terms of $f^{(0)}$. Equation (B.13b) is then used to replace $f^{(1)}$ to find a expression for the first order approximation of the pressure tensor $\hat{P}_{\alpha\beta}^{(1)}$.

$$\begin{aligned} \hat{P}_{\alpha\beta}^{(1)} &= \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) f^{(1)} \\ &= -\omega^{-1} \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) \left[\partial_t^{(1)} f^{(0)} + \xi_\gamma \partial_{x_\gamma}^{(1)} f^{(0)} \right] \end{aligned} \tag{B.14}$$

Due to $f^{(0)}$ depending of t and $\mathbf{x}$ only through ρ and $\mathbf{u}$ and by replacing the temporal with spatial derivatives using equations (B.5a) and (B.5c) $\hat{P}_{\alpha\beta}^{(1)}$ can be written as

$$\begin{aligned} \hat{P}_{\alpha\beta}^{(1)} &= -\omega^{-1} \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) \Big[\partial_\rho f^{(0)} \partial_t^{(1)} \rho + \partial_{u_\gamma} f^{(0)} \partial_t^{(1)} u_\gamma \\ &\quad + \xi_\gamma \partial_\rho f^{(0)} \partial_{x_\gamma} \rho + \xi_\gamma \partial_{u_\gamma} f^{(0)} \partial_{x_\gamma}^{(1)} u_\gamma \Big] \\ &= -\omega^{-1} \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) \Big[-f^{(0)} \partial_{x_\gamma}^{(1)} u_\gamma - f^{(0)} u_\gamma \partial_{x_\gamma}^{(1)} \rho \\ &\quad + (RT)^{-1} (\xi_\gamma - u_\gamma) f^{(0)} \left(u_\delta \partial_{x_\delta}^{(1)} u_\gamma + \rho^{-1} \delta_{\gamma\delta} \partial_{x_\delta}^{(1)} p \right) \\ &\quad + (RT)^{-1} \xi_\gamma (\xi_\delta - u_\delta) f^{(0)} \partial_{x_\gamma}^{(1)} u_\delta + \xi_\gamma f^{(0)} \partial_{x_\gamma}^{(1)} \rho \Big] \\ &= \omega^{-1} \partial_{x_\gamma}^{(1)} u_\gamma \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) f^{(0)} \\ &\quad - \omega^{-1} \partial_{x_\gamma}^{(1)} \rho \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta)(\xi_\gamma - u_\gamma) f^{(0)} \\ &\quad - \frac{u_\delta \partial_{x_\delta}^{(1)} u_\gamma + \rho^{-1} \delta_{\gamma\delta} \partial_{x_\delta}^{(1)} p}{\omega RT} \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta)(\xi_\gamma - u_\gamma) f^{(0)} \\ &\quad - \frac{\partial_{x_\gamma}^{(1)} u_\delta}{\omega RT} \int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) \xi_\gamma (\xi_\delta - u_\delta) f^{(0)} \end{aligned} \tag{B.15}$$

Those integrals can be solved:

$$\int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) f^{(0)} = \rho RT \delta_{\alpha\beta} \tag{B.16a}$$

$$\int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta)(\xi_\gamma - u_\gamma) f^{(0)} = 0 \tag{B.16b}$$

$$\int \mathrm{d}\boldsymbol{\xi} (\xi_\alpha - u_\alpha)(\xi_\beta - u_\beta) \xi_\gamma (\xi_\delta - u_\delta) f^{(0)} = \rho (RT)^2 (\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \tag{B.16c}$$

With those results $\hat{P}^{(1)}_{\alpha\beta}$ can be written as

$$\hat{P}^{(1)}_{\alpha\beta} = -\frac{\rho RT}{\omega}\left[(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma})\partial^{(1)}_{x_\gamma}u_\delta - \delta_{\alpha\beta}\partial^{(1)}_{x_\gamma}u_\gamma\right] \tag{B.17}$$

The divergence of $\hat{P}^{(1)}_{\alpha\beta}$ is:

$$\partial_{x_\alpha}\hat{P}^{(1)}_{\alpha\beta} = \rho\nu\left(\partial^{(1)}_{x_\beta}\partial^{(1)}_{x_\beta}u_\alpha + \partial^{(1)}_{x_\alpha}\partial^{(1)}_{x_\beta}u_\beta\right) \tag{B.18}$$

Here $\nu = RT\omega^{-1}$ is the viscosity.

When putting the results derived for the first and second order approximations back into equation (B.3) and setting $\varepsilon = 1$ the Navier-Stokes equation is obtained:

$$\partial_t\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\rho^{-1}\nabla p + \nu\left(\nabla^2\mathbf{u} + \nabla\left(\nabla\cdot\mathbf{u}\right)\right) \tag{B.19}$$

C. NetCDF C++

NetCDF by Unidata is quite a commonly used file format in geo-sciences. It is possible to store time series of several variables (which can also be multidimensional arrays) in one file. Unfortunately the current version of the C++ interface is not well documented online, which is why some snippets of code are given here to show how netCDF files are initialized and written.

The example given here stores a variable `fl_var` of type `float` and a 2D array `fl_2d` of floats at successive time steps. The first step is to initialize a netCDF file (see listing C.1). In line 3 a new file is opened by creating a new instance of type `NcFile`; the variable `netCDFfile` is a pointer that was defined at an upper level of the code. First the dimensions of the variables have to be defined and added to the file (see lines 6-8). The first argument of the function `addDim` is the name of the dimension, the second argument defines its length (the number of entries along this dimension). If no length is specified (line 8) the dimension is 'unlimited', which means that the length of this dimension is dynamically adjusted when storing variables. This is especially useful for the time dimension, when it is not clear at the stage of initialization how many time slices will have to be stored.

After the dimensions are defined one can add a variable for each dimension to store the coordinates of this dimension. In lines 11-13 these variables are added using the function `addVar`. As a first argument the name of the variable is needed (in case the variable should only store the coordinates of one dimension and its name has to be identical to the name of the dimension), the second argument is the variable type and the last argument is the dimension. In lines 16-28 the coordinates are computed and saved to the particular variable (using `putVar`). It is also possible to add more information to the variables and the file using the `putAtt` function[1]. In this example the function is used to add information about the units of the variables (see lines 31-33 and 40-41).

In lines 36 & 37 the two variables in which the results should later be stored are added. These variables are defined at a higher level of the code (as was also `netCDFfile`) to

[1] There exist a lot of different conventions about the inner structure of netCDF files to make the processing of different datasets easier or even possible. Those conventions outline requirements for the used variable and dimension names but also for the provided attributes. A list of different conventions and their requirements can be found online:
`http://www.unidata.ucar.edu/software/netcdf/conventions.html`

```
1  int init_netCDF(){
2     // Create the file.
3     netCDFfile = new netCDF::NcFile('exampleFile.nc', netCDF::NcFile::replace,
       netCDF::NcFile::nc4);
4
5     //Define dimensions
6     netCDF::NcDim xDim = netCDFfile->addDim("x_dimension", xSize);
7     netCDF::NcDim yDim = netCDFfile->addDim("y_dimension", ySize);
8     netCDF::NcDim timeDim = netCDFfile->addDim("time");
9
10    //Define variables that describe the dimensions
11    netCDF::NcVar xVar = netCDFfile->addVar("x_dimension", netCDF::ncFloat, xDim);
12    netCDF::NcVar yVar = netCDFfile->addVar("y_dimension", netCDF::ncFloat, yDim);
13    timeVar = netCDFfile->addVar("time", netCDF::ncFloat, timeDim);
14
15    //Compute and add values to the dimension variables
16    double *x_scale = new double[xSize];
17    double *y_scale = new double[ySize];
18
19    for (int _x = 0; _x < xSize; _x++)
20            x_scale[_x] = dx *(_x  - (xSize)/2.);
21    for (int _y = 0; _y < ySize; _y++)
22            y_scale[_y] = dx *_y;
23
24    yVar.putVar(y_scale);
25    xVar.putVar(x_scale);
26
27    delete x_scale;
28    delete y_scale;
29
30    //add units and more informations about the dimension variables
31    xVar.putAtt("units", "m");
32    yVar.putAtt("units", "m");
33    timeVar.putAtt("units", "s");
34
35    //create variables
36    fl_2dVar = netCDFfile->addVar("2dVar_name", netCDF::ncFloat, std::vector<
       netCDF::NcDim>{timeDim, yDim, xDim});
37    fl_Var = netCDFfile->addVar("Var_name", netCDF::ncFloat, timeDim);
38
39    //add units and more informations about the variables
40    fl_2dVar.putAtt("units", "kg/m^3");
41    fl_Var.putAtt("units", "m/s");
42
43    return 0;
44 }
```

Listing C.1: Initialization of a netCDF file

keep them valid after the initialization function is finished. Again the function `addVar` is used, but this time the choice of the name is independent of the dimensions, and for multi-dimensional variables a vector containing the dimensions has to be assigned as the third argument (see line 36). The order of the dimensions in that vector can be crucial for the performance of writing the data. To avoid expensive re-sorting of the whole array, the order in which the dimensions are stored in memory[2] should be the same as the order of the dimensions in the netCDF file. The dimensions are sorted by the rate how quick the dimensions vary when going through the array in memory. The dimension varying fastest comes last in that vector, the dimension varying slowest (this

[2] A multi-dimensional array is stored in memory as a long vector. To read or write an element of this array, its coordinates are mapped onto this vector to find its position.

```
1  int saveState(){
2   std::vector<size_t> start_t = {(unsigned int)numOutput};
3   std::vector<size_t> start_2d = {(unsigned int)numOutput, 0, 0};
4   std::vector<size_t> count_2d = {1, (unsigned int)ySize, (unsigned int)xSize};
5   std::vector<ptrdiff_t> stride_2d = {1, 1, 1};
6   std::vector<ptrdiff_t> imap_2d = {1, (unsigned int)xSize,1};
7
8   timeVar.putVar(start_t, t);
9   fl_2dVar.putVar(start_2d, count_2d, stride_2d, imap_2d, fl_2d_array);
10  fl_Var.putVar(start_t, fl_data);
11
12  return 0;
13 }
```

LISTING C.2: Writing data to a netCDF file

is usually the 'record'-dimension, in this example the time) comes first.
After the initialization of the netCDF file is finished, data can be written into the variables. This is done by the function `saveState` in listing C.2. As mentioned above, the time dimension was defined as unlimited. To store the data at the right position one has to keep track of the number of records written (here this is done by the variable `numOutput`). This is the only information needed for writing only a single variable (see lines 2 and 8 in listing C.2). For writing array-like data, one has to define some vectors of the type `unsigned int` to define where and how the data is written into the variable (see lines 3-6). The order of vector elements corresponds to the order of dimensions of the netCDF file. Here is a short description of those vectors:

- **start:** defines the index where the first element is written. Usually the index of the record variable is set to the current number of output `numOutput` and the other dimensions are written from the beginning (index 0).
- **count:** specifies the number of elements along each dimension that should be written.
- **stride:** specifies the sampling interval along each dimension.
- **imap:** defines the mapping of the dimensions to the array in memory. This number is the distance (number of elements) in memory between two adjacent elements along one dimension.

```
1  int reopenNcFile(){
2   netCDFfile = new netCDF::NcFile('exampleFile.nc', netCDF::NcFile::write);
3
4   timeVar = netCDFfile->getVar("time");
5   fl_2dVar = netCDFfile->getVar("2dVar_name");
6   fl_Var = netCDFfile->getVar("Var_name");
7
8   return 0;
9  }
```

LISTING C.3: Opening an existing netCDF file

Once the program is finished, the netCDF file is automatically closed and can then be analyzed. However, it can also be closed manually (for example to provide a pause function to have a look at the preliminary results during run-time) by calling the destructor of the NcFile object (`netCDFfile->~NcFile()`). To add data to an existing file the file has to be opened and the handlers for the variables have to be read (see listing C.3).

D. Additional figures

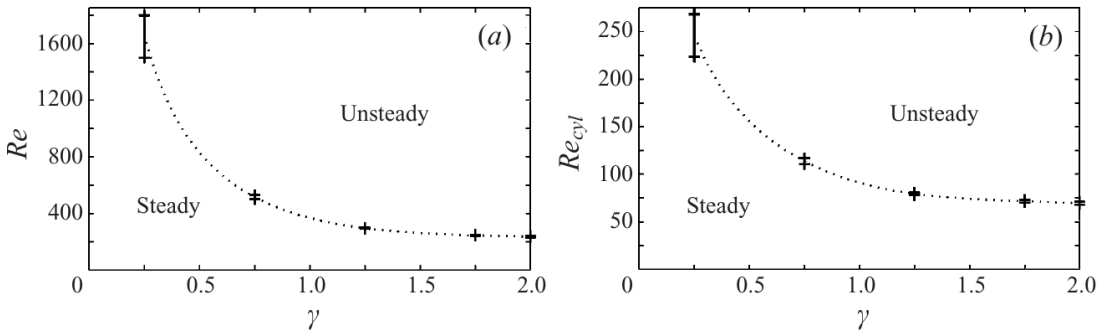

FIGURE D.1.: Figure 2 of Zovatto & Pedrizzetti [2001]. Original caption: *Critical (a) Reynolds number and (b) cylinder-based Reynolds number as a function of the gap parameter.*

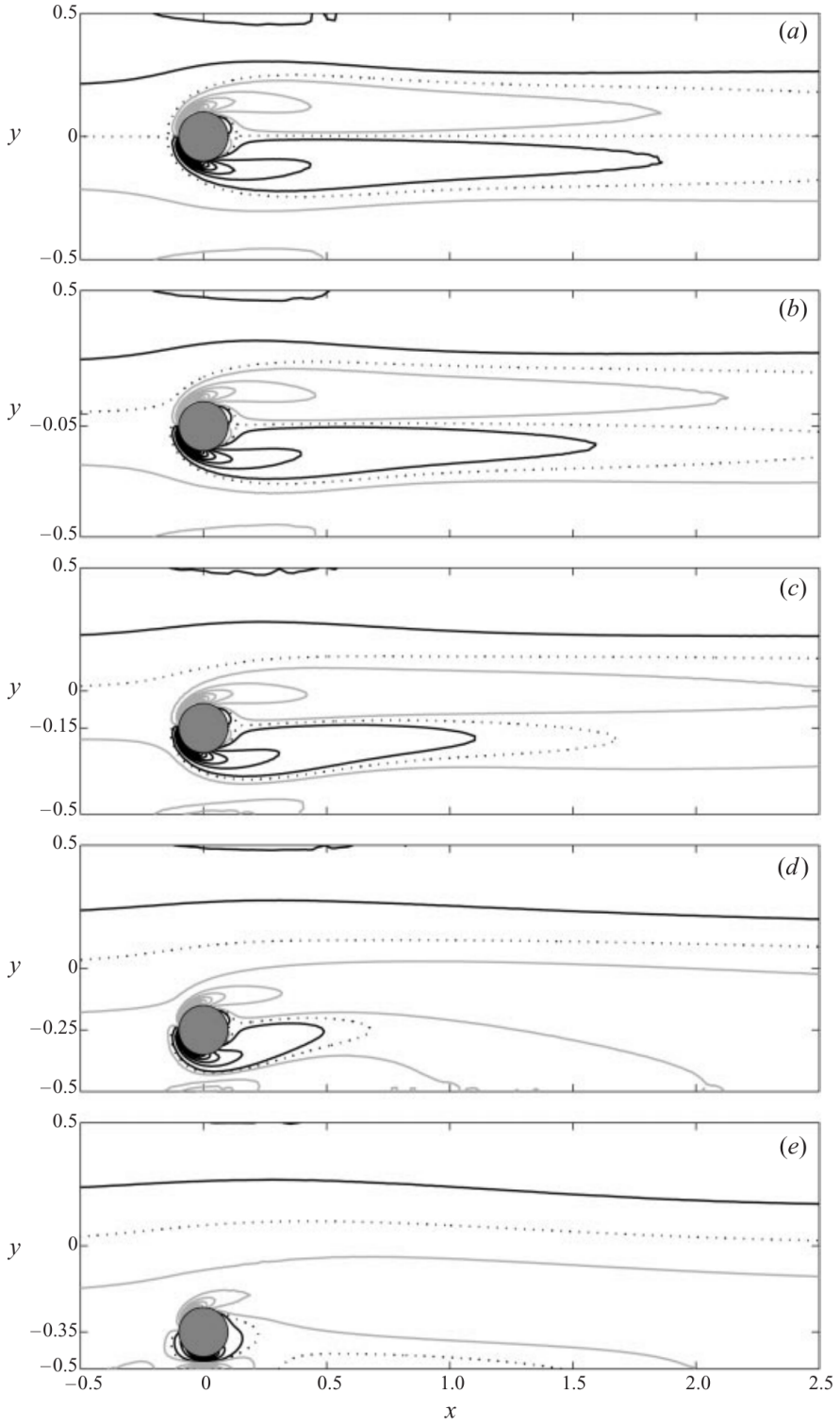

FIGURE D.2.: Figure 3 of Zovatto & Pedrizzetti [2001]. Original caption: *Steady flow vorticity contours for* $Re = 200$*, and (a)* $\gamma = 2$*, (b)* $\gamma = 1.75$*, (c)* $\gamma = 1.25$*, (d)* $\gamma = 0.75$*, (e)* $\gamma = 0.25$*.*

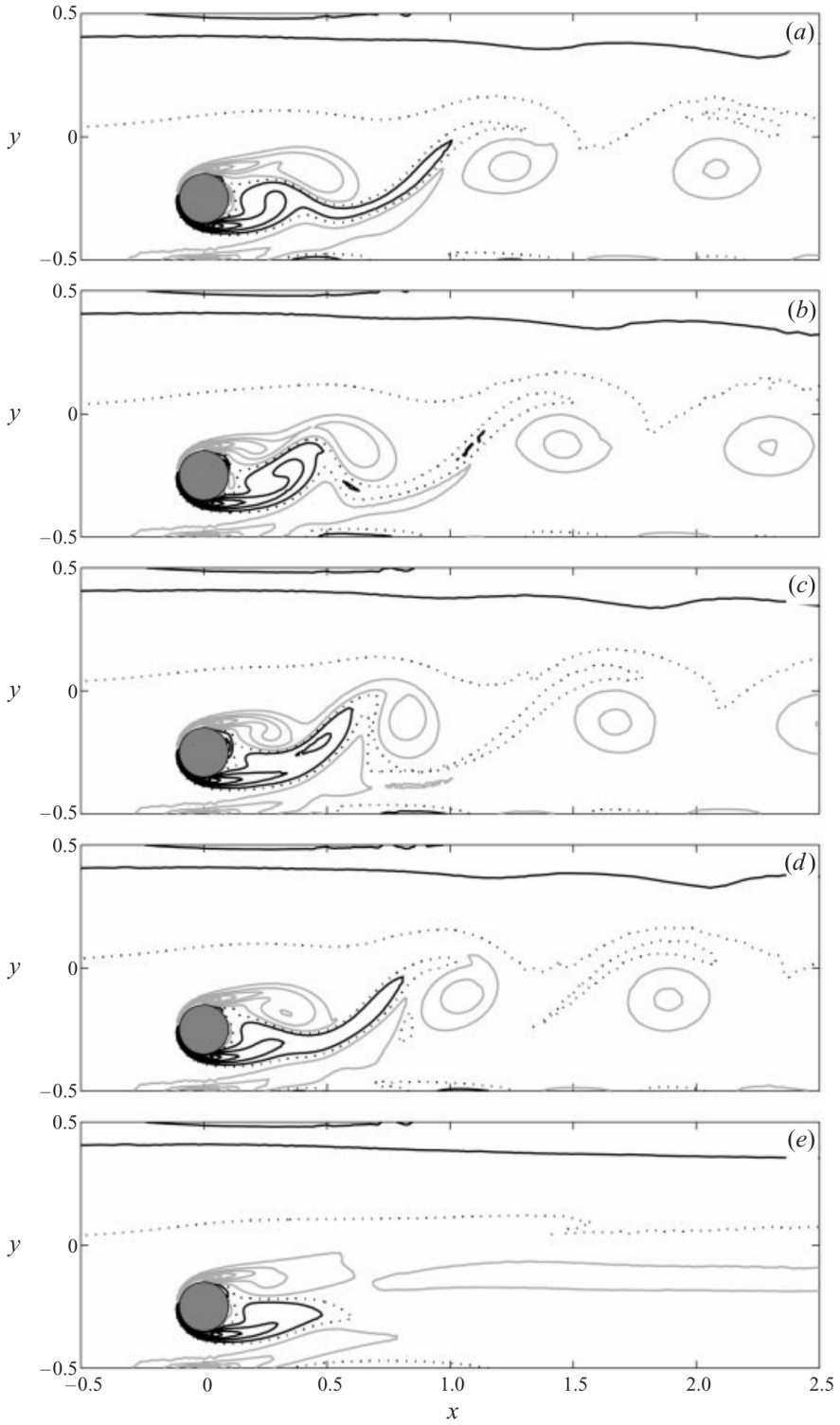

FIGURE D.3.: Figure 7 of Zovatto & Pedrizzetti [2001]. Original caption: *Unsteady flow vorticity contours for* $Re = 1000$ *and* $\gamma = 0.75$*; (a–d) instantaneous vorticity during one period* ($T = 0.78$) *with time increasing from a to d in steps of* 0.2 *time units; (e) period-averaged vorticity.*

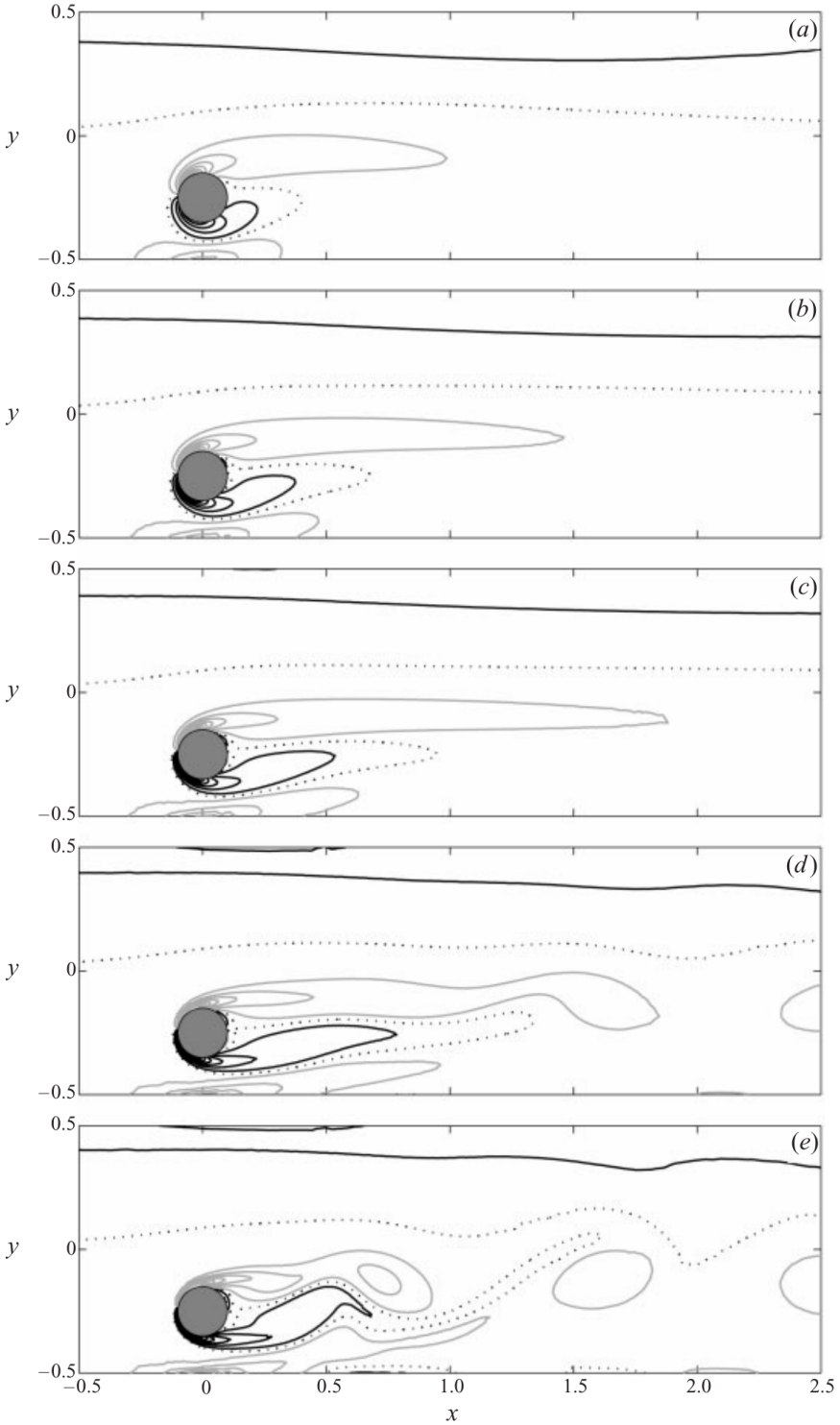

FIGURE D.4.: Figure 9 of Zovatto & Pedrizzetti [2001]. Original caption: *Flow vorticity contours for* $\gamma = 0.75$ *and (a)* $Re = 100$*, (b)* 200*, (c)* 300*, (d)* 500*, (e)* 700*: (a, b, c) are steady fields, (d, e) are instantaneous fields during the periodic regime.*

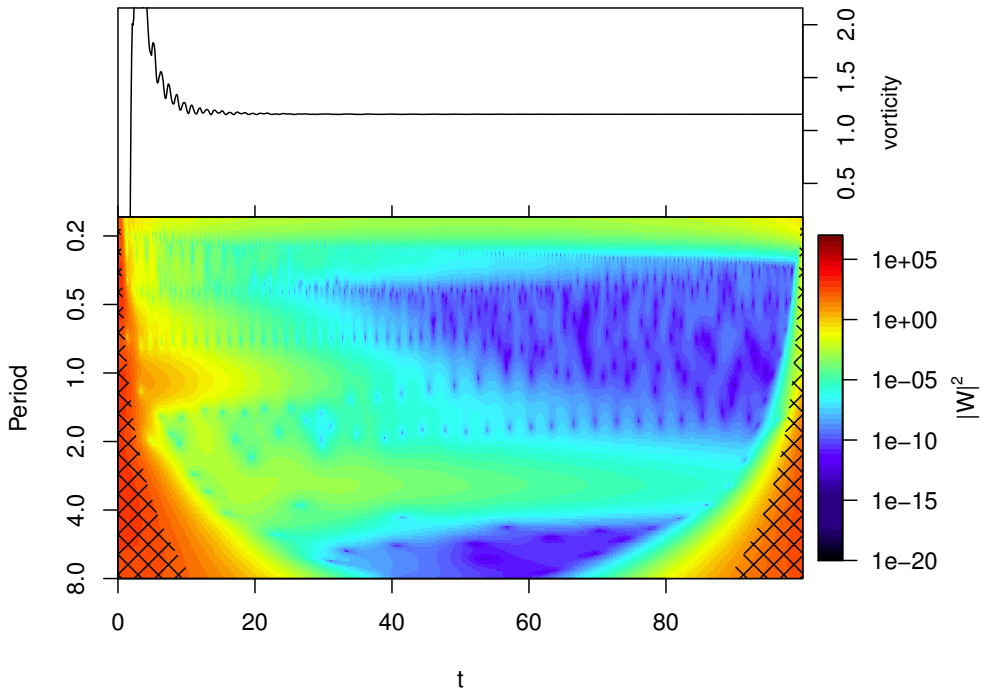

FIGURE D.5.: Temporal evolution of the vorticity at (x_p, y_p) (top) and its wavelet power spectrum (bottom) for $Re = 800$ and $\gamma = 0.25$. The hatched areas mark the cone of influence (see section 4.1.3 for more information).

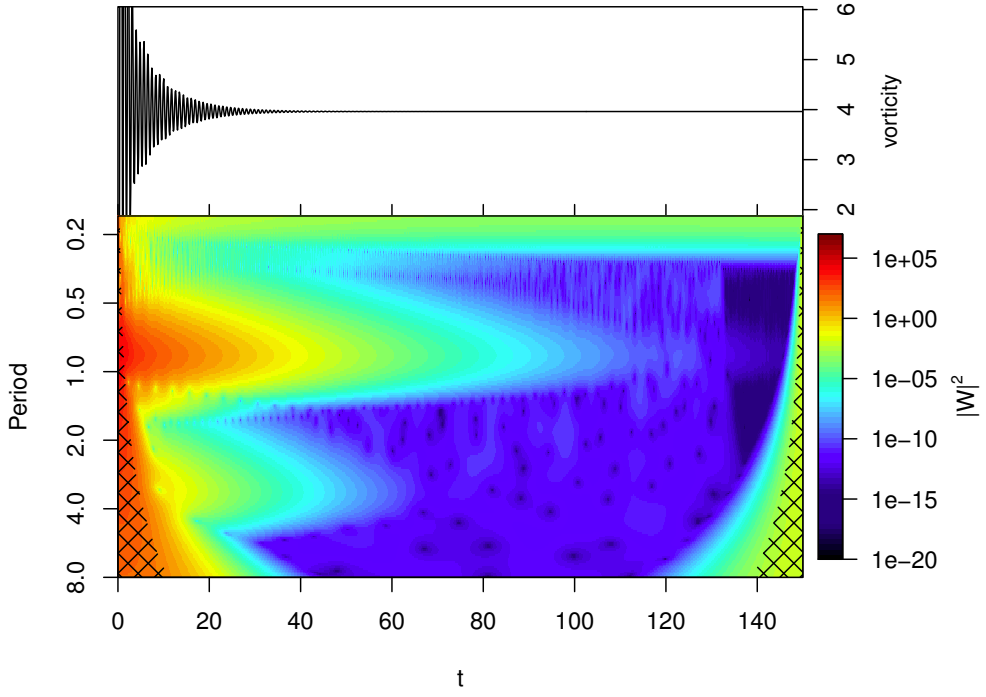

FIGURE D.6.: Temporal evolution of the vorticity at (x_p, y_p) (top) and its wavelet power spectrum (bottom) for $Re = 270$ and $\gamma = 1.25$. The hatched areas mark the cone of influence (see section 4.1.3 for more information).

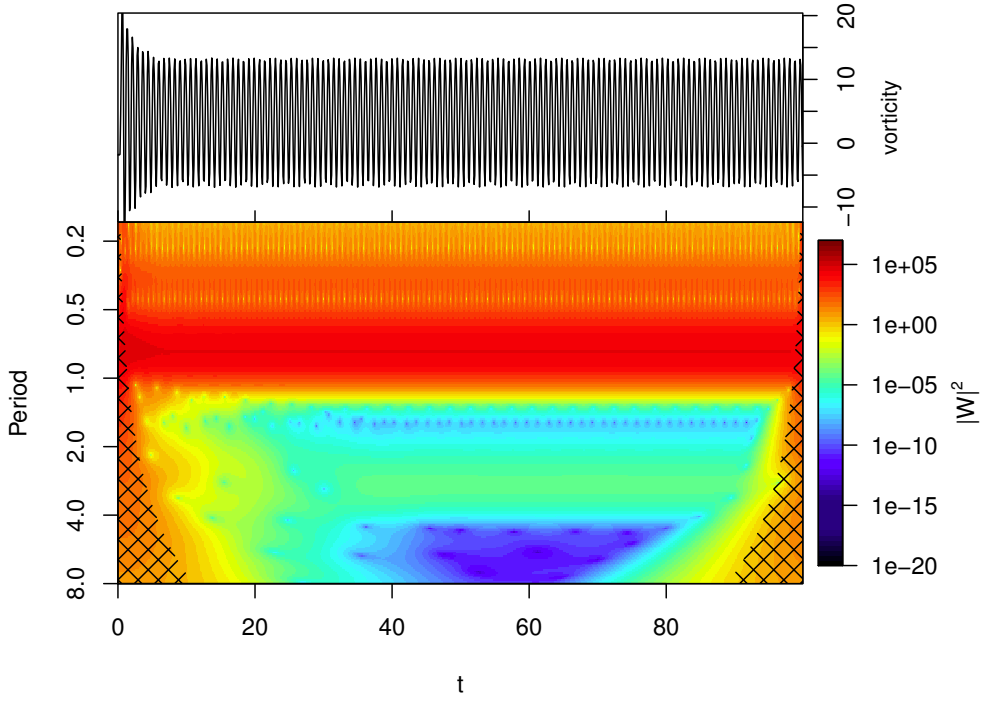

FIGURE D.7.: Temporal evolution of the vorticity at (x_p, y_p) (top) and its wavelet power spectrum (bottom) for $Re = 450$ and $\gamma = 1.25$. The combination of parameters is instable with a period length of $T = 0.77$. The vortex shedding almost instantaneous sets in because of the strong perturbation at the beginning and unsymmetrical placement of the cylinder. The hatched areas mark the cone of influence (see section 4.1.3 for more information).

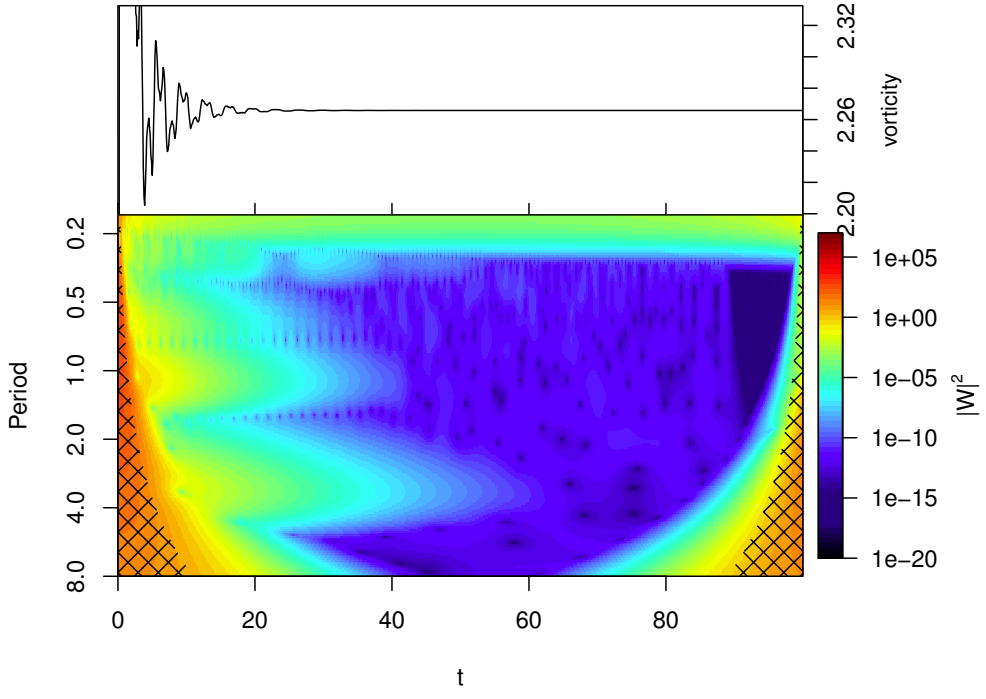

FIGURE D.8.: Temporal evolution of the vorticity at (x_p, y_p) (top) and its wavelet power spectrum (bottom) for $Re = 200$ and $\gamma = 2$. The hatched areas mark the cone of influence (see section 4.1.3 for more information).

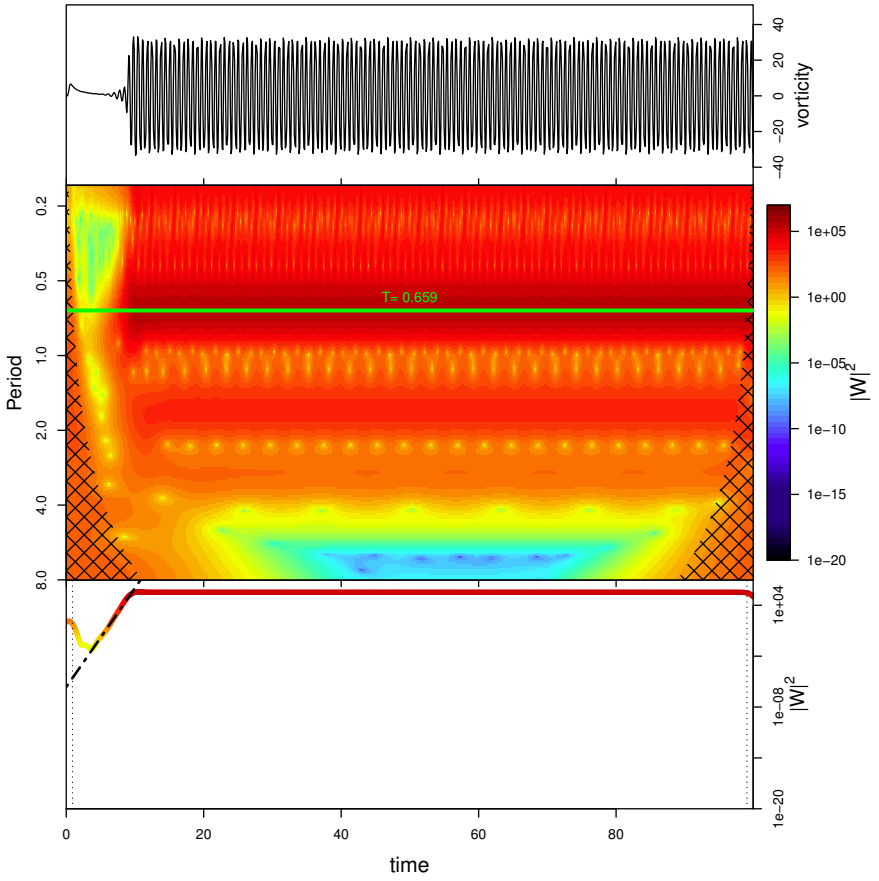

FIGURE D.9.: Temporal evolution of the vorticity at $(x_\mathrm{p}, y_\mathrm{p})$ (top) and its wavelet power spectrum (middle) for $Re = 1000$ and $\gamma = 2$. The combination of parameters is instable with a period length of $T = 0.659$. The vortex shedding sets in quickly but the exponential increase of the characteristic mode can still be seen in the bottom panel, which shows a transect of the power spectrum for the period $T = 0.659$ (marked in the mid panel by the green line). To determine the growth rate k, a line is fitted to the logarithm of the power spectrum in the interval where the increase is exponential ($t \in [4, 9]$). This line can be seen in the bottom panel as the dashed black line. The cylinder is centered in the channel and the shedding only sets in because of a small perturbation in the initial conditions of the velocity field to break the symmetry. The hatched areas in the mid panel mark the cone of influence (see section 4.1.3 for more information).

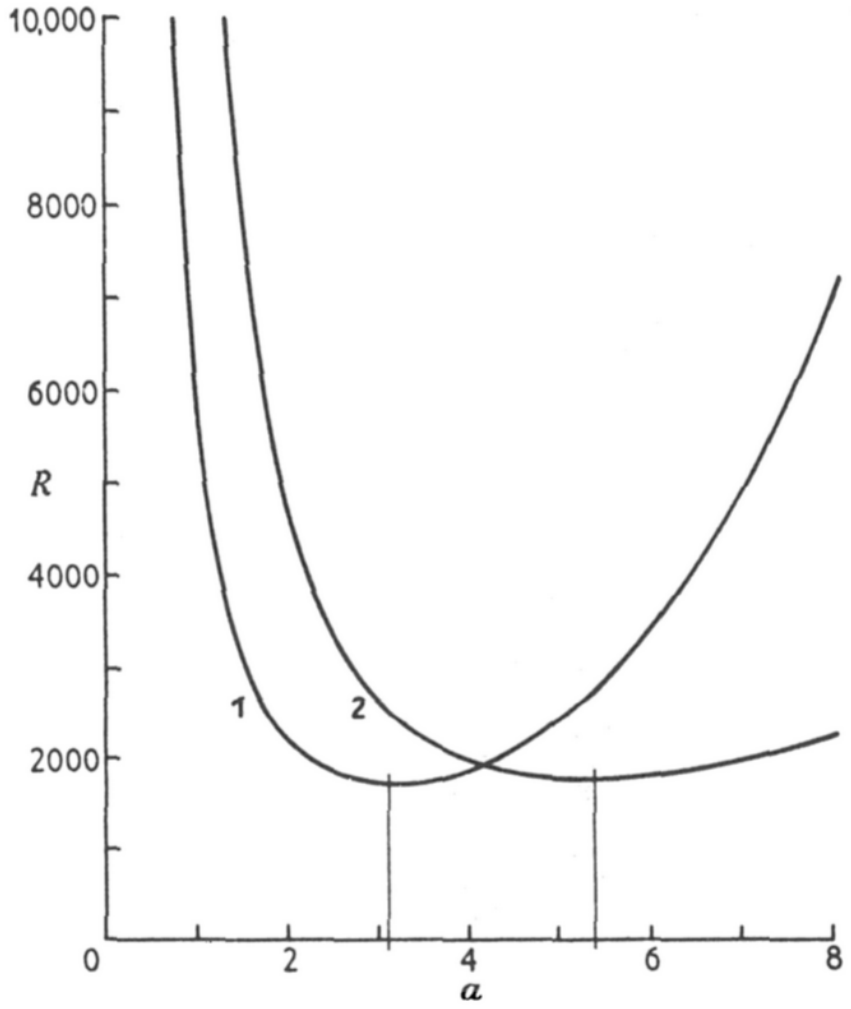

FIGURE D.10.: Figure 2 of Chandrasekhar [1970]. Original caption:
The Rayleigh numbers at which instability sets in for disturbances of different wave numbers a for the first even (curve labelled 1) and the first odd (curve labeled 2) mode.

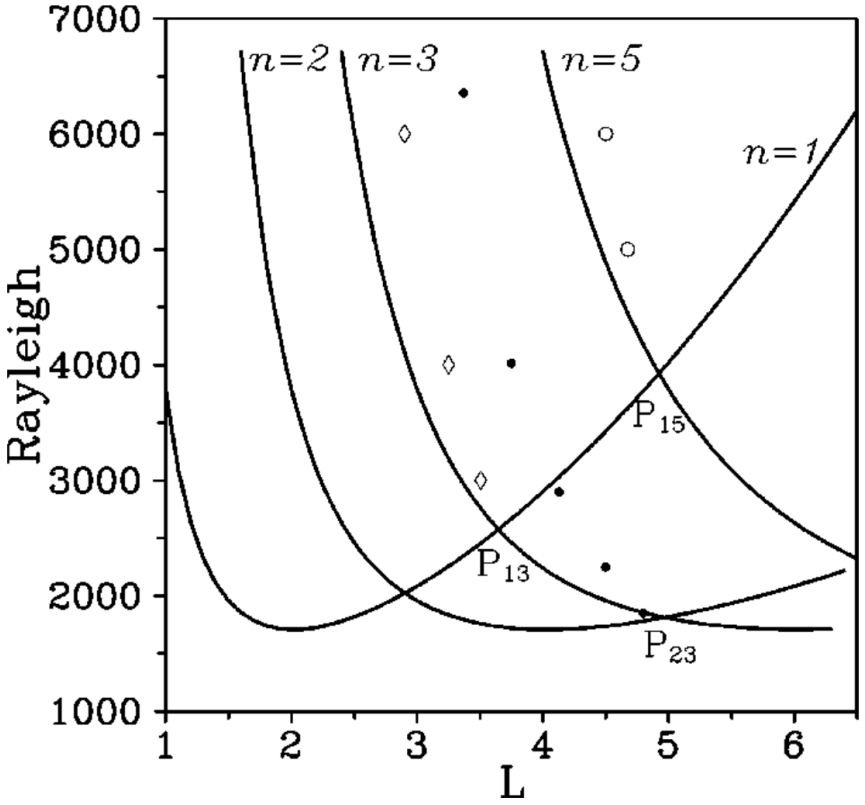

FIGURE D.11.: Figure 6 of Prat et al. [1998]. Original caption:
The marginal stability curves for $n = 1$, $n = 2$, $n = 3$, and $n = 5$ modes (solid lines). Diamonds and open circles denote the points where $n = 3$ and $n = 5$ solutions bifurcate to families of hybrid solutions; full circles denote the points where the $n = 3$ state acquires stability.

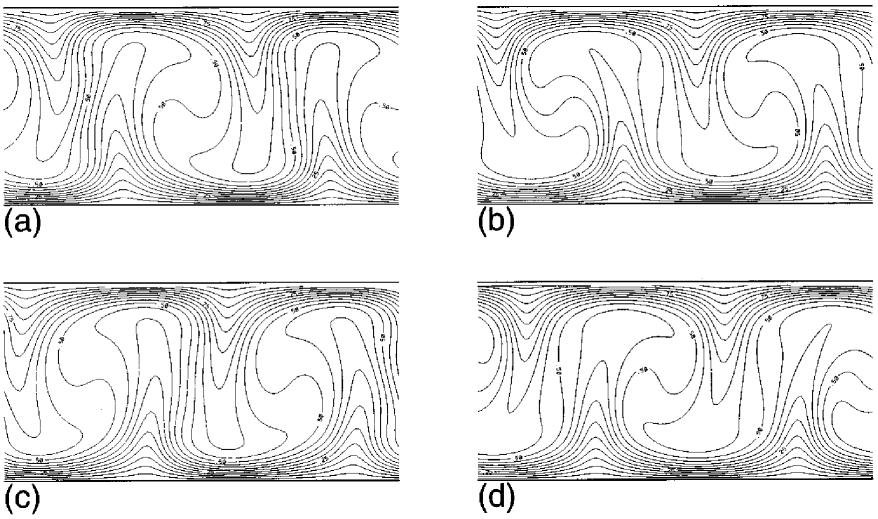

FIGURE D.12.: Figure 8 of Shan [1997]. Original caption:
Isotherms in 2D simulation. The simulation was started from the static conductive state with $R = 50000$. The system evolves into an oscillatory state. The isotherms are taken at (a) the beginning, (b) one-quarter, (c) half, and (d) three-quarters of one oscillation period.

Diese Erklärungen sind in jedes Exemplar der Bachelor- bzw. Masterarbeit mit einzubinden.

Name:_____ Matr.Nr.:_____

Urheberrechtliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine als die angegebenen Quellen und Hilfsmittel verwendet habe.

Alle Stellen, die ich wörtlich oder sinngemäß aus anderen Werken entnommen habe, habe ich unter Angabe der Quellen als solche kenntlich gemacht.

Datum Unterschrift

Erklärung zur Veröffentlichung von Abschlussarbeiten

Die Abschlussarbeit wird zwei Jahre nach Studienabschluss dem Archiv der Universität Bremen zur dauerhaften Archivierung angeboten.

Archiviert werden:

1) Masterarbeiten mit lokalem oder regionalem Bezug sowie pro Studienfach und Studienjahr 10 % aller Abschlussarbeiten

2) Bachelorarbeiten des jeweils ersten und letzten Bachelorabschlusses pro Studienfach und Jahr.

❍ Ich bin damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

❍ Ich bin damit einverstanden, dass meine Abschlussarbeit nach 30 Jahren (gem. §7 Abs. 2 BremArchivG) im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

❍ Ich bin nicht damit einverstanden, dass meine Abschlussarbeit im Universitätsarchiv für wissenschaftliche Zwecke von Dritten eingesehen werden darf.

Datum Unterschrift

References

Anderson, J. D. (1995). *Computational fluid dynamics: the basics with applications*. New York: McGraw-Hill.

Benzi, R., Succi, S., & Vergassola, M. (1992, December). The lattice Boltzmann equation: theory and applications. *Physics Reports*, *222*(3), 145–197.

Bhatnagar, P. L., Gross, E. P., & Krook, M. (1954, May). A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical Review*, *94*(3), 511–525.

Boltzmann, L. (1872). Weitere Studien über das Wärmegleichgewicht unter Gasmolekülen. In *Wiener Berichte* (pp. 275–370). Wien.

Cercignani, C. (1988). *The Boltzmann equation and its applications*. New York, NY: Springer.

Cercignani, C. (1990). *Mathematical methods in kinetic theory*. New York: Plenum Pr.

Chandrasekhar, S. (1970). *Hydrodynamic and hydromagnetic stability*. Oxford: Clarendon P.

Chapman, S. (1916, January). On the law of distribution of molecular velocities, and on the theory of viscosity and thermal conduction, in a non-uniform simple monatomic gas. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *216*(538-548), 279–348.

Chapman, S. (1918, January). On the kinetic theory of a gas. part II: A composite monatomic gas: Diffusion, viscosity, and thermal conduction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *217*(549-560), 115–197.

Chen, J.-H., Pritchard, W. G., & Tavener, S. J. (1995, February). Bifurcation for flow past a cylinder between parallel planes. *Journal of Fluid Mechanics*, *284*, 23–41.

Chen, S., Martínez, D., & Mei, R. (1996, September). On boundary conditions in lattice Boltzmann methods. *Physics of Fluids (1994-present)*, *8*(9), 2527–2536.

Dellar, P. J. (2003, September). Incompressible limits of lattice Boltzmann equations using multiple relaxation times. *Journal of Computational Physics*, *190*(2), 351–370.

d'Humieres, D. (1992). Generalized lattice-Boltzmann equations. In *Rarefied gas dynamics: Theory and simulations* (pp. 450–458). Washington D.C.: American Institute of Aeronautics and Astronautics.

d'Humieres, D., Ginzburg, I., Krafczyk, M., Lallemand, P., & Luo, L.-S. (2002, March). Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *360*(1792), 437–451.

Enskog, D. (1917). *Kinetische Theorie der Vorgänge in mässig verdünnten Gasen.* Uppsala: Almquist & Wiksell.

Gershuni, G. Z., & Zhukhovitskii, E. M. (1976). *Convective stability of incompressible fluids.* Jerusalem; Springfield, Va.: I.P.S.T.

Ginzburg, I. (2005, November). Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation. *Advances in Water Resources*, *28*(11), 1171–1195.

Guo, Z., Shi, B., & Wang, N. (2000, November). Lattice BGK model for incompressible Navier–Stokes equation. *Journal of Computational Physics*, *165*(1), 288–306.

Guo, Z., Shi, B., & Zheng, C. (2002a). A coupled lattice BGK model for the Boussinesq equations. *International Journal for Numerical Methods in Fluids*, *39*(4), 325–342.

Guo, Z., Shi, B., & Zheng, C. (2002b, April). Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E*, *65*(4), 046308.

Guo, Z., & Shu, C. (2013). *Lattice Boltzmann method and its' applications in engineering.* World Scientific Publishing Company Incorporated.

Guo, Z.-L., Chu-Guang, Z., & Bao-Chang, S. (2002, April). Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method. *Chinese Physics*, *11*(4), 366.

Hänel, D. (2004). *Molekulare Gasdynamik: Einführung in die kinetische Theorie der Gase und Lattice-Boltzmann-Methoden.* Berlin: Springer.

He, X., Chen, S., & Doolen, G. D. (1998, October). A novel thermal model for the lattice Boltzmann method in incompressible limit. *Journal of Computational Physics*, *146*(1), 282–300.

He, X., & Luo, L.-S. (1997). Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Physical Review E*, *56*(6), 6811.

Kaminski, D., & Jensen, M. K. (2011). *Introduction to thermal and fluids engineering* (Auflage: Updated. ed.). Hoboken, N.J.: John Wiley & Sons.

Lallemand, P., & Luo, L.-S. (2000). Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, galilean invariance, and stability. *Physical Review E*, *61*(6), 6546.

Landau, L. D., & Lifshitz, E. M. (1987). *Fluid mechanics, second edition: Volume 6* (2edition ed.). Oxford: Butterworth-Heinemann.

Latt, J. (2007). *Hydrodynamic limit of lattice Boltzmann equations* (Unpublished doctoral dissertation). University of Geneva.

Marshall, J., & Plumb, R. A. (2007). *Atmosphere, ocean, and climate dynamics: An introductory text.* Amsterdam ; Burlington, MA: Academic Press.

Nan-Zhong, H., Neng-Chao, W., Bao-Chang, S., & Zhao-Li, G. (2004, January). A unified incompressible lattice BGK model and its application to three-dimensional lid-driven cavity flow. *Chinese Physics*, *13*(1), 40.

Olbers, D., Willebrand, J., & Eden, C. (2012). *Ocean dynamics* (Auflage: 2012 ed.). Heidelberg ; New York: Springer.

Prat, J., Massaguer, J. M., & Mercader, I. (1995). Large-scale flows and resonances in 2-D thermal convection. *Physics of Fluids*, *7*(1), 121.

Prat, J., Mercader, I., & Knobloch, E. (1998). Resonant mode interactions in Rayleigh-Bénard convection. *Physical Review E*, *58*(3), 3145.

Schwabl, F. (2006). *Statistische Mechanik mit 26 Tabellen und 186 Aufgaben.* Berlin [u.a.]: Springer.

Shan, X. (1997). Simulation of Rayleigh-Bénard convection using a lattice Boltzmann method. *Physical Review E*, *55*(3), 2780.

Shi, Y., Zhao, T., & Guo, Z. (2004, December). Thermal lattice Bhatnagar-Gross-Krook model for flows with viscous heat dissipation in the incompressible limit. *Physical Review E*, *70*(6).

Skordos, P. A. (1993). Initial and boundary conditions for the lattice Boltzmann method. *Physical Review E*, *48*(6), 4823.

Stroustrup, B. (2013). *The C++ programming language* (4th ed.). Addison-Wesley Professional.

Succi, S. (2001). *The lattice Boltzmann equation: For fluid dynamics and beyond*. Oxford University Press.

Torrence, C., & Compo, G. P. (1998, January). A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, *79*(1), 61–78.

Tritton, D. J. (1988). *Physical fluid dynamics*. Oxford [England]; New York: Clarendon Press ; Oxford University Press.

Wang, J., Wang, D., Lallemand, P., & Luo, L.-S. (2013, January). Lattice Boltzmann simulations of thermal convective flows in two dimensions. *Computers & Mathematics with Applications*, *65*(2), 262–286.

Watari, M. (2012, February). What is the small parameter ϵ in the Chapman-Enskog expansion of the lattice Boltzmann method? *Journal of Fluids Engineering*, *134*(1), 011401–011401.

Williamson, C. H. K. (1996). Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, *28*(1), 477–539.

Wolf-Gladrow, D. A. (2000). *Lattice-gas cellular automata and lattice Boltzmann models: An introduction*. Springer Science & Business Media.

Ziegler, D. P. (1993, June). Boundary conditions for lattice Boltzmann simulations. *Journal of Statistical Physics*, *71*(5-6), 1171–1177.

Zou, Q., & He, X. (1997, June). On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids (1994-present)*, *9*(6), 1591–1598.

Zovatto, L., & Pedrizzetti, G. (2001). Flow about a circular cylinder between parallel walls. *Journal of Fluid Mechanics*, *440*(1), 1–25.