DOCTORAL THESIS

# TOWARDS LATTICE BOLTZMANN MODELS FOR CLIMATE SCIENCES

## The GeLB programming language with applications

**Dissertation zur Erlangung des**
**Doktorgrades der Naturwissenschaften**

**Dr. rer. nat.**

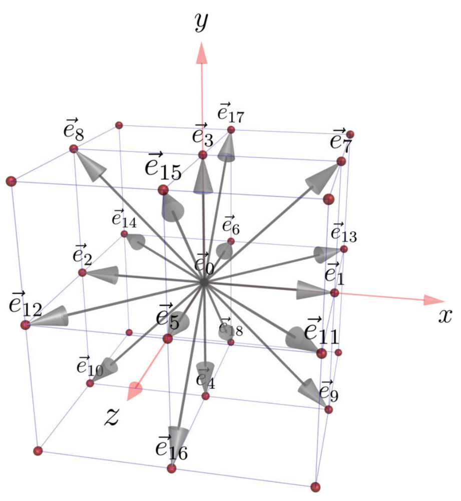
dem Fachbereich Physik/Elektrotechnik der

Universität Bremen

vorgelegt von

DRAGOS BOGDAN CHIRILA

aus Iași



Datum des Kolloquiums: 6. März 2018

1. *Gutachter:* Prof. Dr. Gerrit Lohmann
2. *Gutachter:* Prof. Dr. Dieter Wolf-Gladrow

Universität Bremen

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

**Towards Lattice Boltzmann models for climate sciences. The GeLB programming language with applications**

by **Dragos Bogdan Chirila**

## ABSTRACT

The complexity of *Earth system models* (ESMs) is continuously increasing – both *quantitatively* (higher spatio-temporal resolution for existing models) and *qualitatively* (accounting for additional processes). These trends are sustained by growing capabilities of computers and (equally important) by innovative algorithms. Better algorithms can lead to *more accurate* and/or *more efficient* numerical solutions. Efficiency attracted more attention during the last decade when, due to thermal limitations, the driving force behind increased computing performance has shifted from higher clock-frequencies (lower latencies) to more hardware parallelism (higher throughput). Not all numerical algorithms are suited for the new massively-parallel machines – some established approaches can reach plateaus in terms of performance-scalability, which motivates ongoing research to find alternatives that thrive on the new hardware. In this thesis the potential of the *lattice Boltzmann method* (LBM) is analyzed, as a promising alternative for modeling processes relevant to ESMs. During the last two decades, this relatively new approach was successfully applied to many flow problems in engineering (simulation of multi-phase and multi-component flows, melting processes, flows in porous media, and *direct numerical simulation* (DNS) of turbulence). At the core of any LBM algorithm is a simplified physical landscape inspired by the kinetic theory of gases, with "mesoscopic" particles which interact (collisions) and then propagate freely (streaming). This idealized dynamics (usually with local interactions) leads to algorithms which are particularly suited for parallel execution – a key property, which is also interesting for ESMs. However, the impact of LBM on *Earth system model*s was small so far, due to limitations of the early LBM algorithms.

The method deserves reconsideration, due to recent advances on improving its stability, a simplified implementation of accurate body-forces, and accurate simulation of thermal flows. This thesis adds two main contributions to this direction: (a) From a *computer science* (CS)/technical perspective, the new GeLB *domain-specific language* (DSL) is introduced, to facilitate testing and development of new LBM algorithms. By isolating many of the technical implementation side-issues away from the core physical algorithm, this new tool aims to counteract some of the "fragmentation" of the LBM research, by: (i) shortening the time to develop a *parallel simulation* from an *algorithm idea*, (ii) serving as a basis for objective comparisons of different physical algorithms, and by (iii) facilitating sharing of algorithms. (b) From a physical point of view, several flow-problems related to climate sciences are simulated, taking advantage of the recent progress in the LBM research literature. First, the *Rayleigh-Bénard* (RB) problem is simulated (in 2D and 3D configurations). The evolution of the flow in this problem is driven by buoyancy forces which can trigger convection (similar to convection in the atmosphere, or to the intermittent bursts of deep-reaching convection, which significantly influence the composition and circulation of oceanic water-masses). As a last application, simulation results are shown for the *wind-driven ocean circulation* (WDOC) of an idealized barotropic ocean, to which one of the more recent LBM algorithms is applied for the first time (first with an idealized geometry, then with a realistic global land-mask).

"Call it a clan, call it a network, call it a tribe, call it a family. Whatever you call it, whoever you are, you need one."

Jane Howard

## ACKNOWLEDGMENTS

This thesis summarizes a long journey, during which I was fortunate to meet extraordinary people, who helped me in countless ways. My deepest gratitude goes to Prof. Dr. Gerrit Lohmann (mein Doktorvater) for the inspiring discussions, contagious scientific curiosity, crucial insights and for his continuous support of this project. I also thank Prof. Dr. Dieter Wolf-Gladrow and Dr. Manfred Mudelsee (for their excellent suggestions and encouragement, which I should have used more). Also, near the end of this project, the help and encouragement of Dr. Vadym Aizinger were essential.

My work intersected with many scientific and technical disciplines, and I was fortunate to have within reach experts in many of these fields, who were eager to share their insights – especially Dr. Sergey Danilov, Dr. Vladimir Gryanik and Dr. Dirk Barbi. In addition, I have benefited from the excellent environment at the *Alfred-Wegener-Institut, Helmholtz-Zentrum für Polar- und Meeresforschung* (AWI), from the Paleo-climate Dynamics and Scientific Computing groups. Also at AWI, the very pleasant experience of co-supervising Sebastian Hinck taught me even more about teaching and mentoring than I passed to him about numerical methods and programming.

I would like to acknowledge the help of Prof. Dr. Arzhang Khalili and Dr. Bo Liu, who gave me a first exposure to LBM algorithms. For staying up-to-date with the more recent developments in this field, I am grateful to the work of many researchers, some of which I also had the privilege to meet in person – especially Prof. Dr. Li-Shi Luo, Prof. Dr. Maciej Matyka, Prof. Dr. Pietro Asinari, Prof. Dr. Martin Geier, Prof. Dr. Sauro Succi, Prof. Dr. Paul Dellar, Dr. Nicolas Delbosc, and Dr. Jonas Latt.

For making the simulations of the barotropic *wind-driven ocean circulation* (WDOC) as close to the situation on Earth as permitted by the (many) limitations of the model, the author would like to acknowledge support from members of the excellent climate sciences community at AWI, who were not mentioned above: Dr. Dmitry Sidorenko, Dr. Himansu Kesari Pradhan, Dr. Martin Losch, and Dr. Claudia Wekerle. Any remaining mistakes are my own.

From a personal point of view, I am grateful to my friends (from Bremerhaven, from my hometown Iași, or those scattered around the world), for enriching my life and for making the sunny days even sunnier (and the cloudy days less cloudy). Finally, I thank my family, for their unwavering support, patience, and love – despite the great distance, I will always feel you close to me.

# CONTENTS

1 INTRODUCTION 1

I BACKGROUND 5

2 THEORETICAL FOUNDATIONS 7

- 2.1 Molecular Dynamics (microscopic scale) . . . 7
- 2.2 Kinetic Theory ("mesoscopic" scale) . . . 10
- 2.3 Fluid Dynamics (macroscopic scale) . . . 13
  - 2.3.1 General form of macroscopic equations for fluids . . . 13
  - 2.3.2 Incompressible fluids and the Oberbeck-Boussinesq approximation . . . 18
- 2.4 The Rayleigh-Bénard (RB) problem . . . 19
  - 2.4.1 Governing equations for Rayleigh-Bénard (RB) problem . . . 20
  - 2.4.2 Specification of *boundary conditions* (BCs) and *initial conditions* (ICs) 20
  - 2.4.3 Problem reformulation in non-dimensional units . . . 22
  - 2.4.4 Important theoretical and experimental results for the RB problem 25
- 2.5 Overview of the physics of turbulence . . . 39
  - 2.5.1 Defining turbulent flows . . . 39
  - 2.5.2 Important results from turbulence theory . . . 41
  - 2.5.3 Approaches for quantifying and predicting turbulent flows . . . 45
- 2.6 Flows at the planetary scale . . . 52
  - 2.6.1 Primitive-equations model of the ocean . . . 55
  - 2.6.2 Wind-driven ocean circulation (WDOC) . . . 55

3 COMPUTATIONAL FLUID DYNAMICS (CFD) AND LATTICE BOLTZMANN METHODS (LBMs) 61

- 3.1 Workflow for Numerical Solution of Differential Equations (DEs) . . . 61
- 3.2 Traditional methods in Computational Fluid Dynamics (CFD) . . . 62
  - 3.2.1 Finite difference (FD) methods . . . 62
  - 3.2.2 Finite volume (FV) methods . . . 63
  - 3.2.3 Finite element (FE) methods . . . 63
  - 3.2.4 Spectral methods . . . 63
- 3.3 A "bottom-up" approach to modeling . . . 64
- 3.4 Precursors of LBM . . . 65
  - 3.4.1 Cellular Automata (CA) . . . 65
  - 3.4.2 Lattice Gas Cellular Automata (LGCA) . . . 67
- 3.5 How LBMs work . . . 67

II CORE 69

4 GELB– A TOOL FOR LATTICE BOLTZMANN MODELING 71

- 4.1 Computers and programming languages . . . 71
- 4.2 Previous related work . . . 73

4.3 How GeLB works and what it does . . . 74
4.4 Elements of a GeLB program (GD-file) . . . 76
4.4.1 Comments . . . 76
4.4.2 Declaration of GeLB version . . . 77
4.4.3 `CONFIGURATION` block . . . 77
4.4.4 `LATTICE_PROPERTIES` block . . . 79
4.4.5 `LATTICES` block . . . 79
4.4.6 `FUNCTION` block(s) . . . 80
4.4.7 `NODE_CATEGORIES` block . . . 81
4.4.8 `GEOMETRY` block . . . 81
4.4.9 Data-types supported in the GD-language . . . 82
4.4.10 Non-scalar variables and constants . . . 83
4.5 Automatic generation of parallel code . . . 83
5 APPLICATIONS 85
5.1 Rayleigh-Bénard (RB) convection (2D) . . . 85
5.1.1 Model setup for the 2D case . . . 86
5.1.2 Discretization of the dimensionless equations . . . 87
5.1.3 Numerical method . . . 89
5.1.4 Simulation results . . . 97
5.2 Rayleigh-Bénard (RB) convection (3D) . . . 101
5.2.1 Model setup for the 3D case . . . 101
5.2.2 Discretization of the dimensionless equation . . . 102
5.2.3 Numerical method . . . 102
5.2.4 Simulation results . . . 107
5.3 Wind-driven ocean circulation (2D) . . . 116
5.3.1 Idealized (square) geometry simulation . . . 116
5.3.2 Realistic geometry simulation . . . 120

III SYNTHESIS 127
6 DISCUSSION AND CONCLUSIONS 129
6.1 GeLB– a tool for Lattice Boltzmann modeling . . . 129
6.2 Rayleigh-Bénard (RB) convection . . . 129
6.3 Wind-driven ocean circulation (WDOC) . . . 130
7 OUTLOOK 133
7.1 Possible Improvements to *Generic lattice Boltzmann framework* (GeLB) . . . 133
7.2 LBM work . . . 137

BIBLIOGRAPHY 141

# LIST OF FIGURES

Figure 1 Physical theories for describing ocean flows, arranged from small space- and time-scales (bottom) to large scales (top). Arrows $(A)-(D)$ denote the connections between the theories, along with the references where the derivations are presented. 8

Figure 2 Sketch of the *parameter-space* defined by the control-parameters of a fluid system. . . . . . . . . . . . . . . . . . . . . . . . . . . 27

Figure 3 Plot of the critical Rayleigh number as a function of the wavelength of the perturbations. The dotted lines represent the marginal-stability curves for the even (blue) and odd (red) modes, respectively. The numbered labels are used to mark the different regions of the parameter-space. . . . . . . . . . . 34

Figure 4 Illustration of the flow directly after first transition (source: Lappa (2010)). Original caption: *"Schematic diagram of rolls in Rayleigh-Bénard convection: for a laterally unlimited domain, the fluid motion is regular and organized as a set of horizontal rolls (arrows indicate the direction of fluid flow; the wavelength of the pattern is approximately equal to twice the layer height* $d$*)"* . . . . 35

Figure 5 The Busse balloon (source: Busse (1978)). Original caption: *"Region of stable convection rolls in the three-dimensional* $R-P-\alpha$ *space. The thick curves represent computed stability boundaries for the oscillatory (OS), the skewed varicose (SV), the cross-roll (CR), the knot (KN), and the zig-zag (ZZ) instabilities. The other curves represent approximate interpolations from results of Busse and Clever (1979). The stability boundary for* $P = 300$ *actually represents the computations of Busse (1967) for* $P = \infty$ *which are expected to give a good approximation for* $P = 300$.*"* Note: the parameters $P$, $R$ and $\alpha$ correspond to Pr, Ra and $k$ respectively in this thesis. . 36

Figure 6 Different flow-regimes for the RB problem (source: Busse (1978)). Original caption: *"Transitions in convection as a function of Rayleigh and Prandtl numbers after Krishnamurti (1973) and others. The curves indicate the onset of steady rolls (I), three-dimensional convection pattern (II), time-dependent convection (III) in isolated spots (III(a)) and uniformly throughout the layer (III(b)), and turbulent convection (IV)."* The original figure was annotated with the red dashed lines, indicating the approximate corresponding positions of water (Pr $= 7$) and air (Pr $= 0.71$). . . . . . . . . . 38

Figure 7 Schematic of the energy cascade for high-Re flows (adapted after Fig. 6.2 of Pope (2000)). Note the difference in eddy-sizes and isotropy. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

Figure 8 Sketches of the atmospheric circulation (source: *Encyclopædia Britannica Online*, article “Atmospheric circulation”, accessed 15 December 2017, at https://www.britannica.com/science/atmospheric-circulation). . . . . . . . . . . . . . . . . . . . . . 53
Figure 9 Estimation of the zonal mean eastward wind stress (source: Hellerman and Rosenstein (1983)). Original caption: *“Zonal means of annual $\tau^x$ (dyn cm$^{-2}$) from wind rose data of the Marine Climatic Atlases and Pilot Chart data of the U.S. Navy Hydrographic Office (dashed line) and zonal means of annual $\tau^x$ from TDF-11 data (solid line).”* . . . . . . . . . . . . . . . . . . . . . . . . 53
Figure 10 Sketch of the oceanic surface currents (source: *National Aeronautics and Space Administration* (NASA) / US Navy Oceanographic Office, article “General Characteristics of the World’s Oceans: 4 Ocean Currents”, accessed 15 December 2017, at https://icp.giss.nasa.gov/research/ppa/1997/oceanchars/currents.html). 54
Figure 11 Sketch of the thermohaline circulation of the oceans (source: NASA *Earth Observatory Online*, article “Explaining Rapid Climate Change: Tales from the Ice”, accessed 15 December 2017, at https://earthobservatory.nasa.gov/Features/Paleoclimatology_Evidence/paleoclimatology_evidence_2.php). . . . . . . . . 54
Figure 12 Illustration of the western-boundary current intensification in the oceans (source: NASA *Ocean Motion*, article “Wind driven surface currents: western boundary currents”, accessed 15 December 2017, at http://oceanmotion.org/html/background/western-boundary-currents.htm). . . . . . . . . . . . . . . . . . 56
Figure 13 Top-down versus bottom-up approaches for numerical simulation of fluids. . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
Figure 14 Snapshots of the *superstable line puffer* pattern: $t = 0$ (upper-left), $t = 100$ (upper-right), $t = 150$ (lower-left), and $t = 200$ (lower-right). The figures were produced using the Golly software (Trevorrow and Rokicki, 2009). . . . . . . . . . . . . . . . 66
Figure 15 Flowchart of typical execution flow for *lattice Boltzmann method* (LBM) applications. . . . . . . . . . . . . . . . . . . . . . . . . . . 68
Figure 16 Illustration of the components of the GeLB framework, along with their interactions with compilers for specific *general-purpose languages* (GPLs). . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
Figure 17 Illustration of the typical “dynamics” in *lattice Boltzmann* (LB) algorithms. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75

Figure 18 Illustration of the effects of the three types of functions supported in the *GeLB description* (GD) language: *initializers* (=write-only) are used for imposing ICs onto the domain, *dynamics* (=read & write) are for specifying the core numerical algorithms explicitly, and *gauges* (=read-only) are used for evaluating quantities (e.g. macroscopic moments) which depend on the *distribution functions* (DFs) in the lattice. . . . . . . . . . . . 80

Figure 19 Sketch of geometry for the *Rayleigh-Bénard* (RB) simulations in 2D. For the present simulations, the aspect-ratio was fixed to $\gamma = \frac{2\pi}{k_{cr,1}} \approx 2.0158$. . . . . . . . . . . . . . . . . . . . . . . . . . 86

Figure 20 D2Q5 lattice (for modeling advection-diffusion of temperature) 89

Figure 21 D2Q9 lattice (for modeling isothermal hydrodynamic fields) . 89

Figure 22 The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{max}$ for different values of the Rayleigh number (left) and zoomed-in plot of the same data, for the linear regime (right). All simulations were performed on a $83 \times 41$ grid, with $Ma = 0.1$. . . . . . . . 97

Figure 23 Dimensionless temperature (left) and streamfunction (right) plots for the $Ra = 1730$ simulation (final state, after $t_{Sim}^{(d)} = 450$ diffusive time-scales). . . . . . . . . . . . . . . . . . . . . . . . 98

Figure 24 The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{max}$ (left) and of the total kinetic energy (right) for the $Ra_{cr,2}$ 2D simulations. 99

Figure 25 Spectrogram for the 2D simulation, for $Ra = 6500$. Spectrograms for values of $Ra = 6000$ and of up to $Ra = 9000$ show similar behavior towards the end of the time domain. . . . . . 100

Figure 26 Sketch of geometry for the *Rayleigh-Bénard* (RB) simulations in 3D. For the present simulations, the aspect-ratio was fixed to $\gamma = \frac{2\pi}{k_{cr,1}} \approx 2.0158$. . . . . . . . . . . . . . . . . . . . . . . . . . 101

Figure 27 D3Q7 lattice (for modeling advection-diffusion of temperature) 102

Figure 28 D3Q19 lattice (for modeling isothermal hydrodynamic fields) 102

Figure 29 The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{max}$ for different values of the Rayleigh number (left) and zoomed-in plot of the same data, for the linear regime (right). All simulations were performed on a $83 \times 83 \times 41$ grid, with $Ma = 0.1$. . . . . 107

Figure 30 Velocity vectors and temperature distribution after $t^{(d)} = 25$, for the $Ra = 1730$ experiment with the 3D model. . . . . . . . 108

Figure 31 Velocity vectors and temperature distribution after $t^{(d)} = 25$, for a $Ra = 6500$ experiment with the 3D model, and low aspect-ratio (*unphysical* behavior). . . . . . . . . . . . . . . . . . . . . . . . . 109

Figure 32 The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{max}$ (left) and of the total kinetic energy (right) for the $Ra_{cr,2}$ 3D simulations. 110

Figure 33 Flow-fields for the $Ra = 6000$ simulation (left) and for the $Ra = 6500$ simulation (right), at $t^{(d)} = 1$. . . . . . . . . . . . . . . . . 110

Figure 34 Flow-fields for the Ra $= 6000$ simulation (left) and for the Ra $= 6500$ simulation (right), at $t^{(d)} = 2$. . . . . . . . . . . . . . . . . . 110

Figure 35 Flow-fields for the Ra $= 6000$ simulation (left) and for the Ra $= 6500$ simulation (right), at $t^{(d)} = 2.5$. . . . . . . . . . . . . . . . . 111

Figure 36 Flow-fields for the Ra $= 6000$ simulation (left) and for the Ra $= 6500$ simulation (right), at $t^{(d)} = 3$. . . . . . . . . . . . . . . . . . 111

Figure 37 Flow-fields for the Ra $= 6000$ simulation (left) and for the Ra $= 6500$ simulation (right), at $t^{(d)} = 16$. . . . . . . . . . . . . . . . . . 111

Figure 38 Flow-fields for the Ra $= 6000$ simulation (left) and for the Ra $= 6500$ simulation (right), at $t^{(d)} = 18$. . . . . . . . . . . . . . . . . . 112

Figure 39 Spectrograms of the 3D simulations for testing the behavior in the vicinity of $\text{Ra}_{cr,2}$: (left) Ra $= 6000$ and (right) Ra $= 6500$. Both figures correspond to the pressure time-series sampled at the point $(^{N_x}/_4, ^{3N_y}/_4, ^{N_z}/_4)$. . . . . . . . . . . . . . . . . . . . 112

Figure 40 Snapshots of the traveling-wave oscillating flow-field for the Ra $= 6500$ simulation. From left-to-right, top-to-bottom: $t^{(d)} = 17.00$, $t^{(d)} = 17.38$, $t^{(d)} = 17.50$, $t^{(d)} = 17.63$, $t^{(d)} = 17.75$, and $t^{(d)} = 17.88$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113

Figure 41 Portion of the infinite space-filling tiling that our model simulates for Ra $= 6000$ (top view). . . . . . . . . . . . . . . . . . . . . 114

Figure 42 The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{\max}$ (left) and of the total kinetic energy (right) for the Ra $= 10^6$ 3D simulation. 114

Figure 43 Snapshots showing the evolution of the flow-field towards (and into) the turbulent-convection regime for the Ra $= 10^6$ simulation. From left-to-right, top-to-bottom:$t^{(d)} = 0.0302$, $t^{(d)} = 0.0327$, $t^{(d)} = 0.0352$, $t^{(d)} = 0.2513$, $t^{(d)} = 0.5025$, and $t^{(d)} = 1.0000$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115

Figure 44 Sketch of geometry for the *wind-driven ocean circulation* (WDOC) simulations in 2D. . . . . . . . . . . . . . . . . . . . . . . . . 117

Figure 45 (top) Total kinetic energy (TKE) versus time, for a duration of 15 weeks since the onset of the wind-forcing. (bottom) Visualization of the streamlines at the end of the simulation, for the square-domain *wind-driven ocean circulation* (WDOC) setup. . . 119

Figure 46 Examples of land-masks produced by the coarsening-algorithm: (a) original (`ETOP02v2`) 2-minute grid, (b) coarsened 20-minue grid, (c) coarsened $2^\circ$ grid, and (d) coarsened $10^\circ$ grid. The blue rectangle indicates the resolution which was used for calculating the results presented later. . . . . . . . . . . . . . . . . . 121

Figure 47 Visualization of the streamlines after 15 weeks of integration for the barotropic model of the wind-driven circulation, with a realistic land-mask. The color-bar indicates the velocity-magnitude. *NOTE: This result is not meant to represent a realistic ocean model, due to the limitations of the equations used (see discussion in the text).* 122

Figure 48 Visualization of the streamlines after 15 weeks of integration for the barotropic model of the wind-driven circulation, with a realistic land-mask. The color-bar indicates the velocity-magnitude, while the vectors indicate the local direction of the flows. *NOTE: This result is not meant to represent a realistic ocean model, due to the limitations of the equations used (see discussion in the text).* . . 123

## LIST OF TABLES

Table 1 Wavenumber ($k$) and Rayleigh number ($Ra_{cr}$) for the first even and odd unstable modes in the *Rayleigh-Bénard* (RB) problem. The highlighted values indicate the minimum critical values for the even and odd modes. . . . . . . . . . . . . . . . . . . . . . . . 35
Table 2 Simulation parameters for the 2D numerical experiments for determining $Ra_{cr,1}$. The duration of the simulations ($t_{Sim}$) is measured according to the *diffusive* time-scale. . . . . . . . . . 97
Table 3 Rates of growth/decay of $\log\left(\|u_y^{(d)}(t^{(d)})\|_{max}\right)$ for each numerical experiment. . . . . . . . . . . . . . . . . . . . . . . . . . 98
Table 4 Simulation parameters for the 2D numerical experiments for studying the behavior near $Ra_{cr,2}$. The duration of the simulations ($t_{Sim}$) is still measured according to the *diffusive* time-scale, for consistency. However, note that this duration corresponds to a much larger time in the (more relevant here) *eddy turnover* time-scale, of $t^{\mathcal{E}}_{Sim} \approx 1175$ for simulation $\mathsf{E}_{RB\text{-}2D}$ and of $t^{\mathcal{E}}_{Sim} \approx 1223$ for simulation $\mathsf{F}_{RB\text{-}2D}$. . . . . . . . . . . . . . . . 99
Table 5 Simulation parameters for the 3D numerical experiments for determining $Ra_{cr,1}$. The duration of the simulations ($t_{Sim}$) is measured according to the *diffusive* time-scale. . . . . . . . . . 107
Table 6 Rates of growth/decay of $\log\left(\|u_y^{(d)}(t^{(d)})\|_{max}\right)$ for each numerical experiment. . . . . . . . . . . . . . . . . . . . . . . . . . 108
Table 7 Simulation parameters for the 3D numerical experiments for studying the behavior near $Ra_{cr,2}$. The duration of the simulations ($t_{Sim}$) is still measured according to the *diffusive* time-scale, for consistency. However, note that this duration corresponds to a much larger time in the (more relevant here) *eddy turnover* time-scale, of $t^{\mathcal{E}}_{Sim} \approx 1175$ for simulation $\mathsf{E}_{RB\text{-}3D}$ and of $t^{\mathcal{E}}_{Sim} \approx 1223$ for simulation $\mathsf{F}_{RB\text{-}3D}$. . . . . . . . . . . . . . . . 109

## LISTINGS

Listing 1 Example `configuration` block in GeLB programs . . . . . . . . . 77
Listing 2 Example `lattice_properties` block in GeLB programs . . . . 79
Listing 3 Example `lattices` block in GeLB programs . . . . . . . . . . . 80
Listing 4 Example `dynamic` block in GeLB programs . . . . . . . . . . . . . 81

| | | |
|---|---|---|
| Listing 5 | Example `initializer` block in GeLB programs | 81 |
| Listing 6 | Example `gauge` block in GeLB programs | 81 |
| Listing 7 | Example `node_categories` block in GeLB programs | 81 |
| Listing 8 | Example `geometry` block in GeLB programs | 82 |
| Listing 9 | Example `geometry` block in GeLB programs with *complex* geometries | 82 |

## ACRONYMS

| | |
|---|---|
| **ABB** | *anti-bounce-back* |
| **ACC** | *Antarctic Circumpolar Current* |
| **AST** | *abstract syntax tree* |
| **AWI** | *Alfred-Wegener-Institut, Helmholtz-Zentrum für Polar- und Meeresforschung* http://www.awi.de |
| **BBGKY** | *Bogoliubov, Born, Green, Kirkwood, and Yvon* |
| **BB** | *bounce-back* |
| **BC** | *boundary condition* |
| **BGK** | *Bhatnagar-Gross-Krook* |
| **CA** | *cellular automata* |
| **CFD** | *computational fluid dynamics* |
| **CFL** | *Courant-Friedrichs-Levy* |
| **CI** | *confidence interval* |
| **CPU** | *central processing unit* |
| **CS** | *computer science* |
| **DE** | *differential equation* |
| **DF** | *distribution function* |
| **DNS** | *direct numerical simulation* |
| **DSL** | *domain-specific language* |
| **EBNF** | *Extended Backus-Naur-Form* |

| | |
|---|---|
| **EOS** | *equation of state* |
| **ESM** | *Earth system model* |
| **ESS** | *Earth system science* |
| **FD** | *finite differences* |
| **FE** | *finite elements* |
| **FLOP** | *floating-point operations per second* |
| **FPGA** | *field-programmable gate array* |
| **FV** | *finite volumes* |
| **GCM** | *general circulation model* |
| **GD** | *GeLB description* |
| **GeLB** | *Generic lattice Boltzmann framework* |
| **GoL** | *game of life* |
| **GPL** | *general-purpose language* |
| **GPU** | *graphics processing unit* |
| **HPC** | *high performance computing* |
| **IC** | *initial condition* |
| **IDE** | *integrated development environment* |
| **I/O** | *input/output* |
| **IR** | *intermediate representation* |
| **ISR** | *ideal straight rolls* |
| **K-O** | *Kolmogorov-Obukhov* |
| **LBM** | *lattice Boltzmann method* |
| **LB** | *lattice Boltzmann* |
| **LES** | *large-eddy simulation* |
| **LGCA** | *lattice gas cellular automata* |
| **LHS** | *left-hand side* |
| **LLVM** | *LLVM compiler infrastructure project* |

| | |
|---|---|
| **MDF** | *multiple distribution function* |
| **MPI** | *Message Passing Interface* |
| **MRT** | *multiple-relaxation-times* |
| **NASA** | *National Aeronautics and Space Administration* https://www.nasa.gov |
| **netCDF** | *NETwork Common Data Format* http://www.unidata.ucar.edu |
| **NSE** | *Navier-Stokes equations* |
| **ODE** | *ordinary differential equation* |
| **OpenMP** | *Open MultiProcessing* |
| **OpenCL** | *Open Computing Language* |
| **OpenGL** | *Open Graphics Language* |
| **PDE** | *partial differential equation* |
| **RANS** | *Reynolds-averaged Navier-Stokes* |
| **RA** | *Reynolds averaging* |
| **RB** | *Rayleigh-Bénard* |
| **RHS** | *right-hand side* |
| **r.m.s.** | *root-mean square* |
| **RSM** | *Reynolds-stress model* |
| **SGS** | *subgrid scale* |
| **SIC** | *sensitivity to initial conditions* |
| **SRT** | *single-relaxation-time* |
| **THC** | *thermohaline circulation* |
| **TRT** | *two-relaxation-times* |
| **WDOC** | *wind-driven ocean circulation* |

# 1 INTRODUCTION

"Research is formalized curiosity. It is poking and prying with a purpose"

Zora Neale Hurston

Fluids (gases or liquids) represent one of the most common states of matter in the universe. In order of increasing spatial scales, they are encountered in nanofluidic devices, the interior of living cells, respiratory and circulatory systems of all creatures (Kleinstreuer and Zhang, 2010), in technological devices, household appliances, geophysical phenomena (oceans, atmospheres, and planetary interiors), and even in astrophysics (stellar dynamics, gas giants, or the magnetospheres of planets such as our own). Therefore, in order to understand these systems (especially when considering time-dependent phenomena), we need to consider at least to some degree the relevant fluids (which often dominate the physics).

As for any other natural phenomenon, there are two approaches that can be used to study fluids. First, when the governing equations are not well-known[1], we have to rely on *experimental observations* of the flow (see e.g. Wallace and Vukoslavčević, 2010), where the ICs and BCs also need to be documented carefully. Using these observations (and pre-existing knowledge such as conservation laws which are expected to hold and analogies with similar systems), we try to construct *mathematical models*, which summarize the observations. Reaching this level of description is always preferrable, because it is more general – in addition to explaining existing observations, any useful theory also yields predictions of the system's behavior for other values of the parameters, suggesting additional experiments which can be conducted to test the theory. Also, theoretical models enable a better *understanding* of the underlying phenomena, since they describe the physical reality at higher levels of abstraction.

From the very beginnings of science, computation had a supporting role for both of these approaches – numerical computations are performed to analyze the experimental results, and analytic or numeric computations are used to evaluate the implications of the theoretical models. What has changed in the last century is the *scale* of these computations: while in earlier centuries the amount of experimental data was modest and could be analyzed by hand, we now have installations such as the CERN's Large Hadron Collider, which generates terabytes of raw data each second. Similarly, the early mathematical models could be solved analytically for simple cases (e.g. Newton's equations when applied to the motion of a single planet

1 This can still happen nowadays, when we consider systems with complex interactions between multiple scales.

around the Sun); however, this approach is no longer feasible for many systems that contemporany research focuses on (such as the evolution of the Earth's climate). Therefore, the increase of complexity of science (especially during the last century) was sustainable only because of advances in computing, which happened in parallel.

The advances in computing occured in two directions. First, computing *hardware* has become increasingly powerful – both in terms of theoretical *floating-point operations per second* (FLOP), and in terms of storage capacity (measured in *bits*). At the same time, the manufacturing and operation costs per FLOP and bit have dropped continuously, making large supercomputer installations feasible, and even allowing personal computers to have capabilities comparable to those of supercomputers from just a decade earlier. However impressive the advances in hardware have been, the computing capabilities available today would not be possible without the contribution of the second direction in computing research, which consists of developing and refining computational *algorithms*. Indeed, choosing a better algorithms can often (MacCormick, 2012) speed-up the solution of a problem by orders of magnitude more than a hardware upgrade can.

Research in *computational fluid dynamics* (CFD) was a major beneficiary of these advances in computing, for fundamental reasons rooted into the physics of the problem being studied – unlike the solid matter, one of the main sources of complexity when studying fluids in detail is the fact that most flows are *turbulent* (especially at the scales of technological devices or larger). Intuitively, we can describe this as a tendency of the flow to become *disordered* and *highly-sensitive to initial conditions*, such that even infinitesimal variations in ICs and BCs will eventually lead to significant differences between multiple realisations of an experiment – each realisation is therefore fundamentally *irreproducible* in detail. Fortunately, space- and time-statistics are often well-defined and stable – a property on which most theories of turbulence rely. In addition to disorder, another important characteristic of turbulence is that it *enhances mixing* (homogenization) of fluid properties.

Of all the flows encountered in daily live, the phenomena in the oceans and in the atmosphere are the most extreme examples of turbulent flows. Here, due to the curvature of the surface of the Earth, points at different latitudes also absorb different amounts of energy from the Sun. This meridional gradient of absorbed energy is the main driving mechanism behind the large-scale circulation of the atmosphere which, in turn, is the major force driving the motion of the oceans. Due to the complexity of these systems, analytic methods cannot be used for local quantitative predictions (e.g. for weather prediction). On the other hand, experimental research is limited to passive local observations since (unlike for engineering-scale fluid flow experiments) it is not possible to impose a specific set of boundary- and initial conditions on geophysical flows. Therefore, numerical simulations are of crucial importance for improving our understanding of weather and of climate in general (especially for establishing *causality* between different events).

CONTRIBUTIONS OF THIS THESIS As a contribution towards constructing better simulations in CFD (and, hopefully also in the *Earth system science* (ESS) in the near future), this thesis introduces GeLB, a new *domain-specific programming language* (DSL) which makes it easier to construct models based on the *lattice Boltzmann method* (LBM)[2]. As concrete applications, we then simulate (using LBM) three fluid problems. For the first two applications, the RB problem is simulated using two setups, for 2D and 3D respectively. Both setups exhibit convection currents which are sustained by buoyancy gradients; such simulations are relevant for improving our understanding of atmospheric convection, or of intermittent bursts of deep-reaching convection in the oceans (which are believed to play a key role for the composition and circulation of oceanic water-masses (despite being isolated to relatively small regions, such as the Labrador, Greenland, and northwestern Mediteranean seas). Three-dimensional simulations of such convective flows are particularly valuable, since they are less common in the literature (due to the much higher computational demands compared to the corresponding two-dimensional configurations). As a third (and last) application, the *wind-driven ocean circulation* (WDOC) of an idealized barotropic ocean (in 2D) is simulated, using some of the more recent techniques developed by the LBM research community (this being the first time these techniques are applied to the WDOC problem, to the knowledge of the author). First we show that this approach can simulate a simple (square) domain. More interestingly, we then demonstrate (with promising results) the ability of the model to simulate this problem for the case with complex geometries, corresponding to a realistic, high-resolution land-mask on Earth.

OUTLINE OF THE THESIS This thesis is structured as follows: in Chapter 2, some aspects of kinetic theory, fluid dynamics, ocean dynamics and turbulence theory are briefly reviewed; these provide the necessary context for understanding the LBM, and the applications considered in the thesis (with an expanded discussion of the theory of the RB problem, which is necessary for two of the applications discussed later). Chapter 3 contains an overview of numerical methods for *computational fluid dynamics* (CFD), and of *lattice Boltzmann method*s in general. In Chapter 4, the GeLB DSL is introduced, which represents the first new contribution of this thesis. Chapter 5 deals with the three concrete applications of GeLB:

1. The first application (Section 5.1) consisted of simulating the problem of convection between two parallel plates, when the lower plate is kept at a higher temperature. This configuration is also known as the *Rayleigh-Bénard* (RB) problem, which was originally studied experimentally (Bénard, 1900), and then analyzed theoretically (Rayleigh, 1916) – later literature named the problem after the authors of these two studies. Extensive subsequent studies analyzed this problem using multiple approaches (experimentally, analytically (Clever and Busse, 1974), and numerically (Moore and Weiss, 1973)), as summarized in the excellent literature review of Busse (1978).

2 Although mainly LBM applications are discussed here, the DSL is more general, and can also be used with other algorithms, as long as they are *explicit* in time and they are formulated on a *rectangular spatial grid*.

For this first application, the two-dimensional setup was used, which serves as a good test-case for model validation. Of particular interest in this setup is to determine the conditions under which the (initially stationary) flow transitions to a convective regime, under the influence of small initial perturbations.

2. The second application (Section 5.2) builds upon the first, by expanding the simulation to three spatial dimensions. Many numerical studies focus on 2D results, which are intrinsically of limited applicability to real-world scenarios (such as those typically encountered in ESS). Therefore, this thesis contributes results from the computationally-intensive simulations of the three-dimensional configuration of this problem, which permits capturing of important qualitative changes in the flow-regimes, especially for choices of parameters that lead to time-dependent three-dimensional flow and development of turbulent convection.

3. The third application (Section 5.3) was to simulate the wind-driven circulation in a barotropic ocean, using a 2D numerical algorithm based on LBM. This problem has actually been studied before with LBM (Wolf-Gladrow, 2000), but using the *single-relaxation-time* (SRT) variety of the method. The new contribution of this thesis is to use the *multiple-relaxation-times* (MRT) variety, which recently gained traction in the literature, due to its improved accuracy and stability. Also, we demonstrate the ability of the method to simulate flows within very complex geometries, by incorporating a high-resolution land-mask[3] of the Earth as BCs for simulations.

To synthesize the work, conclusions and discussion are presented in Chapter 6, and an overview of future work is given in Chapter 7.

3 For this last simulation (complex geometry), the author would like to emphasize that the goal was *not* to build an actual ocean model – this would require substantial additional algorithmic and model-calibration work, which is left for future generations.

Part I

# BACKGROUND

# 2 THEORETICAL FOUNDATIONS

"If I have seen further than others, it is by standing upon the shoulders of giants"

Isaac Newton

This chapter provides a brief overview of the physical theories which are most relevant for understanding LBM algorithms (introduced later, in Chapter 3), and our target physical systems (described in Chapter 5).

Our current knowledge of physics can be categorized into several broad theories, which provide different perspectives on the system of interest (generally focusing on different space- and time-scales). For the fluid systems relevant to this thesis, four levels of description are available to us (see Figure 1) – from the smallest to the largest scales, these are: (a) *molecular dynamics* (Newton/Hamilton equations), (b) *kinetic theory* (Boltzmann equation), (c) *continuum fluid mechanics* (Euler/Navier-Stokes equations), and (d) *geophysical fluid dynamics* (primitive equations).

These different descriptions are inter-related – since each theory has its own range of space- and time-scales for which it is applicable, it should be possible to derive the higher-level theories from the lower-level ones. An additional benefit of such analyses is that they allow us to derive constitutive equations for the empirical quantities appearing in the higher-level theories.

There are several approaches that can be used for the transformation between the different levels of physical descriptions. For example, Zwanzig (1960) and Mori (1965) proposed a general formalism, which can be used to construct coarse-grained models directly from the microscopic dynamics for Hamiltonian (i.e. non-dissipative) systems[1], by applying projection operators. Alternatively, within the framework of classical statistical mechanics and kinetic theory, the hydrodynamic equations can be obtained using the Chapman-Enskog theory (Chapman and Cowling, 1970). We will reference mostly the second approach, since it is more directly related to the rest of the thesis.

## 2.1 MOLECULAR DYNAMICS (MICROSCOPIC SCALE)

The theory of molecular dynamics provides the first level of description. The basic premise is to treat the molecules of the fluid as a set of particles with well-defined positions and momenta, which evolve according to the laws of classical mechanics.

1 Zwanzig (1980) later expanded the theory for non-Hamiltonian systems.

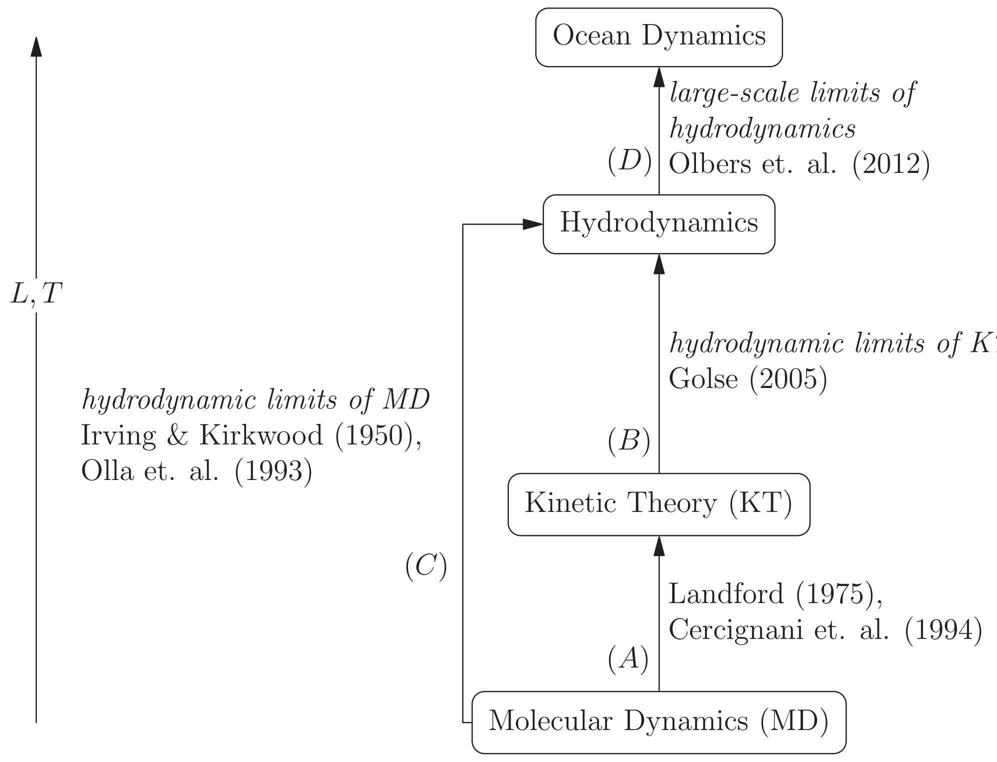

Ocean Dynamics
large-scale limits of hydrodynamics
Olbers et. al. (2012)
(D)
Hydrodynamics
hydrodynamic limits of KT
Golse (2005)
(B)
Kinetic Theory (KT)
Landford (1975), Cercignani et. al. (1994)
(A)
Molecular Dynamics (MD)
hydrodynamic limits of MD
Irving & Kirkwood (1950), Olla et. al. (1993)
(C)
L, T

Figure 1: Physical theories for describing ocean flows, arranged from small space- and time-scales (bottom) to large scales (top). Arrows $(A) - (D)$ denote the connections between the theories, along with the references where the derivations are presented.

This approximation is valid as long as the average distance between the molecules is much larger than the de Broglie wavelength of a molecule, i.e.

$$\left(\frac{V}{N}\right)^{1/3} \gg \frac{\hbar}{\sqrt{2m\kappa_B T}} \tag{1}$$

where $\hbar$ is the reduced Planck constant, $m$ is the mass of one fluid molecule, $\kappa_B$ is the Boltzmann constant, and $T$ is the average temperature of the fluid. This criterion is satisfied for ordinary fluids (such as air or water), which are relevant for this thesis.

The following notations are used in the discussion below:

- $\mathbf{r_i}$: the 3D− position of molecule
- $\mathbf{p_i}$: momentum of molecule
- $\mathbf{z_i} \equiv (\mathbf{r_i}, \mathbf{p_i})$: position of molecule ($\mu$ *phase-space* $\rightsquigarrow$ 6D); $d\mathbf{z_i} \equiv d\mathbf{r_i} d\mathbf{p_i}$
- $\mathbf{r}^N \equiv (\mathbf{r_1}, \dots, \mathbf{r_N})$, $\mathbf{p}^N \equiv (\mathbf{p_1}, \dots, \mathbf{p_N})$
- $\mathbf{z}^N \equiv (\mathbf{z_1}, \dots, \mathbf{z_N})$ position of the system ($\Gamma$ *phase-space* $\rightsquigarrow$ 6ND)
- $\dot{\xi} \equiv \frac{d\xi}{dt}$ time-derivative of quantity $\xi$

Assuming a system consisting of N identical molecules[2], we can write a Hamiltonian function for the system:

$$\mathcal{H}\left(\mathbf{z}^{\mathbf{N}}\right) = \sum_{i=1}^{N} \frac{|\mathbf{p_i}|^2}{2m} + \mathcal{V}_{\text{int}}\left(\mathbf{r}^{\mathbf{N}}\right) + \mathcal{V}_{\text{ext}}\left(\mathbf{r}^{\mathbf{N}}\right), \tag{2}$$

where $\mathcal{V}_{\text{int}}$ is the potential energy due to molecule-interactions, and $\mathcal{V}_{\text{ext}}$ is the potential energy due to external forces (which are assumed not to have explicit time-dependence – e.g. gravity).

The dynamics of the molecules is then governed by the Hamilton equations:

$$\begin{cases} \dot{\mathbf{r}}_{\mathbf{i}} = \dfrac{\partial \mathcal{H}}{\partial \mathbf{p_i}} & \text{(3a)} \\ \dot{\mathbf{p}}_{\mathbf{i}} = -\dfrac{\partial \mathcal{H}}{\partial \mathbf{r_i}} = \mathbf{X}_{\mathbf{i}}^{\text{ext}} + \sum_{j \neq i} \mathbf{X}_{\mathbf{i,j}}^{\text{int}}, & \text{(3b)} \end{cases}$$

where $\mathbf{X}_{\mathbf{i}}^{\text{ext}}$ is the sum of external forces acting on molecule "i", and $\mathbf{X}_{\mathbf{i,j}}^{\text{int}}$ is the force acting on "i" caused by interactions with molecule "j".

Although it may not be immediately obvious, eqs. (3) already contain empirical factors, because there is no universal expression for the inter-molecular force $\mathbf{X}_{\mathbf{i,j}}^{\text{int}}$ (and, therefore, for $\mathcal{V}_{\text{int}}$ in eq. (2)). Instead, the current practice (E, 2011) is to formulate system-specific potentials, by guessing the functional form of the potential, followed by calibration of parameters using experimental data and first-principle quantum-mechanical calculations.

Ignoring the complications due to the empirical inter-molecular potential, we could (in principle) use the framework of classical mechanics to predict the exact evolution of the $\mathbf{r_i}$s and $\mathbf{p_i}$s at any time (assuming we start from a well-defined initial state). This complete knowledge of the system is referred to as a *representative point* (or *microstate*) in Γ phase-space. It is postulated, in turn, that all intensive macroscopic properties of the system can be computed from this hypothetical knowledge of the microstate.

Unfortunately, even with state-of-the-art computers, this theory can at most be used to study processes at sub-micrometer and sub-second scales. While this is not a serious limitation for some fields (e.g. drug discovery in medicine (Durrant and McCammon, 2011)), it renders the approach impractical for the type of flows investigated in this thesis – higher-level theories are necessary in those cases.

As mentioned earlier, it is important for the theories at different levels to be consistent with each other; in our current context, it is interesting to note that various authors successfully linked the equations of molecular dynamics to the macroscopic, continuum equations of fluid dynamics (arrow (C) in Figure 1) – see (Irving and

2 We use the term "molecule" to denote the smallest entity in the fluid – in reality, this may be a single atom (e.g. Ar), or an actual molecule (e.g. $H_2O$). Also, for simplicity, we assume each "molecule" to be spherical.

Kirkwood, 1950) for a physical discussion or (Olla et al., 1993) for a more mathematical view.

It is interesting to note that the time-evolution predicted by eqs. (3) is completely reversible: if we could reverse the sign of all momenta ($\{\mathbf{p_i}\} \mapsto \{-\mathbf{p_i}\}$), the molecules would re-trace the exact trajectories from the past. This raises an apparent problem from the point of view of the higher-level descriptions of the system (kinetic and continuum), which are *irreversible* – this apparent paradox was, however, resolved within the framework of kinetic theory.

## 2.2 KINETIC THEORY ("MESOSCOPIC" SCALE)

The next level of description for studying fluids (see Figure 1) is kinetic theory, which usually focuses (for simplicity) on the so-called *ideal gas*, thought of as a large number of hard spheres, which collide elastically with each other[3]. Unlike the molecular dynamics approach, this is a probabilistic description, created out of necessity, because directly solving the equations of motion for numbers of molecules involved in macroscopic flows (of the order of Avogadro's number, $N_A = 6.023 \times 10^{23}$molecules/mol) is a hopeless task. Fortunately, most of the time we are also not interested in the details of each molecule's motion, but rather in the macroscopic effects, which we can measure with usual instruments (thermometers, pressure gauges, etc.) – the probabilistic approach provides a good theoretical basis for analyzing and describing such effects.

We denote by $\rho^N(\mathbf{z}, t)$ the N-particle probability density, which represents the probability[4] that the state of the system falls in the infinitesimal Γ-volume between $\mathbf{z}^\mathbf{N}$ and $\mathbf{z}^\mathbf{N} + d\mathbf{z}^\mathbf{N}$, at time t.

It can be shown (Huang, 1987; Kardar, 2007) that $\rho^N$ evolves such that the following equation (Liouville's equation) holds:

$$\frac{d\rho^N}{dt} \equiv \frac{\partial \rho^N}{\partial t} + \sum_{i=1}^{N} \left[ \frac{\partial \rho^N}{\partial \mathbf{r_i}} \dot{\mathbf{r}}_\mathbf{i} + \frac{\partial \rho^N}{\partial \mathbf{p_i}} \dot{\mathbf{p}}_\mathbf{i} \right] = 0 \qquad (4)$$

This equation resembles the incompressible continuity equation – this is why it is often said that the phase-space density $\rho^N(\mathbf{z}, t)$ behaves like an incompressible fluid.[5] Intuitively, this property is due to the fact that no phase-points can spontaneously appear or disappear as the system evolves in time.

Taking a closer look at the full phase-space density $\rho^N$, note that it specifies the probability for the 1st particle to have $(\mathbf{r}, \mathbf{p}) \in [\mathbf{r_1}, \mathbf{r_1} + d\mathbf{r_1}] \times [\mathbf{p_1}, \mathbf{p_1} + d\mathbf{p_1}]$ and, for

3 This is a good approximation for real gases at low densities. In addition, the asymptotic behavior of the model (hydrodynamic regime) should not depend on the details of the interaction between the constituent particles.

4 This probability is defined in terms of averages over an imaginary ensemble of identical replicas of the system – this is known as the *Gibbs formulation* of equilibrium statistical mechanics (contrast that with the concept of averaging over a phase-space trajectory, known as the *Boltzmann formulation*).

5 In fact, the terms within the square brackets cancel in pairs (*coordinates* ↔ *conjugate momenta*), so eq. (4) is a special case of incompressibility.

the 2nd particle to *simultaneously* have $(\mathbf{r}, \mathbf{p}) \in [\mathbf{r_2}, \mathbf{r_2} + d\mathbf{r_2}] \times [\mathbf{p_2}, \mathbf{p_2} + d\mathbf{p_2}]$, and so on. However, as far as most macroscopic processes are concerned, this is much more information than necessary – often we only want to know what is the probability of finding *any* particle in a region of the 6−dimensional phase-space ($\mu$−space). The later is known as the *single-particle distribution function*, and is defined by "collapsing" dimensions by integration over redundant portions of the original $\Gamma$−space:

$$f_1(\mathbf{z_1}, t) = \int \cdots \int \rho^N(\mathbf{z}, t) d\mathbf{z_2} \dots d\mathbf{z_N}. \tag{5}$$

In a similar manner, we can define the *two-particle distribution function*

$$f_2(\mathbf{z_1}, \mathbf{z_2}, t) = \int \cdots \int \rho^N(\mathbf{z}, t) d\mathbf{z_3} \dots d\mathbf{z_N}, \tag{6}$$

and, in general, the *s-particle reduced distribution function*:

$$f_s(\mathbf{z_1}, \dots, \mathbf{z_s}, t) = \int \cdots \int \rho^N(\mathbf{z}, t) d\mathbf{z_{s+1}} \dots d\mathbf{z_N} \tag{7}$$

By integrating the Liouville eq. (4), it is possible to derive a series of evolution equations for the reduced distribution functions – this is known as the *Bogoliubov, Born, Green, Kirkwood, and Yvon* (BBGKY) hierarchy ((Yvon, 1935), (Kirkwood, 1946), (Born and Green, 1947), (Bogoliubov, 1962)). Unfortunately, the resulting equations are coupled, as the evolution equation for $f_s$ depends on $f_{s+1}$. To obtain a solvable system, we need to assume that the hierarchy can be truncated after some order, while still resolving the physical phenomena of interest. Such an assumption is usually adopted in kinetic theory, leading to a simplification of the discussion by only focusing on the single-particle distribution function $f(\mathbf{r}, \mathbf{p}, t)$, which is defined such that:

$$f(\mathbf{r}, \mathbf{p}, t) d^3\mathbf{r} d^3\mathbf{p} \tag{8}$$

represents the number of molecules with positions within the $d^3\mathbf{r}$ volume around $\mathbf{r}$ and with momenta within the $d^3\mathbf{p}$ element around $\mathbf{p}$, at time $t$.

In the special case where the molecules are distributed uniformly in space (which is approximately true for fluids in thermodynamic equilibrium, as long as the representative volume contains enough molecules), we can write

$$f(\mathbf{r}, \mathbf{p}, t) d^3\mathbf{p} = \frac{N}{V}, \tag{9}$$

where $N$ is the number of molecules in a volume $V$.

The goal of kinetic theory is therefore to find the distribution function $f(\mathbf{r}, \mathbf{p}, t)$. As a first step, we need to specify the equation which governs the time-evolution of this function (assumed to be continuous). Ignoring (for now) collisions, during an infinitesimal interval of time ($t \to t + \delta t$), a molecule will move from the phase-space point $(\mathbf{r}, \mathbf{p})$ to the point $(\mathbf{r} + \mathbf{v}\delta t, \mathbf{p} + \mathbf{X}\delta t)$, due to the sum $\mathbf{X}$ of external forces acting on the molecule ($\mathbf{v} \equiv \mathbf{p}/m$ is the velocity of the molecule, and $m$ is its mass). The molecules initially located within the ("source") phase-space region $d^3\mathbf{r}d^3\mathbf{p}$ at time $t$ will occupy a new ("destination") phase-space region $d^3\mathbf{r}'d^3\mathbf{p}'$ at time $t + \delta t$. In other words,

$$f(\mathbf{r} + \mathbf{v}\delta t, \mathbf{p} + \mathbf{X}\delta t, t) d^3\mathbf{r}'d^3\mathbf{p}' = f(\mathbf{r}, \mathbf{v}, t) d^3\mathbf{r}d^3\mathbf{p} \tag{10}$$

It can be shown (Kardar, 2007) that *the volume enclosed by a surface moving through phase-space remains constant at all times*.[6] This important result, known as *Liouville's theorem*, allows us to simplify eq. (10) to

$$f(\mathbf{r} + \mathbf{v}\delta t, \mathbf{p} + \mathbf{X}\delta t, t) = f(\mathbf{r}, \mathbf{v}, t) \tag{11}$$

By Taylor-expanding the LHS of eq. (11) and subtracting the RHS we obtain, to $\mathcal{O}(\delta t)$, the equation of motion for the distribution function:

$$\left( \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} + \mathbf{X} \cdot \nabla_{\mathbf{p}} \right) f(\mathbf{r}, \mathbf{p}, t) = 0 \tag{12}$$

In order to also include inter-molecule collisions to the picture, we need to add a correction factor to eq. (12), such that

$$\left( \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} + \mathbf{X} \cdot \nabla_{\mathbf{p}} \right) f(\mathbf{r}, \mathbf{p}, t) \equiv \left( \frac{\partial f}{\partial t} \right)_{\text{coll.}} \tag{13}$$

The new term on the RHS represents the net contribution of collisions to the number of molecules which reach $d^3\mathbf{r}'d^3\mathbf{p}'$ at time $t + \delta t$. This consists of two terms:

- gains: due to collisions, some molecules will end-up in *destination* at time $t + \delta t$, even if they were *not* in *source* at time $t$
- losses: also due to collisions, some molecules will *not* end-up in *destination*, despite starting from *source*; since the phase-space elements are assumed to be small, the simplifying assumption is made that any molecule starting in *source* which undergoes a collision is disturbed so much that it does not reach the normal *destination*

To simplify the collision operator above, Boltzmann (1872) used several assumptions, of which the most famous is the assumption of molecular chaos ("Stosszahlansatz"), which states that the velocities of a pair of interacting particles are uncorrelated (the other two simplifying assumptions are that external forces can be ignored over the duration of the collisions, and that only two-particle collisions are important).

6 In a sense, phase-space behaves like an incompressible fluid. More exactly, this applies to the so-called *phase-space density function*, where an ensemble of realizations of the system are considered. However, the molecules of our gas can be interpreted as such different realizations, since we (for now) assume no collisions – this allows us to consider the *phase-space density function* as equivalent to the *molecular distribution function*.

For many applications, the collision operator can be simplified further, by assuming a simple relaxation towards the local Maxwellian distribution:

$$\left(\frac{\partial f}{\partial t}\right)_{\text{coll.}} \approx -\frac{1}{\tau}\left(f - f^{eq}\right), \tag{14}$$

This is known as the *Bhatnagar-Gross-Krook* (BGK) approximation, which is also used in many LBM models.

## 2.3 FLUID DYNAMICS (MACROSCOPIC SCALE)

To set the context for the first two of the applications discussed in this thesis (the RB convection studies from Sections 5.1 and 5.2), we continue the journey towards increasing space- and time-scales. At the next level beyond kinetic theory we have the continuum models of fluid dynamics (e.g. Euler or Navier-Stokes equations). These models are appropriate for describing flows encountered in everyday applications (vehicles, water pipes, pumps, etc.). We are concerned here mainly with thermal incompressible fluids, whose state is characterized by the velocity vector[7] field $\mathbf{v}(\mathbf{x}, t)$ and by three scalar fields: density $\rho(\mathbf{x}, t)$, pressure $p(\mathbf{x}, t)$, and temperature $T(\mathbf{x}, t)$.

The models can be derived from the conservation laws for mass, momentum, and energy, and the equation of state for the fluid. Since such models do not directly acknowledge the complex interactions between the molecules of the fluid, they incorporate the effects of those smaller scales through parameters and relations fitted experimentally (e.g. viscosity, heat conductivity, and the equation of state).

### 2.3.1 *General form of macroscopic equations for fluids*

In this subsection, we present the general form of the macroscopic equations for fluid flows. In later sections we present some simplifications of these equations, which are appropriate for the problems discussed in this thesis. These equations are written for an infinitesimal control volume, which remains fixed in space (Eulerian formulation).[8]

7 In general $\mathbf{v} \in \mathbb{R}^3$, although for some flows one (or even two) components can be discarded.

8 The alternative formulation, known as *Lagrangian approach*, is to write the equations from the point of view of a particle moving with the flow. The mapping between Eulerian (at a *fixed point*) variations of a quantity $\frac{\partial}{\partial t}$ and the Lagrangian variation $\frac{D}{Dt}$ is given by the *substantive (material) derivative* operator:

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla = \frac{\partial}{\partial t} + u_i \frac{\partial}{\partial x_i}$$

#### 2.3.1.1 *Mass equation (continuity equation)*

The most basic physical law is the conservation of mass – for our control volume, this means that the time-variation of mass inside the volume equals the mass exchanged with the rest of the fluid. In differential form, this reads:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{15}$$

where $\rho$ is the density of the fluid, $u_i$ is the velocity along axis $i \in \{1, 2, 3\}$, and the gradient operator is defined (in 3D) as $\nabla^T \equiv [\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3]$. Also, Einstein's summation notation is used, whereby repetition of an index implies summation over the range of that index.

#### 2.3.1.2 *Momentum equations*

The second physical law is the conservation of momentum. This is nothing else than an application of Newton's second law to our control volume, which says that the time-variation of momentum along direction $j$ has to be balanced by the sum of stresses $\sigma_{ij}$ and total body-force per volume unit ($X_j$ has units of acceleration). Because different fluid particles are present at any fixed point at different times, it is more intuitive to derive this equation from a Lagrangian point of view (following a fluid particle moving with the flow). However, for brevity we provide below the final result, after re-mapping back to Eulerian coordinates:

$$\frac{\partial (\rho u_j)}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_i} = \frac{\partial \sigma_{ij}}{\partial x_i} + \rho X_j \tag{16}$$

Without loss of generality, the *left-hand side* (LHS) of eq. (16) can be simplified, by expanding the terms and using the equation of continuity eq. (15), leading to:

$$\rho \left( \frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} \right) = \frac{\partial \sigma_{ij}}{\partial x_i} + \rho X_j \tag{17}$$

To make eq. (17) more explicit, the stress-tensor $\sigma_{ij}$ needs to be expressed in terms of the tensor of *strain-rates* $e_{ij}$ (also known as *deformation rates*), which are defined in terms of spatial variations[9] of the velocity field:

$$e_{ij} \equiv \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{18}$$

For the fluids considered in this thesis, the dependence $\sigma_{ij}(e_{kl})$ can be assumed to be linear. Using, in addition, the *isotropy* assumption for the fluid (which requires

9 Specifically, only the *symmetric part* $e_{ij}$ of the general *deformation tensor* $\partial u_i/\partial x_j$ is used here – the *anti-symmetric part* refers to local rotation motion, which does not involve viscous transfer of momentum.

invariance of the equations with respect to rotations and translations of the system of coordinates), the constitutive equation for the stresses can be shown to be:

$$\sigma_{ij} = -\delta_{ij} p + \underbrace{2\mu \left( \underbrace{e_{ij} - \frac{1}{3} e_{kk} \delta_{ij}}_{\text{deviatoric strain rate tensor}} \right)}_{\equiv \tau_{ij} \text{ (deviatoric stress tensor)}} + \zeta e_{kk} \delta_{ij} \tag{19}$$

where $\delta_{ij}$ is the Kronecker delta, $\mu$ is the *dynamic (shear) viscosity* (or, simply, *viscosity*), $\zeta$ is the *volume (bulk) viscosity* and $p$ is the *pressure*.

For all applications considered in this thesis, the term containing the bulk viscosity $\zeta$ can be ignored[10], so eq. (19) simplifies to:

$$\sigma_{ij} = -\delta_{ij} p + \tau_{ij} \tag{20}$$

Or, writing the deviatoric stress tensor explicitly, the stresses read:

$$\sigma_{ij} = -\delta_{ij} p + \underbrace{\mu \left[ \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right]}_{\equiv \tau_{ij}} \tag{21}$$

The *deviatoric stress tensor* $\tau_{ij}$ is also known in fluid dynamics as the *viscous stress tensor*. It relates to the microscopic motion of fluid particles, which transfer momentum in the presence of velocity gradients (shear). Intuitively, viscosity $\mu$ is the proportionality factor relating a given amount of shearing (deviatoric strain rate)[11] to the amount of shearing stress necessary to produce that shearing. The remaining part of the stress tensor ($-\delta_{ij} p$ in eq. (20)) accounts for the stresses which are *normal* to the surface-elements of the fluid particle, i.e. the *pressure*.

Plugging eq. (20) into eq. (17) (and dividing the result by $\rho$), we obtain the momentum equation for flows without sound- or shock-waves:

$$\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial p}{\partial x_j} + \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_i} + X_j \tag{22}$$

Equation (22) may be expanded further; to simplify the expressions, we use the (common) assumption of a constant dynamic viscosity $\mu$, which (after some algebra) leads to:

$$\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial p}{\partial x_j} + \nu \left[ \frac{\partial^2 u_j}{\partial x_i \partial x_i} + \frac{1}{3} \frac{\partial}{\partial x_j} \left( \frac{\partial u_i}{\partial x_i} \right) \right] + X_j, \tag{23}$$

where $\nu \equiv \mu/\rho$ is known as *kinematic viscosity*.

10 This term only becomes important when studying propagation of sound or shock waves.

11 It is important to notice that (unlike solids studied in continuum mechanics) a fluid is unable to sustain deviatoric stresses when it is at rest.

As a note on the momentum eq. (23), the pressure $p$ above refers to the so-called *mechanical pressure*, defined in terms of the purely *translational* motion of the molecules. Strictly-speaking, this is different from the *thermodynamic equilibrium pressure*, which is computed from the equation of state (see Section 2.3.1.4), and which also accounts for the fact that molecules may have additional, non-translational degrees of freedom (e.g. rotations or vibrations). However, this thesis only addresses fluid problems where the bulk viscosity is assumed to vanish – in such cases the distinction between the two pressures becomes irrelevant.

**Useful tool for visualizing stationary 2D flows: the streamfunction**

For the special case of stationary (time-independent) 2D flows, we can define the *streamfunction* $\boxed{\psi(\mathbf{x})}$, in terms of the velocity components as follows:

$$\begin{cases} u_1 = -\dfrac{\partial \psi}{\partial x_2} & (24) \\ u_2 = \dfrac{\partial \psi}{\partial x_1} & (25) \end{cases}$$

Assuming the velocity-field is known, these equations may be used (in combination with appropriate BCs at the boundaries of the fluid-domain) to evaluate $\psi(\mathbf{x})$ numerically (usually by transforming eqs. (24) and (25) into a Poisson equation for vorticity). For a 2D, stationary flow, the isocontours of the streamfunction represent the trajectories of the fluid-particles.
We use this quantity to visualize the solution of the 2D RB model, in Section 5.1.

### 2.3.1.3 *Temperature equation*

So far, we discussed the equations resulting from the conservation principles for mass and momentum. Since we are also interested in fluids where temperature plays a role, we also need to consider the *conservation of energy*. It can be shown that this leads to an equation for the evolution of temperature, with the general form:

$$\rho \frac{\partial (c_V T)}{\partial t} + \rho u_i \frac{\partial (c_V T)}{\partial x_i} = -p \frac{\partial u_i}{\partial x_i} + \frac{\partial}{\partial x_i}\left(k \frac{\partial T}{\partial x_i}\right) + \underbrace{2\mu \left(e_{ij} e_{ij} - \frac{1}{3} e_{jj} e_{jj}\right)}_{\Phi} \qquad (26)$$

where $T$ is the temperature, $c_V$ is the specific heat (at constant volume) of the fluid, $k$ is the coefficient of heat conductivity, and $\Phi$ is the rate of viscous heat dissipation. When (as will be the case for the applications in this thesis) $c_V$ and $k$ can be considered constants, the temperature equation simplifies somewhat:

$$\frac{\partial T}{\partial t} + u_i \frac{\partial T}{\partial x_i} = -\frac{p}{\rho c_V} \frac{\partial u_i}{\partial x_i} + \frac{k}{\rho c_V} \frac{\partial^2 T}{\partial x_i \partial x_i} + \frac{2\mu}{\rho c_V}\left(e_{ij} e_{ij} - \frac{1}{3} e_{jj} e_{jj}\right) \qquad (27)$$

#### 2.3.1.4 *Equation of state*

Equations (15), (23) and (27) discussed above give *four* equations for *five* unknowns: the two scalar fields $\rho$ and $p$, plus the three components of the vector field $\mathbf{u}$. An additional equation is necessary for the system of equations to be solvable. There are several approaches for specifying this last equation:

1. The simplest approach, used for incompressible flows is to "demote" density to a material property, assumed to be constant (or with only small variations, which are of little importance for the total mass transfer).

2. In the more general case, when the density is not constant, we need to relate it to other properties of the flow. Such a relation is known as an *equation of state* for the fluid; in general, it is of the form:

$$\rho = \rho(p, T) \tag{28}$$

   Here, we can identify two sub-cases:

   a) ***barotropic* fluid**: if the equation of state can be treated approximately as a function of pressure only, i.e.:

   $$\rho = \rho(p), \tag{29}$$

   the fluid is said to be *barotropic*. In such a case, the equation of state is all we need to close the system of equations. Of the fluids of interest in ESS water is barotropic to a good approximation, while air is not – this is why this approximation is used more in oceanography than in atmospheric sciences. However, even flows of non-barotropic (i.e. "baroclinic") fluids may contain regions where the barotropic approximation holds – these are known as "barotropic layers" in the literature; for example, large-scale atmospheric flows near the tropics (central latitudes) are mostly barotropic.

   b) ***baroclinic* fluid**: in the more general case, when the equation of state depends on temperature also, the fluid is said to be *baroclinic*. In such cases, we need an additional equation for closing the system – this is provided by the *energy equation*.

   For the convection problems in this thesis, the equation of state is first simplified to:

   $$\rho = \rho_0 \left[1 - \alpha \left(T - T_0\right)\right], \tag{30}$$

   where $\rho_0$ is a constant density-value, at temperature $T_0$ (as explained in the next section), and $\alpha$ is the *coefficient of thermal expansion*.[12] This is obviously a *linearization* of the true (non-linear) expression for $\rho$; nonetheless, the approximation is reasonable for the flows studied here.

12 Also known as the *coefficient of volume expansion*.

### 2.3.2 *Incompressible fluids and the Oberbeck-Boussinesq approximation*

As already mentioned, for the flows considered in this thesis, the coefficients of viscosity, of specific heat, and of heat conductivity may be considered to be constants. By considering some additional properties of the flows, the governing eqs. (15), (23) and (27) can be simplified further.

Due to the low values of $\alpha$ in eq. (30), the variation of density can also be ignored[13] in all terms of the governing equations $\underline{\text{except}}$ in the body-force term[14] (because of the high value of the gravitational acceleration constant) – this is known as the *Oberbeck-Boussinesq approximation* (also referred to as simply the *Boussinesq approximation* by some authors)[15] of the momentum equation. Under this assumption, the continuity eq. (15) simplifies to:

$$\boxed{\frac{\partial u_i}{\partial x_i} = 0.} \tag{31}$$

The momentum eq. (23) becomes:

$$\boxed{\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} = -\frac{1}{\rho_0}\frac{\partial p}{\partial x_j} + \nu \frac{\partial^2 u_j}{\partial x_i \partial x_i} + \left[1 - \alpha\left(T - T_0\right)\right] X_j} \tag{32}$$

The temperature equation also simplifies, because the $1^{\text{st}}$ term on the *right-hand side* (RHS) cancels due to eq. (31), while the last term (viscous heat dissipation) can be shown via scale-analysis to be less important for fluids such as water. This leads to:

$$\boxed{\frac{\partial T}{\partial t} + u_i \frac{\partial T}{\partial x_i} = \kappa \frac{\partial^2 T}{\partial x_i \partial x_i}} \tag{33}$$

where we defined the *coefficient of thermal conductivity* as $\kappa \equiv \frac{k}{\rho_0 c_V}$.

We consider eqs. (31) to (33) to be the fundamental equations for the convection applications, which we solve numerically in this thesis. In vector form, they can be re-written as:

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{(continuity eq.)} \tag{34}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \left(\mathbf{u} \cdot \nabla\right) \mathbf{u} = -\frac{1}{\rho_0} \nabla p + \nu \triangle \mathbf{u} + \left[1 - \alpha\left(T - T_0\right)\right] \mathbf{X} \qquad \text{(momentum eq.)} \tag{35}$$

$$\frac{\partial T}{\partial t} + \left(\mathbf{u} \cdot \nabla\right) T = \kappa \triangle T \qquad \text{(temperature eq.)} \tag{36}$$

13 As long as the temperature-gradients are small ($\lesssim 10\,°\mathrm{C}$).

14 Indeed, for two of the applications (Sections 5.1 and 5.2) the fundamental physics of the problems would be lost if all density variations were ignored.

15 Oberbeck (1879) was the first to introduce the approximation, for describing flows driven by horizontal temperature gradients.

where we defined the *Laplace operator* as:

$$\triangle \equiv \nabla^2 \equiv \frac{\partial^2}{\partial x_i \partial x_i}$$

#### 2.3.2.1 *Coupling of hydrodynamic fields to temperature field*

The incompressible hydrodynamic and temperature fields are tightly coupled, via two mechanisms: (a) first, the temperature field is advected by the velocity of the fluid; (b) on the other hand, the fluid is also influenced by the temperature, through variations of the last term in the RHS of eq. (32).

## 2.4 THE RAYLEIGH-BÉNARD (RB) PROBLEM

The setup known as the *Rayleigh-Bénard* (RB) problem corresponds to a layer of fluid bounded by two horizontal planar surfaces, where the lower surface is maintained at a temperature $T_b$, which is higher than the temperature $T_t$ of the upper surface. If the system starts from a rest-state, this adverse temperature gradient imposed at the vertical boundaries causes an *unstable* vertical distribution of density, which eventually causes convective motions to appear, if the temperature-gradient is sufficiently large.

Despite its apparent simplicity, this problem is of great interest, because it can evolve into a fully-turbulent flow, via a series of flow-regime transitions. Therefore, it serves as an excellent test bed for improving our understanding of buoyancy-driven flows in general. Also, being one of the few problems in hydrodynamic stability for which an exact stability criterion was obtained, this setup is of great importance for validating numerical models of fluid flow.

This problem is solved numerically using LBM later in this thesis (in Section 5.1 for the 2D case and Section 5.2 for the 3D case). To prepare the theoretical background for these applications, in this section we proceed with specializing the incompressible equations from the previous section. The discussion below focuses on the (more general) 3D case, for which we obtain the final version of the evolution equations that will be solved numerically.

The final manipulations of the governing equations are given in Section 2.4.1. *boundary condition*s (BCs) and *initial condition*s (ICs) will be given in Section 2.4.2. Afterwards, in Section 2.4.3, we reformulate the entire problem into non-dimensional units, to facilitate the theoretical analysis. Finally, in Section 2.4.4, we present important relevant results from the theory of hydrodynamic instabilities, and from experimental studies (which are necessary for validating the simulations discussed later in the thesis).

### 2.4.1 *Governing equations for Rayleigh-Bénard (RB) problem*

To describe the evolution of the RB problem, we start with the equations for incompressible, Boussinesq fluids (Equations (31) to (33)). Of these, only the momentum eq. (32) needs further manipulations.

First, the acceleration $X_j$ due to the body-force is:

$$\mathbf{X} = -g\hat{y} \Leftrightarrow X_j = -g\delta_{j,y}, \tag{37}$$

where $g$ is the gravitational acceleration and $\delta_{m,n}$ is the Kronecker symbol. The momentum eq. (32) becomes:

$$\frac{\partial u_j}{\partial t} + u_i\frac{\partial u_j}{\partial x_i} = -\frac{1}{\rho_0}\frac{\partial p}{\partial x_j} + \nu\frac{\partial^2 u_j}{\partial x_i \partial x_i} - [1 - \alpha\,(T - T_0)]\,g\delta_{j,y}, \tag{38}$$

where $\rho_0$ is the reference density, taken[16] as the density at the average temperature (initially in the middle of the vertical extent):

$$T_0 \equiv \frac{T_b + T_t}{2} \tag{39}$$

Second, to simplify the subsequent calculations, we absorb the constant part of the body-force into the concept of *modified pressure*[17] ($p^*$), which is defined by:

$$\frac{\partial p^*}{\partial x_j} \equiv \frac{\partial p}{\partial x_j} + \rho_0 g\delta_{j,y} \tag{40}$$

Therefore, the final form of the momentum equation for the RB problem reads:

$$\boxed{\frac{\partial u_j}{\partial t} + u_i\frac{\partial u_j}{\partial x_i} = -\frac{1}{\rho_0}\frac{\partial p^*}{\partial x_j} + \nu\frac{\partial^2 u_j}{\partial x_i \partial x_i} + \alpha g\,(T - T_0)\,\delta_{j,y}} \tag{41}$$

### 2.4.2 *Specification of BCs and ICs*

To complete the specification of the problem, we also need to provide appropriate *boundary conditions* (BCs) and *initial conditions* (ICs).

#### 2.4.2.1 *BCs*

VERTICAL WALLS: For the theoretical discussion in this chapter, the fluid is assumed to extend infinitely along the $x$ and $y$ directions, so no BCs are necessary for the vertical walls.

16 In reality, the dependence of $\rho$ on $T$ is non-linear. However, this particular choice of $T_0$ yields a reasonable approximation for the linearized function, as long as $T_b - T_t$ is not too large.

17 The change from *normal* to *modified* pressure only causes a constant shift in the values of the pressure-gradients.

HORIZONTAL WALLS: For the horizontal walls, the *no-slip* BC is chosen for the hydrodynamic component:

$$\mathbf{u}\left(x, -H/2, z, t\right) = \mathbf{u}\left(x, +H/2, z, t\right) = 0 \tag{42}$$

where $H$ is the distance between the horizontal planes bounding the fluid.

For the thermal component, a constant temperature is set at each wall; Denoting

$$(\Delta T)_0 \equiv T_b - T_t \geqslant 0, \tag{43}$$

we have:

$$T\left(x, -H/2, z, t\right) = T_b \equiv T_0 + \frac{(\Delta T)_0}{2} \tag{44}$$

$$T\left(x, +H/2, z, t\right) = T_t \equiv T_0 - \frac{(\Delta T)_0}{2} \tag{45}$$

### 2.4.2.2 *ICs*

For studying the onset of convection, it is natural to start with the stationary solution, when the fluid is at rest:

$$\mathbf{u}(\mathbf{x}, 0) = 0. \tag{46}$$

We plug this velocity into the temperature eq. (33), to obtain a compatible IC for temperature. Specifically, we obtain a 1D Laplace equation, which can be easily integrated and fitted to the BCs, to yield:

$$T(\mathbf{x}, 0) = T_0 - \frac{(\Delta T)_0}{H} y \tag{47}$$

Physically, this corresponds to the situation where the heat-transfer is entirely due to molecular diffusion.

Finally, to obtain a consistent IC for the modified pressure, we plug eq. (46) and eq. (47) into eq. (41). After integration we obtain an IC for *modified pressure differences*:

$$p^*(\mathbf{x}, 0) - p_0^* = -\frac{\rho_0 \alpha g \left(\Delta T\right)_0}{2H} y^2, \tag{48}$$

where $p_0^* \equiv p^*(x, 0, z, 0)$ is some reference value for the modified pressure, in the middle of the fluid layer (i.e. the same location where we have $T_0$ and $\rho_0$, for consistency).

It can be seen that we derived the initial temperature ($\Rightarrow$ density) and pressure profiles, so that the pressure-gradients cancel exactly the effect of the buoyancy forces – in the upper-half of the domain, the buoyancy force is downwards, while the modified pressure-gradient is upwards (and vice-versa for the lower-half of the domain).

### 2.4.3 *Problem reformulation in non-dimensional units*

Before we proceed, it is useful to re-write the problem in *dimensionless* form. This procedure, formalized by the concept of *dynamic similarity* (Buckingham, 1914), has the benefit of making the subsequent results more generally applicable, by minimizing the number of parameters needed to specify the problem.

For the RB setup, the natural length-scale is provided by the distance ($H$) between the horizontal planes. Because the flow is initially at rest, there is no obvious scale for velocity or time. However, a so-called "diffusive time-scale"[18] can be constructed, by combining the height of the domain and the thermal diffusivity ($H^2/\kappa$). From these two, a velocity scale can be computed ($\kappa/H$).

Because only differences or (spatial/temporal) variations of the modified pressure and temperature only appear in the governing equations (eqs. (33) and (41)), it is sufficient to define scaling-relations for *differences* of these quantities (relative to the reference values, in the middle of the domain when at rest). A scale for the modified pressure-differences is obtained from the reference density and the characteristic velocity ($\rho_0\kappa^2/H^2$). Finally, it is natural to scale temperature-differences by the temperature-difference between the two planes ($(\Delta T)_0$).

To summarize, we have the following scaling-relations:

SPACE SCALING

$$x_j^{(d)} \equiv \frac{x_j}{H} \iff x_j = H x_j^{(d)} \tag{49}$$

TIME SCALING

$$t^{(d)} \equiv \frac{\kappa}{H^2} t \iff t = \frac{H^2}{\kappa} t^{(d)} \tag{50}$$

VELOCITY SCALING

$$u_j^{(d)} = \frac{H}{\kappa} u_j \iff u_j = \frac{\kappa}{H} u_j^{(d)} \tag{51}$$

PRESSURE-DIFFERENCE SCALING

$$(\delta p)^{(d)} = \frac{H^2}{\rho_0\kappa^2}(p^* - p_0^*) \iff p^* - p_0^* = \frac{\rho_0\kappa^2}{H^2}(\delta p)^{(d)}, \tag{52}$$

where for brevity we drop the $\square^*$ from the notation of the modified pressure (in the dimensionless system).

TEMPERATURE-DIFFERENCE SCALING

$$(\delta T)^{(d)} = \frac{1}{(\Delta T)_0}(T - T_0) \iff T - T_0 = (\Delta T)_0 (\delta T)^{(d)} \tag{53}$$

18 We will return to the issue of scales at the end of this section.

With these scalings, the governing equations can be written in dimensionless form as:

$$\frac{\partial u_i^{(d)}}{\partial x_i^{(d)}} = 0 \tag{54}$$

$$\frac{\partial u_j^{(d)}}{\partial t^{(d)}} + u_i^{(d)} \frac{\partial u_j^{(d)}}{\partial x_i^{(d)}} = -\frac{\partial (\delta p)^{(d)}}{\partial x_j^{(d)}} + \mathrm{Pr} \frac{\partial^2 u_j^{(d)}}{\partial x_i^{(d)} \partial x_i^{(d)}} + \mathrm{Ra}\,\mathrm{Pr} (\delta T)^{(d)} \delta_{j,y} \tag{55}$$

$$\frac{\partial (\delta T)^{(d)}}{\partial t^{(d)}} + u_i^{(d)} \frac{\partial (\delta T)^{(d)}}{\partial x_i^{(d)}} = \frac{\partial^2 (\delta T)^{(d)}}{\partial x_i^{(d)} \partial x_i^{(d)}} \tag{56}$$

where the dimensionless coefficients Ra (Rayleigh number) and Pr (Prandtl number) are defined as:

$$\mathrm{Pr} \equiv \frac{\nu}{\kappa} \tag{57}$$

$$\mathrm{Ra} \equiv \frac{\alpha g H^3 (\Delta T)_0}{\kappa \nu} \tag{58}$$

These two dimensionless coefficients dictate the intensity of the viscous term and of the buoyancy term in the momentum equation. The advantage of the dimensionless formulation is that any two physical systems **A** and **B** are *dynamically equivalent*, as long as $\mathrm{Pr_A} = \mathrm{Pr_B}$, $\mathrm{Ra_A} = \mathrm{Ra_B}$, and their dimensionless BCs and ICs are also identical. For example, this is the reason why wind-tunnels are useful – by adjusting the various scales of the flow, the motion of air around a scaled-down model of an airplane can be made dynamically similar to the flow around the real airplane that is being designed. While Ra is dependent on the boundary conditions of the flow, Pr is a *material constant*, which quantifies the ratio of *viscous diffusion* to *thermal diffusion*. Common values for Pr are $0.7 - 0.8$ for air, $7$ for water, or $13.4$ and $7.2$ for seawater at $0°$C and at $20°$C respectively (Marshall and Plumb, 2008).

Returning to the RB setup, the values of the control-parameters Ra and Pr have a profound influence on the flow-regimes – in several ranges of these parameters, the flow will become significantly different, even for small changes of the control-parameters (we will discuss several such transitions shortly). However, it is is interesting to note that the value of the Prandtl number (Pr) does not play a role for the initial transition (from rest-state to convection-rolls), which our subsequent theoretical analysis will also show.[19]

**Mechanisms for transport of fluid properties and characteristic time-scales in the RB problem**

For the RB setup, there are two mechanisms for transport of fluid properties, depending on the Rayleigh number (Ra). Accordingly, two characteristic time-scales are commonly used:

19 Neither the final eq. (110) (which predicts the locus of the states of marginal stability for that transition) nor its corresponding BCs contain any dependence on Pr.

1. diffusive time-scale: In the limit of low Ra, when the fluid is at rest, the dominant mechanism of transport is molecular diffusion, for which a so-called "diffusive time-scale" may be defined as:

$$\tau_{\mathcal{D}} \equiv \frac{H^2}{\kappa} \tag{59}$$

   This scale is appropriate for studying the conditions when convection first sets in (which we discuss later). In the case when convection is already present, $\tau_{\mathcal{D}}$ is still relevant for studying the thermal boundary layers, near the horizontal walls.

2. convective time-scale: For studying flows where convection is present, $\tau_{\mathcal{D}}$ is no longer useful, because it is several orders of magnitude larger than the largest time-scales of the flow. Therefore, in such regimes it is customary (Sakievich et al., 2016) to proceed by first defining a so-called "free-fall velocity":

$$U_{\mathcal{E}} \equiv \sqrt{Hg\alpha(\Delta T)_0}, \tag{60}$$

   which yields the corresponding time-scale:

$$\tau_{\mathcal{E}} \equiv \sqrt{\frac{H}{g\alpha(\Delta T)_0}} \tag{61}$$

   which, in turn, is related to the time required for a typical eddy in the flow to complete a rotation ("eddy-turnover time"). For the case when the heat-flux is fixed (instead of the temperature-difference, as considered in this thesis), the scaling of Deardorff (1970) is commonly used.

   For the regimes with strong convection, a lot of work has been done towards understanding both the behavior of the bulk (convective part) and the thermal boundary-layers (see Grossmann and Lohse (2000) and the references therein).

For our numerical experiments in Sections 5.1 and 5.2 we follow the convention of using the diffusive time-scale (although it would be more appropriate to use the convective time-scale for higher-Ra simulations. Fortunately, given a physical phenomenon which is characterized by a "real-world" time-duration, velocity, and pressure-difference, it is easy to derive an elegant mapping between what would be reported in the two dimensionless systems of units:

$$\begin{cases} t_{\mathcal{D}}^{(d)} = t_{\mathcal{E}}^{(d)} \dfrac{1}{\sqrt{\mathrm{Ra\,Pr}}} \quad , & (62) \\ u_{j,\mathcal{D}}^{(d)} = u_{j,\mathcal{E}}^{(d)} \sqrt{\mathrm{Ra\,Pr}} \quad , \quad \text{and} & (63) \\ (\delta p)_{\mathcal{D}}^{(d)} = \dfrac{(\delta p)_{\mathcal{E}}^{(d)}}{\mathrm{Ra\,Pr}}. & (64) \end{cases}$$

We use the equations above for presenting the simulation data in Sections 5.1 and 5.2. However, because the following theory in this chapter is related to the onset of convection, we focus on the diffusive scales for now, and drop the $\square_{\mathcal{D}/\mathcal{E}}$ subscripts.

#### 2.4.3.1 *BCs and ICs*

To complete the dimensionless formulation of the problem, we also need the scaled expressions for BCs and ICs.

BCs: Because the periodic BCs used at the vertical walls are of a *topological* nature, they can be applied directly at the model-level. Therefore, only the BCs at the horizontal walls need to be mentioned explicitly here.

For the *velocity* field, we have:

$$\mathbf{u}^{(d)}\left(x^{(d)}, -1/2, z^{(d)}, t^{(d)}\right) = \mathbf{u}^{(d)}\left(x^{(d)}, +1/2, z^{(d)}, t^{(d)}\right) = 0, \tag{65}$$

and for the *temperature* field:

$$(\delta T)_b^{(d)} \equiv (\delta T)^{(d)}\left(x^{(d)}, -1/2, z^{(d)}, t^{(d)}\right) \equiv T^{(d)}\left(x^{(d)}, -1/2, z^{(d)}, t^{(d)}\right) - T_0^{(d)} = \frac{1}{2} \tag{66}$$

$$(\delta T)_t^{(d)} \equiv (\delta T)^{(d)}\left(x^{(d)}, +1/2, z^{(d)}, t^{(d)}\right) \equiv T^{(d)}\left(x^{(d)}, +1/2, z^{(d)}, t^{(d)}\right) - T_0^{(d)} = -\frac{1}{2} \tag{67}$$

where $T_0^{(d)}$ is some arbitrary reference temperature-value in the middle of the domain (which we may set to zero, for convenience).

ICs: For the *velocity* field, the ICs becomes:

$$\mathbf{u}^{(d)}\left(\mathbf{x}^{(d)}, 0\right) = 0, \tag{68}$$

for the *temperature* field:

$$(\delta T)^{(d)}\left(\mathbf{x}^{(d)}, 0\right) \equiv T^{(d)}\left(\mathbf{x}^{(d)}, 0\right) - T_0^{(d)} = \frac{1}{(\Delta T)_0}\left(T\left(\mathbf{x}, 0\right) - T_0\right) = -y^{(d)} \tag{69}$$

and for the *pressure* field:

$$(\delta p)^{(d)}\left(\mathbf{x}^{(d)}, 0\right) \equiv p^{(d)}\left(\mathbf{x}, 0\right) - p_0^{(d)} = -\frac{\mathrm{Ra}\,\mathrm{Pr}}{2}\left(y^{(d)}\right)^2 \tag{70}$$

where $p_0^{(d)}$ (analogue to $T_0^{(d)}$) is an arbitrary reference modified-pressure value in the middle of the domain (which we may set to zero, without any loss of generality).

### 2.4.4 *Important theoretical and experimental results for the RB problem*

To conclude our coverage of the theory of the RB problem, in this subsection we briefly introduce the field of hydrodynamic stability (Section 2.4.4.1), which we then use in Section 2.4.4.2, to analyze theoretically the 1st instability for the RB problem. Finally, in Sections 2.4.4.3 and 2.4.4.4, we discuss some related theoretical and experimental results.

#### 2.4.4.1 *Principles of hydrodynamic stability analysis*

As mentioned previously, the state of any fluid system is characterized by several fields (such as velocity, pressure, and temperature). Under certain conditions, these fields do not depend on time, in which case the system is said to be in a *stationary state*. The simplest examples of stationary states are the cases when the velocity field is zero everywhere (i.e. hydrostatics); however, many flows also exhibit *non-trivial* ($\mathbf{u}(\mathbf{x}) \neq 0$) stationary states.

Let us assume that a system is to be studied, starting from such a stationary IC. In practice, however, a *perfectly stationary* IC is impossible to prepare, because some small fluctuations will always be present. The goal of the theory of *hydrodynamic stability analysis* is to determine whether those small fluctuations will have a significant impact on the system or not. Specifically, the perturbations may:

a) *decay*, so that the state of the system will evolve towards the stationary state $\Rightarrow$ the system is *unstable* with respect to the perturbation, or

b) *become amplified*, so that the system will evolve towards another state (which may also be stationary) $\Rightarrow$ the system is *unstable* with respect to the perturbation.

To simplify the theory, stability is defined in a "firm" sense, *with respect to all possible perturbations*: a state is said to be *stable* if *any perturbation will eventually decay*, and *unstable* if *even a single perturbation exists which gets amplified*.

We assume (for generality) that the hydrodynamic problem was formulated in a dimensionless form, where the dynamics is described by a minimal set of control parameters $(X_1, X_2, \ldots, X_n)$. Since the control parameters include the effects of any material constants appearing in the governing equations, as well as of the BCs and the ICs, which of the above scenarios will be followed by the system also depends on the control parameters. Let us denote by *parameter-space* the sub-region of $\mathbb{R}^n$ defined by the physically-valid ranges of all control parameters (the outer solid contour in Figure 2).

Within the parameter-space, there will be sub-regions where a particular IC will be *stable* (green zone in Figure 2), as well as sub-regions where the IC will be *unstable* (red zone in Figure 2). Separating these regions are states of *marginal stability*,[20] which in general satisfy an equation of the form:

$$\mathcal{F}(X_1, X_2, \ldots X_n) = 0 \tag{71}$$

In problems of *hydrodynamic stability*, the goal is to find the function $\mathcal{F}$ (or at least to approximate it numerically).

Near the states of marginal stability, the amplitudes of the perturbations decay or become amplified. These amplitude-changes may take place:

- *aperiodically* (*monotonically*): in such cases, at the onset of instability there will be a pattern of mostly stationary motions, eventually leading to a new state

20 These are sometimes also denoted as states of *neutral stability*.

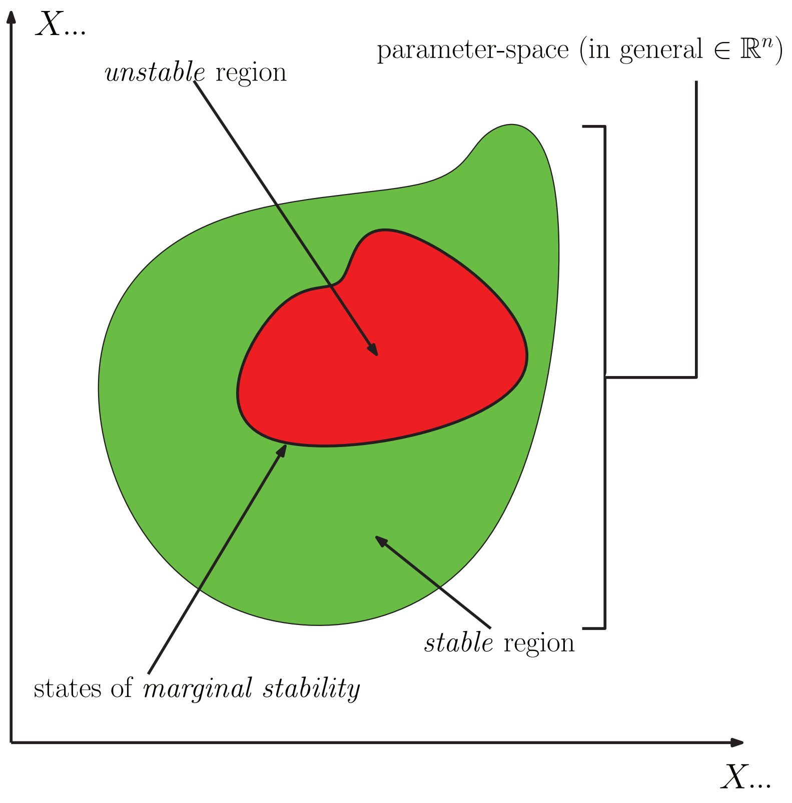

$X...$
*unstable* region
parameter-space (in general $\in \mathbb{R}^n$)
*stable* region
states of *marginal stability*
$X...$

Figure 2: Sketch of the *parameter-space* defined by the control-parameters of a fluid system.

with so-called *secondary flows*. For such situations, it is said that the *principle of the exchange of stabilities* (Davis, 1969) holds.

- *periodically*: in such cases, oscillatory motions will be present at the onset of instability.[21]

The problem of stability can be considered at various orders of approximation:

a) The simplest approach (which is outlined in the remainder of this section) is to linearize the equations (hence the name *linear stability analysis*). Due to the linearization, the stability of the system can only be investigated in the close vicinity of a stationary (or quasi-stationary) state, in response to infinitesimal perturbations. Assuming that the evolution of the system may be described as a superposition of normal modes, the problem of stability can then often be reduced to an eigenvalue problem. Mathematically, the question of stability is related to the sign of the real part of the eigenvalues (stability for negative sign and instability for positive sign).

b) A more general approach to stability analysis is Lyapunov's second method (also known as the "direct" method). This approach works directly with the original equations (instead of the linearized versions), so that the influence of the *finite-amplitude* ICs also enters the picture. However, as can be expected, this increased

21 This scenario is also known as *overstability*, because of restoring forces which oppose the perturbations, but their effects are *overshooting* with respect to the equilibrium.

generality usually comes at the expense of much more complex calculations, which are only practical for simple systems.

We focus on the *linear stability analysis* below. In general, this procedure consists of the following sequence of steps:

1. Starting from ICs corresponding to the stationary state, we apply small perturbations to the fields describing the flow, to obtain the evolution equations for the perturbations. To keep the equations tractable, it is usually necessary to only keep the terms which are linear with respect to the perturbations, which is reasonable if the perturbations are small.

2. To explore the stability of the system with respect to *all* possible perturbations, the perturbation is assumed to consist of a complete superposition of an (infinite) set of *basic modes*. Because of the linearity of the equations, mode-interactions are outside the scope of the theory. Assuming that each basic mode is characterized by a vector $\mathbf{k}$, we write the perturbation for a flow-field $\Psi_i$ as:

$$\Psi_i(\mathbf{x}, t) = \int \Psi_{i,k}(\mathbf{x}) \exp(s_{\mathbf{k}} t) d\mathbf{k} \tag{72}$$

   where

$$s_{\mathbf{k}} \equiv s_{\mathbf{k}}^{(r)} + i s_{\mathbf{k}}^{(i)} \in \mathbb{C} \tag{73}$$

   are constants (depending on the control-parameters) which have to be determined.

3. By imposing the BCs to which the flow is subjected, only certain values of $s_{\mathbf{k}}$ will allow non-trivial solutions, which leads to a *characteristic value problem* in terms of $s_{\mathbf{k}}$.

4. For the flow to be *stable* with respect to any perturbation, we need to have:

$$s_{\mathbf{k}}^{(r)}(X_1, X_2, \ldots, X_n) < 0, \qquad \forall \mathbf{k}. \tag{74}$$

   Conversely, the flow is *unstable* if:

$$\exists\ \mathbf{k}_c \quad \text{such that} \quad s_{\mathbf{k}_c}^{(r)}(X_1, X_2, \ldots, X_n) > 0. \tag{75}$$

   Therefore, the state of *neutral stability* (with respect to mode $\mathbf{k}$) is given by a curve in parameter-space:

$$s_{\mathbf{k}}^{(r)}(X_1, X_2, \ldots, X_n) \equiv \mathcal{F}_{\mathbf{k}}^{(r)}(X_1, X_2, \ldots, X_n) = 0. \tag{76}$$

5. By taking the envelope of the resulting curves for *all* modes, we can finally distinguish the *regions of complete stability* from the *regions of instability*. For every

point along this envelope, the system will become unstable with respect to the particular mode $k$, whose parameter-space curve coincides with this envelope at that point.

#### 2.4.4.2 *Stability analysis for the* $1^s$ *transition*

For the RB problem, the control parameters are the Rayleigh and Prandtl numbers (Ra and Pr, respectively). The 1st transition of this flow (from quiescent state to 2D convection-rolls) is remarkable, because it is one of the few examples of flow-instabilities which can still be studied analytically. In this section, we provide a sketch of this analysis, the results of which will be later used for validating our numerical models. Although our discussion largely follows that of Chandrasekhar (1981), we introduce the dimensionless formulation at an earlier stage, which makes some of the calculations more clear.

Starting from the dimensionless evolution equations (Equations (54) to (56)), we superimpose a perturbation onto the quiescent IC (given by eqs. (68) to (70)):

$$\underbrace{\mathbf{u}'(\mathbf{x},0)}_{\text{perturb. of velocity field}} \neq 0 \tag{77}$$

$$(\delta T)'(\mathbf{x},0) = -y + \underbrace{\theta}_{\text{perturb. of temperature field}} \tag{78}$$

$$(\delta p)''''(\mathbf{x},0) = -\frac{\mathrm{Ra}\,\mathrm{Pr}}{2}y^2 + \underbrace{(\delta p)}_{\text{perturb. of pressure field}} \tag{79}$$

where we dropped the $\square^{(d)}$ superscript for brevity.

The perturbations need to be physically-consistent, which requires that:

$$\frac{\partial u'_i}{\partial x_i} = 0 \tag{80}$$

Plugging the perturbed fields into the dimensionless evolution equations, we obtain (after some algebra, where we discard the higher-order terms w.r.t. the perturbations) the corresponding equations governing the evolution of the perturbations themselves:

$$\frac{\partial u'_j}{\partial t} = -\frac{\partial(\delta p)}{\partial x_j} + \mathrm{Pr}\,\frac{\partial^2 u'_j}{\partial x_i \partial x_i} + \mathrm{Ra}\,\mathrm{Pr}\,\theta\delta_{j,y} \tag{81}$$

$$\frac{\partial \theta}{\partial t} = u'_i\delta_{i,y} + \frac{\partial^2 \theta}{\partial x_i \partial x_i} \tag{82}$$

**Curl using permutation tensor (Levi-Civita symbol) and $\epsilon - \delta$ identity**

Given a vector field $\mathbf{F}(\mathbf{x}) = F_i(\mathbf{x})\hat{e}_i$, the $i^{\text{th}}$ component of the curl of $\mathbf{F}$ is:

$$(\nabla \times \mathbf{F})_i(\mathbf{x}) \equiv \epsilon_{i,j,k} \frac{\partial}{\partial x_j} F_k(\mathbf{x}) \tag{83}$$

where $-1$ if $(i, j, k)$ is an odd permutation of $(1, 2, 3)$.

$$\epsilon_{i,j,k} = \begin{cases} 0 & \text{if any index is repeated,} \\ 1 & \text{if } (i,j,k) \text{ is an \textit{even permutation} of } (1,2,3)\text{, or} \\ -1 & \text{if } (i,j,k) \text{ is an \textit{odd permutation} of } (1,2,3). \end{cases} \tag{84}$$

A very useful relation between $\epsilon$ and the Kronecker $\delta$ symbol is:

$$\epsilon_{r,s,n}\epsilon_{n,k,j} = \delta_{r,k}\delta_{s,j} - \delta_{r,j}\delta_{s,k} \tag{85}$$

We eliminate the pressure-term, by taking the curl of the momentum equation. Using the Levi-Civita symbol, this leads to:

$$\frac{\partial \omega_n}{\partial t} = \mathrm{Ra}\,\mathrm{Pr}\,\epsilon_{n,k,j}\frac{\partial \theta}{\partial x_k}\delta_{j,y} + \mathrm{Pr}\,\frac{\partial^2 \omega_n}{\partial x_i \partial x_i}, \tag{86}$$

where we defined the $n^{\text{th}}$ component of the *vorticity* of the perturbed flow-field as:

$$\omega_n \equiv \epsilon_{n,k,j}\frac{\partial u'_j}{\partial x_k} \tag{87}$$

If we multiply eq. (86) by $\delta_{r,y}$, we obtain an evolution-equation for the *vertical component of the vorticity*, $\zeta \equiv \delta_{r,y}\omega_r$:

$$\frac{\partial \zeta}{\partial t} = \mathrm{Pr}\,\frac{\partial^2 \zeta}{\partial x_i \partial x_i} \tag{88}$$

Taking again the curl of eq. (86), we obtain:

$$\epsilon_{r,s,n}\frac{\partial}{\partial t}\left(\frac{\partial \omega_n}{\partial x_s}\right) = \mathrm{Ra}\,\mathrm{Pr}\,\epsilon_{r,s,n}\epsilon_{n,k,j}\delta_{j,y}\frac{\partial^2 \theta}{\partial x_s \partial x_k} + \mathrm{Pr}\,\epsilon_{r,s,n}\frac{\partial^2}{\partial x_i \partial x_i}\left(\frac{\partial \omega_n}{\partial x_s}\right) \tag{89}$$

Using the identity eq. (85), this becomes:

$$\frac{\partial}{\partial t}\left(\frac{\partial^2 u'_r}{\partial x_k \partial x_k}\right) = \mathrm{Ra}\,\mathrm{Pr}\left(\delta_{r,y}\frac{\partial^2 \theta}{\partial x_k \partial x_k} - \delta_{s,y}\frac{\partial^2 \theta}{\partial x_s \partial x_s}\right) + \mathrm{Pr}\,\frac{\partial^4 u'_r}{\partial x_i \partial x_i \partial x_k \partial x_k} \tag{90}$$

By multiplying this equation by $\delta_{r,y}$, we obtain an equation for the vertical component of the velocity-field, $w \equiv \delta_{r,y} u'_r$:

$$\frac{\partial}{\partial t}\left(\frac{\partial^2 w}{\partial x_k \partial x_k}\right) = \mathrm{Ra}\,\mathrm{Pr} \underbrace{\left(\frac{\partial^2 \theta}{\partial x_k \partial x_k} - \frac{\partial^2 \theta}{\partial y^2}\right)}_{\text{filter-out vertical part of the Laplacean}} + \mathrm{Pr}\,\frac{\partial^4 w}{\partial x_i \partial x_i \partial x_k \partial x_k} \tag{91}$$

To summarize, we need to consider the equations:

$$\begin{cases} \frac{\partial \zeta}{\partial t} & = \mathrm{Pr}\,\frac{\partial^2 \zeta}{\partial x_i \partial x_i}, \\ \frac{\partial}{\partial t}\left(\frac{\partial^2 w}{\partial x_k \partial x_k}\right) & = \mathrm{Ra}\,\mathrm{Pr}\left(\frac{\partial^2 \theta}{\partial x_k \partial x_k} - \frac{\partial^2 \theta}{\partial y^2}\right) + \mathrm{Pr}\,\frac{\partial^4 w}{\partial x_i \partial x_i \partial x_k \partial x_k}, \quad \text{and} \\ \frac{\partial \theta}{\partial t} = w + \frac{\partial^2 \theta}{\partial x_i \partial x_i}, & \end{cases} \tag{92}$$

subject to the BCs:

$$\begin{bmatrix} \zeta \\ w \\ \frac{\partial w}{\partial y} \\ \theta \end{bmatrix}(x, -{}^1\!/_2, z, t) = \begin{bmatrix} \zeta \\ w \\ \frac{\partial w}{\partial y} \\ \theta \end{bmatrix}(x, +{}^1\!/_2, z, t) = 0, \tag{93}$$

where the additional BC for $w(\mathbf{x})$ (which we now need because the $w$-equation is of 2nd-order) results from the continuity equation applied close to the walls.

The next step in the general recipe is the analysis into normal modes – we introduce an Ansatz for the perturbation as a superposition of normal modes, and then analyze the stability of each of these modes. In the present case, we assume that the disturbance for each field can be written as a superposition of 2D periodic waves (each of them characterized by its wavenumber $k$), i.e.:

$$\{\zeta, w, \theta\} \sim \exp\left(i(k_x x + k_z z) + st\right), \tag{94}$$

where:

- $k \equiv \sqrt{k_x^2 + k_z^2}$ is the horizontal wavenumber and
- $s \in \mathbb{C}$ is a constant.

The region of marginal stability is then described by the points where:

$$\mathrm{Re}\left(s(\mathbf{k})\right) = 0 \tag{95}$$

More explicitly, the Ansatz into normal modes reads:

$$\zeta = Y(y)e^{\boxed{\cdots}} \tag{96}$$
$$w = W(y)e^{\boxed{\cdots}} \tag{97}$$
$$\theta = \Theta(y)e^{\boxed{\cdots}} \tag{98}$$

where for brevity we defined $e^{\boxed{\cdots}} \equiv \exp(i(k_x x + k_z z) + st)$.

Any function $\mathcal{F}(\mathbf{x})$ of the above form has some interesting properties, which will be useful later:

$$\frac{\partial \mathcal{F}}{\partial t} = s\mathcal{F}, \tag{99}$$
$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}\right)\mathcal{F} = -k^2\mathcal{F}, \quad \text{and} \tag{100}$$
$$\frac{\partial^2 \mathcal{F}}{\partial x_i \partial x_i} = \left(\frac{d^2}{dy^2} - k^2\right)\mathcal{F} \tag{101}$$

Using the properties above, after plugging eq. (96), eq. (97) and eq. (98) into the evolution-equations, we obtain after some algebra:

$$\begin{cases} sY(y) = \Pr\left(\frac{d^2}{dy^2} - k^2\right)Y(y) & (102) \\ s\left(\frac{d^2}{dy^2} - k^2\right)W(y) = -\mathrm{Ra}\,\Pr\,k^2\Theta(y) + \Pr\left(\frac{d^2}{dy^2} - k^2\right)^2 W(y) & (103) \\ s\Theta(y) = W(y) + \left(\frac{d^2}{dy^2} - k^2\right)\Theta(y) & (104) \end{cases}$$

with the corresponding BCs:

$$\begin{bmatrix} Y \\ W \\ \frac{dW}{dy} \\ \Theta \end{bmatrix}(x, -1/2, z, t) = \begin{bmatrix} Y \\ W \\ \frac{dW}{dy} \\ \Theta \end{bmatrix}(x, +1/2, z, t) = 0. \tag{105}$$

By denoting $D \equiv \frac{d}{dy}$ and re-arranging terms, we can rewrite eqs. (103) and (104) as:

$$\begin{cases} \left(D^2 - k^2\right)\left(D^2 - k^2 - \frac{s}{\Pr}\right)W = \mathrm{Ra}\,k^2\Theta & (106) \\ \left(D^2 - k^2 - s\right)\Theta = -W & (107) \end{cases}$$

Next, we can eliminate $\Theta$ between these two equations, which leads to:

$$\left(D^2 - k^2 - s\right)\left(D^2 - k^2\right)\left(D^2 - k^2 - \frac{s}{\Pr}\right)W = -\mathrm{Ra}\,k^2 W \tag{108}$$

It can be shown that the *principle of exchange of stabilities* holds for the RB problem, which means that the imaginary part of s vanishes, $\Im(s) = 0$. This implies that *for the region of marginal stability* (where by definition we also have $\mathrm{Re}(s) = 0$) we must have:

$$s = \mathrm{Re}(s) + i\Im(s) = 0, \tag{109}$$

which allows us to simplify eq. (108) to:

$$\left(D^2 - k^2\right)^3 W = -\mathrm{Ra}\, k^2 W \tag{110}$$

This equation requires 6 BCs on $W$, of which we already have 4 from eq. (105). Two more BCs can be obtained by plugging the BC values of $\Theta$ near the walls into eq. (106). In the end, we get:

$$\begin{bmatrix} W \\ DW \\ \left(D^2 - k^2\right)^2 W \end{bmatrix}(x, -{}^1\!/_2, z, t) = \begin{bmatrix} W \\ DW \\ \left(D^2 - k^2\right)^2 W \end{bmatrix}(x, +{}^1\!/_2, z, t) = 0. \tag{111}$$

Because the BCs at the lower- and upper-walls are identical, and we have an *even* differential operator in the LHS of eq. (110), the *ordinary differential equation* (ODE) will have *even* ($W_e$) and *odd* ($W_o$) solutions, of the form:

$$W_e(y) = \sum_{i=1}^{3} A_i \cosh(q_i y) \quad \text{and} \tag{112}$$

$$W_o(y) = \sum_{i=1}^{3} B_i \cosh(q_i y), \tag{113}$$

where it can be shown (by plugging the solutions into the ODE) that, the coefficients $q_i$ need to be the roots of:

$$(q^2 - k^2)^3 + k^2\, \mathrm{Ra} = 0 \tag{114}$$

It can be shown (see Reid and Harris (1958) or Chandrasekhar (1981) for details) that, in order to satisfy the BCs, the following equations must hold:

$$-q_0 \tan\left(\frac{q_0}{2}\right) = \frac{\left(q_1 + q_2\sqrt{3}\right)\sinh(q_1) + \left(q_1\sqrt{3} - q_2\right)\sin(q_2)}{\cosh(q_1) + \cos q_2} \quad \text{for even modes, and} \tag{115}$$

$$\frac{q_0}{\tan\left(\frac{q_0}{2}\right)} = \frac{\left(q_1 + q_2\sqrt{3}\right)\sinh(q_1) - \left(q_1\sqrt{3} - q_2\right)\sin(q_2)}{\cosh(q_1) - \cos(q_2)} \quad \text{for odd modes,} \tag{116}$$

where we denote:

- $\lambda \equiv \sqrt[3]{\frac{\mathrm{Ra}^2}{k^6}}$,

- $q_0 \equiv k\sqrt{\lambda - 1}$,
- $q_1 \equiv \frac{k}{\sqrt{2}}\sqrt{\sqrt{1+\lambda+\lambda^2} + \left(1+\frac{\lambda}{2}\right)}$ and
- $q_2 \equiv \frac{k}{\sqrt{2}}\sqrt{\sqrt{1+\lambda+\lambda^2} - \left(1+\frac{\lambda}{2}\right)}$.

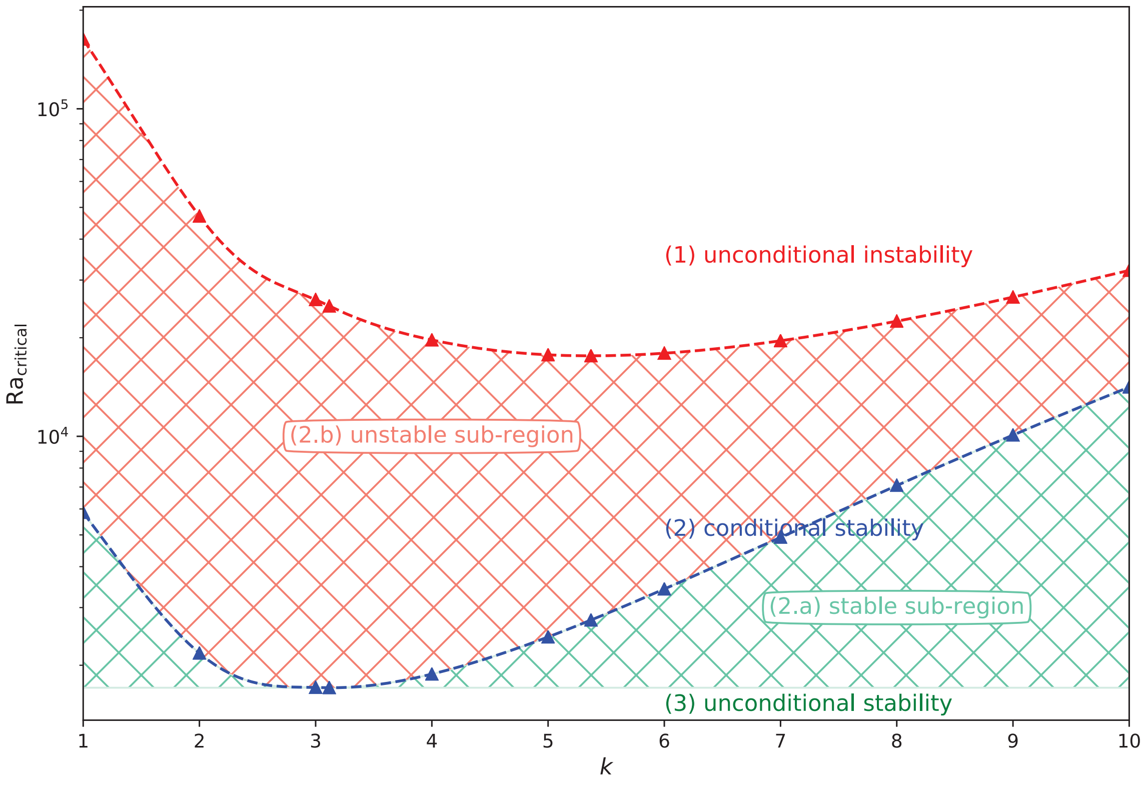


Figure 3: Plot of the critical Rayleigh number as a function of the wavelength of the perturbations. The dotted lines represent the marginal-stability curves for the even (blue) and odd (red) modes, respectively. The numbered labels are used to mark the different regions of the parameter-space.

Equations (115) and (116) are transcendental algebraic equations, and can only be solved numerically. Figure 3 illustrates the dependence of the critical Ra on the wavenumber k (Table 1 lists the exact values).

The main outcome of the analysis is that the quiescent (rest) state becomes unstable for $\boxed{\mathrm{Ra} \geqslant \mathrm{Ra}_{cr} = 1707.762}$, with respect to the even mode with $\boxed{k_{cr} = 3.117}$. These reference values are used in later parts of this thesis, for validation of the numerical models.

This first instability marks the transition from the rest state to a state of stationary (time-*in*dependent) flow. Directly at the transition boundary, the stable convective solution consists of adjacent and parallel rolls, whereby the motion can be considered effectively two-dimensional ( Figure 4). This pattern is also known in the literature as *ideal straight rolls* (ISR).

Another interesting outcome of the analysis is that there is a range (hatched region), where the stability of the flow depends on the wavenumber of the perturbation. Such a situation is common for many flows (Manneville, 2004).

Table 1: Wavenumber (k) and Rayleigh number ($Ra_{cr}$) for the first even and odd unstable modes in the *Rayleigh-Bénard* (RB) problem. The highlighted values indicate the minimum critical values for the even and odd modes.

| | Ra | |
|---|---|---|
| k | *even* modes | *odd* modes |
| 1 | 5854.4848 | 163127.60 |
| 2 | 2177.4121 | 47005.616 |
| 3 | 1711.2771 | 26146.630 |
| $\boxed{3.117}$ | $\boxed{1707.762}$ | 24982.076 |
| 4 | 1879.2560 | 19684.585 |
| 5 | 2439.3217 | 17731.529 |
| $\boxed{5.369}$ | 2747.5681 | $\boxed{17610.412}$ |
| 6 | 3417.9831 | 17933.046 |
| 7 | 4918.5417 | 19575.824 |
| 8 | 7084.5093 | 22461.475 |
| 9 | 10089.629 | 26599.707 |
| 10 | 14134.445 | 32104.101 |

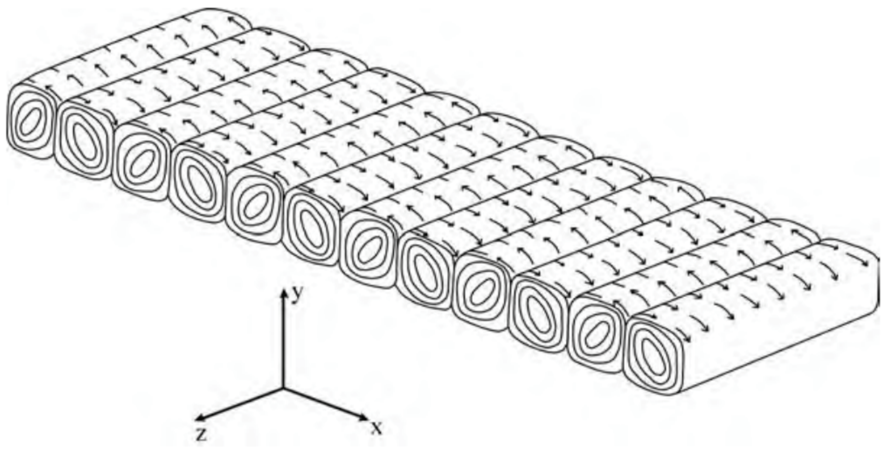


Figure 4: Illustration of the flow directly after first transition (source: Lappa (2010)). Original caption: *"Schematic diagram of rolls in Rayleigh-Bénard convection: for a laterally unlimited domain, the fluid motion is regular and organized as a set of horizontal rolls (arrows indicate the direction of fluid flow; the wavelength of the pattern is approximately equal to twice the layer height* d*)"*

#### 2.4.4.3 *Secondary and higher-order modes for RB convection*

For higher values of the control-parameter (Ra), an IC consisting of parallel rolls also becomes unstable (secondary instabilities), and evolves into more complex states. Similarly, these states may themselves be subject to even higher-order instabilities, which further complicate the dynamics, until the flow attains the chaotic character of turbulent flows.[22]

In principle, the onset of the secondary and higher-order instabilities may also be studied using approaches similar to those outlined in the previous sub-section, for the first instability. In practice, however, this is feasible only for very specific cases, because the mathematical difficulty usually increases exponentially with the degree of the instability. Below we provide a qualitative overview, based on the currently-available literature (see Lappa (2010) for a detailed review).

Busse (1978) was the first to provide a reasonably-complete diagram (Figure 5) of the parameter-ranges where the stationary ISR pattern is stable.

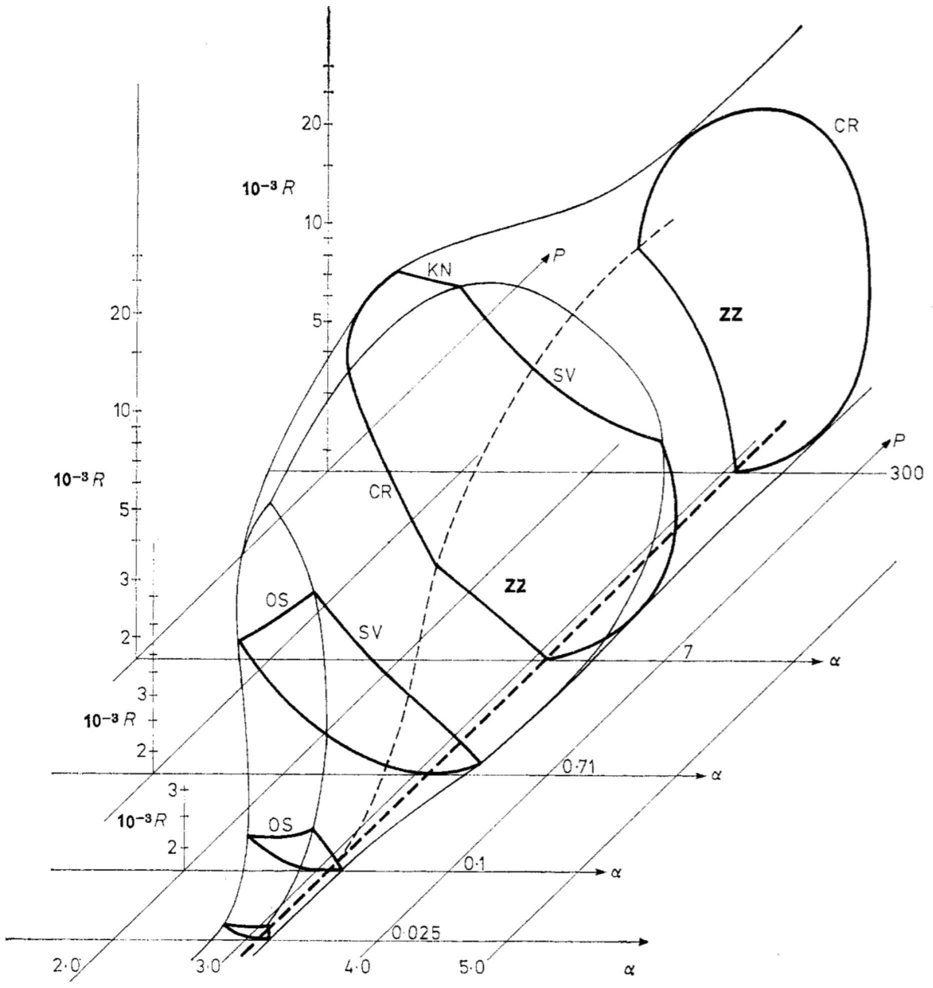


Figure 5: The Busse balloon (source: Busse (1978)). Original caption: *"Region of stable convection rolls in the three-dimensional* $R - P - \alpha$ *space. The thick curves represent computed stability boundaries for the oscillatory (OS), the skewed varicose (SV), the cross-roll (CR), the knot (KN), and the zig-zag (ZZ) instabilities. The other curves represent approximate interpolations from results of Busse and Clever (1979). The stability boundary for* $P = 300$ *actually represents the computations of Busse (1967) for* $P = \infty$ *which are expected to give a good approximation for* $P = 300$.*"* Note: the parameters P, R and $\alpha$ correspond to Pr, Ra and k respectively in this thesis.

22 The more modes we have to consider, the more appealing a purely statistical description becomes.

An important first observation is that, while the value of Pr did not play a role for the first instability, it becomes *crucial* for the secondary- and higher-order instabilities. Another remarkable feature is that, depending on the $(\mathrm{Ra}, \mathrm{Pr}, k)$ values, the instabilities exhibit very different patterns. The most prevalent such patterns are briefly summarized below:

- **Zig-zag (ZZ) instability** (present for medium- to high-values of Pr) This type of instability occurs when the wavenumber of the ISR is less-than-optimal. By deforming the rolls in a zig-zag fashion, the effective wavenumber is increased.
- **Eckhaus (EK) instability** (present for low-values of Pr) This type of instability is characterized by compression and expansion of adjacent rolls.
- **Cross-roll (CR) instability** (present for medium- to high-values of Pr) This type of instability is characterized by the appearance of regions with rolls that are oriented perpendicularly to the axis of the initial rolls. At high Pr, these rolls become very strong inside the thermal boundary layers, leading to a transition to bimodal convection.
- **Oscillatory (OS) instability** (present for low- to medium-values of Pr) This type of instability is similar to zig-zag, except that the bending of the ISR propagate along the roll-axis.
- **Skewed-varicose (SV) instability** (present for low- to medium-values of Pr) This type of instability manifests as a dislocation of the ISR, along a line which is oblique with respect to the original roll-axis.

When comparing simulation or theoretical results with experimental data, it should be noted that these instabilities can be strongly influenced by defects in the initial ISR pattern (which are always present in nature). These effects can complicate the actual dynamics even further (see Lappa (2010) for a comprehensive review of the patterns that may be present in such cases).

#### 2.4.4.4 *Transition to turbulence in RB convection*

As already mentioned, thermal convection is a very convenient system for studying the transition to turbulence, because the flow becomes more-and-more complex, through relatively well-defined transitions, as the Rayleigh number is increased. Therefore, before closing our theoretical discussion of the RB problem, it is fitting to provide at least a qualitative overview of the cascade towards turbulent flow. For this purpose, the summary of Busse (1978) is sufficient.

As seen in Figure 6, the Prandtl number becomes important, as soon as Ra exceeds the $\mathrm{Ra}_{\mathrm{cr}}$ value of the first transition (from pure diffusion to the steady ISR pattern). The physical reason for this situation is that the nonlinear terms in the temperature equation are dominating for high-Pr fluids, whereas the nonlinear terms in the momentum equation are more important for low-Pr fluids. Therefore, it is appropriate to distinguish at least the following two regimes:

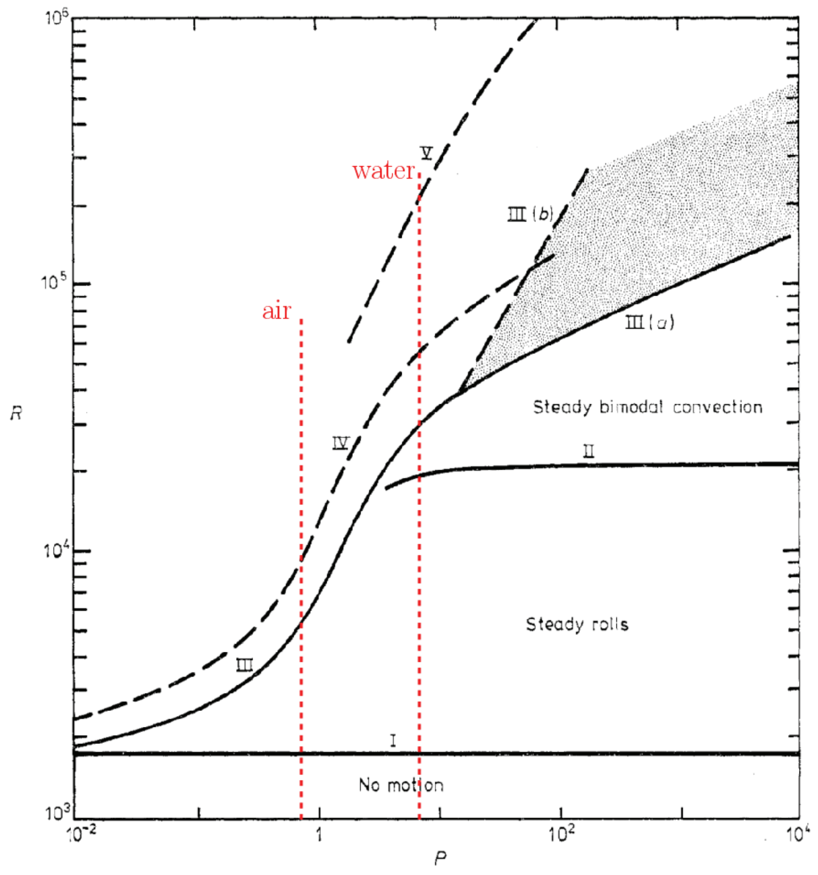


Figure 6: Different flow-regimes for the RB problem (source: Busse (1978)). Original caption: *"Transitions in convection as a function of Rayleigh and Prandtl numbers after Krishnamurti (1973) and others. The curves indicate the onset of steady rolls (I), three-dimensional convection pattern (II), time-dependent convection (III) in isolated spots (III(a)) and uniformly throughout the layer (III(b)), and turbulent convection (IV)."* The original figure was annotated with the red dashed lines, indicating the approximate corresponding positions of water (Pr = 7) and air (Pr = 0.71).

a) **High** Pr **fluids**: Historically, convection in fluids fitting this category was investigated first, because the longer time-scales make experiments easier to setup.

   As Ra is increased, such fluids tend to develop the stationary ISR pattern (first transition), as usual. Then, a time-independent three-dimensional pattern appears (known as "bimodal convection"), where secondary rolls (oriented at 90° relative to the ISR) appear in the thermal boundary-layers.

   For higher Ra, the flow becomes oscillatory, whereby the thermal boundary layers periodically break down and re-appear.

   Finally, for even higher Ra, the periodicity is lost, and the flow is said to have become turbulent.

b) **Low- to medium-**Pr **fluids**: As seen in Figure 6, the succession of regimes occurs in much narrower Ra-intervals for lower values of Pr. In fact, these intervals may even be said to be overlapping, which leads to much more complex dynamics – already for the secondary transitions there are more types of instabilities that can occur (refer back to Figure 5). This complexity has caused difficulties for experimental and numerical investigations, causing our understanding of these regimes to be less complete.

Generally-speaking, for such fluids we have thin viscous boundary-layers near the horizontal plates, which are required to satisfy the no-slip BCs.

As Ra is increased, these fluids also develop the stationary ISR pattern. However, for higher values of Ra, three-dimensional periodicity sets in (skipping the time-*in*dependent regime that was present for high-Pr). Beyond this point, the flows quickly become turbulent, even for small increases of Ra.

## 2.5 OVERVIEW OF THE PHYSICS OF TURBULENCE

As we observed in the previous discussion about the RB problem, fluid flows can exhibit qualitatively different regimes as the control-parameters (such as Ra) are varied: no motion, laminar flow, periodic regime, quasi-periodic and (eventually) turbulent. In this section, we provide a brief overview of the current status of research on the latter regime (turbulent), and of approaches for modeling such phenomena numerically. These topics are directly relevant for analyzing the results of simulations presented in Chapter 5.

Considering again eqs. (31) to (33), it is well-known that in special cases (e.g. for sufficiently-low velocities[23]) they admit *laminar* solutions, when the flow occurs in "sheets" that slide past one another, often with no dependence on time. In such situations it is sometimes even possible to obtain complete analytic solutions for the flow-fields. However, most flows encountered in nature and in engineering problems are *turbulent*, where the flow-fields are very complex and time-dependent. Understanding turbulent flow is therefore very important, due to the vast range of applications, as well as an intellectual challenge.

Although this phenomenon has been recognized for a long time[24] and the governing equations have been known for over a century, our current understanding of turbulence still leaves much room for improvement.[25]

### 2.5.1 *Defining turbulent flows*

A precise definition of turbulent flow is still lacking – what we have instead are several intuitive descriptions. For example, a popular (physical) formulation is attributed to Richardson (1922):

23 How low the velocities have to be depends on specific details of the problem, such as the geometry and the transport-coefficients of the fluid.

24 For example, Leonardo da Vinci provided a qualitative description of what he called "turbolenza", more than 500 years ago.

25 The "problem of turbulence" is often said to be the last open problem of classical physics.

**Richardson's description of turbulence**

"Big whorls have little whorls,[a]
which feed on their velocity;
And little whorls have lesser whorls,
And so on to viscosity
(in the molecular sense)."

[a] The word "whorl" is equivalent to the more modern term "eddy", for which a precise definition is also missing (intuitively, and eddy is defined as *"a turbulent motion, localized within a region of size* $\mathfrak{l}$, *that is at least moderately coherent over this region. The region occupied by a large eddy can also contain smaller eddies."* Pope (2000)).

For several decades, the (statistical) description due to Hinze (1959) was also popular:

**Hinze's description of turbulence**

"Turbulent fluid motion is an irregular condition of the flow in which the various quantities show a random variation with time and space coordinates, so that statistically distinct average values can be discerned."

Finally, a more modern description (which takes into account recent progress in the theory of nonlinear dynamical systems) was given by Chapman and Tobak (1985):

**Chapman's description of turbulence**

"Turbulence is any chaotic solution to the 3-D Navier-Stokes equations that is sensitive to initial data and which occurs as a result of successive instabilities of laminar flows as a bifurcation parameter is increased through a succession of values."

Despite the apparent lack of consensus on what exactly turbulence is, several characteristics of the phenomenon *are* universally-accepted:

- ***sensitivity to initial conditions* (SIC)**: When the flow is turbulent, even minute variations of the ICs or BCs can lead to vastly different flow-fields at later times. Since such variations are unavoidable in reality, experiments involving turbulent flows are fundamentally non-repeatable in a strict sense (although some statistical properties usually remain the same for the various repetitions).

- **chaotic behavior and intermittency**: If, for example, we measure fluid-properties at a specific location, the resulting time-series will be chaotic (irregular). For decades, this caused many investigators (Hinze, 1959) to treat turbulence as a random process. However, it should not be forgotten that (within the limits of the continuum hypothesis) the phenomenon is still deterministic. Also, some flows may exhibit turbulence only in some localized regions (so-called "turbulent patches") and/or only sometimes – a phenomenon known as *intermittency*.

A famous example of a system which exhibits both chaos and intermittency (despite being completely deterministic) is the Lorenz system (Lorenz, 1963), which is based on a simplified model for convection in the atmosphere.

- **wide range of space- and time-scales, and an "energy cascade"**: As explicitly acknowledged in Richardson's description, a turbulent flow normally contains vortices of various sizes; mechanical energy (which is needed to sustain the turbulent regime) generally enters the system at the larger space-scales (where advection-effects dominate) and is eventually lost in the form of heat at the smaller scales (where dissipation-effects dominate).
- **enhanced mixing (diffusivity)**: From a practical point of view, a crucial characteristic of turbulence is its increased efficiency for mixing (tendency to homogenize) momentum, temperature, or solvent concentrations, which is e.g. desirable for many industrial applications, or for reducing the toxicity of emissions to the atmosphere.
- **enhanced dissipation**: Turbulent flow are also much more effective at dissipating mechanical energy than laminar flows, which can be a nuisance for some applications (e.g. fluid transport through pipes), but also beneficial (e.g. viscous dissipation of turbulence kinetic energy in storms (Businger and Businger, 2001)).
- **three-dimensionality**: Last but not least, it is commonly-accepted that turbulent flows need to be three-dimensional, because the primary mechanism for propagation of turbulent behavior is *vortex stretching* (which cannot occur in 2D).[26]

### 2.5.2 *Important results from turbulence theory*

#### 2.5.2.1 *Existence of critical control-parameters*

Some of the first quantitative observations of turbulence are due to Reynolds (1883), who studied the evolution of flows in cylindrical pipes. By inserting a streak of dye into the flow, Reynolds observed that in some cases the dye formed a smooth filament (with very little diffusion). By increasing the velocity ($U$) of the flow, the filament began to oscillate downstream; for even higher velocities, the filament lost its identity, and the dye became spread throughout the volume of the fluid. By experimenting with fluids of different viscosities ($\nu$) and pipe-diameters ($D$), Reynolds found that which of these regimes is realized only depends on the dimensionless quantity:

$$\mathrm{Re} \equiv \frac{UD}{\nu}, \tag{117}$$

26 Note, however, that this does not imply that chaotic solutions cannot be obtained for 2D flows – on the contrary, these are commonplace e.g. in numerical models which are restricted to two dimensions. However, such chaotic results should be regarded more as model artifacts, and not as physically-relevant turbulence results.

which is now known as the *Reynolds number*. Re is an example of a *control parameter* which (w.r.t. transitions to turbulence) is analogous to the Rayleigh number (Ra) from the previous section. Physically, it can be interpreted as the ratio of inertial- to viscous-forces within the fluid (which explains intuitively why high-Re flows tend to be turbulent, while low-Re flows tend to be laminar).

#### 2.5.2.2 *Breaking of symmetries and their restoration in the statistical sense*

Throughout the earlier discussion on the RB problem, we mentioned the tendency of the flows to become increasingly complex, as the value of Ra is increased; such tendencies are also encountered for other fluid-flow problems. From a topological point of view, the increase of complexity may be associated with a sequence of symmetry-breaking events, which make the flows increasingly difficult to predict analytically. Interestingly however, as the flows become more turbulent (high Ra or high Re limits), there is a tendency to restore these symmetries *in a statistical sense* – see Frisch (1995) for a detailed discussion. This observation has encouraged research into these regimes (known as *fully developed turbulence*), due to the physicists' penchant for symmetries but (more importantly) also because such regimes are ubiquitous in nature. Some of these results are presented in the next section.

It should be mentioned that many shear flows (e.g. plane Couette flow and pipe flow) have been found to be turbulent even for parameter-ranges where the linear stability theory predicts a laminar flow. This phenomenon was recently analyzed by Grossmann, Eckhard and their co-workers (see e.g. Grossmann (2000) and Eckhardt (2012) and the references therein). An important outcome of these studies was the discovery of an alternative route to turbulence, which is driven from the start by nonlinear interactions between perturbations and the basic laminar flows.

#### 2.5.2.3 *The energy-cascade and the Kolmogorov-Obukhov theory for high-*Re *flows*

Richardson's intuitive picture of turbulence as a hierarchy of eddies was first put on more quantitative foundations by Kolmogorov (1941b,a,c) and Obukhov (1941a,b). Here we provide a summary[27] of their results, which are relevant for analyzing the turbulent-flow numerical results presented in this thesis.

As already mentioned, a turbulent flow consists of a superposition of eddies of different sizes. To be more precise, for each eddy we associate a *characteristic length-scale* $\boxed{l}$ and a *characteristic velocity* $\boxed{u(l)}$; by combining these scales, we can also define a *characteristic time-scale* $\boxed{\tau(l) \equiv l/u(l)}$ and (by also making use of the viscosity of the fluid) also an eddy-scale Reynolds number $\boxed{\mathrm{Re}_l \equiv \frac{u(l)l}{\nu}}$.

In the *Kolmogorov-Obukhov* (K-O) theory, the eddy motions are classified into three scale-ranges (see Figure 7), with mechanical energy being *introduced into the system* at the largest (energy-containing) range, and eventually being *lost as heat* at the smallest

27 See Pope (2000) for a more detailed discussion. Also, Frisch (1995) provides a more complete overview of this theory, along with expanded proofs.

(dissipative) range. The intermediate (inertial) range acts effectively as a "channel" (in frequency-space) between the other two scales.

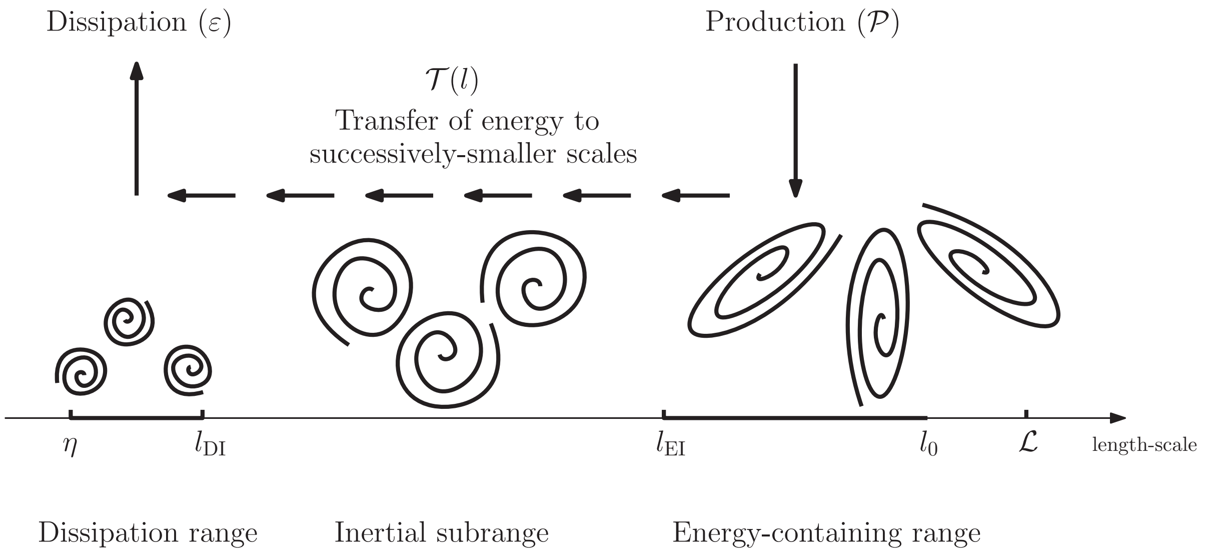


Figure 7: Schematic of the energy cascade for high-Re flows (adapted after Fig. 6.2 of Pope (2000)). Note the difference in eddy-sizes and isotropy.

THE LARGE (ENERGY-CONTAINING) RANGE Eddies belonging to this range have *length-scales* of the same order of magnitude as the geometric scale of the domain ($l_0$). The *velocity-scale* is e.g. similar to the velocities imposed by the BCs. Since both of these scales are usually large, the eddy-local Re is also large, which means that viscosity effects are negligible in this range. Also important to note is that at this scale the eddies are usually anisotropic, due to the influence of the geometry.

THE SMALL (DIFFUSIVE) RANGE When the system is in equilibrium (in a statistical sense), the energy injected at the large scales will have to be eventually dissipated as heat. This diffusion-dominated range is characterized by small eddies, with small velocities. Kolmogorov was the first to provide estimates for these scales (starting from a hypothesis of local similarity), namely:

$$\begin{cases} \eta = \left(\dfrac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}} \quad , & (118) \\ u_\eta = (\varepsilon\nu)^{\frac{1}{4}} \quad , \quad \text{and} & (119) \\ \tau_\eta = \left(\dfrac{\nu}{\varepsilon}\right)^{\frac{1}{2}} \quad , & (120) \end{cases}$$

where $\varepsilon$ is the *rate of dissipation of turbulent kinetic energy* (also referred to as "the dissipation"). Physically, it represents the rate at which larger eddies provide energy to smaller eddies.

The quantities $\eta$, $u_\eta$ and $\tau_\eta$, which are now known as the *Kolmogorov scales*, are directly related to the need for turbulence modeling in numerical simulations, because they provide estimates for the space- and time-resolution requirements for capturing all physical scales in the flow. To see this more clearly, we can estimate $\varepsilon$ based on the energy-containing scales, namely:

$$\varepsilon \approx \frac{u_0^3}{l_0} \quad , \tag{121}$$

which leads to the following scale-ratios:

$$\begin{cases} \dfrac{l_0}{\eta} = \mathrm{Re}^{3/4} \quad , & (122) \\ \dfrac{u_0}{u_\eta} = \mathrm{Re}^{1/4} \quad , \quad \text{and} & (123) \\ \dfrac{\tau_0}{\tau_\eta} = \mathrm{Re}^{1/2} \quad . & (124) \end{cases}$$

The scale-ratios for space and time in particular indicate that, the higher Re is, the larger the number of mesh-points becomes. As the number of mesh-points is increased, it is usually also necessary to increase the temporal resolution, to keep the simulations stable according to the *Courant-Friedrichs-Levy* (CFL) criterion. Equation (124) also gives such a requirement, but with a smaller exponent.

Finally, it is interesting to note that the Reynolds number for eddies in the diffusive range is of the order of unity:

$$\mathrm{Re}_\eta = \frac{u_\eta \eta}{\nu} = 1 \tag{125}$$

THE INTERMEDIATE (INERTIAL) RANGE Between the energy-containing- and diffusive-ranges, we have an intermediate range, for which the convective term of the *Navier-Stokes equations* (NSE) is still at least an order-of-magnitude larger than the viscous term. While this is also the case for the energy-containing-range, the eddies in this so-called inertial range have a more universal character (i.e. they are independent of the details of the flow-geometry).

In the inertial range, the only important parameter is the dissipation-rate ($\epsilon$). Using just this one parameter, it is not possible to derive characteristic scales for space, time, and velocity. However, it is possible to construct scaling-laws for the characteristic velocity- and time-scales, for a given eddy length-scale ($l$) in the inertial-range:

$$u(l) = (\varepsilon l)^{\frac{1}{3}} = u_\eta \left(\frac{l}{\eta}\right)^{\frac{1}{3}} \tag{126}$$

$$\tau(l) = \left(\frac{l^2}{\varepsilon}\right)^{\frac{1}{3}} = \tau_\eta \left(\frac{l}{\eta}\right)^{\frac{2}{3}} \tag{127}$$

Notice that both scales decrease as $l$ decreases.

Another important result (derived independently by both Kolmogorov and Obukhov in 1941) concerns the distribution of energy as a function of the wavenumber $\kappa = \frac{2\pi}{l}$ for the inertial range:

$$E(\kappa) = C_K \varepsilon^{2/3} \kappa^{-5/3}, \qquad \text{for} \quad \frac{1}{l_0} \ll \kappa \ll \frac{1}{\eta}, \tag{128}$$

where $C_K$ is empirical quantity (known as the *Kolmogorov constant*). The original derivations (by Kolmogorov and Obukhov) of eq. (128) suffered from relying on some assumptions which are not necessarily true for many flows; however, the equation still holds (and was later derived from more rigorous principles by Frisch (1995)).

### 2.5.3 *Approaches for quantifying and predicting turbulent flows*

#### 2.5.3.1 *Statistical tools for analyzing flows*

Although the deterministic NSE are widely accepted as a good model for describing the evolution of fluid flows, it is beneficial to introduce a *statistical* description of these phenomena. This is motivated by practical considerations, which are ultimately rooted in the *sensitivity to initial conditions* (SIC) property of turbulent flows – even if we would be able to obtain a complete knowledge[28] of the instantaneous flow-fields for one experiment, this would tell us nothing about the instantaneous flow-fields for a repetition of the experiment. On the other hand, this does not mean that efforts to obtain high-resolution data (numerically or experimentally) are worthless endeavours – due to the *ergodic hypothesis*,[29] such high-resolution data can be used to infer statistical properties of an *ensemble of experiments* (which is often what we are ultimately interested in).

In the interest of the later discussion about turbulence modeling, the definitions of some of these statistical tools is provided below (see (Lumley, 1970) for an extensive review).

STATISTICAL MOMENTS OF AN ENSEMBLE If we consider an *ensemble* (set) of N identically-prepared repetitions of the experiment, the $m^{th}$ statistical moment of the random variable U at space-time position $(\mathbf{x}, t)$ is defined as:

$$\overline{U^m}(\mathbf{x}, t) \equiv \frac{1}{N} \sum_{i=1}^{N} U^m(\mathbf{x}, t : i), \tag{129}$$

where $U(\mathbf{x}, t : i)$ is the $i^{th}$ realization of the experiment.

28 Such complete knowledge is impossible, because: a) closed-form mathematical solutions for turbulent flows are not existing, b) it is impossible to measure (and to store) experimentally the flow-fields with infinite accuracy and for continuous time-intervals, while c) numerical solutions are always imprecise, due to the finite sizes of the meshes, numerical errors, and limited accuracy with which floating-point numbers are represented in the computer.

29 This hypothesis states that time-averages converge to ensemble means, as the averaging time-intervals tend to infinity.

In the (purely theoretical) limit of an infinite number of ensemble-members, we obtain the *expectation value* of the moment, which we denote by angle-brackets:

$$\langle U(\mathbf{x},t)\rangle \equiv \lim_{N\to\infty} \overline{U^m}(\mathbf{x},t) \tag{130}$$

Interestingly, the ensemble-averaging operation commutes with a) differentiation and integration, with respect to space and time, b) addition, and c) multiplication by a constant (but *not* with multiplication with another random variable). These properties will be used extensively in later calculations shown in this thesis.

**Estimating moments from a single experiment**

Because in many situations (e.g. large-scale measurements of climate variables) it is not possible to prepare nearly-identical ensembles, we might still want to estimate the moments even based on a single experiment. This is possible in certain circumstances:

- If the random variable U is *stationary in time*, we can replace the ensemble-average in eq. (129) with a time-average:

$$\overline{U^m}(\mathbf{x},t) \approx \frac{1}{2\Delta t}\int_{t-\Delta t}^{t+\Delta t} U^m(\mathbf{x},t)dt'. \tag{131}$$

- Similarly, if the random variable is *stationary in space*, we can use volume-averaging:

$$\overline{U^m}(\mathbf{x},t) \approx \frac{1}{\Delta V}\int_{\Delta V} U^m(\mathbf{x},t)dV. \tag{132}$$

In practice, the most used moment is the first one (also known as *mean*, or *average*), which gives us *general information about the value of* U:

$$\overline{U}(\mathbf{x},t) \equiv \frac{1}{N}\sum_{i=1}^{N} U(\mathbf{x},t:i) \tag{133}$$

However, typically we are also interested in the *properties of the turbulent fluctuations*. This information is contained within the higher-order moments – for example, the *root-mean square* (r.m.s.) value is derived from the 2nd moment:

$$U_{rms} \equiv \sqrt{\overline{U^2}} \tag{134}$$

To focus on the variability, it is useful to factor-out the information about the mean, by defining the *higher-order central moments*:

$$\overline{(U-\overline{U})^m} \equiv \frac{1}{N}\sum_{i=1}^{N}\left(U(\mathbf{x},t:i) - \overline{U(\mathbf{x},t)}\right)^m \tag{135}$$

The most important central moments of order higher than one[30] are (2nd moment) the *variance*[31] $\overline{(U-\overline{U})^2}$, (3rd moment) the *skewness* $\overline{(U-\overline{U})^3}$, and (4th moment) the *kurtosis* $\overline{(U-\overline{U})^4}$.

CORRELATION, AUTOCORRELATION AND SPECTRA While the moments defined above (normal or central) do provide valuable information about the random variable, they do not tell us how values at different times or at different positions in space are related to each other. For such purposes, we need to study the so-called correlations.

Given two random variables $U_i$ and $U_j$, sampled at different space-time locations $(\mathbf{x}_1, t_i)$ and $(\mathbf{x}_2, t_2)$, their *covariance* is usually defined as the ensemble-average of the product of the fluctuating parts of the variables:

$$R_{ij}\left(\mathbf{x}_1, t_1, \mathbf{x}_2, t_2\right) \equiv \overline{\left(U_i(\mathbf{x}_1, t_1) - \overline{U_i}\right)\left(U_j(\mathbf{x}_2, t_2) - \overline{U_j}\right)}. \tag{136}$$

By normalizing the covariance by the product of the standard deviations ($\sigma_{U_i}$ and $\sigma_{U_j}$, respectively), we obtain the *correlation coefficient*:

$$r_{ij}\left(\mathbf{x}_1, t_1, \mathbf{x}_2, t_2\right) \equiv \frac{R_{ij}\left(\mathbf{x}_1, t_1, \mathbf{x}_2, t_2\right)}{\sigma_{U_i}\sigma_{U_j}}. \tag{137}$$

The more specialized terms *auto-correlation* and *cross-correlation* are used when $i = j$ and $i \neq j$, respectively. The variables are said to be: (a) *strongly correlated (or strongly anti-correlated)* if $r_{ij} \lessapprox 1$ (or $r_{ij} \gtrapprox -1$), (b) *weakly correlated (or weakly anti-correlated)* if $0 < r_{ij} \ll 1$ (or $-1 \ll r_{ij} < 0$), and (c) *uncorrelated* if $\|r_{ij}\| \approx 0$.

The concepts above may be further specialized, depending on the characteristics of the flow, and on which aspects we are interested in:

- If both random variables are *statistically-stationary in time*, it is interesting to look at the correlation between them at fixed points in space, in which case the notation for the covariance may be simplified to:

$$R_{ij}(\tau) = \overline{\left(U_i(t) - \overline{U_i}\right)\left(U_j(t+\tau) - \overline{U_j}\right)}, \tag{138}$$

  The correlation becomes:

$$r_{ij}(\tau) = \frac{R_{ij}(\tau)}{\sigma_{U_i}\sigma_{U_j}}, \tag{139}$$

  where $\tau \equiv t_2 - t_1$ is the *time lag*. In this case, $r_{ij}(\tau)$ is also known as the *lag-dependent correlation coefficient*, for which we have in general $\lim_{\tau\to\infty} r_{ij}(\tau) = 0$. The value $\tau = t_c$ where the *auto-correlation* $r_{11} = 0$ is known as the *correlation time*, which gives an estimate for the equivalent number of ensemble-members (when time-averaging is used instead of ensemble-averaging), namely

30 By definition, the first moment is zero.

31 The square-root of the variance is known as the *standard deviation*: $\sqrt{\overline{(U-\overline{U})^2}}$.

$N \approx \Delta t / t_c$. Other interesting time-scales that may be derived are the *integral time-scale* $\Lambda_t$ and the *Taylor microscale* $\lambda_t$ (see e.g. Kundu et al. (2012) for details), which are measures of the *memory* of the turbulent flow.

- Similarly, if both random variables are *statistically-stationary in space*, it is interesting to study the correlation between them at a fixed time, when the notation for the covariance may be simplified to:

$$R_{ij}(\mathbf{r}) = \overline{\left(U_i(\mathbf{x}) - \overline{U_i}\right)\left(U_j(\mathbf{x}+\mathbf{r}) - \overline{U_j}\right)}, \tag{140}$$

where $\mathbf{r} \equiv \mathbf{x}_2 - \mathbf{x}_1$.

The correlation then becomes:

$$r_{ij}(\mathbf{r}) = \frac{R_{ij}(\mathbf{r})}{\sigma_{U_i}\sigma_{U_j}}, \tag{141}$$

Returning to the general case, the covariance defined by eq. (136) above may be used to extract information about the scales of eddies within the flow. A first step is to compute the *velocity spectrum tensor* $\Phi_{ij}(\boldsymbol{\kappa}, t)$, which is defined as the Fourier transform of the two-point correlation:

$$\Phi_{ij}(\boldsymbol{\kappa}, t) \equiv \frac{1}{(2\pi)^3} \iiint\limits_{\mathbb{R}^3} \exp^{-i\boldsymbol{\kappa}\cdot\mathbf{r}} R_{ij}(\mathbf{r}, t)\, d^3\mathbf{r}, \tag{142}$$

where $d^3\mathbf{r} \equiv dr_1 dr_2 dr_3$. The spatial Fourier mode $\exp^{-i\kappa\cdot x}$ is a periodic function (with wavelength $l = \frac{2\pi}{|\boldsymbol{\kappa}|}$, related to the size of the corresponding eddies), which varies sinusoidally along the direction of $\boldsymbol{\kappa}$ (where $\boldsymbol{\kappa}$ is the *wavenumber vector*).

By integration over spherical layers in $\boldsymbol{\kappa}$-space and dividing by two, we obtain the *energy spectrum function*, which depends only on the magnitude ($\kappa$) of the wavevector:

$$E(\kappa, t) \equiv \frac{1}{2} \iiint\limits_{\mathbb{R}^3} \Phi_{ii}(\boldsymbol{\kappa}, t)\delta\left(|\boldsymbol{\kappa}| - \kappa\right) d^3\kappa, \tag{143}$$

where $d^3\boldsymbol{\kappa} \equiv d\kappa_1 d\kappa_2 d\kappa_3$.

#### 2.5.3.2 *The Reynolds-averaged Navier-Stokes (RANS) equations*

In addition to establishing experimentally the existence of critical control-parameters (such as the Reynolds number), another important contribution of Reynolds was to adopt a statistical approach for analyzing turbulent flow, by decomposing the flow-fields into mean an fluctuating parts. This approach (which was later improved by Taylor (1935) by introducing correlations and Fourier transforms) formed the basis for much of the following efforts on modeling turbulence.

The fields that specify the state of the fluid are written as:

$$\begin{cases} \mathbf{u} = \langle \mathbf{u} \rangle + \mathbf{u}' & (144) \\ p = \langle p \rangle + p' & (145) \\ T = \langle T \rangle + T' & (146) \end{cases}$$

where $\langle \chi \rangle$ denotes the (ensemble-averaged) *mean part*, and $\chi'$ the *fluctuating part*, $\forall \chi \in \{\mathbf{u}, p, T\}$.

For many applications, it is sufficient to obtain information about the mean parts only (which are less costly to compute than the fluctuating parts). To obtain the equations for the mean components, we introduce the Reynolds decomposition eqs. (144) to (146) into the dynamic eqs. (31), (33) and (41).

In order to make the discussion more directly relevant for the RB convection simulations presented later in this thesis, we will also include the forcing-term into the analysis, as we sketch the derivation of the corresponding *Reynolds-averaged Navier-Stokes* (RANS) equations. Also, we use the *physical* coordinates here, which will make the interpretation of the fluctuating terms more clear towards the end of the discussion.

CONTINUITY EQUATION By inserting the decomposition into the continuity eq. (31), and using the properties of the ensemble-averaging operator, we obtain:

$$\frac{\partial \langle u_i \rangle}{\partial x_i} + \frac{\partial u_i'}{\partial x_i} = 0 \tag{147}$$

By taking the ensemble-average of this equation, we eventually obtain:

$$\frac{\partial \langle u_i \rangle}{\partial x_i} = 0 \tag{148}$$

Finally, by subtracting eq. (148) from eq. (147), we obtain:

$$\frac{\partial u_i'}{\partial x_i} = 0 \tag{149}$$

In other words, the *zero-divergence condition* holds for both the *mean* and for the *fluctuating* parts of the velocity individually.

MOMENTUM EQUATION Similarly, by inserting the decomposition into the momentum eq. (41), and using the properties of the ensamble-averaging operator, we obtain (after some algebra) the momentum equation for the mean part:

$$\frac{\partial \langle u_j \rangle}{\partial t} + \langle u_i \rangle \frac{\partial \langle u_j \rangle}{\partial x_i} + \frac{\partial \langle u_i' u_j' \rangle}{\partial x_i} = -\frac{1}{\rho_0} \frac{\partial \langle p \rangle}{\partial x_j} + \nu \frac{\partial^2 \langle u_j \rangle}{\partial x_i \partial x_i} + \alpha g \left( \langle T \rangle - T_0 \right) \delta_{j,y} \quad , \tag{150}$$

and for the fluctuating part:

$$\frac{\partial u_j'}{\partial t} + \langle u_i \rangle \frac{\partial u_j'}{\partial x_i} + u_i' \frac{\partial \langle u_j \rangle}{\partial x_i} + u_i' \frac{\partial u_j'}{\partial x_i} - \frac{\partial \langle u_i' u_j' \rangle}{\partial x_i} = -\frac{1}{\rho_0} \frac{\partial p'}{\partial x_j} + \nu \frac{\partial^2 u_j'}{\partial x_i \partial x_i} + \alpha g T' \delta_{j,y} \quad (151)$$

Returning to the momentum eq. (150) for the mean part, we can write them in a more standard form (by moving the last term on the LHS to the RHS, and by expressing the viscous part in terms of the *viscous stress-tensor*):

$$\frac{\partial \langle u_j \rangle}{\partial t} + \langle u_i \rangle \frac{\partial \langle u_j \rangle}{\partial x_i} = -\frac{1}{\rho_0} \frac{\partial \langle p \rangle}{\partial x_j} + \frac{1}{\rho_o} \frac{\partial}{\partial x_i} \left[ \tau_{ji} \quad \underbrace{-\rho_0 \langle u_i' u_j' \rangle}_{\text{Reynolds'stresses}} \right] + \alpha g \left( \langle T \rangle - T_0 \right) \delta_{j,y} \quad (152)$$

We now see that the resulting (Reynolds-averaged) momentum equation has the same form as the original momentum equation, except for the $\boxed{-\rho_0 \langle u_i' u_j' \rangle}$ terms, which account for the momentum-transfer due to the turbulent fluctuations in the flow. These terms may be viewed as additional stresses (which is why they are also known as "Reynolds' stresses"). Unfortunately, these terms lead to a *closure problem*, because there is no universal[32] expression for them. This spawned the entire field of *turbulence modeling* which, during the last few decades, resulted in several closure approaches (as discussed later).

TEMPERATURE EQUATION Finally, Reynolds' decomposition can also be applied to the temperature eq. (33), yielding the temperature equation for the mean part:

$$\frac{\partial \langle T \rangle}{\partial t} + \langle u_i \rangle \frac{\partial \langle T \rangle}{\partial x_i} + \frac{\partial \langle u_i' T' \rangle}{\partial x_i} = \kappa \frac{\partial^2 \langle T \rangle}{\partial x_i \partial x_i} \quad (153)$$

and for the fluctuating part:

$$\frac{\partial T'}{\partial t} + u_i' \frac{\partial \langle T \rangle}{\partial x_i} + \langle u_i \rangle \frac{\partial T'}{\partial x_i} - \frac{\partial \langle u_i' T' \rangle}{\partial x_i} = \kappa \frac{\partial^2 T'}{\partial x_i \partial x_i} \quad (154)$$

Similar to what we did for the Reynolds-averaged momentum equation, we can re-express eq. (153) as:

$$\frac{\partial \langle T \rangle}{\partial t} + \langle u_i \rangle \frac{\partial \langle T \rangle}{\partial x_i} = \frac{\partial}{\partial x_i} \left[ \kappa \frac{\partial \langle T \rangle}{\partial x_i} - \langle u_i' T' \rangle \right] \quad , \quad (155)$$

which is formally identical to the original temperature equation, except for the $\boxed{-\langle u_i' T' \rangle}$ terms, known as the *turbulent scalar fluxes*. Similar to the momentum equation, these new terms introduce a *closure problem*, and have to be approximated based on the mean-flow quantities if we want to get a complete system of RANS equations.

32 I.e., valid for all turbulent flows

NONLINEARITY AND TURBULENT FLUXES It is interesting to notice that turbulent fluxes (such as were obtained above for the momentum and temperature equations) only occur when there are nonlinearities in the equations to which the *Reynolds averaging* (RA) procedure is being applied (in the present case, the nonlinearity is due to the presence of advection terms in the LHS of the original equations).

### 2.5.3.3 *Turbulence modeling*

During the last century, several approaches for simulating turbulent flows were developed, which we briefly mention below.

THE RANS SIMULATION APPROACH In order to bring the averaged momentum eq. (152) and temperature eq. (155) in a solvable form, we need to provide appropriate expressions for the turbulent fluxes $\boxed{-\rho_0\langle u_i' u_j'\rangle}$ and $\boxed{-\langle u_i' T'\rangle}$. In principle, we could attempt to obtain equations for these terms by multiplying eqs. (151) and (154) with $u_i'$ and taking the ensemble-average. Unfortunately, this procedure can be shown to lead to equations for the turbulent fluxes which depend on higher-order turbulent fluxes, which depend on even higher-order turbulent fluxes, and so on. Therefore, this approach leads to an infinite hierarchy of equations, which is analogue to the BBGKY hierarchy from kinetic theory (see Section 2.2).

For concrete computations, this infinite hierarchy needs to be truncated at some point, by estimating the missing terms based on information about the averaged fields ($\langle u_i\rangle$, $\langle p\rangle$ and $\langle T\rangle$). Various expressions (known as *turbulence models*) exist for various types of applications, ranging from the simple *algebraic models* (which normally use information about velocity-gradients to compute so-called "eddy viscosities" and "eddy diffusivities", respectively, in addition to some *tuning parameters* based on Prandtl's idea of *mixing length*), to the more general *Reynolds-stress models* (RSMs) (which consist of transport equations for the turbulent flux tensors, with a minimum of tuning parameters). Good overviews on this topic are provided e.g. by Pope (2000) and Wilcox (2006).

THE LARGE-EDDY SIMULATION LES APPROACH An alternative to the ensemble-averaging procedure is to apply *spatial filtering*, which allow us to explicitly separate the prognostic fields into *resolved* and *sub-grid* components. Using the universality of the small-scale eddies (according to the theory of Kolmogorov), the effect of these scales may then be simulated using so-called *subgrid scale* (SGS) models. Similar to the RANS approach, there are many SGS models to chose from. The simplest (and most popular) such model was proposed by Smagorinsky (1963). In general, the *large-eddy simulation* (LES) approach leads to more accurate computations in situations where the time-dependence of the flow is critical (Pope, 2000).

THE DIRECT NUMERICAL SIMULATION DNS APPROACH The simplest (conceptually) but, at the same time, also the hardest (computationally) approach for simulating turbulent flows is to have grid-spacings and time-steps which are small-enough to simulate even the smallest eddies in the flow. This approach (which is also used in this thesis, for the RB convection simulation) has the advantage that it does not require any empirical parameters. Also, the simulation results usually show excellent agreement with experimental data (e.g. (Dong and Karniadakis, 2005)), which makes DNS very valuable, both as a tool for basic research, as well as for calibrating RANS and LES models.

## 2.6 FLOWS AT THE PLANETARY SCALE

In the previous sections, we considered convection-driven flows which occur at human space- and time-scales (e.g. as encountered in engineering applications). Finally, in order to provide the theoretical background for the last application presented in this thesis, we briefly discuss the physics of flows at much larger (planetary) scales in the ocean. While a reasonably-complete description of the RB theory could be presented in the previous section, the theory we present for the large-scale ocean flows is more qualitative. This is due to the much more diverse range of phenomena at these scales, an exhaustive discussion of which is outside the scope of the thesis (see e.g. Pedlosky (1996) or Olbers et al. (2012) for details).

Interestingly, convection also plays a crucial role for driving flows at larger (planetary) scales, in the oceans and in the atmosphere. Indeed, the primary driving mechanism for both atmospheric and oceanic flows is the uneven warming of the Earth's surface by the Sun (i.e. more at the equator, and less at the poles). In an "attempt" to diminish this imbalance, the atmosphere and the oceans transport a significant amount of heat, from the equator towards the poles. Although currents in the ocean are much slower than winds in the atmosphere, the global[33] oceanic and atmospheric poleward heat transports are of comparable magnitude.

In the atmosphere, the differential heating leads to differences in air-pressure which, in turn, drive the surface winds (Figures 8 and 9).

In the oceans, long-lived flow structures may be observed at the surface region (i.e. down to a depth of $\sim 100\text{m}$), known as *oceanic surface currents*. These features (sketched in Figure 10 ) are driven by the surface winds in the atmosphere. Due to the Coriolis force, the currents are deflected to the right of the wind-direction in the northern hemisphere (causing predominantly clockwise flows), and to the left of the wind-direction in the southern hemisphere (causing predominantly anti-clockwise flows).

Variation of seawater-density (caused by differential heating by the Sun and by variations in salinity due to rivers and melting/formation of sea-ice) is another important factor driving the ocean circulation; this causes downwelling and upwelling of water-masses (Marshall and Schott, 1999), and drives the circulation of the deep

33 On a more local scale, the ratio of the zonally-averaged heat transports has a strong dependence on latitude (Czaja and Marshall, 2006), with the oceans dominating in the tropics and the atmosphere dominating in mid-to-high latitudes.

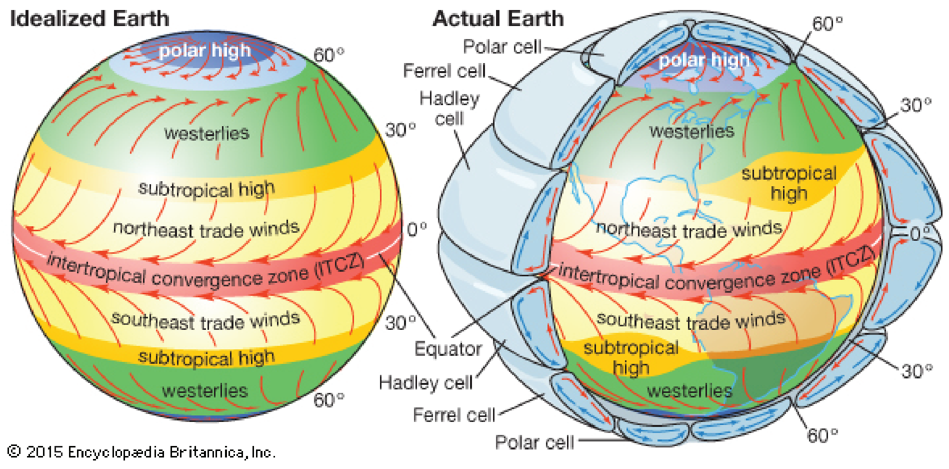


Figure 8: Sketches of the atmospheric circulation (source: *Encyclopædia Britannica Online*, article "Atmospheric circulation", accessed 15 December 2017, at `https://www.britannica.com/science/atmospheric-circulation`).

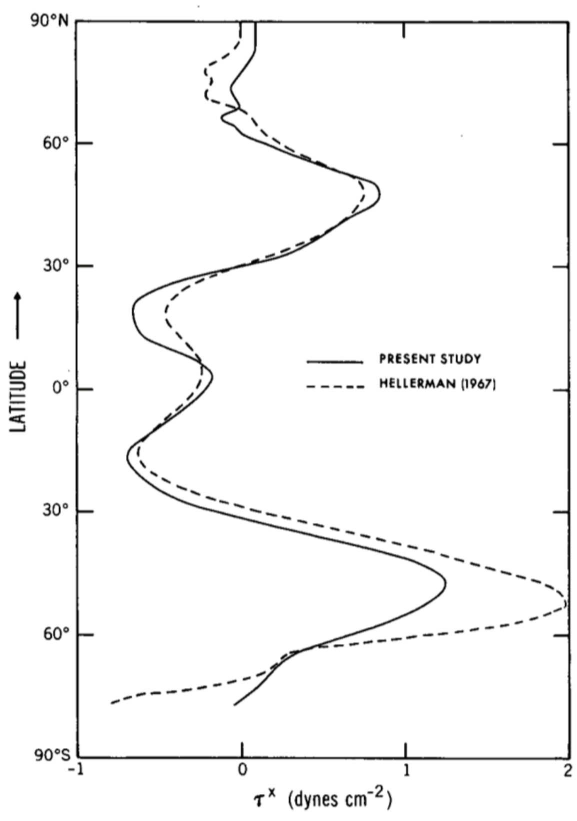


Figure 9: Estimation of the zonal mean eastward wind stress (source: Hellerman and Rosenstein (1983)). Original caption: *"Zonal means of annual $\tau^x$ (dyn cm$^{-2}$) from wind rose data of the Marine Climatic Atlases and Pilot Chart data of the U.S. Navy Hydrographic Office (dashed line) and zonal means of annual $\tau^x$ from TDF-11 data (solid line)."*

currents. These effects are denoted by the term *thermohaline circulation* (THC) (Figure 11).

The third type of external forcing upon the ocean consists of the gravitational pull due to the Moon, which causes the tides. The tidal motion influences the general

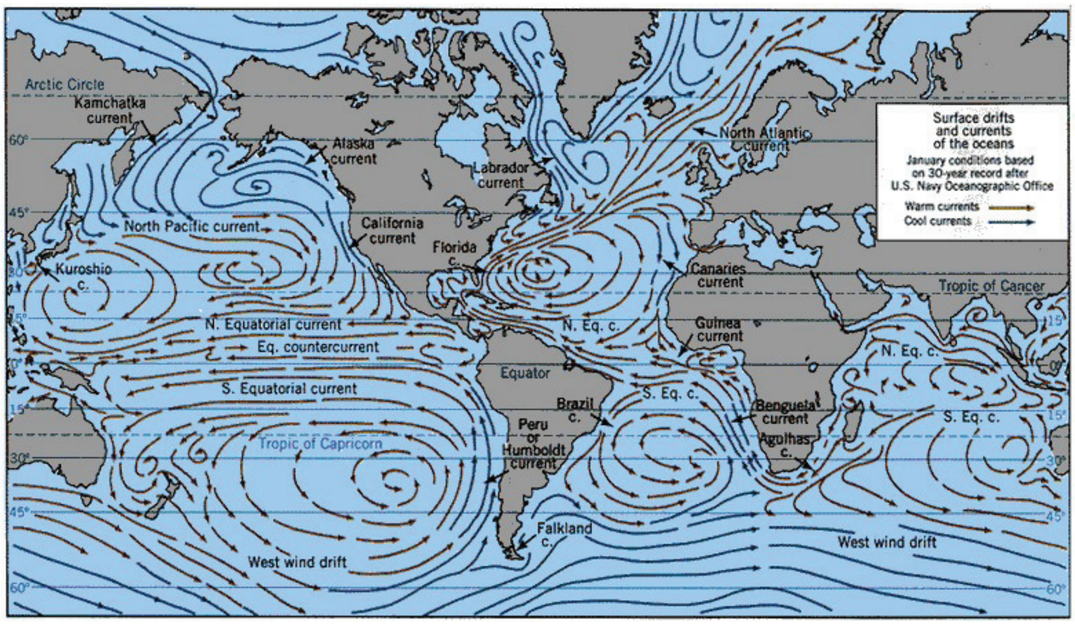


Figure 10: Sketch of the oceanic surface currents (source: NASA / US Navy Oceanographic Office, article "General Characteristics of the World's Oceans: 4 Ocean Currents", accessed 15 December 2017, at https://icp.giss.nasa.gov/research/ppa/1997/oceanchars/currents.html).

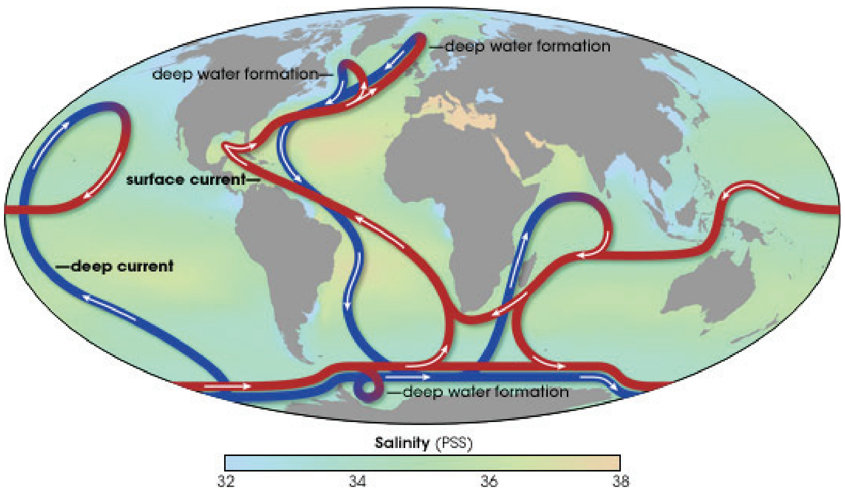


Figure 11: Sketch of the thermohaline circulation of the oceans (source: NASA *Earth Observatory Online*, article "Explaining Rapid Climate Change: Tales from the Ice", accessed 15 December 2017, at https://earthobservatory.nasa.gov/Features/Paleoclimatology_Evidence/paleoclimatology_evidence_2.php).

circulation by increasing the bottom friction, and by enhancing the vertical mixing in the shallow regions (Moon, 2005).

Another important aspect for planetary-scale flows is the Coriolis *pseudo*-force, due to the rotation of the Earth, which induces an additional acceleration:

$$\mathbf{a}^C = -2\boldsymbol{\Omega} \times \mathbf{u} \Longleftrightarrow a_j^C = -2\epsilon_{j,k,l}\Omega_k u_l \quad , \tag{156}$$

where $\mathbf{\Omega} = 7.29 \times 10^{-5}\text{s}^{-1}$ is the angular velocity of the Earth's rotation, $\mathbf{u}$ is the velocity of the fluid parcel, and $\epsilon_{j,k,l}$ is the permutation tensor defined earlier.

While this is not a source of energy *per se* (the mechanical work of the Coriolis force is zero), this strongly deflects the currents in the atmosphere and oceans, as long as both the spatial and the temporal scales are sufficiently-large.

In addition to the factors mentioned above, the dynamics of fluids at the planetary scale is further complicated by a myriad phenomena, such as phase-transitions of water (precipitation and evaporation, or formation and melting of ice-sheets), influences due to the carbon cycle, or feedback-effects (e.g. the ice-albedo feedback).

### 2.6.1 *Primitive-equations model of the ocean*

The discussion in the previous section was more qualitative than the earlier parts of this chapter. The reason for this, as mentioned previously, is that the phenomena occurring at the planetary scales are too diverse to be satisfactorily captured by a small set of equations (as is the case in human-scale fluid flows). This is due to the very large scales and the small vertical-to-horizontal aspect ratios, but also because many additional processes need to be considered to describe the complete physics. Therefore, instead of a unified set of equations, it is more feasible to consider separate models for studying specific phenomena, which can be later coupled as needed.

One of the more general versions of the dynamical equations of motion in the ocean (which is also used by many *general circulation model*s (GCMs)) is the system of the so-called *primitive equations*. These equations can be derived[34]from the usual fluid-mechanics equations discussed previously (with the addition of a salinity-transport equation), by assuming the flows to be incompressible (i.e. ignoring phenomena on the acoustic time-scales, and employing the Boussinesq approximation) and by neglecting terms that are rendered small by the small vertical-to-horizontal aspect ratio. Additionally, the *Reynolds averaging* (RA) procedure is normally applied, to obtain the final equations.

### 2.6.2 *Wind-driven ocean circulation (WDOC)*

For ocean GCMs, the primitive equations are usually written in *spherical coordinates* (or in oblate spheroidal coordinates, for a more accurate representation of the Earth's surface). Because in that form the equations contain many geometric factors which are not essential for our current purposes (see (Olbers et al., 2012) or (Pedlosky, 1987) for details), we do not write them here. Instead, we start with a simplification of the primitive equations known as the *barotropic (vertically-integrated) ocean approximation*, which is useful for understanding qualitatively the shape of the wind-driven surface ocean currents. For the remainder of this chapter, we present the theory behind this model, which is then simulated using the LBM approach in Section 5.3.

34 The derivation is outside the scope of this thesis – see e.g. (Müller, 2006) for details.

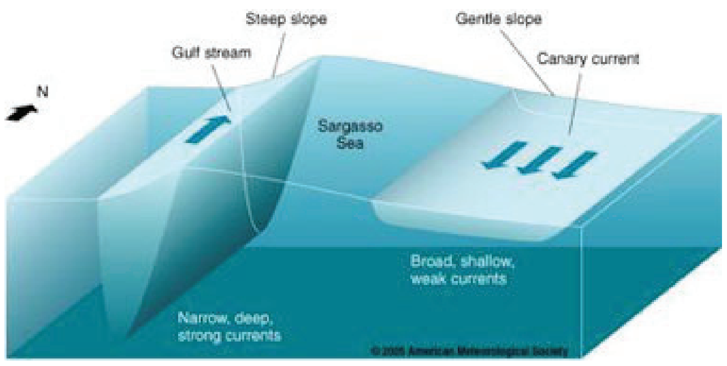

Steep slope
Gentle slope
Gulf stream
Canary current
N
Sargasso Sea
Broad, shallow, weak currents
Narrow, deep, strong currents
© 2005 American Meteorological Society

Figure 12: Illustration of the western-boundary current intensification in the oceans (source: NASA *Ocean Motion*, article "Wind driven surface currents: western boundary currents", accessed 15 December 2017, at http://oceanmotion.org/html/background/western-boundary-currents.htm).

A salient feature of the WDOC (which our numerical model will have to reproduce) is the intensification of the currents at the western boundaries (e.g. the Gulf Stream, the Kuroshio, or the Aghulas Current), as illustrated in Figure 12. This phenomenon puzzled oceanographers until the first half of the 20th century. While significant contributions (especially for understanding the flow in the interior of the ocean basins) were made by Ekman (1905, 1923) and by Sverdrup (1947), it was Stommel (1948) and Munk (1950) who showed that the western-boundary current intensification is caused by the strengthening of the Coriolis effects with increasing latitude. These studies provided analytical solutions for the *linearized* problem, under the assumption of *time-independent* flow. Such solutions serve as a good starting-point for assesing solutions for the *nonlinear* and/or *time-dependent* problem (which require use of numerical models).

### 2.6.2.1 *The primitive equations (β-plane approximation)*

A simple form of the primitive equations (which still retains the terms that are important for capturing the wind-driven circulation) is the *locally-Cartesian* formulation (see e.g. (Miller, 2007)):

$$\begin{cases} \dfrac{\partial u_i}{\partial x_i} = 0 & (157)\\ \dfrac{\partial u_1}{\partial t} + u_i \dfrac{\partial u_1}{\partial x_i} = -\dfrac{1}{\rho_0}\dfrac{\partial p}{\partial x_1} + f u_2 + F_1 & (158)\\ \dfrac{\partial u_2}{\partial t} + u_i \dfrac{\partial u_2}{\partial x_i} = -\dfrac{1}{\rho_0}\dfrac{\partial p}{\partial x_2} - f u_1 + F_2 & (159)\\ p(x_3) = p_0 + g\displaystyle\int_z^0 \rho\, dz' & (160)\\ \dfrac{\partial \theta}{\partial t} + u_i \dfrac{\partial \theta}{\partial x_i} = F_\theta & (161)\\ \dfrac{\partial s}{\partial t} + u_i \dfrac{\partial s}{\partial x_i} = F_s & (162) \end{cases}$$

where $p_0$ is the surface pressure, $\theta$ is the *potential temperature*, $s$ is the salinity. A hydrostatic pressure distribution is assumed. The quantities $F_1$ and $F_2$ represent the effects of turbulent mixing, which are often parameterized using the eddy-viscosity approach (with different coefficients for the vertical and for the horizontal components):

$$F_i = A_V \frac{\partial^2 u_i}{\partial x_3 \partial x_3} + A_H \Delta_H u_i \qquad \forall i \in \{1, 2\} \quad , \tag{163}$$

where $\Delta_H \equiv \frac{\partial^2}{\partial x_1 \partial x_1} + \frac{\partial^2}{\partial x_2 \partial x_2}$ is the *horizontal Laplacian*, and $A_V \sim 10^{-3} m^2/s$ ($A_H \sim 2 \times 10^5 m^2/s$) is the vertical (horizontal) eddy-viscosity coefficient (numerical values after (Lohmann, 1996)).

For the terms $F_\theta$ and $F_s$, representing turbulent mixing of potential temperature and of salinity, Bryan and Cox (1967) proposed parameterizations based on *convective adjustment*:

$$F_\chi = \frac{\partial}{\partial x_3}\left(\frac{A_{\chi,V}}{\delta}\frac{\partial \chi}{\partial x_3}\right) + A_{\chi,H}\Delta_H \chi \qquad \forall \chi \in \{\theta, s\} \quad , \tag{164}$$

where $A_{\chi,V} \sim 10^{-4} m^2/s$ ($A_{\chi,H} \sim 2 \times 10^3 m^2/s$) is the vertical (horizontal) eddy-diffusivity coefficient for the passive scalar $\chi \in \{\theta, s\}$ (numerical values after (Lohmann, 1996)). By taking the parameter $\delta$ of the form:

$$\delta(\rho) = \begin{cases} 1 & \text{if } \frac{\partial \rho}{\partial x_3} < 0 \\ 0 & \text{if } \frac{\partial \rho}{\partial x_3} > 0 \end{cases} , \tag{165}$$

the concentration of the passive scalar is assumed to be instantly equalized vertically whenever a situation of unstable stratification occurs (while usual diffusion is active for situations of stable stratification).

To model the effect of the Coriolis force, the so-called $\beta$-plane approximation is used, whereby the *Coriolis parameter* $f$ (also known as *Coriolis frequency* or *Coriolis coefficient*) is assumed to vary linearly with the locally-Cartesian northward coordinate ($x_2$):

$$f(x_2) \approx \underbrace{2\Omega \sin\phi_0}_{f_0} + \underbrace{\frac{2\Omega\cos\phi_0}{R}}_{\beta \equiv \left(\left.\frac{\partial f}{\partial x_2}\right|_{\phi_0}\right)} x_2 \quad , \tag{166}$$

where $R \approx 6367.4$ km is the mean radius of the Earth and $\phi_0$ is some reference latitude. Equation (166) is a compromise between:

a) the much simpler $f$-plane approximation

$$f = \text{const.} = \underbrace{2\Omega \sin\phi_0}_{f_0} \quad , \tag{167}$$

which does *not* capture important effects (such as the intensification of the western-boundary currents and the Rossby waves), and

b) the exact dependence on latitude

$$f(\phi) = 2\Omega \sin(\phi) \quad , \tag{168}$$

which makes analytic solutions much more difficult to obtain.

Equations (157) to (162) need to be supplemented by an equation of state, which gives the dependency of the density on temperature and salinity. While the general equation of state is highly-nonlinear, the following linearized version is used frequently:

$$\rho(\theta, s) = \rho_0 \left[1 - \alpha(\theta - \theta_0) + \sigma(s - s_0)\right] \quad , \tag{169}$$

where $\sigma$ is the *haline contraction coefficient*.

### 2.6.2.2 *Governing equations for a barotropic ocean*

By vertically-integrating the equations presented in Section 2.6.2.1 and changing the *equation of state* (EOS) to a constant (fixed density $\rho_0$), it is possible to obtain an even simpler, two-dimensional model, which is formally identical to the two-dimensional incompressible *Navier-Stokes equations* (NSE):

$$\begin{cases} \dfrac{\partial u_i}{\partial x_i} = 0 & (170) \\ \dfrac{\partial u_j}{\partial t} + u_i \dfrac{\partial u_j}{\partial x_i} = -\dfrac{1}{\rho_0} \dfrac{\partial p}{\partial x_j} + A \dfrac{\partial^2 u_j}{\partial x_k \partial x_k} + \underbrace{T_j + a_j^C}_{\text{body-forces}} \; , & (171) \end{cases}$$

where, in contrast to the RB momentum equations, the coefficient of *kinematic viscosity* ($\nu$) is replaced by the *horizontal eddy viscosity coefficient* ($A \approx 10^4 \, \mathrm{m^2/s} \gg \nu \approx 10^{-6} \, \mathrm{m^2/s}$), as used by Munk (1950), who followed the approach of Boussinesq (1877) of modeling momentum transfer due to turbulent eddies in an analogous way as momentum transfer due to molecular diffusion (except that the former is orders-of-magnitude more efficient at mixing momentum).

In eq. (171) **T** represents the effect of the surface wind-stress (which appears in our 2D model as a body-force). For simplicity, we can assume a zonal wind, of the form:

$$\mathbf{T} = T_0 \begin{pmatrix} -\sin^2\left(\frac{\pi}{L} x_2\right) \\ 0 \end{pmatrix} , \tag{172}$$

which corresponds to easterly winds at low latitudes and to westerlies at mid latitudes.

For the Coriolis force, the previously-mentioned *beta plane approximation* is used:

$$\mathbf{a}^{\mathbf{C}} = -f(x_2)\hat{\mathbf{u}} = -(f_0 + \beta x_2)\hat{\mathbf{u}} = \begin{pmatrix} (f_0 + \beta x_2) u_2 \\ -(f_0 + \beta x_2) u_1 \end{pmatrix} , \tag{173}$$

with $f_0$ and $\beta$ same as in eq. (166), and where we denoted by $\hat{\mathbf{u}}$ the velocity vector-field $\mathbf{u}$, *rotated anticlockwise* (by 90°):

$$\hat{\mathbf{u}} \equiv \begin{pmatrix} -u_2 \\ u_1 \end{pmatrix}, \tag{174}$$

#### 2.6.2.3 *BCs and ICs for the barotropic ocean model*

As usual, to complete the specification of the model, we need to provide *boundary condition* (BC) and ICs.

BCs For simplicity, we only consider *closed* domains, where the no-slip BC is assumed to hold at the ocean-land interface:

$$\mathbf{u}(\mathbf{x}, t)\big|_{\mathbf{x}\in\{\text{ocean-land interface}\}} = 0 \tag{175}$$

ICs As for the ICs, we simply assume the ocean to be initially at rest:

$$\mathbf{u}(\mathbf{x}, 0) = 0, \qquad \forall \mathbf{x} \in \{\text{ocean}\} \tag{176}$$

#### 2.6.2.4 *Problem reformulation in non-dimensional units*

It is useful to re-write the equations in *dimensionless* form, as we did for the RB problem. In the present case, we denote by L the characteristic length-scale (horizontal) and U as the characteristic velocity.[35] From these, we can construct $\frac{L}{U}$ as a time-scale and $\rho_0 U^2$ as a scale for pressure-differences. By plugging these scales into eqs. (170) and (171) we obtain (after some algebra):

$$\begin{cases} \dfrac{\partial u_i}{\partial x_i} = 0 & (177) \\[2ex] \dfrac{\partial u_1}{\partial t} + u_i \dfrac{\partial u_1}{\partial x_i} = -\dfrac{\partial(\delta p)}{\partial x_1} + \dfrac{1}{\mathrm{Re}}\dfrac{\partial^2 u_1}{\partial x_k \partial x_k} + \dfrac{1}{\mathrm{Ro}}\left(\Gamma_{\mathrm{Ro}} + x_2\right) u_2 - \dfrac{1}{\mathrm{Fr}^2}\sin^2(\pi x_2) & (178) \\[2ex] \dfrac{\partial u_2}{\partial t} + u_i \dfrac{\partial u_2}{\partial x_i} = -\dfrac{\partial(\delta p)}{\partial x_2} + \dfrac{1}{\mathrm{Re}}\dfrac{\partial^2 u_2}{\partial x_k \partial x_k} - \dfrac{1}{\mathrm{Ro}}\left(\Gamma_{\mathrm{Ro}} + x_2\right) u_1 & (179) \end{cases}$$

35 We will provide concrete values for these scales in Section 5.3.

**NOTE (notation)**

To make the structure of the equations more readable, we *dropped the* $\square^{(d)}$ *superscript* in the equations. The reader should therefore keep in mind that the quantities $u_i$, $x_i$, $t$ and $(\delta p)$ in the eqs. (177) to (179) above actually refer to the dimensionless counterparts $u_i^{(d)}$, $x_i^{(d)}$, $t^{(d)}$ and $(\delta p)^{(d)}$ respectively.

In the eqs. (177) to (179) we introduced the following *dimensionless numbers*:

- **Reynolds number**

$$\mathrm{Re} \equiv \frac{UL}{A} \equiv \frac{\text{advection}}{\text{dissipation}} \tag{180}$$

- **Froude number**:

$$\mathrm{Fr} \equiv \frac{U}{\sqrt{T_0 L}} \equiv \frac{\text{inertia}}{\text{strength of the external force-field}} \tag{181}$$

- **Rossby number**:

$$\mathrm{Ro} \equiv \frac{U}{\beta L^2} \equiv \frac{\text{advection}}{\text{strength of the } \beta\text{-term in Coriolis force}} \tag{182}$$

- **Coriolis "β-ratio"**:

$$\Gamma_{\mathrm{Ro}} \equiv \frac{f_0}{\beta L} \equiv \frac{\text{strength of constant term in Coriolis force}}{\text{strength of } \beta\text{-term in Coriolis force}} \tag{183}$$

We will use these results in Section 5.3, when we discuss the numerical solution of this problem.

# 3 COMPUTATIONAL FLUID DYNAMICS (CFD) AND LATTICE BOLTZMANN METHODS (LBMs)

"An algorithm must be seen to be believed"

Donald Knuth

Owing to the simplicity of the algorithms and to their suitability for massively-parallel computations, the *lattice Boltzmann method* (LBM) has attracted an active research community during the last two decades, which developed the method on several fronts, and eventually made LBM a serious competitor to methods with a longer tradition (such as *finite differences* (FD), *finite volumes* (FV), *finite elements* (FE) or spectral methods).

The LBM has been successfully applied to traditional CFD problems (Wolf-Gladrow (2000), Succi (2001), Sukop and Thorne (2006), Luo et al. (2010), Guo and Shu (2013)). In addition, the approach seems very promising for emerging classes of applications, such as fluid-structure interactions (Tian et al., 2011), microfluidics (Zhang, 2011), and multiphase flows (Fox, 2011).

There is a variety of different schemes that are based on the LB equations, most of which are summarized in the excellent reviews of Benzi et al. (1992), Chen et al. (1998), Aidun and Clausen (2010), Luo et al. (2010), or in the books of Wolf-Gladrow (2000), Succi (2001), Sukop and Thorne (2006), Guo and Shu (2013), and Krüger et al. (2017). Many of these models differ with respect to how they implement the collision operator – for example, SRT models (Qian et al., 1992), MRT models (D'Humieres, 1992), regularized models (Latt and Chopard, 2006), entropic models (Ansumali and Karlin, 2000), or the more recent cumulant LB model (Geier et al., 2015).

In this chapter, we only briefly present the conceptual ideas behind LBM. Because many good references covering the early developments of the field already exists (e.g. (Wolf-Gladrow, 2000) and (Succi, 2001)), we do not repeat this material here. However, some of the newer variations of the method, which were used for our concrete simulations are described in Chapter 5, together with the problems to which they are applied.

## 3.1 WORKFLOW FOR NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS (DEs)

Although there are many algorithms to choose from for solving any given fluid flow problem numerically, there are three general phases, which are independent of the chosen algorithm:

1. **Pre-processing**

- specifying the problem (governing equations, fluid properties, *initial conditions* (ICs), *boundary conditions* (BCs)) in *physical* units
- casting the problem in *non-dimensional* form
- *discretizing* the non-dimensional equations; this step involves several substeps:
  - setting-up the geometry
  - specifying a system of equations which needs to be solved numerically
  - choosing the simulation parameters (such that both *stability* and *accuracy* are ensured)
- *initializing* the fields in the simulation

2. **Solution**
   - solving the algebraic equations
3. **Post-processing**
   - evaluating derived (secondary) quantitites (if any)
   - producing graphical representations of the results

## 3.2 TRADITIONAL METHODS IN COMPUTATIONAL FLUID DYNAMICS (CFD)

To understand what makes LBM-like numerical schemes different, this section summarizes the methods of *finite differences* (FD), *finite elements* (FE), *finite volumes* (FV), as well as the spectral method, which are the more "traditional" approaches for constructing numerical algorithms.

### 3.2.1 *Finite difference (FD) methods*

Perhaps the most natural approach for simulating *differential equations* (DEs) numerically is that of FDs methods.[1] The main idea for FD methods is to replace the continuous derivatives in the target DEs with discrete approximations; this leads to a system of *difference equations*, which can be solved numerically. The approximations for the derivatives are usually constructed by combining different Taylor expansions of the unknown function.

FD methods are ideally suited for regular meshes, where the usual Taylor expansions are directly applicable. However, their implementation becomes more complicated for irregular meshes, because in those cases it is necessary to transform the equations before the discretization step (or to use interpolation).

1 From an algorithmic point of view, the *lattice Boltzmann methods* (LBMs) can also be considered to be FD schemes. However, the interpretation of the stencils is less direct.

### 3.2.2 *Finite volume (FV) methods*

Another approach for numerical simulation of DEs is that of *finite volumes* (FV) methods (Roe, 1981; LeVeque, 1997, 1998). These methods are based on an *integral* formulation of the governing equations (in contrast to FD methods, which are based on an *differential* formulation). Similar to FD methods, the idea is to compute the values of the quantities of interest at *discrete points*. However, the interpretation of the resulting numerical values is different: instead of representing the values of the quantities at that particular point in space, it represents the *average value* of the quantities, over a finite spatial sub-volume ("cell"), for which the point is the centroid. In the FV methods, it is assumed that we have a *piece-wise linear* variation of the quantities of interest over the volume of each cell. To study time-dependent problems, FV methods focus on how the values at the centroid change due to exchange of fluxes with the neighbouring cells, by evaluating the fluxes at the cell-interfaces (using the divergence theorem, the volume integrals from the governing equations are converted into surface integrals, representing the fluxes). By construction, this method has the useful property of being *conservative*, since the flux entering one cell is equal to the flux leaving the adjacent cell – the fluxes also have a direct physical significance in most applications (which is not necessarily the case e.g. for *finite elements* (FE) methods).

While the *conservative* nature of FV methods is an advantage in many applications, this also makes them less general than FD methods (which can also be applied to problems which are not based on physical conservation laws).

Because there is a lot of freedom in choosing the shape of the cells, this method is generally well-suited for applications requiring irregular meshes. However, the calculation of the fluxes can also become very complicated for general irregular meshes.

### 3.2.3 *Finite element (FE) methods*

A third approach for solving DEs numerically is that of *finite elements* (FE) methods. Originally introduced as a method for *structural analysis* (of bridges, ships, etc.) in engineering applications (where it is also known as *FE analysis*), this method is based on the idea of dividing the computational domain into a number of small sub-domains (the "elements"). The flow solutions are then computed on each of these elements (for example using the Galerkin method). In a sense, the FE method can be viewed as a local version of the spectral methods, which are discussed in the next section.

### 3.2.4 *Spectral methods*

A fourth general approach for solving DEs numerically is that of *spectral methods* (Orszag, 1969, 1971; Canuto et al., 1988). In this approach, the solution of the DE is written as a summation over a space of basis functions (e.g. Fourier series), where each such function is multiplied by a coefficient. These coefficients are then computed, such that the DE is satisfied as accurately as possible. For solving systems of ODEs, the spectral approach leads[2] to a system of algebraic equations for the coefficients, while for time-dependent *partial differential equation*s (PDEs) the result is a system of ODEs, which can be integrated in time using methods such as Runge-Kutta.

The idea of writing the solution in terms of the basis functions is common to FE methods and spectral methods. However, while FE methods define these basis functions on small sub-domains (the "elements"), spectral methods define *global* basis functions, for the entire computational domain. This implies that the resulting solution at each point will implicitly depend on the entire computational domain (unlike FD methods for example, where only information from neighbouring points is used). Due to this, spectral methods have *exponential convergence* for problems where they are applicable, which makes them the "golden standard", against which other methods are often evaluated – in the context of LBM, see e.g. Martínez et al. (1994) and Peng et al. (2010). However, due to the Gibbs phenomenon, spectral methods lose their advantages for problems with complex geometries. Also, they are generally considered inappropriate for problems with strong shocks (though this is not so important in ESS).

## 3.3 A "BOTTOM-UP" APPROACH TO MODELING

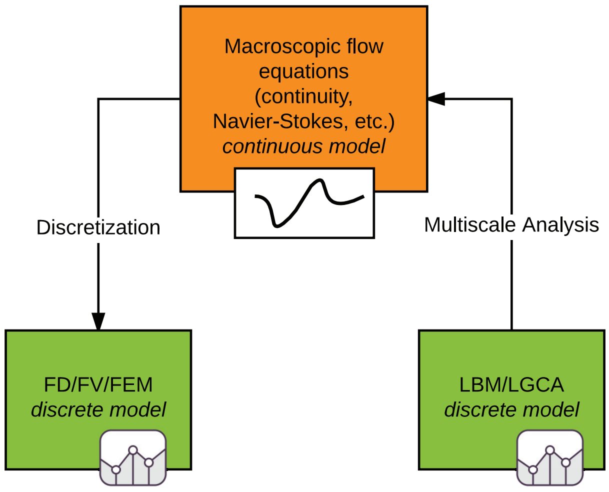


Figure 13: Top-down versus bottom-up approaches for numerical simulation of fluids.

As discussed in Chapter 2, a fluid can be described by several physical theories, of different granularities. The fact that we can, in principle, recover the phenomena predicted by the coarse-grained theories from solutions of the fine-grained theories also suggests a non-conventional way of constructing numerical algorithms for simulating fluid flows (see Figure 13): instead of directly modeling the coarse-grained equations (i.e. Navier-Stokes equations for human-scale flows), we can construct a *simplified model* of the fine-grained equations, which will exhibit the same behavior at the larger scales. We emphasized "simplified model" above, by which we mean that

2 The equations for the coefficients are the result of a transformation from the physical space (continuous) to the coefficient space (discrete), usually via the Fourier transform.

the these models are not meant to provide quantitatively accurate solutions of the fine-grained equations (which is the costly procedure that we want to avoid). Instead, the idea is to capture *qualitatively* the essential physics at these scales, which are important for recovering the correct large-scale dynamics. It can be shown, nonetheless, that the coarse-grained dynamics simulated by methods following this approach (such as *lattice gas cellular automata* (LGCA) or LBM) *can be* qualitatively accurate, assuming that the parameters in the models are chosen appropriately.

Although this approach seems convoluted at first, it turns out to have several surprising benefits:

- From an algorithmic point of view:
  - the resulting numerical schemes are often easier to implement than those of classical approaches, and
  - they are also more suitable for parallel computing (which is important in today's many-core hardware landscape).
- Also, because the algorithms are based on more "fundamental" principles than the Navier-Stokes equations, they can also be extended more easily to simulate more complex physics beyond Navier-Stokes.

It should also be noted that, despite these advantages, this idea also brings some unique challenges. First, while there are several methods (Watari, 2012; Caiazzo et al., 2009) for deriving the macroscopic equations that are actually solved by a given LBM/LGCA discrete model, there is currently no general procedure for deriving the discrete model for arbitrary systems of macroscopic PDEs. The most common approach for deriving LBM-like models is to add/remove terms to the evolution equation of an existing LBM model, followed by tuning of the degrees of freedom in the model (at the end of the multiscale-analysis), to match the target PDEs as close as possible. This approach is very intuitive (and is actually a strength of LBM) when the goal is to model effects which are also present at the small scales (such as body forces or multiphase flows). However, it becomes an inconvenience if the goal is to model arbitrary systems of PDEs, because the natural tendency of LBM algorithms is to model equations which are similar to Navier-Stokes. Owing to continuing interest in LBM methods, these problems will hopefully be addressed by future research.

## 3.4 PRECURSORS OF LBM

To understand the *lattice Boltzmann method* (LBM), it is useful to briefly consider the models which inspired the method: *cellular automata* (CA) and *lattice gas cellular automata* (LGCA).

### 3.4.1 *Cellular Automata (CA)*

Historically, the first numerical models following the philosophy outlined above were the CA. These are discrete models, consisting of entities ("cells") which typi-

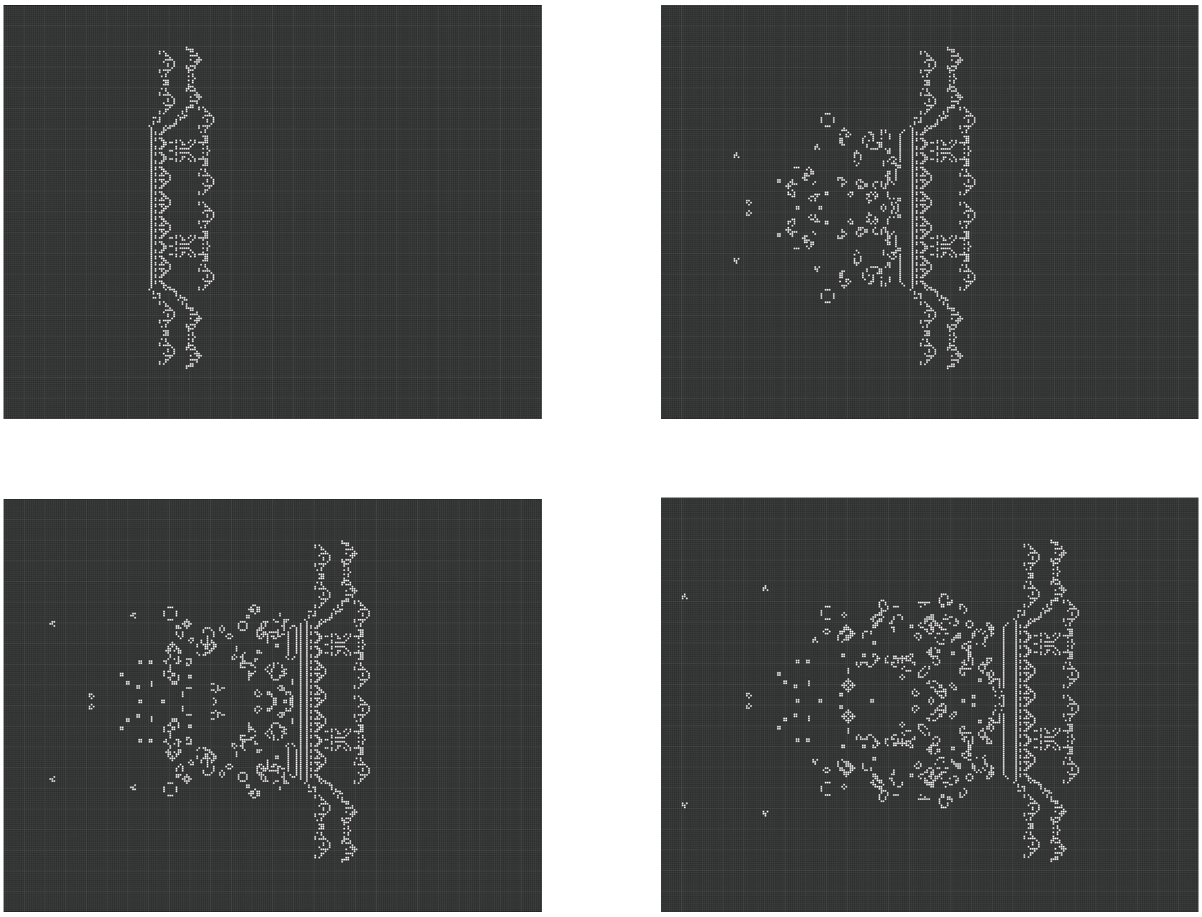


Figure 14: Snapshots of the *superstable line puffer* pattern: $t = 0$ (upper-left), $t = 100$ (upper-right), $t = 150$ (lower-left), and $t = 200$ (lower-right). The figures were produced using the `Golly` software (Trevorrow and Rokicki, 2009).

cally reside on a regular grid (lattice). The state of each cell is updated from time $t$ to $t + 1$, based on some simple evolution rule, which takes into account the state (at time $t$) of the cell itself and of some of its neighbors:

$$N(\mathbf{x}, t+1) = f\left(\{N(\mathbf{x}+\boldsymbol{\delta}_i, t) : (\forall \boldsymbol{\delta}_i)\,, \|\boldsymbol{\delta}_i\| \leqslant r\}\right), \tag{184}$$

where $N(\mathbf{x}, t)$ represents the state of the cell at discrete location $\mathbf{x}$ and discrete time $t$, and $r_S$ is

Typically, the state of each cell is characterized by a boolean ("dead" or "alive") or integer value. This is appealing from a computational point of view, since it implies that the evolution rule is free of numerical rounding-errors.

Most remarkably, despite the simplicity of the evolution rules, CA are capable of producing very complex, chaotic patterns. For example, Figure 14 shows the fascinating evolution of a pattern called the "line puffer", which moves from left to right as the CA is integrated in time, leaving a complex "wake" behind. Interestingly, the *game of life* (GoL) automaton is known to be *Turing complete* (Berlekamp et al., 2003), which is to say that it can be used to compute any result that is computable in theory.

### 3.4.2 *Lattice Gas Cellular Automata (LGCA)*

The direct ancestors of LB methods are the *lattice gas cellular automata* (LGCA). Here, instead of directly simulating each atom/molecule in the fluid, "virtual particles" are simulated, each of these representing a group of atoms/molecules. The grouping corresponds to coarse-graining in space but, equally important, also to coarse-graining in *velocity space* (since only a finite number of velocities are allowed).

The virtual particles then move in discrete steps, each according to its velocity, and all of their possible positions trace a regular grid (the "lattice"). This process, known as *streaming* (or *propagation*), is inspired by the fact that atoms/molecules in a gas move freely for considerable periods of time.

Also inspired by kinetic theory, when two or more particles meet at the same lattice node, they interact according to a set of rules (which are designed to satisfy the physical conservation laws for mass, momentum, and energy).

The first LGCA model was proposed by Hardy et al. (1973), which used a square lattice. However, while this recovered qualitatively fluid-like behavior, there were some unphysical effects. These were corrected by subsequent versions of the LGCA ((Frisch et al., 1986), (Wolfram, 1986)).

Despite these improvements, LGCA models fell out of favor, due to the inherent noise of the simulations, which requires coarse-graining to get physically-meaningful macroscopic fields. Instead, they gave way to LBM models, which replaced the boolean state-vector with real probability distribution functions. To account for this change, LBMs also use different collision operators compared to LGCAs.

## 3.5 how LBMs work

The typical steps in LBMs simulations are shown in Figure 15. The signature pattern is the succession of collision and streaming steps, which are inspired by simplifications of the real processes described in kinetic theory. For the boundaries of the domain, there are also specialized rules that need to be applied (usually after the collision step), to enforce the desired values for the macroscopic fields.

lattice-topologies used in this thesis In LBM algorithms there are several possible choices of velocity-space discretizations, depending on the requirements of the model (e.g. whether it is one-, two-, or three-dimensional, whether the energy-equation needs to be recovered fully, etc.). Out of these choices, only four lattices are used for the applications discussed in this thesis. These velocity-space discretizations are illustrated in Chapter 5, along with the specific algorithms using them – specifically D2Q5 (Figure 20), D2Q9 (Figure 21), D3Q7 (Figure 27), and D3Q19 (Figure 28), with lattice-names following the DdQq convention which is common in the literature, where where $\boxed{\text{d}}$ has to be replaced by the dimensionality of the lattice (one-, two-, or three-dimensional), and $\boxed{\text{q}}$ stands for the number of DFs at each node (including the rest velocity, if any).

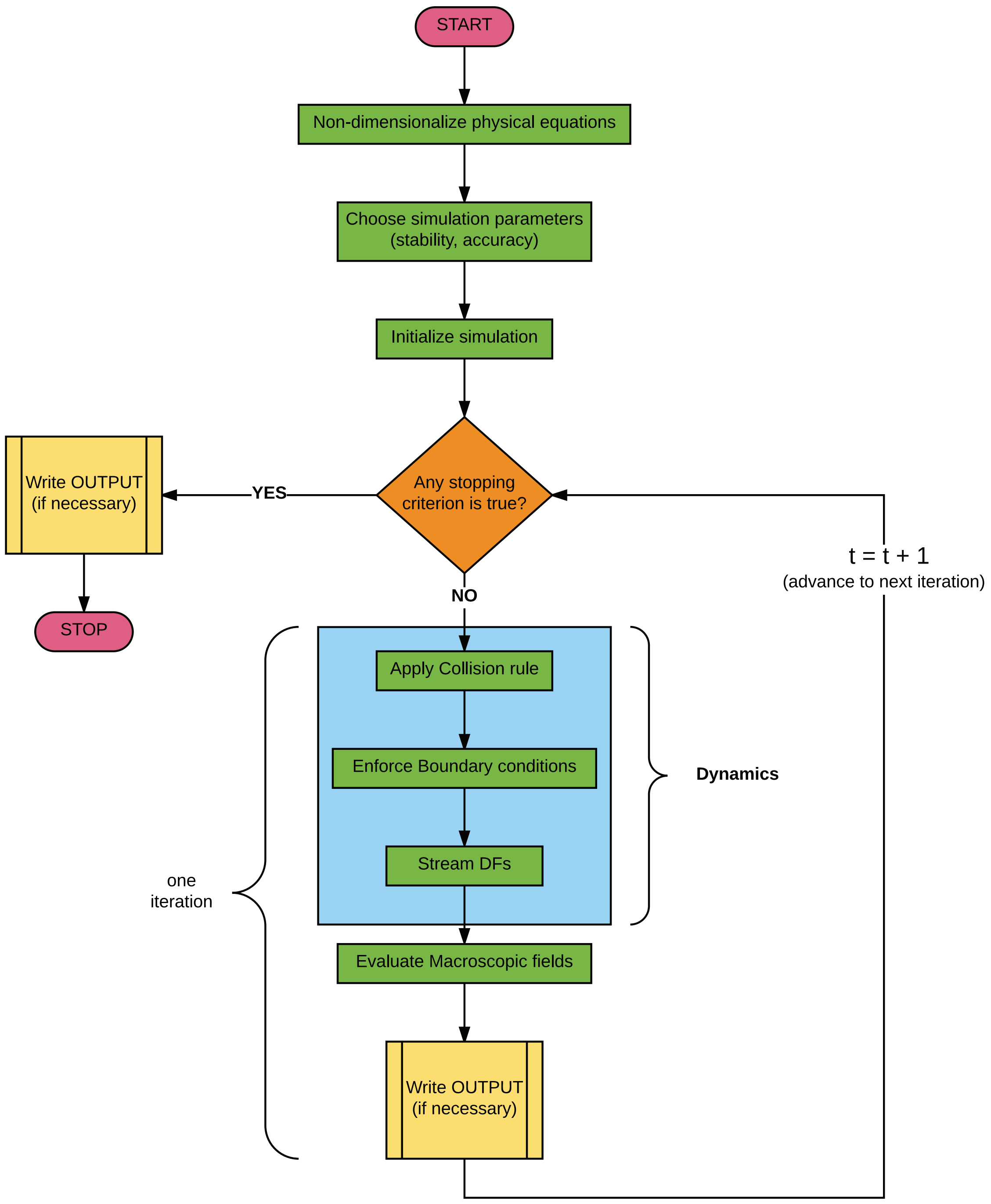

START
Non-dimensionalize physical equations
Choose simulation parameters (stability, accuracy)
Initialize simulation
Any stopping criterion is true?
YES
Write OUTPUT (if necessary)
STOP
NO
Apply Collision rule
Enforce Boundary conditions
Stream DFs
Dynamics
one iteration
Evaluate Macroscopic fields
Write OUTPUT (if necessary)
t = t + 1
(advance to next iteration)

Figure 15: Flowchart of typical execution flow for *lattice Boltzmann method* (LBM) applications.

Part II

# CORE

# 4 GeLB – A TOOL FOR LATTICE BOLTZMANN MODELING

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

Albert Einstein

In this chapter, the GeLB *domain-specific language* (DSL) is presented, as a first outcome of the work withing the scope of this thesis. After a brief discussion about general-purpose programming languages (Section 4.1), the notion of a DSL is explained, highlighting its benefits. Next (Section 4.2), previous related work is summarized, to provide a context and motivation for the current work. An overview of the internal processing-steps is then provided (Section 4.3), which is good to keep in mind for potential users of this tool; at that stage, some of the implementation-techniques used for the framework are also mentioned, along with alternative approaches (some of which were tried by the author). Finally, the chapter closes by describing the *GeLB description* (GD) programming language, which provides the means for specifying LBM simulations using GeLB (Section 4.4).

## 4.1 COMPUTERS AND PROGRAMMING LANGUAGES

From a very high-level perspective, digital computers are devices which store some internal state (in some form of memory). Based on the state of the memory and on a stream of instructions, a new result is computed (which is often also stored in memory). Unfortunately, while mathematically appealing, this view is not very useful in practice, because it is too removed from the practical problems that need to be solved with the help of computers. Therefore, as soon as the first programmable computers were invented[1], the first programming languages also appeared, which allowed humans to more easily specify the computations to be carried-out by the machines. This is because humans do not like (and neither should they need to) think about the very low-level processes occuring inside the computer – instead, the idea of *abstraction* is used, to hide the irrelevant details. Over the years, programming languages evolved to align better with this goal, oferring increasingly-powerful abstractions. An illustrative example is the evolution of FORTRAN, which is one of the first general-purpose programming languages: although it is not considered to be at the

1 Charles Babbage's second computer (the "analytical engine") is considered to be the first *programmable* computer; accordingly, Ada Lovelace, who wrote some notes on how to use this machine to compute the Bernoulli numbers, is considered the first "programmer".

forefront of programming language design, FORTRAN added (Chirila and Lohmann, 2015) many techniques in response to the needs of its community of users.

Some examples of techniques that GPLs provide for increasing the level of abstraction are:

- **structured programming:** Here, instead of having a long stream of instructions, the code is organized into independent subroutines, that may even be re-used in other programs.

- **object-oriented programming:** In this approach, the code is designed around the concept of *entities* (modeling e.g. some real-world objects). Each entity has an associated *internal state*, and also a set of *actions* that it can perform.

- **generic programming**: For this approach, the main idea is to write code which is applicable to different types of entities, by leveraging properties that those types of entities have in common at a more abstract level – for example, all numbers support addition, no matter if they are integers or reals. This approach is implemented in different ways by various programming languages – for example using the idea of *templates* or of *polymorphism*.

Combinations of these techniques have been successfully used for raising the level of abstractions in many application-domains, e.g. by constructing *software libraries* which are specialized for some part of the problem that needs to be solved. This approach has the benefit of being easier to adopt (for the users which are already familiar with the parent GPL). However, for expressing numerical algorithms for *high performance computing*s (HPCs) applications, a library makes it difficult to target multiple parallel backends, because the library lacks an abstract implementation of the algorithm that the user wants to implement. Also, using the library approach exposes (in the opinion of this author) users to concepts of the GPL which may not be directly relevant to their goal at hand (for example, many C++ libraries force users to learn about templates). An alternative approach is to design and implement a *domain-specific language* (DSL), which exposes to the user a more useful set of high-level concepts pertaining to the problem that needs to be solved.

### *The DSL approach for raising the level of abstraction*

As the name implies, a DSL is a programming language which is designed to increase productivity for users when they interact with a specific, well-defined, problem domain (Mernik et al., 2005). This approach is fundamentally different from that of *general-purpose language*s (GPLs), which aim to serve many communities of users, who may have very different needs. Both approaches have their strengths and inevitable weaknesses: while GPLs may become overwhelming for some users because they contain too many features they probably do not need (C++ is a good example), DSLs can feel limiting in some cases, because they lack some features that some users have come to expect (for example, they generally restrict the direct access to the lower-level system).

Although the term may seem exotic, DSLs are actually very common in normal computer usage. In fact, there are many types of DSLs – for *markup* (e.g. the LaTeX language used to typeset this thesis), *modeling* (e.g. UML diagrams of software systems), or *programming*. Other examples are the HTML format which powers the world wide web, the Bourne shell and its derivatives (used for automation in Unix environments), or the GNU Make language for automating compilation of software.

In the case of GeLB, the problem domain is that of implementing numerical simulations based on LBM algorithms.[2] As specific features which are handled by the DSL "behind the scenes", there is the issue of algorithm-parallelization (which is becoming progresssively harder to grasp, due to increase in *heterogeneity* of computer-architectures), efficient *input/output* (I/O) or implementation of support for pausing/resuming simulations.

Before closing this section, it is worth noting some of the advantages and disadvantages of DSLs:

- **advantages**
  - a new domain-specific programming language may allow its users to express solutions for problems in that domain more clearly, and
  - with less "boilerplate" (verbose) code
- **disadvantages**
  - DSLs are hard to design and to implement, and
  - users need to invest some time to learn to use them

## 4.2 PREVIOUS RELATED WORK

The idea of using DSLs to automatically generate code for numerical simulation is not new – for example, ATMOL (van Engelen, 2001) is an interesting DSL for formulating and implementing atmospheric models, while Pochoir (Tang et al., 2011) is a DSL for generating optimized C++ code from descriptions of FD kernels.

However, to the best knowledge of the author, this approach has not been proposed before in the LBM community. Instead, for LBM software libraries are common, where the core computational algorithm may be hidden under many layers (therefore discouraging experimentation with new algorithms). While such systems do have the benefit of being able to provide better performance for specific targets (e.g. the TheLMA framework of Obrecht et al. (2011)), they may also be restrictive in terms of choices of algorithms (which may or may not be a problem, depending on the needs of each user). Popular frameworks in the LBM community are for example the OpenLB C++ library (Heuveline and Latt, 2007), LBflow (Llewellin, 2010a,b), or WaLBerla (Feichtinger et al., 2011).

2 However, the framework can also be used for implementing more traditional *finite differences* (FD) algorithms, of which LBM can be viewed as a subset (at least from the point of view of the final implementation – the way LBM algorithms are derived is fundamentally different from that of FD.

## 4.3 how GeLB works and what it does

A high-level overview of the workflow with GeLB is shown in Figure 16. The user writes a so-called *GeLB description* (GD) file (shown in blue), which is then processed by the GeLB framework, eventually resulting in a *general-purpose language* (GPL) source-code file containing the code for the LBM simulation. Typically, the generated file is approximately one order of magnitude larger than the GD file that the user has to write. This size-difference is due to the technical infrastructure which GeLB implements to support the simulation, and it represents the main benefit of using GeLB. Finally, the generated GPL file is processed by a compiler suitable for the target machine, resulting in the final executable file (shown in red) that needs to be run to perform the simulation.

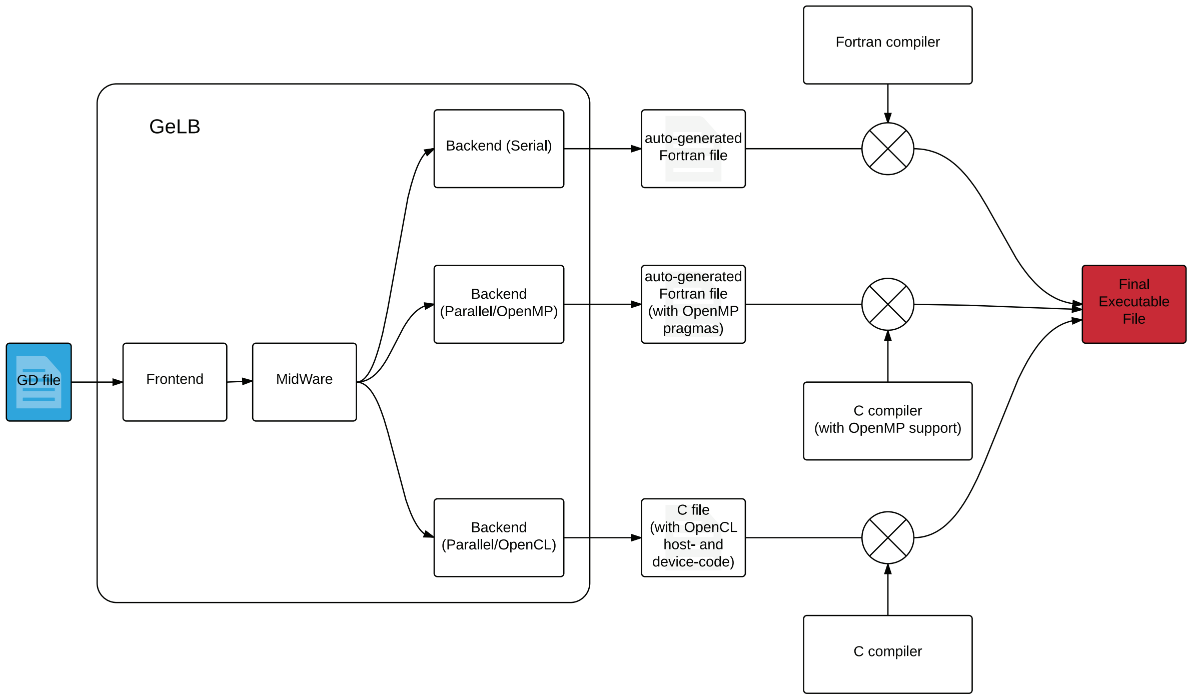

Fortran compiler
GeLB
Backend (Serial)
auto-generated Fortran file
Backend (Parallel/OpenMP)
auto-generated Fortran file (with OpenMP pragmas)
Final Executable File
GD file
Frontend
MidWare
C compiler (with OpenMP support)
Backend (Parallel/OpenCL)
C file (with OpenCL host- and device-code)
C compiler

Figure 16: Illustration of the components of the GeLB framework, along with their interactions with compilers for specific GPLs.

To get a better idea of how the GPL file is generated, it is useful to consider Figure 17, which illustrates how the numerical kernel (the "dynamics") typically acts upon the DFs in the lattice: in general terms, during each time-iteration a subset of the DFs at each lattice node is read from the lattice; these values are then used in computations (e.g. for computing of macroscopic fields and/or for applying the relaxation operator), after which another set of DFs is written back to the lattice (normally at the neighbouring nodes, to perform the streaming operation).

GeLB accounts for these steps, by allowing the user to write the equations in a compact notation, which is as close as possible to the mathematical formulation of the algorithms (as described in research papers). This compact notation is then parsed internally, to extract the exact geometric information about which DFs are altered by each kind of "dynamics". This information is then exploited by the backend, which

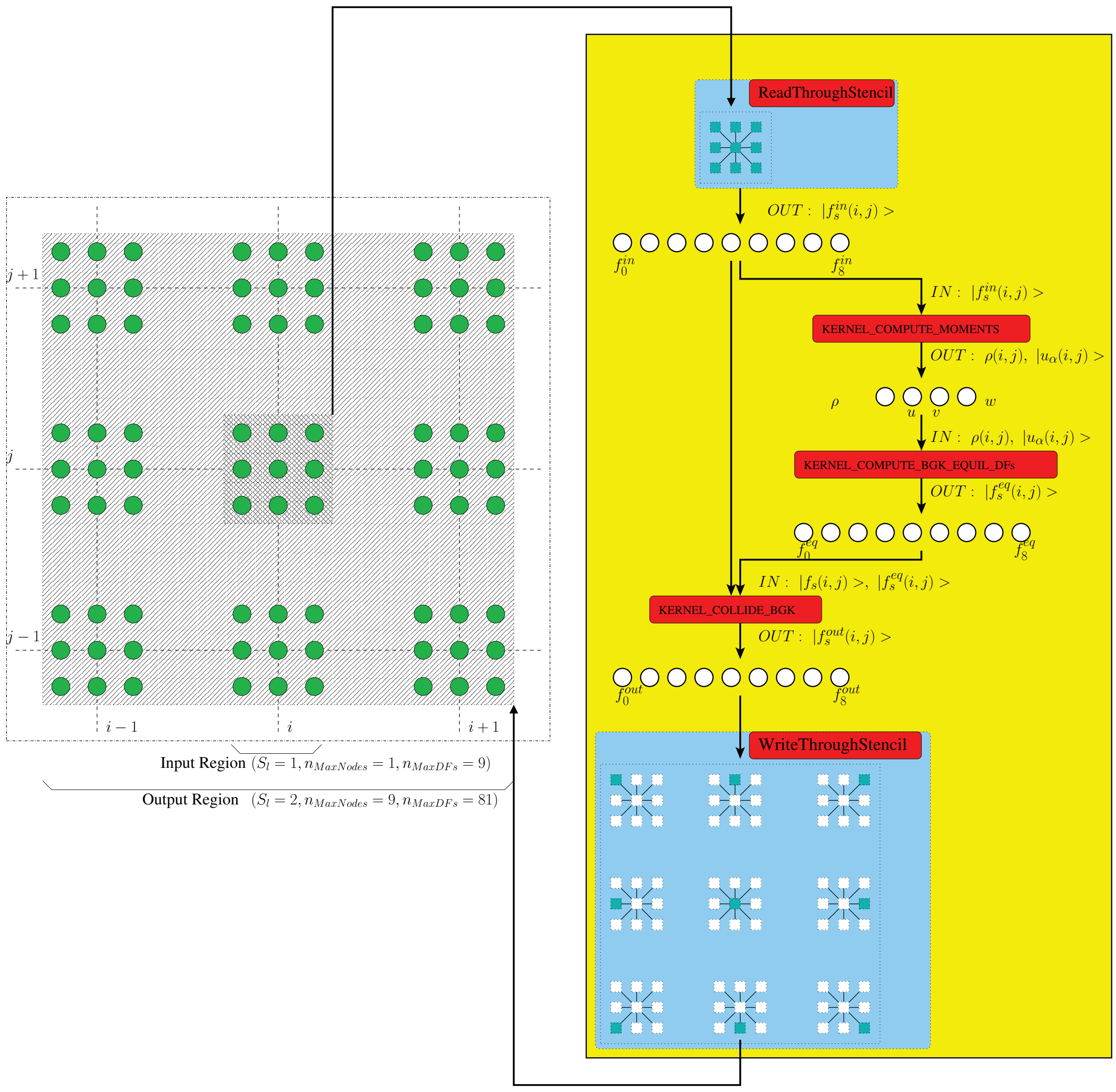


Figure 17: Illustration of the typical "dynamics" in LB algorithms.

decides how to perform the domain-decomposition for parallelization. Also, because it is very easy to make mistakes when writing down the algorithm, GeLB incorporates some sanity-checks, to ensure e.g. that the read/written DFs are not outside the lattice.

*Some notes on implementation details*

As shown in Figure 16, GeLB consists internally of three types of modules:

- **Frontend:** This module is responsible for parsing the GD program provided by the user, and for creating an *internal representation* of that program – to use compiler terminology, this representation is known as the *abstract syntax tree* (AST) of the program, and is designed to be *well-specified* (in the sense that the same program will always lead to the same AST after passing through the frontend). The frontend is based on a parser that was generated using the ANTLR4[3] framework (Parr, 2013), from a grammar-file describing the GD-language. Because the frontend has this knowledge about the grammar-file built-in, it is also the component which performs some initial validation of the program.

- **MidWare:** The component in the middle of the GeLB framework consists of several stages, which operate on the AST generated by the frontend, for example by expanding some of the compact notations from the GD language into more concrete forms, that can be understood by the backends.
- **Backends:** After the processing from the **MidWare** stages, the AST is finally translated to code in a *general-purpose language* (GPL) by the backends. Because each backend is specific to a particular type of parallelization, there are several backends in GeLB (of the ones shown in the figure, only the *Fortran-serial* and *Fortran-parallel-OpenMP* are currently implemented, with the *C-OpenCL* backend being developed at the time of this writing).

## 4.4 ELEMENTS OF A GELB PROGRAM (GD-FILE)

As already mentioned, GeLB operates on source-code files with the `.gd` extension. Such files, also known as *GeLB description* (GD) files, need to adhere to the syntax defined by the GeLB *domain-specific language* (DSL). During the design of the language, it was attempted to balance *simplicity* (to accelerate learning for new users) and *expressivity* (to reduce amount of boilerplace code needed for writing a simulation).

This section provides an overview of the elements that appear in GD files, explaining the purpose of each element.

### 4.4.1 *Comments*

Owing to the simplicity of the language, GD programs are hopefully self-explaining. However, it is highly recommended to add comments for explaining nontrivial aspects of the simulation.[4]

GD files support `C`-style comments, with the two variants:

1. **single-line comments**: such comments start with a double-slash (`//`), and continue until the end of the line:

```
... code (optional) ... // comment extends until end of line
```

3 Although most users do not need to be concerned with these details, it is maybe interesting to note that the ANTLR4 tool generates so-called *recursive-descent parsers*, composed of a set of recursive functions, one for each rule in the grammar-file.

4 A good guideline is to provide comments when it is not immediately clear *why* a certain choice was made. For example, in the case of LBM simulations, it is a good idea to motivate why a certain collision operator was chosen, to motivate the algorithms for implementing BCs, etc.

2. **multi-line comments**: such comments start with a `/*` and end with a `*/`; they may occupy part of a line, or even several lines:

```
/* comment on
   multiple lines */
```

All the text within comments is ignored by the GeLB compiler (`gelbc`), and will also not appear in the source-code file generated by the backends.

### 4.4.2 *Declaration of GeLB version*

To account for possible changes in syntax in future versions of the GD language, each simulation needs to specify explicitly the versions of the language being used. This thesis describes version 1, so the following line is included in all programs (by convention, as the first line in the file which is not a comment):

```
gelb_version = 1;
```

### 4.4.3 *CONFIGURATION block*

Each simulation needs a `configuration` block, which contains two mandatory sub-blocks: `simulation_metadata` and `constants`. Below is an example of a complete `configuration` block:

```
configuration {
  simulation_metadata {
    short_name = "2d_rb_mrt Ra 1900 Pr 7.1";
    description = R"
Simulate the 2D Rayleigh-Benard convection, with:
- no-slip + constant temperature BC at the horizontal walls, and
- periodic BCs at the vertical walls.
Ra = 1900
Pr = 7.1
";
  }

  constants {
    // sizes of the discrete domain
    int NX = 256;
    int NY = 256;
    // for ICs
    real RHO_0 = 1;
    real UV_0[2] = 0;
    real T_0 = 0;
    real RA = 1900.;
    real PR = 7.1;
    real MAX_MACH = 0.3;
  }
}
```

Listing 1: Example `configuration` block in GeLB programs

The sub-components are explained below.

#### 4.4.3.1 *SIMULATION_METADATA sub-block*

The `simulation_metadata` sub-block specifies two items:

- `short_name` This serves as an identifier, so that users can distinguish between different simulations. GeLB also uses this string during the construction of the *output file-names* (if any).

  > **NOTE**
  >
  > Because some operating-systems (particularly Unix descendants) do not cope well with file-names containing space characters, GeLB will replace any whitespace in this string to underscores when naming output files.

- `description` For documentation purposes, it is recommended to provide a more detailed description of each simulation. This is the role of the `description` field, which may extend over multiple lines. If the target output file-format supports metadata strings (e.g. the *NETwork Common Data Format* (`netCDF`) format), GeLB will insert this string as metadata into all output files.

  > **NOTE**
  >
  > Description strings need to start with `R"` and to end with a `"` character. This syntax is mostly due to historical reasons (compatibility with `C++11` raw string literals), and may be replaced in future releases of GeLB by a syntax closer to `Python`.

#### 4.4.3.2 *`CONSTANTS` sub-block*

The `constants` sub-block allows the user to specify an arbitrary number of global constants, which can be referred to from other parts of the simulation. In the listing above there are two constants of type `int` and six constants of type `real` (of which `UV_0` is a vector with two components).

The syntax for declaring individual constants is similar to other programming languages such as `C/C++` – the only difference is the absence of the `const` keyword, which is implied (declarations of normal variables are not allowed in the `constants` sub-block).

For the precise definition of the data-types `int` and `real` (and for the other options available) see Section 4.4.9.

#### 4.4.3.3 *Overriding configuration data with command-line arguments*

In many situations, it is useful to run a series of simulations, which differ only in the value of one (or a few) constants. To account for this frequent usage-pattern, it is possible to override values from the `constants`-block, by passing new values when `gelbc` is invoked. This idea was inspired by a similar feature in the `Chapel` programming language (Chamberlain, 2015). For example, assuming that `RA` is a constant in the program `my_simulation.gd`, the value from the simulation file can be overriden by invoking `gelbc` as follows:

```
gelbc my_simulation.gd -o backend_file_out_basename -DRA=1700.0
```

### 4.4.4 *LATTICE_PROPERTIES block*

Each simulation needs one (or more) `lattice_properties` block(s). These blocks contain declarations of constants which are specific to the type of lattice defined. Some constants are *mandatory*, while the others are *optional*.

The two *mandatory* constants, which each `lattice_properties` block needs to define, are `int D` and `int Q`. These values define the dimensionality of the lattice and the length of the state-vector at each node, respectively.

An example `lattice_properties` block is given below:

```
lattice_properties D2Q5 {
  int D = 2;
  int Q = 5;
  // DF direction-vectors
  array<int, {D, Q}> E = {{  0,  0 },  // 0
                          {  1,  0 },  // East
                          {  0,  1 },  // North
                          { -1,  0 },  // South
                          {  0, -1 }}; // West
  // matrix for mapping DFs -> moments
  array<real, {Q, Q}> N = {{  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 }};
  array<real, {Q, Q}> N_INV = inverse(N);
}
```

Listing 2: Example `lattice_properties` block in GeLB programs

### 4.4.5 *LATTICES block*

A GD simulation may use several lattices with the same `lattice_properties`. This is why it is necessary to explicitly specify what kind of lattices are used in the simulation, by listing them in the `lattices` block. This block contains one (or more) lattice declaration(s), where each declaration contains:

- the *name* of the lattice (as a variable-name),
- `properties` of the lattice (i.e. the name of one of the `lattice_properties` blocks defined in the simulation),
- `df_type` (which selects the data-type of the state-vectors at each node of the lattice), and
- `df_alias` (which is a shorter alias for the lattice, useful for making the code more concise).

An example `lattices` block for a simulation using two lattices is shown below:

```
lattices {
  lattice fluid(properties = D2Q9, df_type = real, df_alias = f);
  lattice temperature(properties = D2Q5, df_type = real, df_alias = g);
}
```

Listing 3: Example `lattices` block in GeLB programs

### 4.4.6 *`FUNCTION` block(s)*

GeLB supports four different types of functions. The first three types (`initializer`, `dynamic` and `gauge`) directly interact with the lattice(s) (as illustrated in Figure 18), while the fourth type is that of normal `function`s (as encountered in most programming languages).

**NOTE: Restrictions in GeLB**

To make optimization and parallelization possible, the GD language does not support flow-control structures (loops, if-clauses, etc.). This choice is *by design*, to make the problem of automatic parallelization tractable. From this point of view, all types of functions in GeLB are "pure", in the sense that the shape of the output of the functions and their exit-points are always the same (which facilitates generation of high-performance code within GeLB).

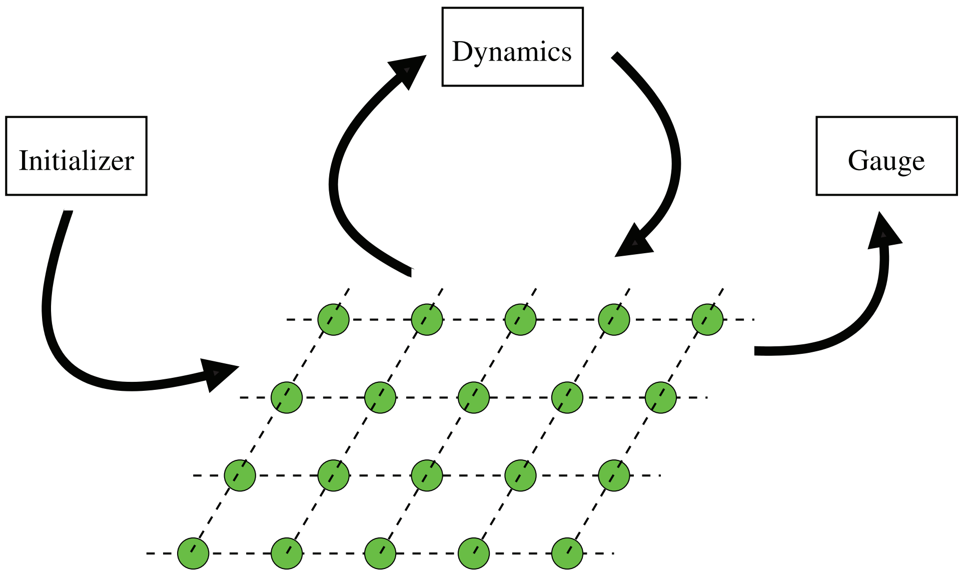


Figure 18: Illustration of the effects of the three types of functions supported in the GD language: *initializers* (=write-only) are used for imposing ICs onto the domain, *dynamics* (=read & write) are for specifying the core numerical algorithms explicitly, and *gauges* (=read-only) are used for evaluating quantities (e.g. macroscopic moments) which depend on the DFs in the lattice.

To give some illustrative examples, the BGK *single-relaxation-time* (SRT) operator is expressed with the `dynamic`:

```
dynamic CollideBGK {
  auto rho = evalRho();
  auto velo = evalVelo(rho);
  auto fEq = evalFEqBGK(rho, velo);
  f.out[x, y | i = {:}] = f.in[x, y | i] - (f.in[ x, y | i] - fEq[i]) * BGK_TAU_INV;
}
```

Listing 4: Example `dynamic` block in GeLB programs

A sample `initializer` is:

```
initializer ToEquil {
  f.out[x, y | i = {:}] = evalFEqBGK(RHO_0, U_0, V_0);
}
```

Listing 5: Example `initializer` block in GeLB programs

An example `gauge` is given below:

```
gauge RhoVelo[out(real rho, array<real, {2}> velo)] {
  rho = evalRho();
  velo = evalVelo(rho);
}
```

Listing 6: Example `gauge` block in GeLB programs

### 4.4.7 *NODE_CATEGORIES block*

As mentioned previously, the GD language does not support flow-control structures. However, some form of flow-control is necessary for deciding what type of functions to apply at each lattice node. This is achieved in GeLB with the `node_categories` block, where each "color" in the mesh is associated to a specific set of functions:

```
node_categories {
  BULK = category(ToEquil, {CollideBGK, Bulk_StreamIN}, RhoVelo);

  // VELOCITY (moving lid)
  NW_CORNER = category(ToEquil, {CollideBGK, BC_NW_Corner}, RhoVelo);
  N_EDGE = category(ToEquil, {CollideBGK, BC_N_Edge}, RhoVelo);
  NE_CORNER = category(ToEquil, {CollideBGK, BC_NE_Corner}, RhoVelo);

  // NO-SLIP (static walls)
  W_EDGE = category(ToEquil, {CollideBGK, BC_W_Edge}, RhoVelo);
  SW_CORNER = category(ToEquil, {CollideBGK, BC_SW_Corner}, RhoVelo);
  S_EDGE = category(ToEquil, {CollideBGK, BC_S_Edge}, RhoVelo);
  SE_CORNER = category(ToEquil, {CollideBGK, BC_SE_Corner}, RhoVelo);
  E_EDGE = category(ToEquil, {CollideBGK, BC_E_Edge}, RhoVelo);
}
```

Listing 7: Example `node_categories` block in GeLB programs

### 4.4.8 *GEOMETRY block*

To specify the geometry of the simulations, the `geometry`-block is used. Currently, GeLB supports two methods for specifying the geometry:

1. The first method for specifying the geometry is based on a "painting" metaphor, where each line assigns a specific kind of "color" for the nodes selected on

the LHS. This approach should be used for simulations with *simple* geometries (because it allows the `gelbc` compiler to generate higher-performance code).

For example, for the lid-driven cavity problem, we may have:

```
geometry {
  set_grid_size(NX = 256, NY = 256);
  // LEFT (WEST)
  node[ x = 0, y = 0 ] = SW_CORNER;
  node[ x = 0, y = NY - 1 ] = NW_CORNER;
  node_range[ x = {0}, y = {1 : NY - 2} ] = W_EDGE;
  // MIDDLE
  node_range[ x = {1 : NX - 2}, y = {0} ] = S_EDGE;
  node_range[ x = {1 : NX - 2}, y = {NY - 1} ] = N_EDGE;
  node_range[ x = {1 : NX - 2}, y = {1 : NY - 2} ] = BULK;
  // RIGHT (EAST)
  node[ x = NX - 1, y = 0 ] = SE_CORNER;
  node[ x = NX - 1, y = NY - 1 ] = NE_CORNER;
  node_range[ x = {NX - 1}, y = {1 : NY - 2} ] = E_EDGE;
}
```

Listing 8: Example `geometry` block in GeLB programs

2. A problem with the first method for specifying the geometry is that it does not scale well for *complex geometries*. Therefore, GeLB also supports an alternative method, based on reading the geometry-data directly from a file. For example, in the case of the *wind-driven ocean circulation* (WDOC) simulations with the realistic land-mask (presented in Section 5.3), the geometry block is simply:

```
geometry {
  read_geometry_from_file(''wdoc_geometry.nc'');
}
```

Listing 9: Example `geometry` block in GeLB programs with *complex* geometries

### 4.4.9 *Data-types supported in the GD-language*

The following data-types are currently supported for *variables* and *constants* in GD programs.

- **boolean**: `bool` (`true` or `false`)
- **integer**: signed two's complement integer (explicit # of bits): `int8`, `int16`, `int32`, `int64`

  By default (# of bits omitted) we have `int ≡ int64`.
- **floating-point** (explicit # of bits): `real16`, `real32`, `real64`

  By default (# of bits omitted) we have `real ≡ real64`.

It is also possible (and recommended, whenever possible, for performance) to define data of *constant type*, by appending `_const` to the name of the type (e.g. `real_const`).

### 4.4.10 *Non-scalar variables and constants*

Scientific computing applications operate more often on *vectors*, *matrices*, or *multi-dimensional arrays*. Therefore, the GD language allows the user to declare variables and constants to be of arbitrary dimensionality (as long as the shape is "rectangular", and it does not change at runtime). This is achieved with the `array` type.

As concrete examples, to declare a one-dimensional vector with 4 `real` elements, we would use:

```
array<real, {4}> my_array;
```

and to declare a *constant* three-dimensional array with 2, 3 and 4 `int` elements along the $x$, $y$ and $z$ axes respectively, we would use:

```
array<real_const, {2, 3, 4}> my_const_array;
```

## 4.5 AUTOMATIC GENERATION OF PARALLEL CODE

As mentioned already, GeLB functions are expected to be "pure", with static inputs and outputs, and single exit-points. Because of this restriction, the `gelbc` compiler can determine the stencils for the `initializer`s, `dynamic`s and `gauge`s at every node of the lattice. By adding to this the information from the `node_categories` and from the `geometry` blocks, the complete data-dependencies between the *distribution functions* (DFs) in the model-state is known, which allows (in principle) automatic generation of parallel code. The `gelbc` compiler uses this information to generate parallel code when possible.

# 5 APPLICATIONS

"There are no such things as applied sciences, only applications of science"

Louis Pasteur

The GeLB framework, which was presented in Chapter 4, was built with the aim of being useful for quickly writing simulations based on the *lattice Boltzmann method* (LBM). Although the framework itself may appeal to readers with *computer science* (CS) inclinations, it is also necessary to demonstrate concrete applications to problems in fluid dynamics. This chapter discusses three such applications, starting with the two-dimensional RB problem (Section 5.1). Because of the simple 2D geometry, this example is also the easiest to implement, and not too demanding from a computational point of view. Next (Section 5.2) we extend the setup from two to three spatial dimensions. Although the problem is very similar, it can already be classified as a respectable setup in HPC, owing to the $\mathcal{O}(n^2) \rightsquigarrow \mathcal{O}(n^3)$ change in scaling of number of nodes with resolution. For the third and last application (Section 5.3), a new numerical model was developed (based on the *multiple-relaxation-times* (MRT) LBM approach), to simulate the wind-driven circulation in a barotropic ocean. Although this application is not very demanding from a computational point of view, it is a good test for studying how well LBM methods can simulate flows which are directly relevant to oceanography. Also, in the context of GeLB, this example demonstrates how to impose a complex flow-geometry (in this case, representing a realistic ocean land-mask).

## 5.1 RAYLEIGH-BÉNARD (RB) CONVECTION (2D)

Here we study the evolution of a quasi-incompressible fluid, contained between two horizontal plates (where the temperature of the lower plate is *higher*).[1] The temperature gradient in the problem creates a small density-gradient which, if strong-enough, can cause convection to occur. Compared to heat diffusion at the molecular level, convection dramatically enhances the heat transfer, and is therefore important for many applications (from engineering to *Earth system science* (ESS)). As one of the simplest setups where convection can develop, the physics of this problem is of great interest for understanding how such flows shift between regimes which are qualitatively very different (ranging from no motion at all to turbulent convection). Interest-

1 The alternative scenario, where the temperature gradient is reversed, is *unconditionally* stable, and therefore not an interesting problem to study.

ingly, it also attracted much attention from the dynamical systems community, due to the tendency of the flow to self-organize into distinct patterns.

### 5.1.1 *Model setup for the* 2D *case*

The discussion of the RB problem in Section 2.4 was framed in the physical 3D-space. However, several problems in fluid dynamics may also be considered to behave like lower-dimensional (e.g. 2D) systems, at least for certain ranges of the control-parameters.[2] This is also the case for the RB problem, where we have a region in parameter-space where the flow may be considered to be 2D. Strictly-speaking, this assumption is difficult to reproduce experimentally, due to the ever-present noise in the apparatus (which usually depends on $z$ also). However, as discussed in Section 2.4.4, the first transition of the flow-regime (from a stationary state to convective flow) is two-dimensional in nature, so even the highly-idealized 2D model can provide valuable insights into the physics of the problem. On the other hand, outside this parameter-space region this approximation breaks in a fundamental way (as the numerical experiments presented later in this thesis will indicate).

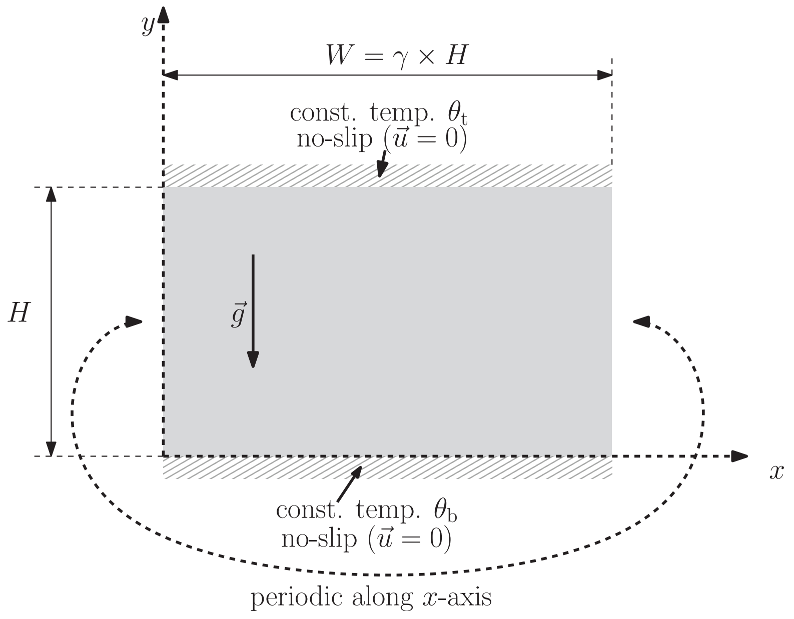


Figure 19: Sketch of geometry for the *Rayleigh-Bénard* (RB) simulations in 2D. For the present simulations, the aspect-ratio was fixed to $\gamma = \frac{2\pi}{k_{cr,1}} \approx 2.0158$.

As shown in Figure 19, for this first model setup we assume a rectangular geometry, with $y \in [0, H]$ and $x \in [0, W = \gamma H]$, where we take the *aspect-ratio* $\gamma = 2.0158$.[3]

2 Using the jargon of the theory of dynamical systems, it can be said that the higher-dimensional system still includes some of the bifurcations of the lower-dimensional system, as a *center manifold* (Olbers et al., 2012).

3 This particular value of the aspect-ratio was chosen, to make sure that the domain is sufficiently-large horizontally to allow for the first instability to develop (as predicted by the theory). While this may seem an artificial choice, it is necessary due to the periodic lateral BCs used. Note that this does not affect the validity of the numerical experiments, since the quantity of interest here is the value of the Rayleigh number for which the mode with wavenumber $k_{cr,1}$ becomes amplified.

It is assumed that all gradients along the $z$-direction vanish, so that the problem can be considered to be two-dimensional. Also, the problem is assumed to be periodic in the $x$-direction, effectively "wrapping" the domain into a cylinder-topology. For the flow-regimes studies later in this section, this is a good approximation of a domain which is "infinite" along this axis, which is how the RB problem is normally formulated in analytic studies.[4]

### 5.1.2 *Discretization of the dimensionless equations*

We mentioned in Section 2.4.3 that in most[5] branches of fluid dynamics the flow-results are reported in dimensionless units, which facilitates comparison between experiments, theory, and simulation. However, numerical algorithms are often formulated in yet another system of units, referred to as "the *numerical system*". The numerical system is designed to map as naturally as possible to the final computer implementation, where we prefer, for example, to discuss in terms of spatial indices of the nodes in the numerical mesh and of iteration-numbers. However, the numerical results have to be transferred back to the dimensionless units (in the output- and pre-processing steps).

The transition from the dimensionless system to the numerical system coincides with the discretization of space-time, which consists of chosing (a) the number of lattice-nodes $N_y$ to represent the unit of length, and (b) the number of time-iterations $N_t$ to represent the unit of time. In the numerical system, then, $N_y$ and $N_t$ act as the reference length and time, respectively. Relative to the dimensionless system, the transformation to the numerical system proceeds along the same lines as for the physical system, as summarized below:

$$x_j^{(d)} = \frac{x_j^{(n)} - 1/2}{N_y} \Longleftrightarrow x_j^{(n)} = 1/2 + N_y x_j^{(d)} \tag{185}$$

The $1/2$-term in eq. (185) above is due to the BCs: (a) for the horizontal walls, the so-called *bounce-back* (BB) algorithm for implementing the no-slip velocity BC effectively places the wall half-way between the last fluid node and the first solid node, and (b) the periodic domain-wrapping along $x$ also introduces a shift of equal magnitude.

4 However, it also has the disadvantage of limiting the wavenumbers of the permitted oscillations.
5 Some exceptions are accepted sometimes in the climate- and weather-simulation communities, because we currently have only one planet to consider.

Continuing with the transformations, we have:

$$t^{(d)} = \frac{1}{N_t} t^{(n)} \iff t^{(n)} = N_t t^{(d)} \tag{186}$$

$$u_j^{(d)} = \frac{N_t}{N_y} u_j^{(n)} \iff u_j^{(n)} = \frac{N_y}{N_t} u_j^{(d)} \tag{187}$$

$$(\delta p)^{(d)} = \frac{N_t^2}{\rho_0^{(n)} N_y^2} (\delta p)^{(n)} \iff (\delta p)^{(n)} = \frac{\rho_0^{(n)} N_y^2}{N_t^2} (\delta p)^{(d)} \tag{188}$$

$$(\delta T)^{(d)} = \frac{(\delta T)^{(n)}}{(\Delta T_0)^{(n)}} \equiv \frac{T^{(n)} - T_0^{(n)}}{(\Delta T)^{(n)}} \iff T^{(n)} = T_0^{(n)} + (\Delta T_0)^{(n)} (\delta T)^{(d)} \tag{189}$$

To simplify the numerics in the equations above, it is common to take $\rho_0^{(n)} = 1$ and also $(\Delta T_0)^{(n)} = 1$. Also, for LBM models of temperature it is generally better to keep the temperature-ranges positive (because of the interpretation of the DFs as particle-densities, it is better to ensure that they are always positive, to be on the safe side); therefore, we choose $T_0^{(n)} = +\frac{1}{2}$, which transforms the scaling for temperature to:

$$T^{(n)} = (\delta T)^{(d)} + \frac{1}{2} \iff (\delta T)^{(d)} = T^{(n)} - \frac{1}{2} \tag{190}$$

Plugging the scaling equations above into the dimensionless governing equations, the following equations are obtained for the model parameters:

$$\nu^{(n)} = \frac{\mathrm{Pr}\, N_y^2}{N_t} \tag{191}$$

$$(\alpha g)^{(n)} = \frac{\mathrm{Ra}\,\mathrm{Pr}\, N_y}{N_t^2} \tag{192}$$

$$\kappa^{(n)} = \frac{N_y^2}{N_t} \tag{193}$$

Finally, it is necessary to specify the BCs and ICs for the simulations. Because there is no non-zero velocity which is to be enforced at the boundaries, it is only necessary to perform the scaling for the temperature (BCs and ICs) and for pressure (only ICs). Using the relations above, the BCs for temperature become:

$$T^{(n)}\left(x^{(n)}, \frac{1}{2}, t^{(n)}\right) = 1 \tag{194}$$

$$T^{(n)}\left(x^{(n)}, N_y + \frac{1}{2}, t^{(n)}\right) = 0 \quad , \tag{195}$$

and the ICs for temperature and pressure are:

$$T^{(n)}\left(x^{(n)}, y^{(n)}, 0\right) = 1 - \frac{1}{2N_y}\left(2y^{(n)} - 1\right) \tag{196}$$

$$(\delta p)^{(n)}\left(x^{(n)}, y^{(n)}, 0\right) = \frac{(\alpha g)^{(n)}}{2N_y}\left(y^{(n)} - \frac{1}{2}\right)\left(N_y + \frac{1}{2} - y^{(n)}\right) \tag{197}$$

## 5.1.3 *Numerical method*

To solve numerically eqs. (54) to (56), we use a LBM based on the MRT approach (Wang et al., 2013). The hydrodynamic and temperature equations are solved on two separate lattices: D2Q9 (Figure 21) and D2Q5 (Figure 20) respectively.

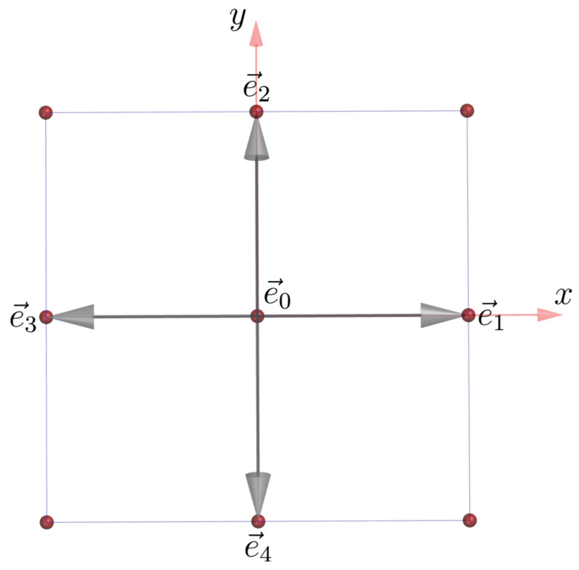


Figure 20: D2Q5 lattice (for modeling advection-diffusion of temperature)

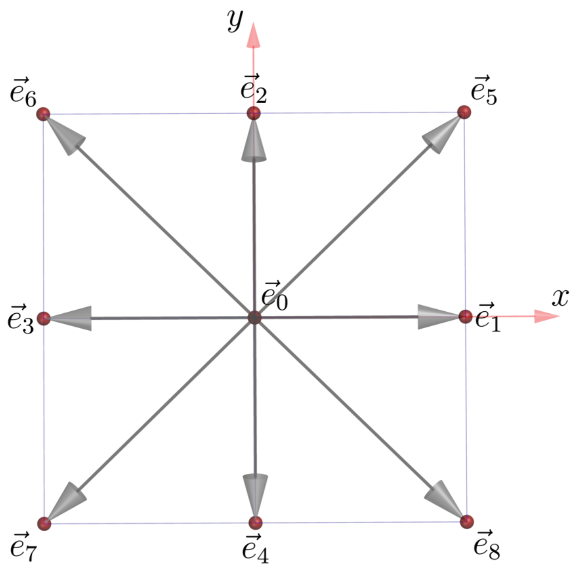


Figure 21: D2Q9 lattice (for modeling isothermal hydrodynamic fields)

### 5.1.3.1 *Model for the fluid component*

The DFs for the fluid solver evolve according to the rule:

$$\underbrace{f_i\left(\mathbf{x}^{(n)} + \mathbf{e}_i\delta_t^{(n)},\ t^{(n)} + \delta_t^{(n)}\right) = f_i\left(\mathbf{x}^{(n)},\ t^{(n)}\right)}_{\text{streaming}} - \left(\underbrace{M^{-1} \quad \underbrace{S\left[\mathbf{m} - \mathbf{m}^{eq}\right]}_{\text{relaxation of moments}}}_{\text{collision operator}}\right)_i \tag{198}$$

with $i \in \{0, \ldots, 8\}$.

The 9 discretized velocities associated to each DF at all lattice nodes are shown below, packed in a matrix $\tilde{e}$:

$$\tilde{e} \equiv \begin{pmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{pmatrix} \equiv (\mathbf{e}_0, \dots, \mathbf{e}_8) = c^{(n)} \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{pmatrix} \tag{199}$$

The scaling factor $c^{(n)}$, also known as the "basic lattice speed", depends on the lattice spacing $\delta_x^{(n)}$ and time step $\delta_t^{(n)}$:

$$c^{(n)} \equiv \frac{\delta_x^{(n)}}{\delta_t^{(n)}} \tag{200}$$

Based on the discussion about discretization from Section 5.1.2, we have $\delta_x^{(n)} = 1$ and also $\delta_t^{(n)} = 1$ by definition.

In eq. (198), the vector $\mathbf{m}$ consists of *moments* of the local DFs, defined by the linear transformation $\tilde{M}$:

$$\mathbf{m} \equiv \left(\rho^{(n)}, j_x^{(n)}, j_y^{(n)}, e^{(n)}, p_{xx}^{(n)}, p_{xy}^{(n)}, q_x^{(n)}, q_y^{(n)}, \epsilon^{(n)}\right)^\dagger = \tilde{M}\mathbf{f} \tag{201}$$

The physical significance of each moment is given below:

- $\rho^{(n)}$ – fluid density
- $j_x^{(n)}$, $j_y^{(n)}$ – $x$- and $y$-components of the fluid momentum
- $e^{(n)}$ – fluid energy
- $p_{xx}^{(n)}$, $p_{xy}^{(n)}$ – diagonal and off-diagonal components of the symmetric traceless viscous stress tensor
- $q_x^{(n)}$, $q_y^{(n)}$ – $x$- and $y$-components of the energy flux
- $\epsilon^{(n)}$ – related to square of the fluid energy

During the collision step, all moments are relaxed towards their corresponding equilibrium values $\mathbf{m}_i^{eq}$, defined as:

$$\begin{cases} m_0^{eq} = (\delta\rho)^{(n)} \\ m_1^{eq} = \rho_0^{(n)} u_x^{(n)} \\ m_2^{eq} = \rho_0^{(n)} u_y^{(n)} \\ m_3^{eq} = -2(\delta\rho)^{(n)} + 3\rho_0^{(n)} \left[ \left(u_x^{(n)}\right)^2 + \left(u_y^{(n)}\right)^2 \right] \\ m_4^{eq} = \rho_0^{(n)} \left[ \left(u_x^{(n)}\right)^2 - \left(u_y^{(n)}\right)^2 \right] \\ m_5^{eq} = \rho_0^{(n)} u_x^{(n)} u_y^{(n)} \\ m_6^{eq} = -\rho_0^{(n)} u_x^{(n)} \\ m_7^{eq} = -\rho_0^{(n)} u_y^{(n)} \\ m_8^{eq} = (\delta\rho)^{(n)} - 3\rho_0^{(n)} \left[ \left(u_x^{(n)}\right)^2 + \left(u_y^{(n)}\right)^2 \right] \end{cases} \tag{202}$$

where the macroscopic variables $\rho^{(n)}$ and $u_j^{(n)}$ are evaluated from the local DFs:

$$\rho^{(n)} = \rho_0^{(n)} + (\delta\rho)^{(n)} \equiv \rho_0^{(n)} + \sum_{i=0}^{8} f_i \tag{203}$$

$$u_j^{(n)} = \frac{1}{\rho_0^{(n)}} \sum_{i=0}^{8} e_{i,j} f_i, \quad \text{with} \quad j \in \{x, y\}. \tag{204}$$

The transformation matrix is obtained through a variant of Gram-Schmidt orthogonalization (Bouzidi et al., 2001):

$$\tilde{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{205}$$

In Section 5.1.2, the numerical value for pressure in the numerical system has been referred to. However, it should be noted that the LBM algorithm used does not recover pressure directly – instead, this quantity is computed from the small density-fluctuations, using an ideal-gas-like equation of state:

$$(\delta p)^{(n)} = \left(c_s^{(n)}\right)^2 (\delta\rho)^{(n)} = \frac{1}{3}\left(c_s^{(n)}\right)^2 (\delta\rho)^{(n)}, \tag{206}$$

where $c_s^{(n)}$ is the (lattice-dependent) pseudo-speed-of-sound, which for this model (D2Q9 lattice) is:

$$c_s^{(n)} = \frac{c^{(n)}}{\sqrt{3}}. \tag{207}$$

SIGNIFICANCE OF DENSITY-FLUCTUATIONS IN THE LBM SOLVER Although the physical problem to be solved is incompressible, the LBM algorithm solves it as a quasi-compressible problem (and, as seen in eq. (206), there is even a gas-like equation of state which is internal to the model). However, perhaps unintuitively, the model is intended to simulate *incompressible* flows (and is *not suitable* for compressible flows as it stands), because the "speed of sound" is chosen (indirectly, through the choice of discretization parameters), to be much lower than that of real gases – the quasi-compressibility is used only to avoid solving a global Poisson problem for pressure, which is a common obstacle to parallelization in more traditional CFD algorithms. In addition, the model was constructed such that the "pseudo-compressibility" errors scale proportionally to $(\mathrm{Ma})^2$, where Ma is the "lattice Mach number" (i.e. ratio of the numerical velocities to the "pseudo-speed-of-sound"). Therefore, if we ensure that the maximum Mach number in the flow does not exceed some conservative threshold (e.g. $\mathrm{Ma} \leqslant 0.1$), we can keep the pseudo-compressibility errors small.

The rate at which each moment is relaxed is given by the (diagonal) relaxation matrix $\tilde{S}$ is:

$$\tilde{S} = \mathrm{diag}(0,\ 1,\ 1,\ s_e,\ s_\nu,\ s_\nu,\ s_q,\ s_q,\ s_\epsilon) \tag{208}$$

The coefficients of this matrix are chosen to optimize stability (by relaxing non-hydrodynamic moments faster than the hydrodynamic ones). One possible choice (which is used in this thesis) is the *two-relaxation-times* (TRT) model, for which:

$$s_\nu = s_e = s_\epsilon, \tag{209}$$

leading to:

$$\tilde{S} = \mathrm{diag}(0,\ 1,\ 1,\ s_\nu,\ s_\nu,\ s_\nu,\ s_q,\ s_q,\ s_\nu) \tag{210}$$

where the adjustable parameter $s_\nu$ determines the kinematic viscosity of the model:

$$\nu^{(n)} = \frac{1}{3}\left(\frac{1}{s_\nu} - \frac{1}{2}\right) \tag{211}$$

and $s_q$ is related to $s_\nu$ via:

$$s_q = 8\frac{2 - s_\nu}{8 - s_\nu} \tag{212}$$

For physical reasons (to prevent a negative viscosity), it is necessary (Ginzburg and D'Humieres, 2003; Wang et al., 2013) to have:

$$\{s_\nu, s_q\} \in [0, 2) \times [0, 2) \tag{213}$$

BODY FORCES The LBM evolution eq. (198) does not take into account any body-forces to which the fluid may be subjected. A common approach for including such effects (especially in SRT models) is to add some force-terms to the RHS. Such corrections for the forcing term at the DF-level have to be carefully constructed (Guo et al., 2002b), to recover the correct equations at the Navier-Stokes level. Fortunately, for the MRT models, the force can be added directly to the corresponding moment[6], which is much more natural. Dellar (2013) showed that such an implementation of the force-term is 2nd-order accurate if a procedure known as "Strang splitting" is used, according to which (a) half of the force-term is added *before* the collision, and (b) the other half *after* the collision.

BCs As already mentioned, the periodic BCs used at the horizontal walls are enforced directly at the implementation-level, because there are no parameters to consider for this type of BCs (they only lead to a "wrapping" of the mesh-topology). More specifically, the streaming of the DFs along the $y$-axis is constrained to produce results within the domain, using a modulo-operation. Therefore, only the implementation of the *no-slip* BCs along the horizontal walls needs to be mentioned explicitly. This type of Neumann (1887) BC is normally implemented in LBM using the so-called *bounce-back* (BB) scheme, according to which the post-collision DFs that would be moved to a solid node by normal streaming are copied instead to the local node, but with the *opposite orientation*. Mathematically, this process can be written as:

$$f_{\bar{i}}^{\text{pre-collision}}\left(\mathbf{x}_f^{(n)}, t^{(n)} + \delta_t^{(n)}\right) = f_i^{\text{post-collision}}\left(\mathbf{x}_f^{(n)}, t^{(n)}\right) \tag{214}$$

where the overline is used to denote the discrete vector with opposite orientation:

$$-\mathbf{e}_i = \mathbf{e}_{\bar{i}} \tag{215}$$

and $\mathbf{x}_f^{(n)}$ is the position of the fluid node adjacent to the solid boundary.

Since no DFs are "lost" or "gained", this approach has the important advantage of ensuring conservation of mass and momentum. This scheme is also local (good for parallelization), and easy to implement (hence its popularity). Unfortunately, it is in general only $\mathcal{O}\left(\delta_x^{(d)}\right)$. However, for planar boundaries which are also axis-aligned

6 Specifically, for the present model the force contributes to the $m_2$ moment, since the gravitational force acts along the $y$-axis.

(such as in this application), it can be shown that the BB scheme becomes $\mathcal{O}\left(\left(\delta_x^{(d)}\right)^2\right)$, <u>if</u> the wall is positioned halfway between the last fluid node and the neighbouring wall node [7]. For more generic applications, other schemes may be preferred, which are second-order accurate in space even when the boundaries are not axis-aligned (or not planar in the first place) – some good options are mentioned e.g. by Mei et al. (1999, 2000) and Guo et al. (2002a).

ICs At the beginning of simulations, the velocity- and pressure-fields need to be initialized to the values specified in the previous sections. For LB methods, this ultimately means that some initial values need to be assigned to the DFs. A common approach is to set the DFs to their equilibrium Maxwellian values. However, for the MRT model used here it is easier to calculate the initial DFs from the moments, and applying the inverse transformation operator $\tilde{M}^{-1}$:

$$\mathbf{f}_{\text{initial}} = \tilde{M}^{-1}\mathbf{m}_{\text{initial}}, \tag{216}$$

where $\mathbf{m}_{\text{initial}}$ is evaluated from the initial macroscopic fields (and setting the other moments to zero).

### 5.1.3.2 *Model for the temperature component*

For solving the temperature advection-diffusion eq. (56), also a MRT model (which is similar in nature to the fuild model) is used (Wang et al., 2013). However, because the temperature equation does not involve higher-order quantities (such as the stress-tensor that needs to be resolved in the hydrodynamic model), a model with a lower number of DFs (specifically, the D2Q5 lattice – Figure 20) is sufficient. This ultimately saves computer memory and, indirectly, computation-time (by putting less pressure on the memory sub-system). The DFs for this temperature solve evolve according to the equation:

$$\underbrace{g_i\left(\mathbf{x}^{(n)} + \mathbf{e}_i\delta_t^{(n)}, t^{(n)} + \delta_t^{(n)}\right) = g_i\left(\mathbf{x}^{(n)},\ t^{(n)}\right)}_{\text{streaming}} - \left(\underbrace{N^{-1}\quad\underbrace{Q\left[\mathbf{n} - \mathbf{n}^{eq}\right]}_{\text{relaxation of moments}}}_{\text{collision operator}}\right)_i, \tag{217}$$

with $i \in \{0, \dots, 4\}$.

The 5 discretized velocities for the model are given in the matrix $\tilde{e}$:

$$\tilde{e} \equiv \begin{pmatrix} \mathbf{e}_x \\ \mathbf{e}_y \end{pmatrix} \equiv (\mathbf{e}_0, \dots, \mathbf{e}_4) = e^{(n)}\begin{pmatrix} 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{pmatrix} \tag{218}$$

7 This displacement of the boundary relative to the last fluid node needs to be taken into account during the initialization and postprocessing stages (as discussed in Section 5.1.2 above).

where the same space-discretization is used as for the fluid model, s.t. $e^{(n)} = \delta_x^{(n)}/\delta_t^{(n)}$, with $\delta_x^{(n)} = 1$ and $\delta_t^{(n)} = 1$ by definition.

To obtain the temperature moments $\mathbf{n}$ from the DFs of this model, a linear transformation $\tilde{N}$ is applied:

$$\mathbf{n} = \tilde{N}\mathbf{g} \tag{219}$$

with:

$$\tilde{N} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -4 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 1 & -1 \end{pmatrix} \tag{220}$$

During the collision step, all moments are relaxed towards their corresponding equilibrium values, which are:

$$\mathbf{n}^{eq} = (T^{(n)}, u_x^{(n)}T^{(n)}, u_y^{(n)}T^{(n)}, aT^{(n)}, 0)^{\dagger} \tag{221}$$

where:

- the macroscopic temperature is evaluated from the DFs:

$$T^{(n)} = \sum_{i=0}^{4} g_i \tag{222}$$

- $a$ is a model parameter which influences the thermal diffusivity
- $u_x^{(n)}$, $u_y^{(n)}$ represent the local components of the velocity, as evaluated from the fluid solver

The relaxation rates form the diagonal matrix $\tilde{Q}$:

$$\tilde{Q} = \mathrm{diag}(0,\ \sigma_\kappa,\ \sigma_\kappa,\ \sigma_e,\ \sigma_\nu) \tag{223}$$

The thermal diffusivity for this model is determined by the parameters $\sigma_\kappa$ and $a$ (which appeared in eq. (221)).

$$\kappa^{(n)} = \frac{4+a}{10}\left(\frac{1}{\sigma_\kappa} - \frac{1}{2}\right) \tag{224}$$

Similar to the fluid MRT model, the parameters $\sigma_i$ and $a$ are tuned to optimize stability and accuracy. In this study, they are set to:

$$\sigma_\kappa = 3 - \sqrt{3}, \tag{225}$$

$$\sigma_\nu = 2(2\sqrt{3} - 3), \quad \text{and} \tag{226}$$

$$\sigma_e = \sigma_\nu, \tag{227}$$

which simplifies the matrix of relaxation-rates to:

$$\tilde{Q} = \mathrm{diag}(0,\ \sigma_\kappa,\ \sigma_\kappa,\ \sigma_\nu,\ \sigma_\nu) \tag{228}$$

Also, the thermal diffusivity of the model is only controlled by the parameter $a$:

$$\kappa^{(n)} = \frac{\sqrt{3}}{60}(4+a), \quad \text{with} \quad -4 < a < 1 \tag{229}$$

BCs Several methods are available for imposing the constant temperature BC at the horizontal walls. The obvious approach of setting the DFs at the boundaries to a Maxwellian-equilibrium distribution can lead to unphysical heat-fluxes along the walls (which break the 2nd-order accuracy). To avoid such effects, the so-called *anti-bounce-back* (ABB) scheme is used instead (Kuo and Chen, 2009):

$$g_{\bar{i}}^{\text{pre-collision}}(\mathbf{x}_f^{(n)}, t^{(n)} + \delta_t) = -g_i^{\text{post-collision}}(\mathbf{x}_f^{(n)}, t^{(n)}) + 2\sqrt{3}\kappa^{(n)} T_{\text{wall}}^{(n)} \tag{230}$$

where the overline denotes the reversed direction vector.

ICs As for the fluid model, the initial DFs for the temperature model are re-constructed from the components of the moments $\mathbf{n}$ corresponding to the macroscopic variables:

$$\mathbf{g}_{\text{initial}} = \tilde{N}^{-1}\mathbf{n}_{\text{initial}} \tag{231}$$

### 5.1.3.3 *Trigger for symmetry-breaking*

Whereas in real-world fluid dynamics experiments there is always some unavoidable noise, in CFD simulations it can also be the case that the simulation is too stable (from a physical point of view), because the initial trigger is missing. This is also the case for the simulations presented later, especially when the periodicity is considered. Therefore, to avoid this situation and to allow the flow to develop like in the real world, a small amount of white noise is added to the initial temperature-field drawn from a uniform distribution with values in the interval $[0, 10^{-8}]$. This noise signal ensures that the flow has a wide range of spatial modes to choose from, so that the system can amplify the physically-realizable spatial frequencies, and dampen the ones which are not consistent with the particular Rayleigh number that is prescribed.

In addition, to make the plots easier to interpret, a larger perturbation of $10^{-5}$ is applied at the point $\left(x^{(n)}, y^{(n)}\right) = \left(\frac{N_x}{2} + 1, 2\right)$. The role of this larger perturbation is to set the alignment of the resulting convection cells (for the cases when convection develops).

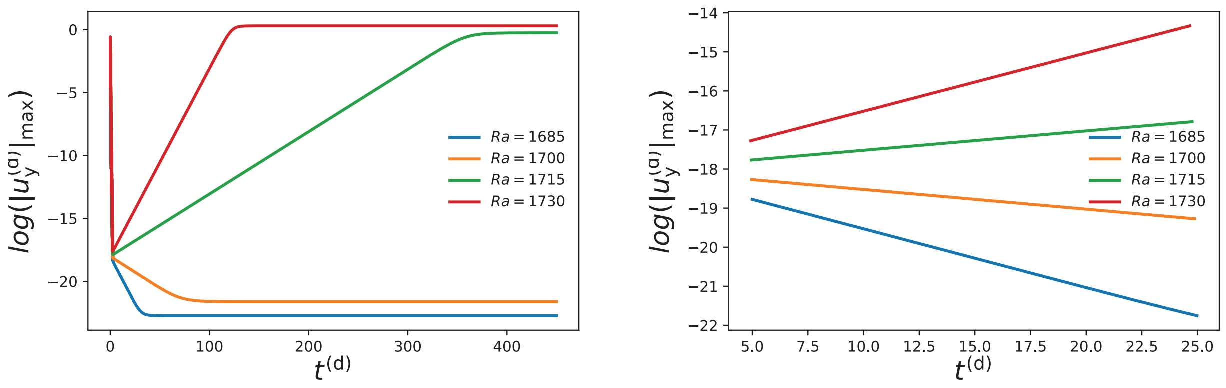


Figure 22: The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{max}$ for different values of the Rayleigh number (left) and zoomed-in plot of the same data, for the linear regime (right). All simulations were performed on a $83 \times 41$ grid, with $Ma = 0.1$.

### 5.1.4 *Simulation results*

#### 5.1.4.1 *Determination of* $Ra_{cr,1}$

As a first validation-test for our model, it is important to check whether it can reproduce the result for the first critical Rayleigh number ($Ra_{cr,1}$), which was predicted by the theory in Section 2.4. To this end, four numerical experiments were performed, with the simulation-parameters summarized in Table 2.

| Experiment ID | Rayleigh number (Ra) | $N_y$ | $t_{Sim}$ | $\gamma$ | $Ma_{max}$ |
|---|---|---|---|---|---|
| $\mathbf{A}_{RB\text{-}2D}$ | $1.685 \times 10^3$ | 41 | 450 | $\frac{2\pi}{k_{cr,1}}$ | 0.1 |
| $\mathbf{B}_{RB\text{-}2D}$ | $1.700 \times 10^3$ | | | | |
| $\mathbf{C}_{RB\text{-}2D}$ | $1.715 \times 10^3$ | | | | |
| $\mathbf{D}_{RB\text{-}2D}$ | $1.730 \times 10^3$ | | | | |

Table 2: Simulation parameters for the 2D numerical experiments for determining $Ra_{cr,1}$. The duration of the simulations ($t_{Sim}$) is measured according to the *diffusive* time-scale.

Figure 22 shows the measured dependence of $\|u_y^{(d)}(t^{(d)})\|_{max}$ (semi-log scale) on the dimensionless time, for experiments $\mathbf{A}_{RB\text{-}2D}$-$\mathbf{D}_{RB\text{-}2D}$. After some initial equilibration-period (left side of the plot), the experiments show that $\log\left(\|u_y^{(d)}(t^{(d)})\|_{max}\right)$ starts to vary linearly with time, until the steady-state (horizontal part of each curve) is reached, when the fluid is either stationary[8] or in a (time-independent) rotating regime.

For the determination of $Ra_{cr,1}$, the slopes of this linear portion have to be computed for each experiment. To select the regime with the linear variation, only the

8 For the $Ra \leqslant Ra_{cr,1}$ flows, one could argue that even the part of the curve which appears to be horizontal cannot represent the final state, because the steady-state corresponds to velocities which are *exactly* zero. However, in numerical flow simulations the best result that can be achieved is a velocity close to the floating-point rounding error, to which our model velocities (in the *numerical* system of units) are very close.

| Rayleigh number (Ra) | growth/decay rate (c) |
|---|---|
| 1685 | −0.1499 |
| 1700 | −0.0504 |
| 1715 | 0.0496 |
| 1730 | 0.1492 |

Table 3: Rates of growth/decay of $\log\left(\|u_y^{(d)}(t^{(d)})\|_{\max}\right)$ for each numerical experiment.

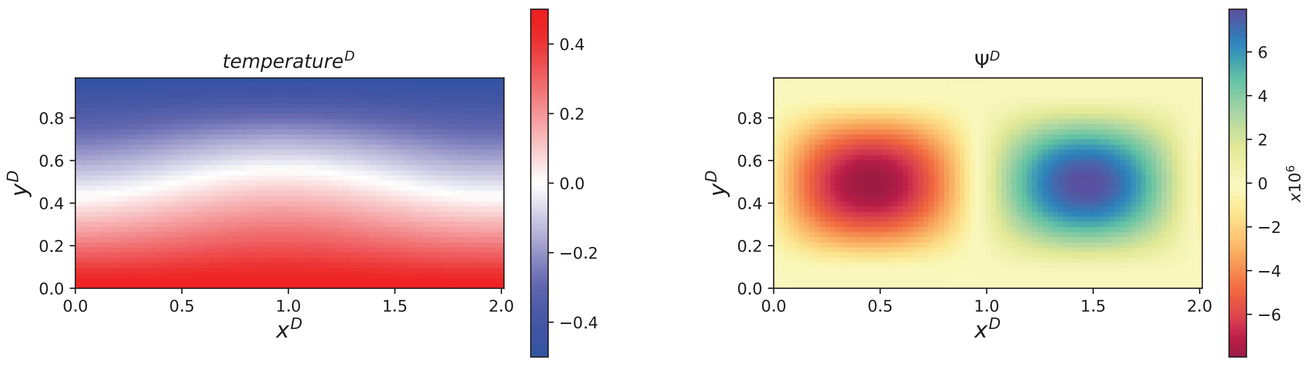


Figure 23: Dimensionless temperature (left) and streamfunction (right) plots for the $Ra = 1730$ simulation (final state, after $t_{Sim}^{(d)} = 450$ diffusive time-scales).

data from the time-interval $t^{(d)} \in [5,\ 25]$ was used. The resulting slope for the least-squares fit in the case of each experiment is given in Table 3. A linear regression based on these values then predicts a growth-factor of zero to be achieved for a Rayleigh number (with 95% *confidence intervals* (CIs)) of:

$$\boxed{Ra_{cr,1}^{numerical} = 1707.5595 \pm 0.0564} \tag{232}$$

Although (strictly-speaking) the result does not agree with the theoretically-predicted value of 1707.762 within the CIs[9], due to the small relative error ($\approx 0.012\%$) we conclude that the model correctly captures the fundamental physics of the problem. For Ra close to (but higher than) $Ra_{cr,1}^{numerical}$, the final state of the fluid will be one of time-independent, roll-like motion[10], as indicated by Figure 23.

### 5.1.4.2 *Behavior near* $Ra_{cr,2}$

With the model passing the first "safety-check", it is interesting to simulate whether this 2D model is also able to correctly reproduce the higher-order instabilities, when

9 A better fit may be obtained with higher-resolution meshes, but that was not the focus of the current study.
10 Of course, being 2D, the present model can only capture the projection of these rolls onto the plane perpendicular to them – this is one of the motivation for the 3D model presented later.

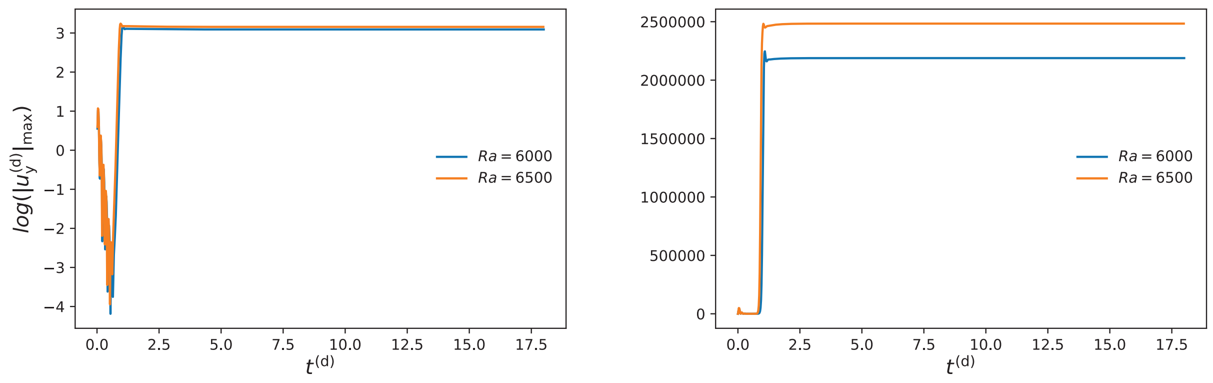


Figure 24: The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{max}$ (left) and of the total kinetic energy (right) for the $Ra_{cr,2}$ 2D simulations.

the motion becomes time-dependent (and, for sufficiently-high Ra, turbulent). To this end, two simulations were performed, with parameters given in Table 4.

| Experiment ID | Rayleigh number (Ra) | $N_y$ | $t_{simulation}$ | $\gamma$ | $Ma_{max}$ |
|---|---|---|---|---|---|
| $\mathbf{E}_{RB\text{-}2D}$ | $6.000 \times 10^3$ | 41 | $18^{(d)}$ | 7 | 0.3 |
| $\mathbf{F}_{RB\text{-}2D}$ | $6.500 \times 10^3$ | | | | |

Table 4: Simulation parameters for the 2D numerical experiments for studying the behavior near $Ra_{cr,2}$. The duration of the simulations ($t_{Sim}$) is still measured according to the *diffusive* time-scale, for consistency. However, note that this duration corresponds to a much larger time in the (more relevant here) *eddy turnover* time-scale, of $t_{Sim}^{\mathcal{E}} \approx 1175$ for simulation $\mathbf{E}_{RB\text{-}2D}$ and of $t_{Sim}^{\mathcal{E}} \approx 1223$ for simulation $\mathbf{F}_{RB\text{-}2D}$.

Notice that a higher aspect-ratio $\gamma = 7$ was used. This was motivated by some initial tests with the 3D model (discussed later, in Section 5.2.4), which showed the setup to be unphysically-stable (with respect to the higher-order instabilities) for an aspect-ratio $\gamma = \frac{2\pi}{k_{cr,1}}$.

Figure 24 indicates that a steady-state was reached after $t_{Sim}^{(d)} \sim 1$. This is confirmed by a spectrogram of the pressure field, sampled at the point $(^{N_x}/_4, {}^{3N_y}/_4)$ (Figure 25).

The important feature to note in the spectrogram is that, after the initial oscillations (due to the onset of the convection) fade-out, no subsequent oscillatory signals are present in the system. This confirms the conclusion that this model does *not* reproduce the secondary instability, which is expected to occur for $Ra \geqslant 6000$.

#### 5.1.4.3 *Behavior for higher*-Ra

Although some numerical experiments (not shown here) indicate chaotic behavior even in the 2D model for high Ra-values (e.g. $Ra = 10^6$), we do not perform a detailed analysis of the model-results in this regime – since this model (unlike the 3D model, discussed later) fails to reproduce the correct physical behavior near the

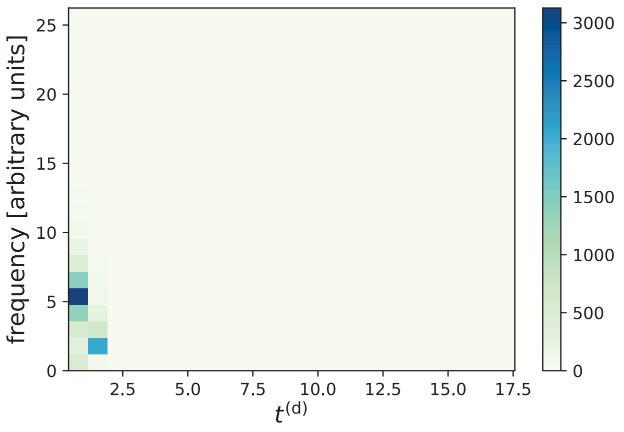


Figure 25: Spectrogram for the 2D simulation, for Ra $=6500$. Spectrograms for values of Ra $=6000$ and of up to Ra $=9000$ show similar behavior towards the end of the time domain.

second instability, we can directly conclude that the Ra $\gg \mathrm{Ra}_{cr,1}$ regime is outside the scope of this model.

## 5.2 RAYLEIGH-BÉNARD (RB) CONVECTION (3D)

As a second application, the setup from the 2D *Rayleigh-Bénard* (RB) problem (Section 5.1) is extended, by also taking into account the third spatial direction. For the simulations, a *multiple distribution function* (MDF) MRT LBM model is formulated, based on the work of D'Humieres et al. (2002) and of Yoshida and Nagaoka (2010). Similar to the 2D model of Wang et al. (2013), the current model uses the Strang-splitting approach for a more accurate representation of the body-force (using the method of Dellar (2013)).

### 5.2.1 *Model setup for the* 3D *case*

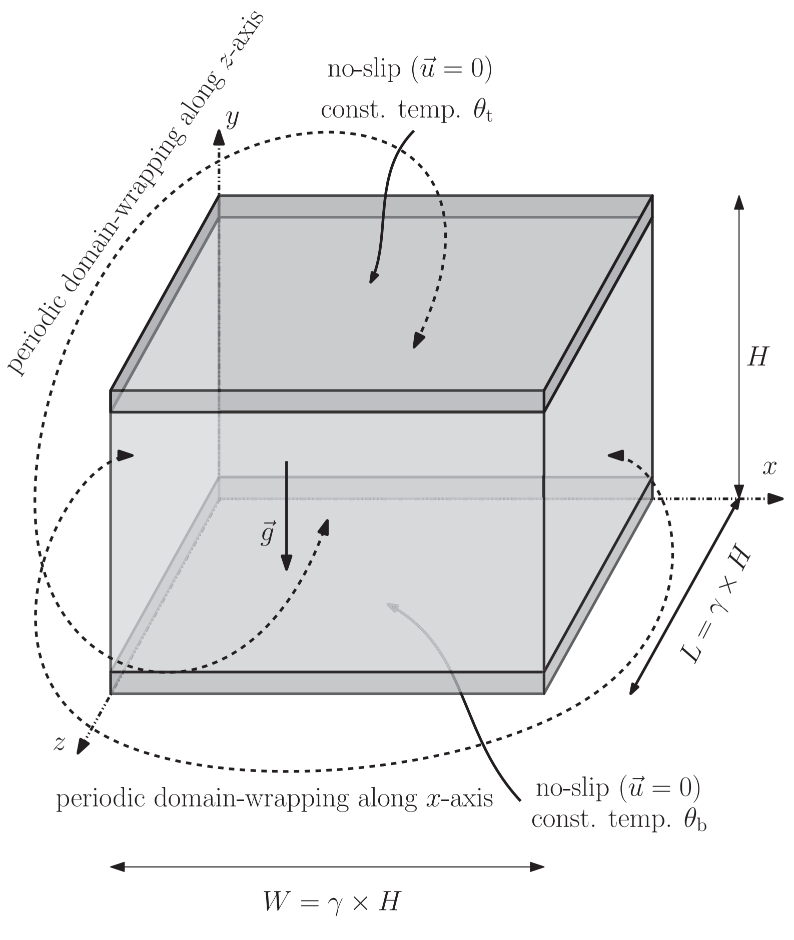


Figure 26: Sketch of geometry for the *Rayleigh-Bénard* (RB) simulations in 3D. For the present simulations, the aspect-ratio was fixed to $\gamma = \frac{2\pi}{k_{cr,1}} \approx 2.0158$.

Because this problem is very similar to the 2D analogue, the discussion from Section 5.1.1 also applies here – the only difference is the new axis $z$ in the horizontal plane, which leads to: (a) one additional momentum equation, and (b) various terms which need to be added to all equations, to account for the $z$ axis. Since the theoretical discussion in Section 2.4 already provided the most general equations, we do not need to discuss them again here.

The geometry is shown in Figure 26. In addition to the setup for the 2D case, now we also consider periodic domain-wrapping along the new coordinate $z$. The horizontal extents of our "convection cell" are the same, $W = L = \gamma H$, with the aspect-ratio $\gamma = 2.0158$. Note that the vertical direction is still along the $y$-axis.

NOTE ON ARRANGEMENT OF AXES In order to re-use the notations and calculations for scaling from Section 5.1, the $y$-axis is kept in the vertical direction; therefore, the new axis ($z$) is in the horizontal plane, which makes the notation somewhat non-standard. However, this does not change anything in the physical principles of the problem.

### 5.2.2 *Discretization of the dimensionless equation*

The discussion on the discretization of the dimensionless equations from Section 5.1.2 also applies to the present model.

### 5.2.3 *Numerical method*

Analogously to Section 5.1.3, the numerical solver used for this setup is also a LBM based on the MRT approach, but specialized for 3D flows (D'Humieres et al., 2002). The hydrodynamic and temperature equations are solved on two separate lattices: D3Q19 (Figure 28) and D3Q7 (Figure 27) respectively. The details of the model are given below, because they are important for setting the physical parameters appropriately. Also, a different type of implementation for body-forces is used relative to the work of D'Humieres et al. (2002), to improve the accuracy of the model.

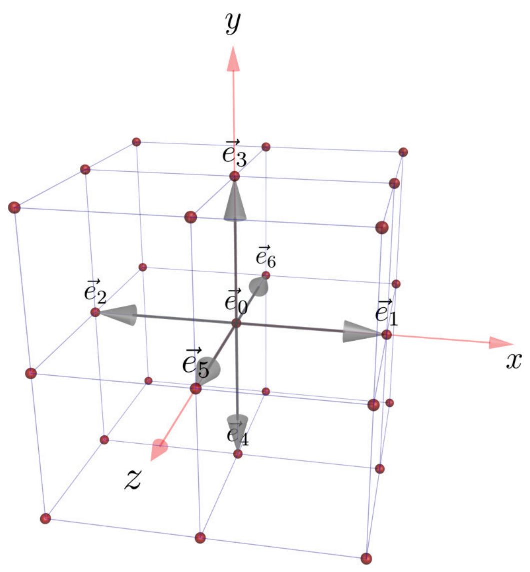


Figure 27: D3Q7 lattice (for modeling advection-diffusion of temperature)

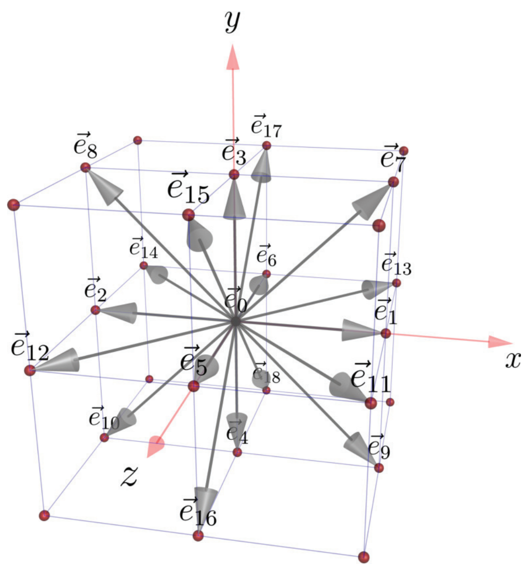


Figure 28: D3Q19 lattice (for modeling isothermal hydrodynamic fields)

#### 5.2.3.1 *Model for the fluid component*

The solver for the fluid component is based on the original work of D'Humieres et al. (2002). The update-rule for the DFs is similar to the 2D case:

$$f_i\left(\mathbf{x}^{(n)} + \mathbf{e}_i\delta_t^{(n)},\ t^{(n)} + \delta_t^{(n)}\right) = f_i\left(\mathbf{x}^{(n)},\ t^{(n)}\right) - \left(M^{-1}S\left[\mathbf{m} - \mathbf{m}^{eq}\right]\right)_i \tag{233}$$

where $i \in \{0, \dots, 18\}$ in this case.

The discretized lattice velocities associated to the DFs are:

$$\tilde{e} \equiv \begin{pmatrix} \mathbf{e}_x \\ \mathbf{e}_y \\ \mathbf{e}_z \end{pmatrix} \equiv c^{(n)} \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \end{pmatrix} \tag{234}$$

with $c^{(n)} \equiv \delta_x^{(n)}/\delta_t^{(n)}$, with $\delta_x^{(n)} = 1$ and $\delta_t^{(n)} = 1$ by definition.

The vector $\mathbf{m}$ represents the moments:

$$\mathbf{m} \equiv \left(\rho,\ e,\ \epsilon,\ j_x,\ q_x,\ j_y,\ q_y,\ j_z,\ q_z,\ 3p_{xx},\ 3\pi_{xx},\ p_{ww},\ \pi_{ww},\ p_{xy},\ p_{yz},\ p_{xz},\ m_x,\ m_y,\ m_z\right), \tag{235}$$

where the most relevant moments are:

- $\rho$ – density,
- $j_x \equiv \rho u_x$, $j_y \equiv \rho u_y$, and $j_z \equiv \rho u_z$ – the components of the momentum.

The mapping from DFs to the moments is defined by the matrix $\tilde{M}$:

$$\mathbf{m} = \tilde{M}\mathbf{f} \tag{236}$$

with the transformation-matrix defined as:

$$\tilde{M} \equiv \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -30 & -11 & -11 & -11 & -11 & -11 & -11 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & -4 & -4 & -4 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & -4 & 4 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\ 0 & -4 & -4 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\ 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & -2 & 2 & 2 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{pmatrix} \tag{237}$$

The vector $\mathbf{m}^{eq}$ for the equilibrium moments has the components: $m_0^{eq} = \rho$, $m_1^{eq} = -11\rho + 19\frac{j_x^2+j_y^2+j_z^2}{\rho_0}$, $m_2^{eq} = 3\rho - \frac{11}{2}\frac{j_x^2+j_y^2+j_z^2}{\rho_0}$, $m_3^{eq} = j_x$, $m_4^{eq} = -\frac{2}{3}j_x$, $m_5^{eq} = j_y$, $m_6^{eq} = -\frac{2}{3}j_y$, $m_7^{eq} = j_z$, $m_8^{eq} = -\frac{2}{3}j_z$, $m_9^{eq} = \frac{2j_x^2-(j_y^2+j_z^2)}{\rho_0}$, $m_{10}^{eq} = -\frac{1}{2}\frac{2j_x^2-(j_y^2+j_z^2)}{\rho_0}$, $m_{11}^{eq} = \frac{j_y^2-j_z^2}{\rho_0}$, $m_{12}^{eq} = -\frac{1}{2}\frac{j_y^2-j_z^2}{\rho_0}$, $m_{13}^{eq} = \frac{j_x j_y}{\rho_0}$, $m_{14}^{eq} = \frac{j_y j_z}{\rho_0}$, $m_{15}^{eq} = \frac{j_x j_z}{\rho_0}$ and $m_{16}^{eq} = m_{17}^{eq} = m_{18}^{eq} = 0$.

As can also be seen from the moments above, the evaluation of the macroscopic variables $\rho^{(n)}$ and $u_j^{(n)}$ is slightly different than for the 2D model:

$$\rho^{(n)} = \sum_{i=0}^{18} f_i \tag{238}$$

$$u_j^{(n)} = \frac{1}{\rho^{(n)}} \sum_{i=0}^{18} e_{i,j} f_i, \quad \text{with} \quad j \in \{x, y\}. \tag{239}$$

However, the same gas-like equation of state for pressure holds (along with the same interpretation, which is not repeated here).

The relaxation of the moments is governed by the diagonal matrix $\tilde{S}$:

$$\tilde{S} = \text{diag}\left(s_0,\ s_1,\ s_2,\ s_3,\ s_4,\ s_5,\ s_6,\ s_7,\ s_8,\ s_9,\ s_{10},\ s_{11},\ s_{12},\ s_{13},\ s_{14},\ s_{15}\right) \tag{240}$$

The values of these parameters for optimal stability are given by:

$$s_0 = s_3 = s_5 = s_7 = 1, \tag{241}$$

$$s_1 = 1.19, \tag{242}$$

$$s_2 = s_8 = s_{10} = s_{12} = 1.4, \tag{243}$$

$$s_4 = s_6 = s_8 = 1.2, \tag{244}$$

$$s_{16} = s_{17} = s_{18} = 1.98, \quad \text{and,} \tag{245}$$

$$s_9 = s_{11} = s_{13} = s_{14} = s_{15} = \frac{1}{3\nu^{(n)} + {}^1\!/_2}. \tag{246}$$

By inverting the last equation, we obtain the viscosity of the model as a value of the $s_9$ relaxation-rate:

$$\nu^{(n)} = \frac{2 - s_9}{6 s_9} \tag{247}$$

Obviously, to prevent unpysical negative viscosities, it is necessary to have:

$$s_9 = s_{11} = s_{13} = s_{14} = s_{15} \in [0, 2) \tag{248}$$

BODY FORCE, BCs, ICs These are implemented in the same way as for the 2D model (the procedures described there are easily-generalizeable to the 3D case).

#### 5.2.3.2 *Model for the temperature component*

For the temperature DFs, the model of Yoshida and Nagaoka (2010) is used, which relies on the D3Q7 lattice (Figure 27). The equation for updating the DFs in the bulk of the domain is:

$$g_i\left(\mathbf{x}^{(n)} + \mathbf{e}_i\delta_t^{(n)}, t^{(n)} + \delta_t^{(n)}\right) = g_i\left(\mathbf{x}^{(n)},\ t^{(n)}\right) - \left(N^{-1}Q\left[\mathbf{n} - \mathbf{n}^{eq}\right]\right)_i, \tag{249}$$

with $i \in \{0, \ldots, 7\}$.

The discrete lattice velocities are a subset of those from the D3Q19 lattice used for the fluid solver:

$$\tilde{e} \equiv \begin{pmatrix} \mathbf{e}_x \\ \mathbf{e}_y \\ \mathbf{e}_z \end{pmatrix} \equiv c^{(n)} \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}, \tag{250}$$

where $c^{(n)}$ has the same significance as for the fluid model.

The operator for mapping DFs $\mathbf{g}$ to the corresponding moments $\mathbf{n}$ becomes:

$$\mathbf{n} = \tilde{N}\mathbf{g} \tag{251}$$

where:

$$\tilde{N} \equiv \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ -6 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & -2 & -2 \end{pmatrix} \tag{252}$$

The equilibrium moments are:

$$\mathbf{n}^{eq} = \left(T^{(n)},\ u_x^{(n)}T^{(n)},\ u_y^{(n)}T^{(n)},\ u_z^{(n)}T^{(n)},\ aT^{(n)},\ 0,\ 0\right)^{\dagger}, \tag{253}$$

where the constant $a = -3/4$.

The macroscopic variable which is of most interest for this model is the temperature, for which the explicit expression reads:

$$T^{(n)} = \sum_{i=0}^{7} g_i \tag{254}$$

While the model of Yoshida and Nagaoka (2010) also applies to *anisotropic* convection-diffusion problems, it is used in an isotropic regime here, s.t. the relaxation matrix is again diagonal:

$$\tilde{Q} = \mathrm{diag}(1,\ s_a,\ s_a,\ s_a,\ s_{t4},\ s_{t5},\ s_{t6}), \tag{255}$$

where the optimal values for the coefficients are:

$$s_a = \frac{2}{8\kappa^{(n)} + 1} \tag{256}$$

$$s_{t4} = s_{t5} = s_{t6} = 0 \quad . \tag{257}$$

The thermal diffusivity in the model is obtained finally by inverting the first equation above:

$$\kappa^{(n)} = \frac{1}{4}\left(\frac{1}{s_a} - \frac{1}{2}\right), \tag{258}$$

from which it is clear that we must have $s_a \in (0, 2]$, because the diffusivity should be positive.

BCs In general case, for when the walls of the domain are not planar or not situated half-way between the last fluid and first solid node of the lattice, it is necessary to use the approach of Li et al. (2013) to avoid destroying the 2nd-order spatial accuracy of the model through the BCs. However, for the simpler geometry used in the current study the original Dirichlet (1850) BC of Yoshida and Nagaoka (2010) for the temperature at the horizontal walls is still 2nd-order accurate. This

$$g_{\bar{i}}^{\text{pre-collision}}\left(\mathbf{x}_f^{(n)}, t^{(n)} + \delta_t\right) = -g_i^{\text{post-collision}}\left(\mathbf{x}_f^{(n)}, t^{(n)}\right) + \epsilon_D T_{\text{wall}}^{(n)} \tag{259}$$

where the overline denotes the reversed direction vector.

ICs The initial DFs are computed from the initial (linearly-varrying) temperature field, using the same method as for the fluid model, by inverting the moment-evaluation equation.

### 5.2.3.3 *Trigger for symmetry-breaking*

For the same reasons as discussed in Section 5.1.3.3, it is necessary to add some perturbations, to break the symmetry of the flow (so that it behaves similar to the real-world scenario, where such perturbations are always present). As for the 2D simulations, we achieve this by applying some random perturbations (drawn from an uniform distribution for the interval $[0, 10^{-8}]$) in the entire fluid-domain. In addition, to make the setup as close as possible to the 2D case (which makes comparisons more meaningful), we also impose a larger perturbation ($10^{-5}$), along a line with $\left(x^{(n)}, y^{(n)}, z^{(n)}\right) \in \left(\frac{N_x}{2} + 1, w, \overline{1, N_z}\right)$.

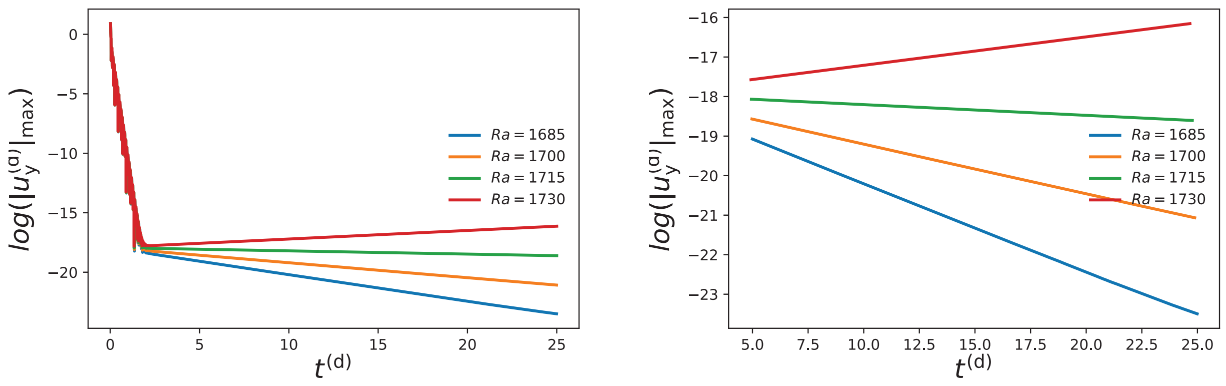


Figure 29: The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{max}$ for different values of the Rayleigh number (left) and zoomed-in plot of the same data, for the linear regime (right). All simulations were performed on a $83 \times 83 \times 41$ grid, with $Ma = 0.1$.

### 5.2.4 *Simulation results*

#### 5.2.4.1 *Determination of* $Ra_{cr,1}$

Following the same steps as for the 2D model (Section 5.1.4.1), we first check whether the 3D model is able to reproduce the theoretical result for the first critical Rayleigh number ($Ra_{cr,1}$), given in Section 2.4. To check this, we performed four simulations, with the parameters summarized in Table 5.

| Experiment ID | Rayleigh number (Ra) | $N_y$ | $t_{Sim}$ | $\gamma$ | $Ma_{max}$ |
|---|---|---|---|---|---|
| $\mathbf{A}_{RB\text{-}3D}$ | $1.685 \times 10^3$ | 41 | 25 | $\frac{2\pi}{k_{cr,1}}$ | 0.1 |
| $\mathbf{B}_{RB\text{-}3D}$ | $1.700 \times 10^3$ | | | | |
| $\mathbf{C}_{RB\text{-}3D}$ | $1.715 \times 10^3$ | | | | |
| $\mathbf{D}_{RB\text{-}3D}$ | $1.730 \times 10^3$ | | | | |

Table 5: Simulation parameters for the 3D numerical experiments for determining $Ra_{cr,1}$. The duration of the simulations ($t_{Sim}$) is measured according to the *diffusive* time-scale.

Figure 29 shows the measured dependence of $\|u_y^{(d)}(t^{(d)})\|_{max}$ (semi-log scale) on the dimensionless time, for experiments $\mathbf{A}_{RB\text{-}3D}$-$\mathbf{D}_{RB\text{-}3D}$. Similar to the 2D case ,there is an initial equilibration-phase (left side of the plot), after which $\log\left(\|u_y^{(d)}(t^{(d)})\|_{max}\right)$ starts to vary linearly with time. To determine $Ra_{cr,1}$ for the model, the slopes of this linear portion

For the determination of $Ra_{cr,1}$, the slopes of this linear portion have to be computed for each experiment, based on the selection of the data from the time-interval $t^{(d)} \in [5,\ 25]$ (see right side of Figure 29).

The resulting slope for the least-squares fit in the case of each experiment is given in Table 6. The least-squares procedure is applied on these results again, yielding the final value of:

$$\boxed{Ra_{cr}^{numerical} = 1719.0841 \pm 0.0898} \tag{260}$$

| Rayleigh number (Ra) | growth/decay rate (c) |
|---|---|
| 1685 | −0.222482 |
| 1700 | −0.125593 |
| 1715 | −0.027108 |
| 1730 | 0.071921 |

Table 6: Rates of growth/decay of $\log\left(\|u_y^{(d)}(t^{(d)})\|_{\max}\right)$ for each numerical experiment.

This result is less accurate compared to the 2D model, but the relative error compared to the theoretical value is still small ($\approx 0.66\%$). The (highly-enlarged) velocity vectors and temperature distribution at the end of the simulation are shown in Figure 30.

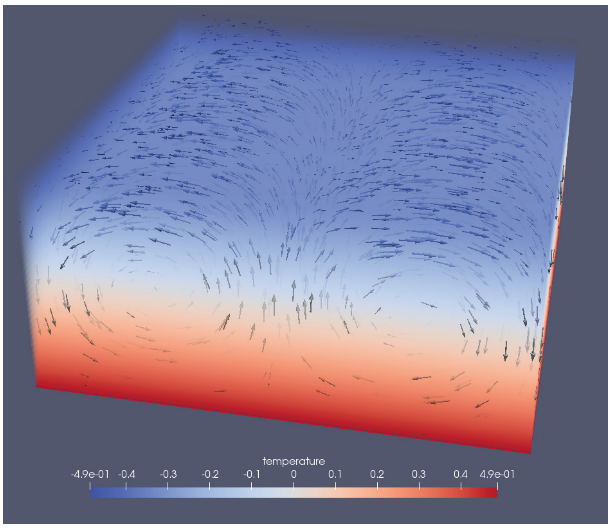


Figure 30: Velocity vectors and temperature distribution after $t^{(d)} = 25$, for the $Ra = 1730$ experiment with the 3D model.

It can be readily observed that two counter-rotating rolls appear, which disturb the iso-surfaces of temperature.

### 5.2.4.2 *Behavior near* $\mathrm{Ra}_{cr,2}$

With the model passing the first "safety-check", it is interesting to simulate whether this 3D model is also able to reproduce correctly the higher-order instabilities, when the motion becomes time-dependent (and, for sufficiently-high Ra, turbulent).

**Un-physical effect of small aspect-ratios (when periodic BCs are used)**

Initial tests with this 3D model showed un-physically stable behavior of the flow for Ra close to $Ra_{cr,2} \approx 6000$. For example, the final state after $t^{(d)} = 25$ for a $Ra = 6500$ simulation, with the same aspect-ratio $\gamma = \frac{2\pi}{k_{cr,1}}$ used for studying the first instability is shown in Figure 31.

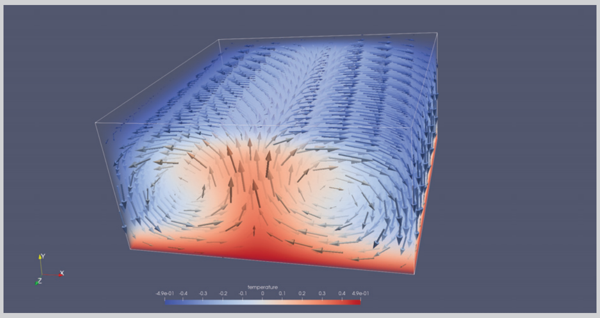


Figure 31: Velocity vectors and temperature distribution after $t^{(d)} = 25$, for a $Ra = 6500$ experiment with the 3D model, and low aspect-ratio (*un-physical* behavior).

This failure to detect the oscillatory behavior that is expected for $Ra \gtrsim Ra_{cr,2}$ was initially thought to be caused by errors in the implementation of the model, or due to insufficient spatial resolution. Interestingly, however, this was not the case – instead, the "artificial stability" was due to the too small aspect-ratio which, when periodic BCs are used, inhibits some of the higher-order instabilities. Due to this reason, all simulation results which follow are using an aspect-ratio of $\gamma = 7$.

For studying the second instability with the 3D model, two simulations were performed, with parameters given in Table 7.

| Experiment ID | Rayleigh number (Ra) | $N_y$ | $t_{Sim}$ | $\gamma$ | $Ma_{max}$ |
|---|---|---|---|---|---|
| $\mathbf{E}_{RB\text{-}3D}$ | $6.000 \times 10^3$ | 41 | 18 | 7 | 0.3 |
| $\mathbf{F}_{RB\text{-}3D}$ | $6.500 \times 10^3$ | | | | |

Table 7: Simulation parameters for the 3D numerical experiments for studying the behavior near $Ra_{cr,2}$. The duration of the simulations ($t_{Sim}$) is still measured according to the *diffusive* time-scale, for consistency. However, note that this duration corresponds to a much larger time in the (more relevant here) *eddy turnover* time-scale, of $t^{\mathcal{E}}_{Sim} \approx 1175$ for simulation $\mathbf{E}_{RB\text{-}3D}$ and of $t^{\mathcal{E}}_{Sim} \approx 1223$ for simulation $\mathbf{F}_{RB\text{-}3D}$.

The plots for the maximum vertical velocity and the total kinetic energy in Figure 32 already indicate much more complex evolutions compared to the 2D case. To compare the two simulations, we will present snapshots of the fields at different moments, side-by-side.

At $t^{(d)} = 1$ (Figure 33), both simulations look the same, with four plumes (eight rolls) being formed, and with the velocity-field being slightly larger for the $Ra = 6500$ case.

At $t^{(d)} = 2$, an anomaly appears in the middle rolls, which starts to affect the entire flow-field (disturbing the other rolls).

By $t^{(d)} = 2.5$ the middle cylinders disappear in the middle of the domain for the $Ra = 6500$ simulation, a phenomenon which also takes place in the $Ra = 6000$ simulation (at $t^{(d)} \approx 3.1$, not shown here).

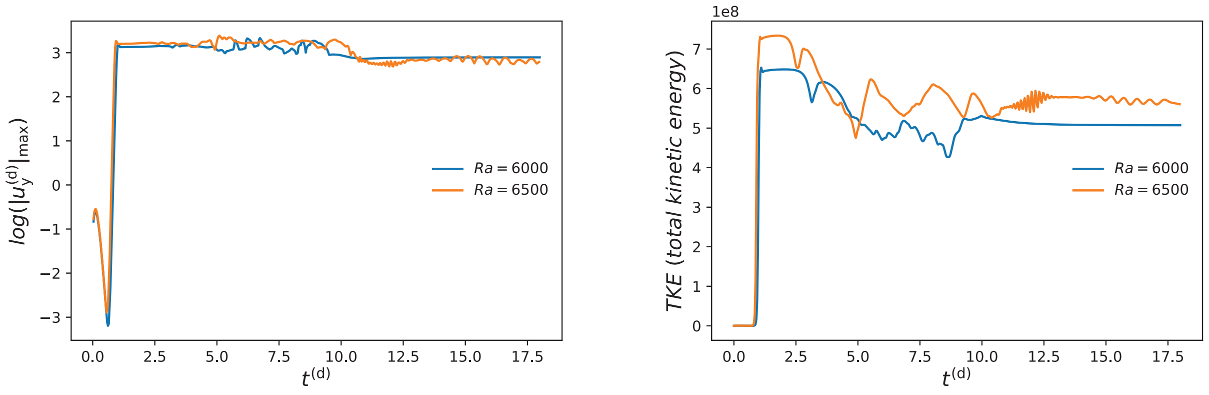


Figure 32: The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{\max}$ (left) and of the total kinetic energy (right) for the $Ra_{cr,2}$ 3D simulations.

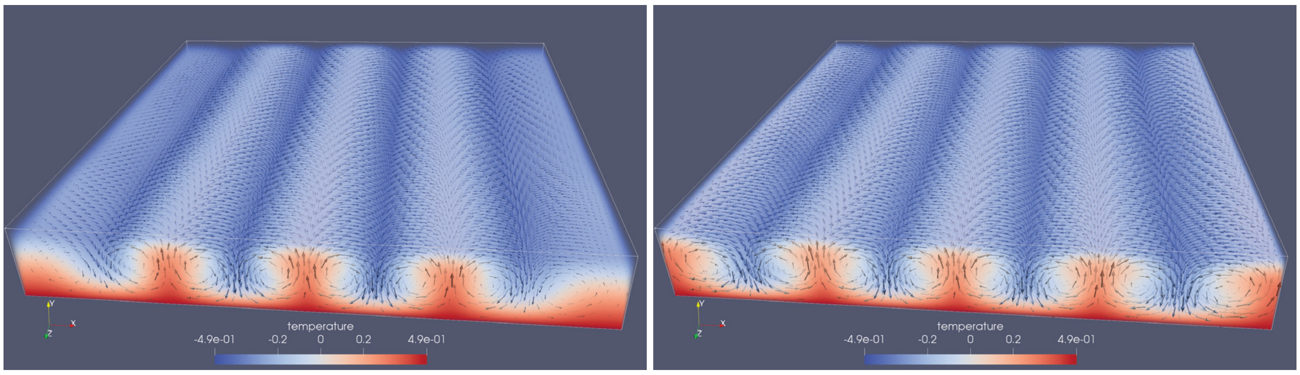


Figure 33: Flow-fields for the $Ra = 6000$ simulation (left) and for the $Ra = 6500$ simulation (right), at $t^{(d)} = 1$.

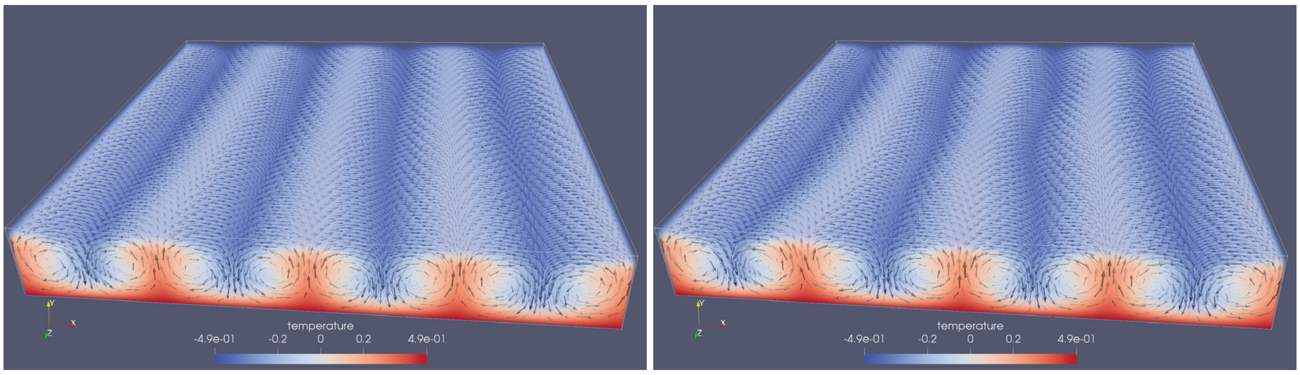


Figure 34: Flow-fields for the $Ra = 6000$ simulation (left) and for the $Ra = 6500$ simulation (right), at $t^{(d)} = 2$.

During the time-interval $t^{(d)} \in [3, 16]$, a period of intense flow-restructuring takes place in both simulations, which leads to a different outcome in each case.

By $t^{(d)} = 16$, the $Ra = 6000$ simulation reaches a steady-state (cellular convection), while the $Ra = 6500$ simulation enters a time-dependent regime (oscillatory convection), as expected for $Ra_{cr,2}$.

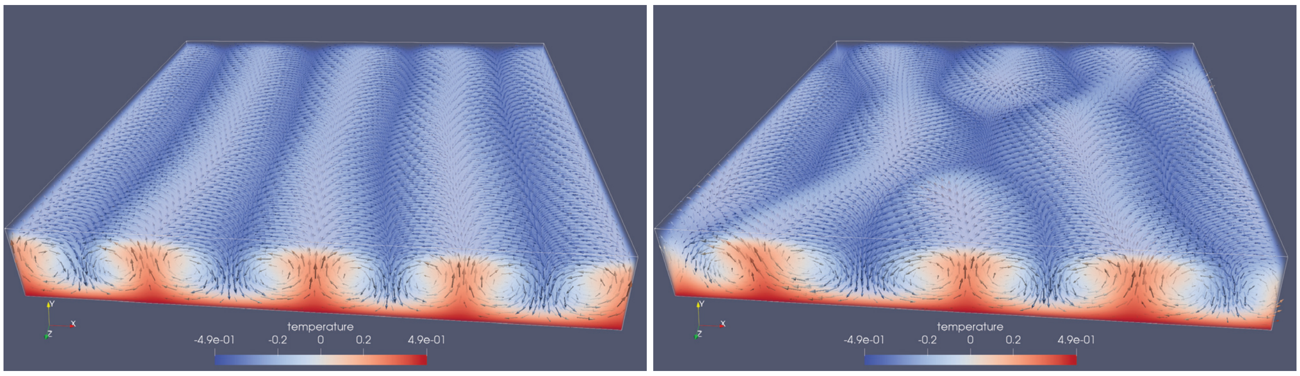


Figure 35: Flow-fields for the Ra $= 6000$ simulation (left) and for the Ra $= 6500$ simulation (right), at $t^{(d)} = 2.5$.

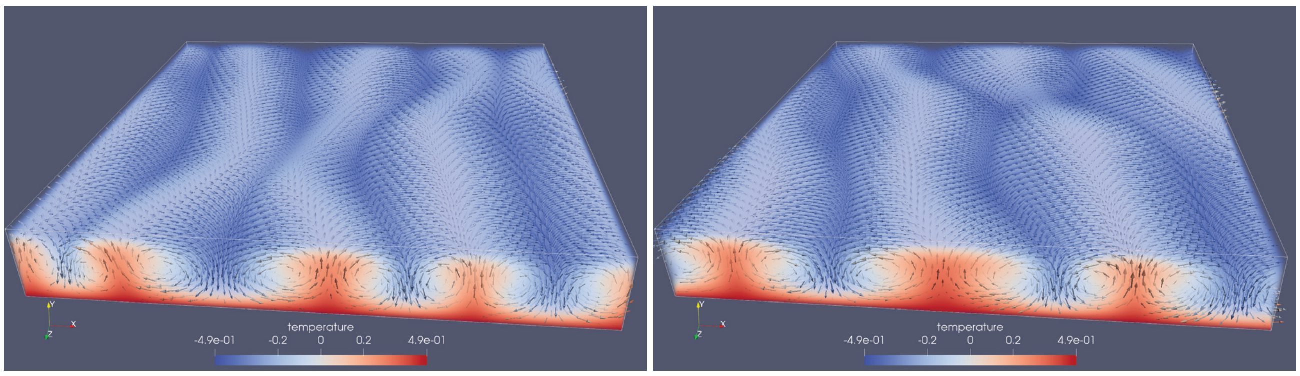


Figure 36: Flow-fields for the Ra $= 6000$ simulation (left) and for the Ra $= 6500$ simulation (right), at $t^{(d)} = 3$.

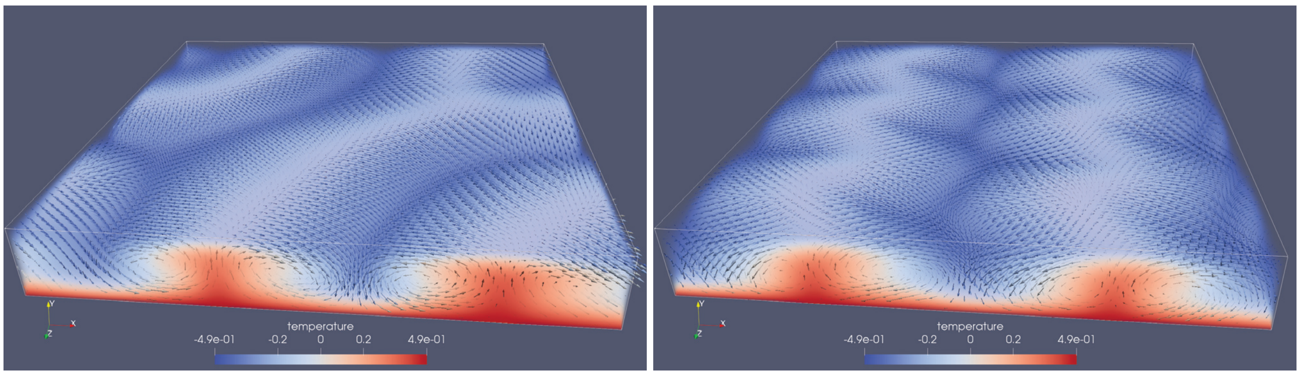


Figure 37: Flow-fields for the Ra $= 6000$ simulation (left) and for the Ra $= 6500$ simulation (right), at $t^{(d)} = 16$.

As further evidence for the distinct final states, the corresponding spectrograms are shown in Figure 39. It can be observed that, after the very energetic phase of flow-restructuring, all oscillations are dampened in the Ra $= 6000$ case, while for

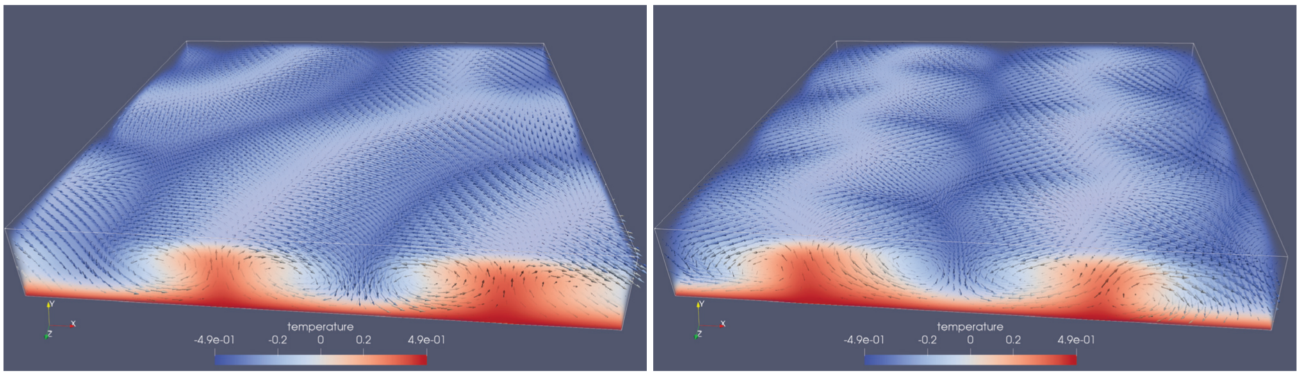


Figure 38: Flow-fields for the $Ra = 6000$ simulation (left) and for the $Ra = 6500$ simulation (right), at $t^{(d)} = 18$.

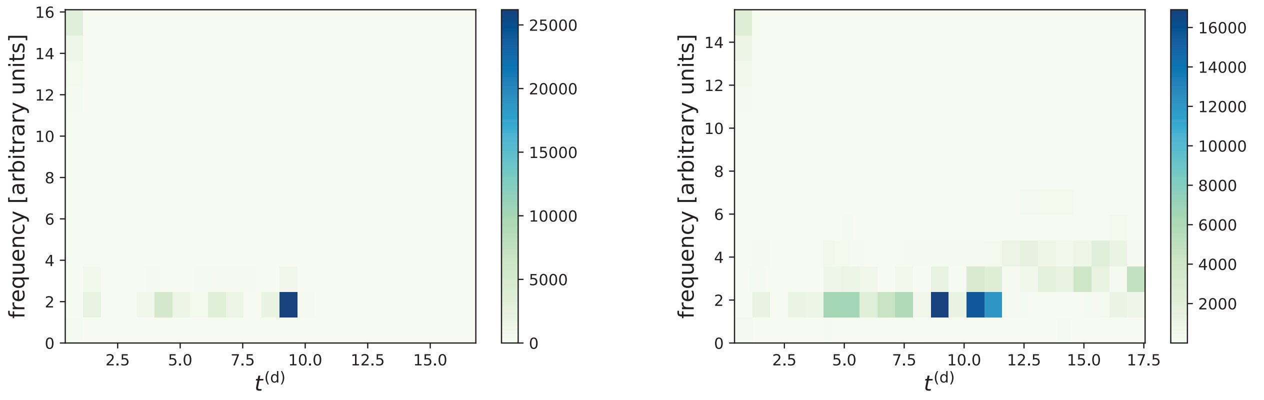


Figure 39: Spectrograms of the 3D simulations for testing the behavior in the vicinity of $Ra_{cr,2}$: (left) $Ra = 6000$ and (right) $Ra = 6500$. Both figures correspond to the pressure time-series sampled at the point $(N_x/4, 3N_y/4, N_z/4)$.

$Ra = 6500$ there is still a strong oscillation (with $f \in [3,4]$[11]) which survives – this corresponds to the wave traveling along the rolls.

The dominant oscillation for the $Ra = 6500$ simulation corresponds to waves traveling along the rolls, in the negative Z direction. Several snapshots of this wave-phenomenon are given in Figure 40.

It can be concluded that the 3D MRT LBM model above for simulating convective flows under the Boussinesq approximation also captures the correct physical behavior near the secondary instability (with an accuracy of 8% or better[12]).

As a final note about these two simulations, it is interesting to say a few words about the stable pattern which emerges at the end of the $Ra = 6000$ simulation. Because the domain is periodic in the horizontal directions, it can be considered that

11 The resolution of the spectrograms is, unfortunately, rather coarse. For a better result, more frequent output at the sampling point would be required. However, because the simulations were performed on a *graphics processing unit* (GPU), significant additional effort would be required for implementing this more frequent sampling (without affecting the performance of the simulations). Therefore, this is left for future work.

12 A more extensive study, with several additional simulations (and with higher spatial resolution) would be required to precisely determine $Ra_{cr,2}$ for the model.

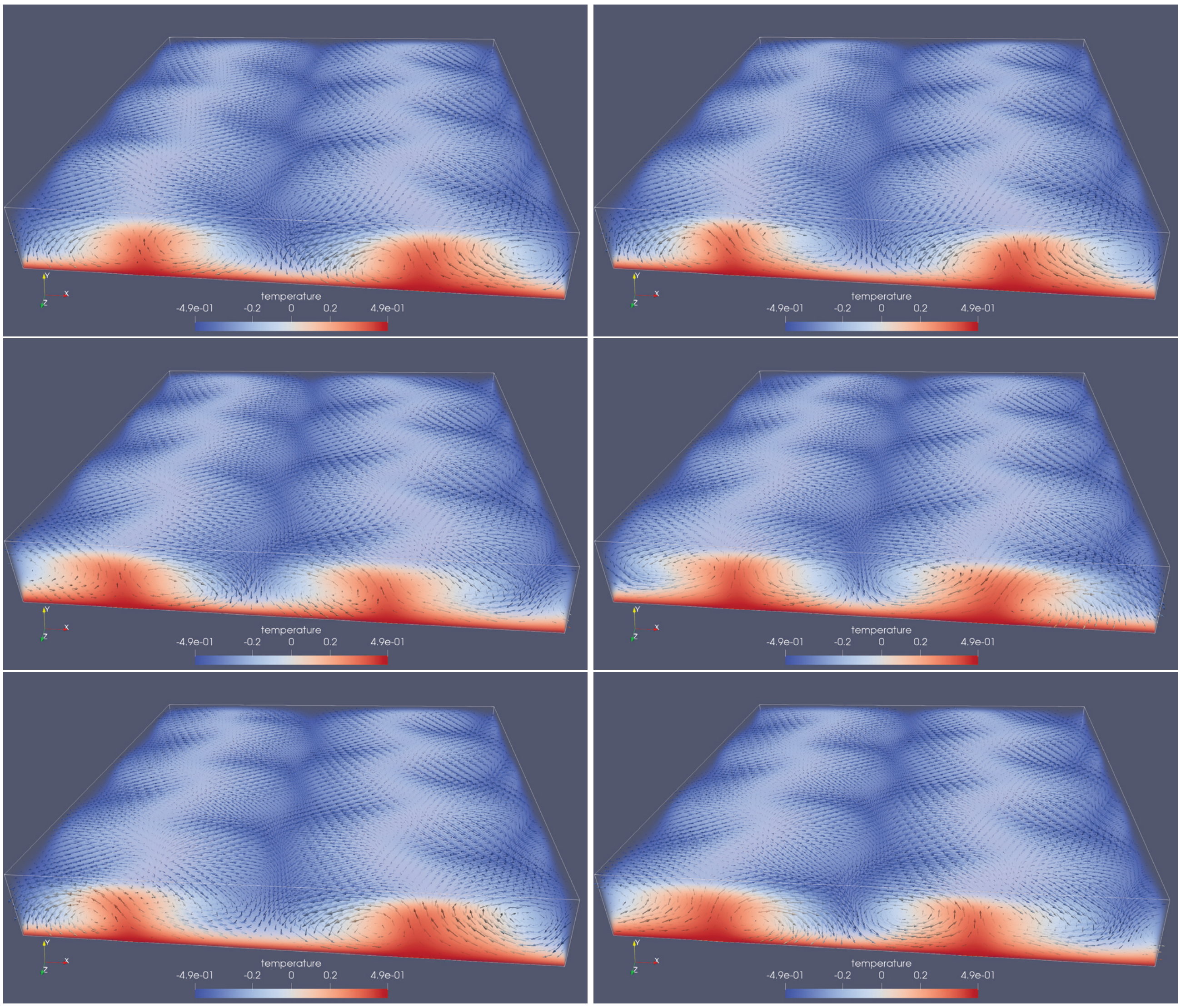


Figure 40: Snapshots of the traveling-wave oscillating flow-field for the $\mathrm{Ra} = 6500$ simulation. From left-to-right, top-to-bottom: $t^{(d)} = 17.00$, $t^{(d)} = 17.38$, $t^{(d)} = 17.50$, $t^{(d)} = 17.63$, $t^{(d)} = 17.75$, and $t^{(d)} = 17.88$.

our simulations provide data for a "unit cell", which is repeated *ad infinitum* in a hypothetical unbounded domain. A portion of this "tiling" (top view) is shown in Figure 41. Similar patterns were also found by previous investigators (see e.g. Getling and Brausch (2003) and Lappa (2010)). It appears that the initial configuration consisting of parallel rolls is *unstable* at these higher Rayleigh numbers, which causes the system to "search" for other local extrema, which are more efficient at transporting the heat.

### 5.2.4.3 *Behavior for higher*-Ra

With a final simulation, we investigate the behavior of the 3D model in the high-Ra regime. Specifically, we consider the $\mathrm{Ra} = 10^6$ case, which is simulated on the same $287 \times 41 \times 287$ grid as for the near-$\mathrm{Ra}_{cr,2}$ simulations.

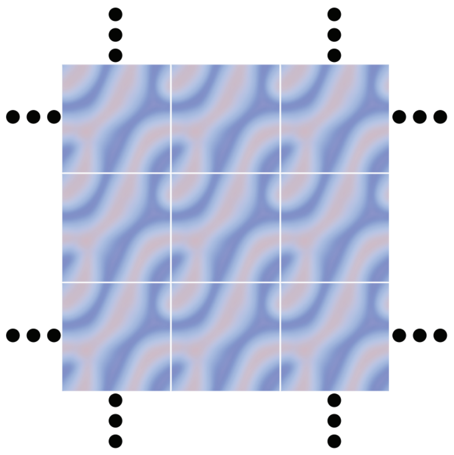


Figure 41: Portion of the infinite space-filling tiling that our model simulates for Ra = 6000 (top view).

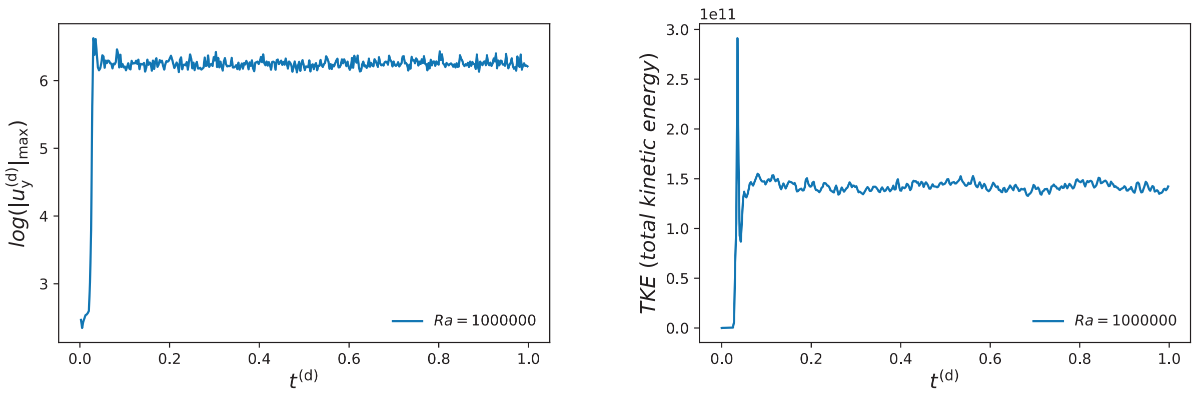


Figure 42: The dependence of the evolution of $\|u_y^{(d)}(t^{(d)})\|_{max}$ (left) and of the total kinetic energy (right) for the $Ra = 10^6$ 3D simulation.

The evolution of the maximum vertical velocity, and of the total kinetic energy is shown in Figure 42. These plots indicate a chaotic regime of turbulent convection from an early stage, which can also be seen from the snapshots in Figure 43. The total simulation-time was $t_{Sim}^{\mathcal{D}} = 1$ in terms of the *diffusive* time-scale. However, in terms of the *convective* (*eddy-turnover*) time-scale (which is more relevant for the high-Ra regimes) we have $t_{Sim}^{\mathcal{E}} \approx 843$, which is more-than-adequate for the flow to become fully-developed.

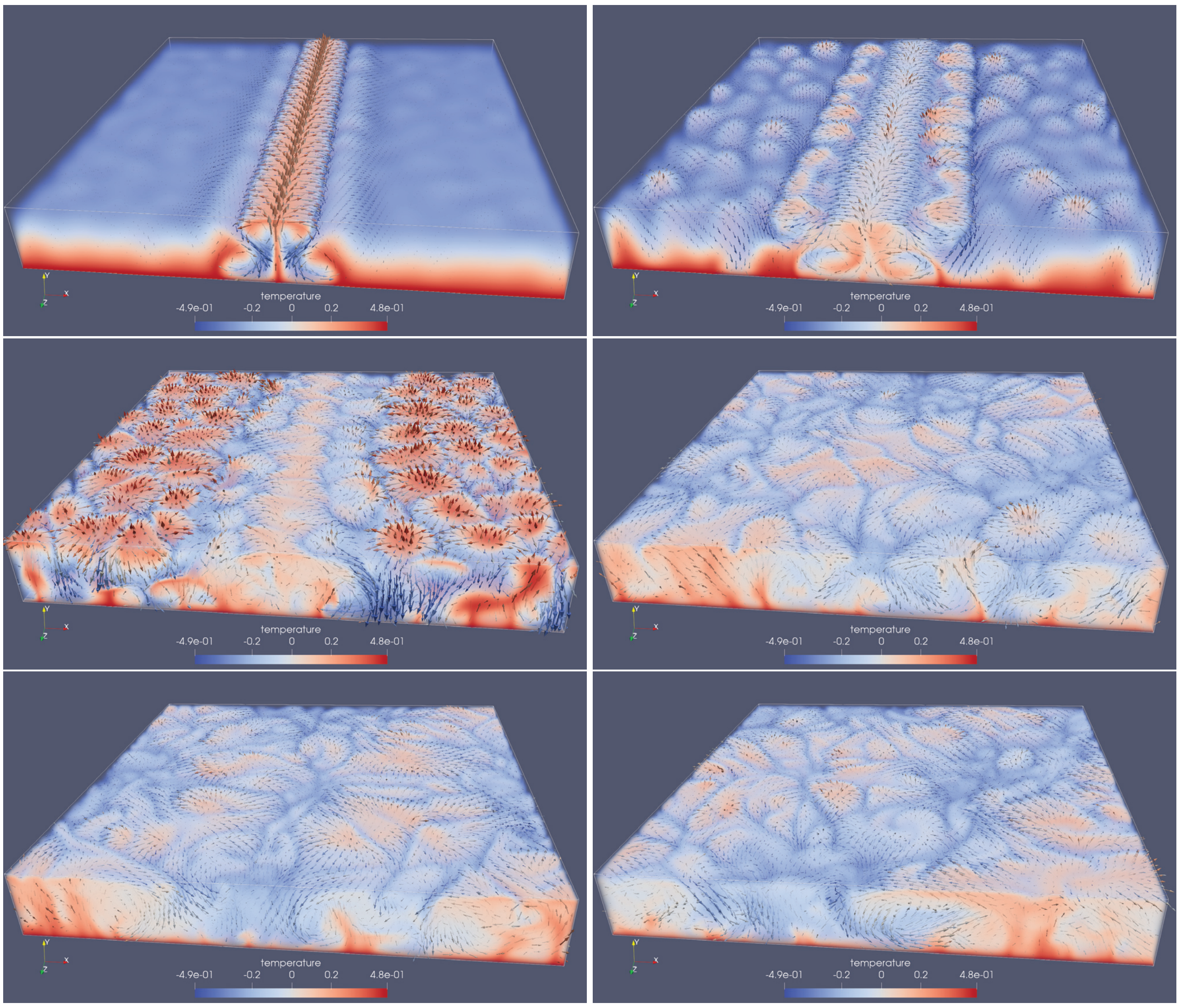


Figure 43: Snapshots showing the evolution of the flow-field towards (and into) the turbulent-convection regime for the $Ra = 10^6$ simulation. From left-to-right, top-to-bottom:$t^{(d)} = 0.0302$, $t^{(d)} = 0.0327$, $t^{(d)} = 0.0352$, $t^{(d)} = 0.2513$, $t^{(d)} = 0.5025$, and $t^{(d)} = 1.0000$.

## 5.3 WIND-DRIVEN OCEAN CIRCULATION (2D)

As a last application of GeLB (which is more directly relevant to climate simulation), we integrate numerically the equations of the barotropic (i.e. vertically-integrated) ocean model, where the bottom-topography is assumed to be flat. As discussed in Section 2.6.2.2, this model is sufficient for studying the most important features of the wind-driven motion of the surface ocean currents, which is a classical problem in oceanography. In addition to the analytic solutions mentioned in Section 2.6.2.2, this problem has been solved numerically by various authors, using *finite differences* (FD) (Bryan, 1963; Veronis, 1966), and even using LBM (Salmon, 1999; Wolf-Gladrow, 2000). Compared to the previous LBM studies, which used the BGK approach, this section uses the newer LBM-MRT method to solve the same problem, with a more complex geometry (experiment 2).

Our new numerical model is first *validated* for the *non-linear*, *no-slip* case (following the work of Wolf-Gladrow (2000), who discussed an BGK-LBM model for the same setup). Then, we apply the same numerical algorithm to a more *realistic, global land-mask*, which is directly derived from the ETOPO dataset (Amante and Eakins, 2008). Although our numerical model is limited by the 2D assumption (which automatically prohibits accurate simulations of the real surface-currents in the oceans), this second oceanographic simulation is valuable as a showcase for the ability of LBM algorithms (which is further enhanced by GeLB) to easily incorporate the complex features of the ocean land-mask. To the best of our knowledge, LBM algorithms have not yet been applied for this particular geometry. Also important is the implementation of the Coriolis force, which normally presents some difficulties (due to the fact that it depends on the velocity-field).

Because we discuss two experiments (one with a *square geometry* and another one with a *realistic geometry*), we separate the discussion into two subsections, addressing each experiment.

### 5.3.1 *Idealized (square) geometry simulation*

#### 5.3.1.1 *Model setup*

The geometry for this experiment consists of a square domain with side $L = 4000$ km, as sketched in Figure 44. The domain corresponds to a portion of the Earth's surface (centered around $\phi_0 = 30°$N). Because we use a 2D model here, any type of vertical motion is ignored. While clearly unrealistic for the real ocean, this is a useful rough approximation for studying the wind-driven circulation at the surface (which is our purpose here). Unlike the RB setup, there is no periodic domain-wrapping – the velocity-field is assumed to satisfy the *no-slip* BC on all sides of the domain.

In ESS it is also customary to provide the physical units used. To facilitate comparison of results with Wolf-Gladrow (2000), we choose:

- $L = 4000$ km: side-length of the ocean domain

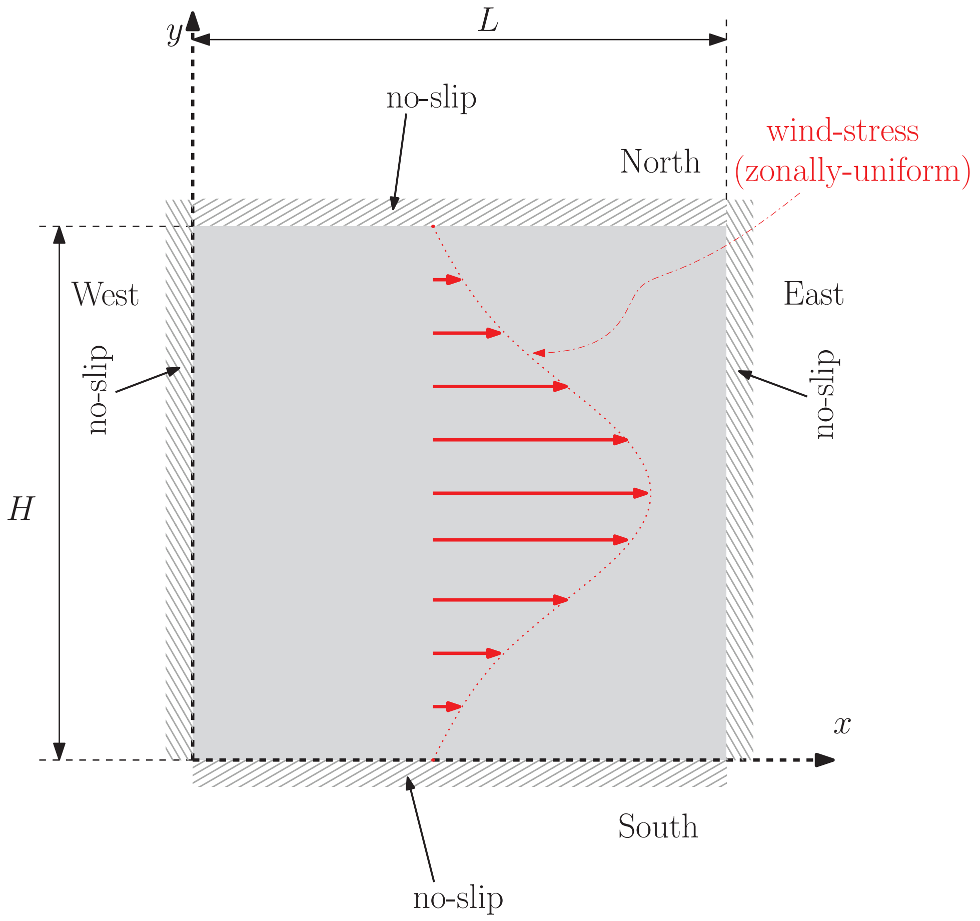


Figure 44: Sketch of geometry for the *wind-driven ocean circulation* (WDOC) simulations in 2D.

- $R = 6371$ km: mean radius of the Earth
- $\Omega = 7.29 \times 10^{-5}$ s$^{-1}$: angular velocity of the Earth
- $A = 2.03 \times 10^{4}$ m$^{2}$s$^{-1}$: horizontal eddy viscosity coefficient
- $\phi_0 = 30°$ N: reference latitude for the $\beta$-plane
- $T_0 = 8.0 \times 10^{-7}$ m/s$^{2}$: wind-stress amplitude
- $U_0 = 0.4$ m/s: typical horizontal ocean in the wind-driven layer of the ocean
- $t_{\text{simulation}} = 15$ *weeks*: total simulated time

### 5.3.1.2 *Discretization of the dimensionless equation*

Similar to how we proceeded for the RB setup, we need to transform the equations from the *dimensionless* units to the *model* units. These transformations (eqs. (185) to (188)) are the same as for the 2D RB setup with the simplification that we no longer need to consider the temperature (because the fluid is assumed to be athermal).

However, we need to select new values for $N_y$ and $N_t$, to reflect the different physics of the model:

- For selecting $N_y$, we need to consider what is the smallest spatial feature that our simulation is supposed to capture. In our case, we want to resolve the *Munk layer*, which has a width $W_M \equiv \left(\frac{A}{\beta}\right)^{1/3} \approx 80$ km for the physical parameters listed above. We want $W_M^{(n)}$ to be at least twice as large as $\delta_x^{(n)}$, which leads to $N_x = N_y = 100$.

- The selection of $N_t$ is based on the CFL criterion for the stability of the solution, which essentially dictates that the time-step should be smaller than the time it takes for the fastest wave to travel a distance equal to $\delta_x$. In our present case, the fastest waves are the *Rossby waves*, the phase-velocity of which can be shown to have the magnitude:

$$\left|v_{\text{ph, Rossby}}\right| = \frac{\beta L^2}{4\pi^2} \approx 8 \text{ m/s} \tag{261}$$

  Simple arithmetic shows that this leads to a maximum time-step of 5000 s. However, to improve the accuracy of the simulations, we take $\delta_t = 400$ s, i.e. $N_t = 2.5 \times 10^4$ (where the characteristic time was calculated as $\frac{L}{U_0}$).

Plugging the scaling equations into the dimensionless governing eqs. (177) to (179), the following equations are obtained for the model parameters:

$$\begin{aligned}
\nu^{(n)} &= \frac{N_y^2}{\text{Re}\, N_t} && (262)\\
a_1^{(n)} &= +\frac{1}{\text{Ro}\,\Gamma_{\text{Ro}} N_t}\left[1 + \frac{\Gamma_{\text{Ro}}}{N_y}\left(x_2^{(n)} - \frac{1}{2}\right)\right] u_2^{(n)} - \frac{N_y}{\text{Fr}^2 N_t^2}\sin^2\left[\frac{\pi}{N_y}\left(x_2^{(n)} - \frac{1}{2}\right)\right] && (263)\\
a_2^{(n)} &= -\frac{1}{\text{Ro}\,\Gamma_{\text{Ro}} N_t}\left[1 + \frac{\Gamma_{\text{Ro}}}{N_y}\left(x_2^{(n)} - \frac{1}{2}\right)\right] u_1^{(n)} \quad , && (264)
\end{aligned}$$

where $a_i^{(n)}$ and $a_2^{(n)}$ are the acceleration-terms due to the body-forces (Coriolis force and wind-stress), along the X (West-East) and Y (South-North) directions, respectively.

ICs AND BCs Because there is no temperature component, the specification of BCs and ICs is even simpler. As an IC, the idealized ocean is considered to start from a motionless state, with:

$$u_j^{(n)}\left(x^{(n)}, y^{(n)}, 0\right) = 0 \tag{265}$$

$$(\delta p)^{(n)}\left(x^{(n)}, y^{(n)}, 0\right) = 0 \quad , \tag{266}$$

where the last condition is equivalent to setting $(\delta\rho)^{(n)}\left(x^{(n)}, y^{(n)}, 0\right) = 0$ initially in the LBM solver. For the square-geometry setup, a no-slip BC is taken for all sides of the domain.

### 5.3.1.3 *Numerical method*

The model developed here is based on the D2Q9 lattice (Figure 21). We re-use the fluid-part from the 2D model for the RB problem, which was already described in Section 5.1.3.

### 5.3.1.4 *Simulation results*

The simulation results after a period of $t = 15$ weeks $\equiv t^{(d)} = 1$ are given in Figure 45.

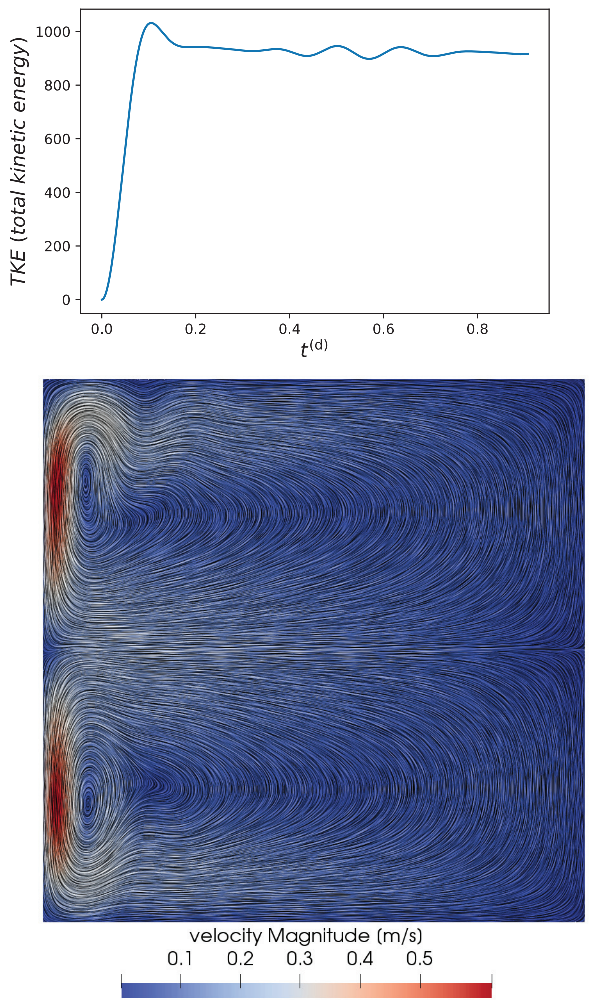


Figure 45: (top) Total kinetic energy (TKE) versus time, for a duration of 15 weeks since the onset of the wind-forcing. (bottom) Visualization of the streamlines at the end of the simulation, for the square-domain *wind-driven ocean circulation* (WDOC) setup.

Because the Munk (1950) theory only applies to the *linear* regime (while our model simulated the *nonlinear* problem), a quantitative comparison with the analytic solutions is not possible. However, qualitatively the results are in agreement with the theory, with the intensification of the ocean-currents (due to the variation of the Coriolis force with latitude) at the western boundary being correctly reproduced. Also, due to the nonlinear effects, the structure of the flow is more complex compared to the analytic solutions (which, for example, predict more straight streamlines on the left sides closet to the northern and southern boundaries, rather than the sinuous profiles computed by our model for those regions).

### 5.3.2 *Realistic geometry simulation*

For a final simulation, we apply the algorithm for simulating the wind-driven ocean circulation, by incorporating a realistic land-mask of the ocean. Because of the complexity of the geometry, this is a good demonstration of the ability of LBM to simulate flows with very fine-scale features, without increasing the complexity of the algorithm excessively.

**Limitations of the current model**

Before we discuss the setup and the results, the reader should keep in mind that this was *not* an attempt to build an ocean model. Indeed, this model has many limitations, which would need to be addressed before it can make any quantitative predictions:

- The *equirectangular projection* (also known as the *equidistant cylindrical* or *geographic* projection) was used, and the 2D RANS (with horizontal eddy-viscosity) equations were "postulated" to hold onto the resulting planar surface. This introduces large distortions near the poles.
- Being a vertically-integrated model (as assumed by the theory of Munk (1950)) with an uniform ocean-depth (equal to the average depth of the global oceans, i.e. $\sim 4$ km), all vertical motions are ignored (which are essential for resolving the meridional overturning circulation).
- Finally, an idealized wind-forcing was used.

Nonetheless, despite this limitations, the model is able to capture some essential features of the major surface currents, as will be demonstrated later.

#### 5.3.2.1 *Modifications to the forcing terms*

To make this global simulation more realistic, the following modifications were used for the forcing terms (relative to the square-domain simulation):

- more accurate expression for the Coriolis force: While the β-plane approximation for the Coriolis force was sufficient for the square domain, this is no longer usable for the global setup (where it would cause the Coriolis force to be unrealistically-large in higher latitudes). Instead, we use the exact expression of the *horizontal* Coriolis acceleration in the *local Cartesian* coordinate system:

$$\mathbf{a}^C = 2\Omega \sin\phi \begin{pmatrix} +u_2 \\ -u_1 \end{pmatrix}, \tag{267}$$

  where $\phi$ is the latitude.

- global-scale approximation for the zonal wind-forcing: For the wind-forcing, we use a square-sinusoidal approximate expression, symmetric around the equator:

$$\mathbf{T} = T_0 \begin{pmatrix} \sin^2(2\phi) \\ 0 \end{pmatrix}, \tag{268}$$

with $T_0 = 8 \times 10^{-7}\ {}^{m}/{}_{s^2}$.

#### 5.3.2.2 *Geometry-coarsening algorithm*

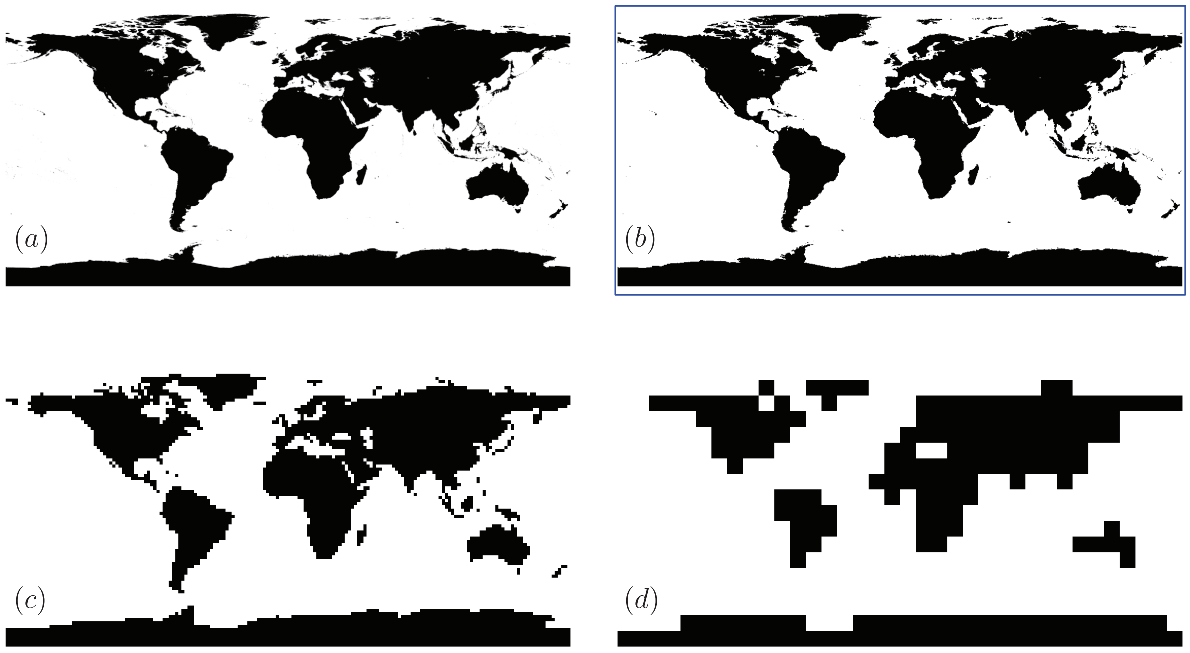


Figure 46: Examples of land-masks produced by the coarsening-algorithm: (a) original (ETOPO2v2) 2-minute grid, (b) coarsened 20-minue grid, (c) coarsened 2° grid, and (d) coarsened 10° grid. The blue rectangle indicates the resolution which was used for calculating the results presented later.

For preparing the geometry of these complex-geometry runs, the ETOPO2v2[13] 2-minute gridded global relief data was used, which provides elevation data on the same type of lon-lat grid as used by the LBM model which was developed for these simulations. However, the native mesh-size of this ETOPO dataset ($10800 \times 5400$ nodes) proved to be impractical for simulations (mostly due to the limited amount of memory attached to the GPUs available to the author). Therefore, a flexible mesh-coarsening algorithm was developed, which constructed lower-resolution land-mask grids, based on the simple criterion of considering a node on the coarse grid to be "land", if more than 50% of the surrounding nodes on the ETOPO grid were land. Figure 46 shows the land-masks produced by this algorithm, for various values of the coarsening-factor. All black nodes are to be taken as walls (no-slip BC), and periodic domain-wrapping is assumed for the X (West-East) direction.

13 https://www.ngdc.noaa.gov/mgg/global/etopo2.html

#### 5.3.2.3 *Simulation results*

A visualization of the streamlines and of the velocity-magnitude after 15 weeks of integration is given in Figure 47 . This figure demonstrates that, remarkably, even our modest 2D model is able to capture the direction (and, most importantly, the western-boundary intensification) of the major ocean currents.

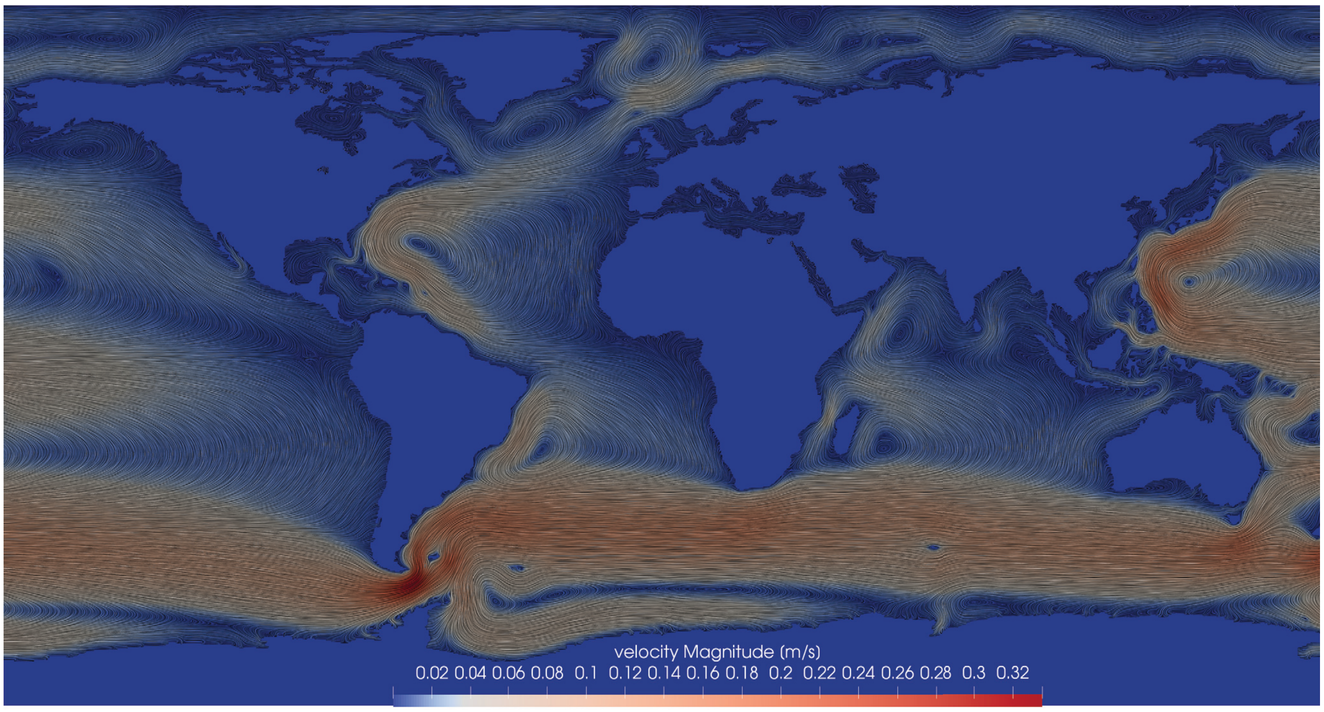


Figure 47: Visualization of the streamlines after 15 weeks of integration for the barotropic model of the wind-driven circulation, with a realistic land-mask. The color-bar indicates the velocity-magnitude. *NOTE: This result is not meant to represent a realistic ocean model, due to the limitations of the equations used (see discussion in the text).*

To illustrate the direction of the flow, we plot the same result with velocity-vectors as an overlay (Figure 48). Separated by the corresponding ocean basin, the model resolves (qualitatively) the following surface currents sketched in Figure 10:

- Atlantic Ocean: In the *northern hemisphere*, the model reproduces the Gulf Stream (including even features of the Florida Current), the North Atlantic Current, and the North Equatorial Current. However, the Labrador Current is shifted too much towards Iceland (and does not disturb the Gulf Stream on the eastern coast of North America).

  In the *southern hemisphere*, the model reproduces the South Equatorial Current, the Falkland Current and the Benguela Current. Although the Brazil Current can be identified, it has an unrealistic shape (shifted too much to the north, and centered too close to the coast of South America). The Guinea Current is not reproduced.

- Pacific Ocean: In the *northern hemisphere*, the model reproduces the Kuroshio Current and the Alaska Current. Although the North Equatorial Current is

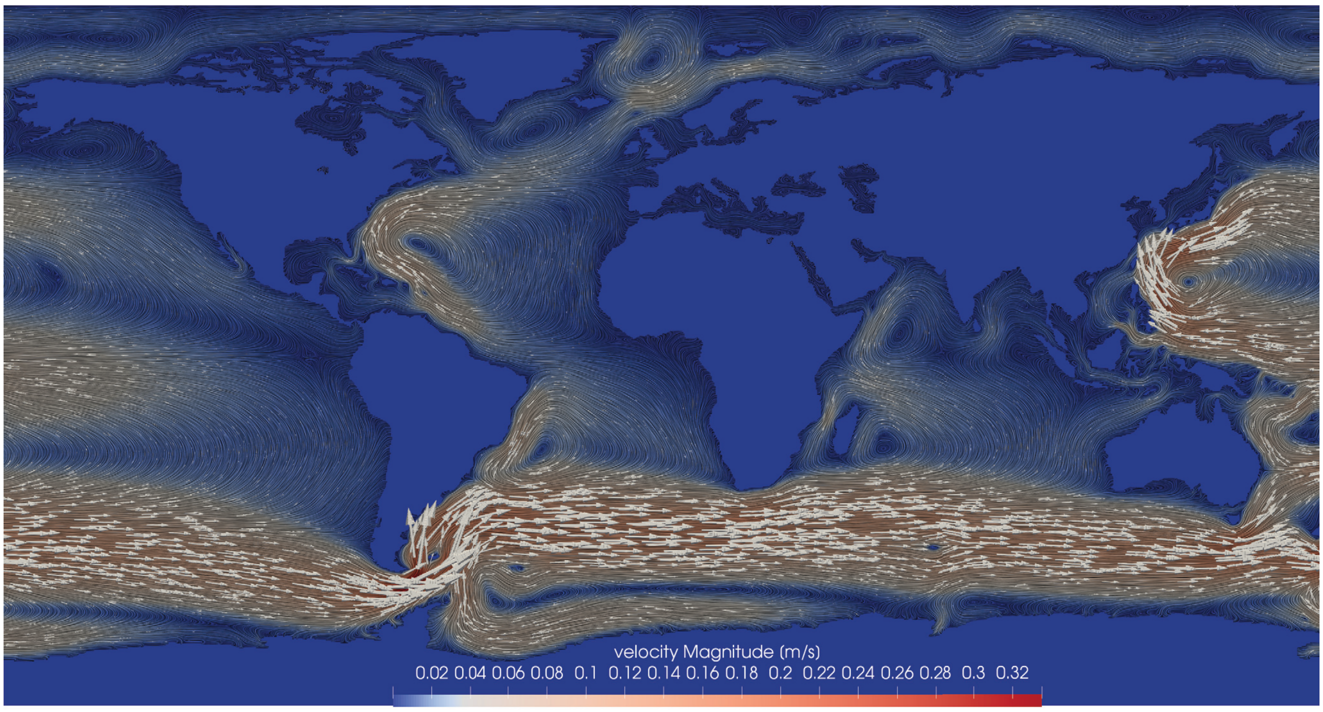


Figure 48: Visualization of the streamlines after 15 weeks of integration for the barotropic model of the wind-driven circulation, with a realistic land-mask. The color-bar indicates the velocity-magnitude, while the vectors indicate the local direction of the flows. *NOTE: This result is not meant to represent a realistic ocean model, due to the limitations of the equations used (see discussion in the text).*

present, it is shifted too much towards the south, and there is no Equatorial Countercurrent. The East Australian Current is absent, being replaced by an outflow from the *Antarctic Circumpolar Current* (ACC) (also known as the West Wind Drift), which produces a flow in the opposite direction. Also, the Kanchatka Current is absent.

In the *southern hemisphere*, the Peru Current is reproduced. Although the South Equatorial Current also appears, it is shifted too much to the south.

- Indian Ocean: In the *northern hemisphere*, the Equatorial Countercurrent is reproduced, but the North Equatorial Current does not appear. Also, the results for the East India Coastal Current are inconclusive.[14]

  In the *southern hemisphere*, the South Equatorial Current is reproduced. Although the direction of the flow between Africa and Madagascar is correct, the Agulhas Current does not appear (probably due to a too strong ACC). The West Australian Current does appear, but it is too close to Madagascar.

- Southern Ocean: Here, the ACC is well-reproduced, although it appears to be too strong (causing disturbances of the currents mentioned above, from the ad-

14 Because this current changes orientation throughout the year due to the monsoon cycle (Shankar et al., 1996), it is a good example of features that cannot be reproduced by our simplified model, for which a time-independent surface wind-stress was prescribed.

jacent ocean basins). The most probable cause of the stronger ACC is the fact that our model effectively assumes a constant ocean depth, while in reality there are several regions (e.g. in the Drake Passage) where the ocean is relatively shallow (which would normally restrain the flow). Also, as compounding factors we have the distortions due to the chosen projection (which gives too much importance to the polar regions in general).

- Arctic Ocean: Finally, here we observe that the North Atlantic Drift is reproduced, as well as the Greenland Current. However, there are several "gyres" which are not physical (for example, to the east of the Svalbard islands, above the Laptev Sea, above the Kamchatka peninsula, and in the Beaufort Sea). The most probable reason for these artifacts is the geometry used (caused by the projection), which causes the North Pole to be actually a line (with no-slip BCs). Also, the flow in the Arctic regions is influenced by distortions due to the projection (to an even greater extent than for the Antarctic regions, where the high-latitude regions are occupied by Antarctica).

Because of the unrealistic representation of the curvature of the Earth, this simulation should only be considered a preliminary result, presented mainly to demonstrate the ability of the *lattice Boltzmann method* (LBM) to simulate domains with very complex geometries. Nonetheless, it can be observed that even this model is able to capture (with the correct direction) most of the major surface ocean currents. Also, the currents which are reproduced also show a pronounced intensification of the flows near the western boundaries, as expected from the theory of Munk (1950) and Stommel (1948).

One of the problems with the shortcomings of the model, which can be observed when zooming-in on the streamlines close to some of the continent boundaries (especially on the western boundaries, e.g. on the western coast of South America), is that the streamlines are not always parallel to the contours of the continent (as they should be). This is due to a combination of two effects:

- Artifacts due to the plotting routine

  To make the model-output compatible with the visualization software, the model was programmed to write values for the velocity even for the nodes inside the continent (a value of zero is produced in that case).

- Accuracy-limitations of the *bounce-back* (BB) *boundary condition* (BC)

  A shortcoming of the BB BC is that it is only $1^{\text{st}}$-order accurate for the general (non-axis-aligned) case. This leads to small errors in the velocities near the boundaries (where the velocity should be *exactly* zero in theory, due to the no-slip BC). This leads to small errors in the velocity near the boundaries, which are exaggerated by the plotting artifacts mentioned above.

However, these defects in the streamlines are most visible in the regions where the velocities are small. Therefore, they should not have (in principle) a significant effect on the overall appearance of the major currents. Nonetheless, some possible approaches for reducing these effects are given in Chapter 7.

Part III

# SYNTHESIS

# 6 DISCUSSION AND CONCLUSIONS

"Science is spectral analysis. Art is light synthesis"

Karl Kraus

## 6.1 GeLB – a tool for lattice Boltzmann modeling

The new *Generic lattice Boltzmann framework* (GeLB) *domain-specific language* (DSL) was designed and implemented, to facilitate testing, inter-comparison, and development of new numerical simulations using algorithms based on the *lattice Boltzmann* (LB) approach. Because it allows expressing the numerical algorithms in a language which is closer to the problem-domain (compared, for example, to *general-purpose languages* (GPLs)), this tool increases the productivity of the user, by abstracting-away many of the time-consuming technical concerns usually encountered in practical numerical simulations. On the other hand, the DSL still preserves internally the topological information which is necessary for generating code with good performance (although making extensive use of this knowledge is still a work in progress). To demonstrate its usefulness, the new tool was used to simulate three concrete physical problems.

## 6.2 Rayleigh-Bénard (RB) convection

The RB problem has been simulated numerically, using both 2D and 3D versions of a *multiple-relaxation-times* (MRT) *lattice Boltzmann method* (LBM) model. The dynamics of this type of flow is determined by the relative importance of *destabilizing* and *stabilizing* mechanisms. In the first category we have the buoyancy force, which causes the fluid from the lower regions to tend to rise, and the fluid from the upper regions to sink. The second category consists of the diffusion of temperature and of the viscous forces. These competing effects make this problem *conditionally stable*. From an energetic point of view, it can be said that in some situations *heat conduction alone* is sufficient for transferring the heat from the lower- to the upper-regions, while in other cases *convection* is also needed, to enhance the heat transfer. The system exhibits several bifurcations, which in theory depend only on the Rayleigh (Ra) and the Prandtl (Pr) number. Analytic results from linear stability theory (Chandrasekhar, 1981) predict that the onset of the first bifurcation depends only on the Rayleigh number – with no-slip BCs at the horizontal walls (same as used in our numerical experiments), the theory predicts the rest-state to be stable when $\text{Ra} \leqslant \text{Ra}_{\text{cr},1}^{\text{theoretical}} = 1707.762$ (see discussion in Section 2.4). This is a simple metric, which can be used to check if numerical models are reproducing the correct physics of the problem. Both model presented in this thesis were found to reproduce $\mathrm{Ra}_{cr,1}$ with good accuracy (relative error $< 1\%$), with a slight advantage for the 2D model. The next bifurcation also depends on the Prandtl number; for the fixed value of $\mathrm{Pr} = 0.71$ used in the present work, the general consensus is that $\mathrm{Ra}_{cr,2} \approx 6000$ (more exactly, Willis and Deardorff (1970) found experimentally that $\mathrm{Ra}_{cr,2} = 5800$, while the theoretical analysis of Clever and Busse (1974) yielded a value of $\mathrm{Ra}_{cr,2} = 6000$). Preliminary testing with the 3D model shown here demonstrated a value of $\mathrm{Ra}_{cr,2}^{\text{numerical}} \in [6000, 6500]$, which is a very encouraging result (establishing more strict bounds for the $\mathrm{Ra}_{cr,2}$ will be the topic of further work). On the other hand, the 2D model was shown to be unable to reproduce $\mathrm{Ra}_{cr,2}$, which indicates that the $\mathrm{Ra} \gg \mathrm{Ra}_{cr,1}$ regimes should *not* be investigated with this model. The reason for this limitation of the 2D model is obvious: for $\mathrm{Ra} \geqslant \mathrm{Ra}_{cr,1}$, the flow becomes strongly three-dimensional. Interestingly, this three-dimensionality was even observed below $\mathrm{Ra}_{cr,2}$, where the fluid seems to settle into (time-independent) patterns which are topologically different from the rolls to which the 2D model is limited.

An unexpected outcome of our work was to show the importance of the aspect-ratio for reproducing the correct behavior (i.e. transition to time-dependent flow) near $\mathrm{Ra}_{cr,w}$. This is an important effect, to be considered in future investigations of this problem, for the high-Ra regimes (even if the computational demands scale proportionally with the square of the aspect-ratio).

Based on the good results for $\mathrm{Ra}_{cr,1}$ and $\mathrm{Ra}_{cr,2}$, we could also apply the 3D model for the high-Ra regime – for example, a simulation at $\mathrm{Ra} = 10^6$ indicated a regime of turbulent convection (which is also in line with the conclusions of other authors (Busse, 1978)). Here, one remarkable feature of the MRT algorithms was observed, namely the high degree of stability, even with coarse grids (however, the method probably behaves as an "implicit LES" algorithm in such situations).

## 6.3 WIND-DRIVEN OCEAN CIRCULATION (WDOC)

To demonstrate the potential applicability of the newer MRT LBM algorithms to ESS research, we simulated numerically the *wind-driven ocean circulation* (WDOC) of a barotropic (i.e. vertically-integrated) ocean in the nonlinear regime, both for the classical setup (square geometry) and for a new setup with complex geometry (based on the currently-available land-mask data). Both simulations yielded very encouraging results, with the first one being able to reproduce the final flow-features (and, especially, the western-boundary current intensification) reported by previous work due to Veronis (1966) and Wolf-Gladrow (2000).

Rather remarkably (considering the limitations of the model) the second simulation (with the realistic geometry) also able to reproduce (at least, qualitatively) most of the important currents which are known to occur in the real oceans.

However, the complex-geometry simulation can only be considered preliminary, because several imporant features are still incorrect. For example, the *Antarctic Circumpolar Current* (ACC) is somewhat stronger than in reality (probably because the non-uniform depth of the bathymetry is not considered). Also, the circulation in the polar regions in general is affected by several simplifying assumptions in the geometry, which artificially increase the importance of the currents in the higher latitudes, and create unphysical gyres near the North Pole. Finally, time-dependent effects (such as the reversal of the East India Coastal Current) are not captured, most probably due to the constant wind-forcing prescribed for the simulation. All of these issues need to be addressed by further model-development work.

# 7 OUTLOOK

"Prediction is very difficult, especially if it's about the future"

Niels Bohr

The work documented in this thesis combined aspects from several disciplines of science and engineering. Therefore, it is useful to separate these aspects when discussing the possible future improvements – Section 7.1 discusses the technical/computer-science-related improvements to the GeLB model framework, while Section 7.2 discusses the natural extensions of the work from the point of fluid-dynamics modeling and of applications in *Earth system science* (ESS).

## 7.1 POSSIBLE IMPROVEMENTS TO GELB

The *GeLB description* (GD) programming language which is at the core of GeLB is still very young, and there is much room for improvement. Because some of the possible improvements require resources or expertise that the author does not posses currently, it would be ideal to attract a group of open-source developers to help with some of this work.

Some of the possible areas for improvement are discussed in the next subsections.

### *Further optimize performance of generated code*

The LBM and CS literature contains a wide range of code-optimization techniques. Some of the techniques (such as combining the streaming- and collision-phases into a single spatial sweep) were already used for this thesis. However, there are additional optimization techniques, which may be considered in the future:

1. The worst "performance sin" of the GeLB version used for this thesis was the fact that it used two separate lattices for holding the old and the new *distribution functions* (DFs). This makes it easier to ensure that the code is correct (which was the first priority). However, this approach obviously also doubles the memory-consumption, which: (a) limits the sizes of the lattices that can be used for high-resolution simulations (especially on GPUs, where memory is less plentiful), and (b) also increases the pressure on the cache-hierarchy (for both *central processing units* (CPUs) and GPUs).

   To alleviate this problem, it is (in principle) possible to schedule the node-updates in such a way that a single lattice is used for iteration $\boxed{n}$ and $\boxed{n+1}$, with some buffer-space (for storing $\boxed{n}$-state which is still needed for updating nodes that are yet to be brought to $\boxed{n+1}$-state). Such an optimization was discussed e.g. by Mattila et al. (2007) and Wittmann et al. (2013), for the case of relatively simple geometries. However, this problem is more difficult to solve in the general case (for the complex geometries supported by GeLB). A complicating factor is the fact that, in order to obtain good performance (especially on GPUs) it is often better to leave the hardware some freedom in terms of the scheduling of node-updates. The author has a prototype algorithm for this general case which, however, still needs to be tested extensively.

2. The cache-utilization of the code could be improved with more advanced data-structures (Demaine, 2002; Kumar, 2003; Strumpen and Frigo, 2006; Nitsure et al., 2006; Bader, 2012), and space-sweeping algorithms (Wellein et al., 2009). This issue is closely related to the previous optimization.

3. Finally, the I/O throughput could be improved by implementing *parallel* I/O. This can be useful for the case when output is to be written frequently to the disk (in which case this operation can become a serious bottleneck).

### *Add more backends*

Version 1 of GeLB, which was described and used in this thesis, only implements the `Fortran` - OpenMP backend, which is suitable for running simulations on single-node, multi-core machines. This target is very popular, but it does not reflect the current state-of-the-art in computing hardware available to scientists – in particular, it does not account for multi-node machines and for computing accelerators such as GPUs or *field-programmable gate arrays* (FPGAs). To fix this shortcoming, `Fortran` - MPI and `C` - OpenCL backends are currently under development. In the future, even hybrid backends (e.g. `C` - MPI - OpenCL) may be considered, depending on the interest of users.

### *Implement a "true" compiler*

The `gelbc` program is the part of the GeLB framework which translates GD programs into the lower-level source-code that is eventually compiled into executables. Because it does *not* directly generate executable code, `gelbc` should be better named a "translator" (some authors also use the term "transpiler"). In principle, there is no reason why `gelbc` could not generate executable code directly. This would allow for better-optimizing code, and it would also simplify usage, by eliminating one compilation step in daily-usage.

One possible approach for makign this transition to a "true" compiler would be to make the *intermediate representation* (IR) currently produced by `gelbc` compatible with the IR of the *LLVM compiler infrastructure project* (LLVM). Because LLVM can already target many of the hardware-platforms planned for future releases of GeLB, this would eliminate the need for implementing backends manually (Section 7.1). However, it could also limit the choice of backends, if they are not currently supported by LLVM (for example, *Message Passing Interface* (MPI) and FPGAs).

*Add a tool for* visual, iterative *simulation-development*

When writing simulations, it would be useful to obtain "live" visual feedback, for example to visualize the in-out stencils of dynamics, or the geometry of the simulation (using a "colouring" of the nodes in the lattice). Such a tool could also remind the user if the simulation-program has any syntax errors, or if it is missing some of the mandatory building-blocks. Most importantly, this tool could highlight errors or inconsistencies related to in-out stencils of the dynamics, which can cause bugs that are difficult to track otherwise (trying to write to a non-existent *distribution function* (DF), attempting reading an un-initialized DF, etc.).

The author developed a rough prototype for such a tool, using `Python` and OpenGL, which may be included in a future release.

*Auto-generate some node-categories*

Currently, the GD language requires users to manually specify distinct functions for each distinct node-category. However, it is easy to notice that many categories are related, and could be (in principle) generated automatically as soon as a category from the same "family" is defined by the user. Specifically, two mechanisms for "category-collapsing" come to mind:

- Make use of rotation/reflection symmetries:

  In many applications using LBM, the categories defined for the corners of the domain are related: once the dynamics for one of the corners is defined, the dynamics of the other three corners could be obtained automatically, by rotation of the in-out stencils in increments of $90°$. Similarly, the dynamics of the bottom edge could be auto-generated from those of the top edge, and those of the right edge from those of the left edge (in these cases, by $180°$ rotations of the in-out stencils). To give an example, for simulating the well-known lid-driven cavity problem in 2D, instead of having to define 9 distinct dynamics, we would only need to define 4, reducing the length of the corresponding GeLB program by $\sim 50\%$. The reduction of boilerplate-code would be even more substantial for 3D simulations with complex wall boundaries.

- Add support for parameterized node-categories:

  A second possible mechanism for reducing the number of categories is to allow dynamics to accept some parameters (similar to templates in `C++`). This is useful when the simulation contains categories for which the dynamics differ only with respect to the magnitude of some parameters – for example, in simulations of thermal fluids where we want to specify a varying *temperature profile*

> along some straight walls, it would be very inconvenient to have to define a new category for each node of the wall. There is currently a way to overcome this problem, by using the node-position arguments that are available to the functions defining the dynamics. However, this mechanism has to be refined.
>
> For example, in the applications for simulating the flows presented in this thesis, the dynamics of the top and bottom nodes are very similar – it would be possible to derive one from the other, using a rotation by 180° and parameterization with respect to the wall-temperature (to impose the different temperatures at the horizontal walls). In these cases, the reduction of the number of categories to be defined manually is more modest – from 3 to 2. However, for more complex geometries the benefits could be more significant.

Because it can significantly reduce the size of GD programs, this feature would be very useful, increasing clarity of programs and decreasing the slope of the "learning curve" for new users.

Using this feature in practice requires an *iterative* approach – some GeLB tools have to monitor what kind of node-categories have been defined already and to compute (based on the geometry "colouring") which ones can be generated automatically. Therefore, this feature should be added to the tool for *visual*, *iterative* simulation-development mentioned above (Section 7.1).

*Refinements of language-syntax*

Designing the syntax for a programming language is, to some extent, an "art" – in the end, some of the language elements are subjective, and determined by the background of the authors. Because version 1 of the GD language was developed by the author of this thesis alone, it probably contains some elements which may not be clear to other users. Therefore, it is valuable to get more feedback from a wider community of users, to discover aspects that could be made more clear in future versions of the language.

*Improve error-messages*

Although the GD language is formally described by the *Extended Backus-Naur-Form* (EBNF) grammar used by the parser in the frontend, this does not mean that error-messages issued by the parser when it encounters syntax- or logic-errors are obvious. Therefore, to increase the productivity of users, it would be very useful to include additional layers of error-handling, with error messages which clearly explain what went wrong. This feature could also be coupled with the work on the visual tool for simulation developent (Section 7.1).

*Add support for GD in text-editors and IDEs*

As a purely technical (but nonetheless important) area of future improvement, it is useful to add support for the GD language in some of the most popular programming text editors and *integrated development environments* (IDEs). The first steps would be to add support for syntax-highlighting and for auto-formatting of code.

This feature could also accelerate any subsequent improvements to the language syntax.[1]

## 7.2 LBM WORK

From a fluid dynamics point of view, there are many opportunities for further studies, both for the specific applications considered in this thesis, as well as for other problems which were beyond the current study.

*Further exploration of the presented applications*

Of the presented applications, the 3D *Rayleigh-Bénard* (RB) convection is one of the most interesting to research further. In particular, the author plans to focus on three important aspects:

1. First, it would be interesting to study how the structure of the initial trigger for breaking the symmetry (Sections 5.1.3.3 and 5.2.3.3) influences the evolution of the setup.

2. A second aspect is the dependence of the results on the geometry. In particular, we would like to perform a more systematic study of the influence of the aspect-ratio on the value of the critical Rayleigh numbers, to understand how this factor constrains the flows.

3. The last topic is tightly coupled with the previous two aspecsts. The idea is to systematically study numerically the rich variety of flow-regimes in the supercritical regimes, and the transition to fully-developed turbulent convection and to compare these simulations with the results of other authors (Busse, 1978, 1981; Koschmieder, 1993; Getling and Brausch, 2003).

*Extension of the work to large-scale modeling in climate sciences*

Although the WDOC simulation in Section 5.3 showed encouraging results, there is still a lot that can be done to have a complete LBM-based ocean model. Specifically, four limitations would need to be overcome:

1 The author designed the GD language to be similar to the `C++` programming language, so that he could re-use the editor-support for that language. However, having dedicated support for the GD language would "lift" this artificial reason for keeping the syntax close to `C++`.

1. Develop a more accurate space-discretization:

   Although we used a general implementation for the Coriolis force in the complex-geometry ocean simulations shown in this thesis, the surface of the Earth was effectively projected onto a cylinder, and then we assumed that the 2D (Cartesian) RA NSEs are valid on the surface of that cylinder. Obviously, this approximation leaves out many geometric factors, which would be required if we would want to accurately account for the curvature of the Earth.

   A promising idea for solving this problem (which would allow re-use of existing and well-tested LBM algorithms) would be to use a "cubed-sphere" (Ronchi et al., 1996) spatial discretization. This approach has the advantage that the mesh (with an appropriate mask) can still be considered regular. Also, there is no singularity at the poles, and even adaptive grid-refinement would become a tractable problem.

2. Solve the problem of mesh-anisotropy for the 3D simulations:

   One of the main obstacles (Salmon, 2009) which prevents the LBM methods from being applied to simulations of large-scale flows in ESS is the common LBM assumption that the underlying computational mesh (i.e. the "lattice") is spatially isotropic (i.e. a *square* grid in 2D and a *cubic* grid in 3D). Unfortunately, this assumption is inadequate for 3D simulations at the planetary scale, where the flow typically has a very high horizontal-to-vertical aspect ratio,[2]. Although the mesh-anisotropy is not the only feature distinguishing planetary-scale flows (Coriolis effects and stratification also play major roles), it is a basic requirement for potential numerical methods.

   Initial work to overcome this limitation in LBM was based on interpolation (Filippova and Hänel, 1998a,b) (which, however, increases artificial dissipation), or solving equations which were not fully consistent hydrodynamically (Bouzidi et al., 2001). Hegele Jr et al. (2013) derived interpolation-free models for rectangular lattices, by using additional DFs (which unfortunately increases the already-high memory footprint of simulations, especially for thermal flows).

   However, an interesting approach that the author recently became aware of is that of Zhou (2012), who derived a rectangular model which has anisotropic fluid viscosity. Despite being detrimental for classical CFD simulations, it may be beneficial for large-scale ESS applications, where it is customary to parameterize the vertical- and horizontal-viscosity (or, more exactly, diapycnal and isopycnal viscosity) differently, to account for the various sub-scale physical effects.

   Also, a new model for simulations on an *anisotropic* mesh was proposed very recently (Zong et al., 2016) which, remarkably, exploits the flexibility of the MRT-LBM formulation to avoid the need for interpolations of for additional DFs

2 For example, the equatorial circumference of the Earth is $\approx 40,075$ km, while the average depth of the ocean is of only $\approx 3.68$ km).

for simulating 2D flows. An extension of this model to 3D would be a very good opportunity for applying LBM to ESS problems.

3. Use more realistic wind-forcing data:

   For the last simulations presented in this thesis, an approximate analytical expression was used for the zonal wind-stress, which was assumed to be time-independent. An obvious next step would be to assimilate real measurement data (e.g. different wind-forcing for each month of the year, as a first approximation). Alternatively, we could also consider coupling this model to a real atmosphere model, such as `ECHAM` (Stevens et al., 2013).

4. Use more accurate *boundary conditions* (BCs) schemes:

   As mentioned in Section 5.3.2.3, the *bounce-back* (BB) *boundary condition* (BC) used for imposing the no-slip condition at the continental boundaries are only 1st-order accurate in space for boundaries with arbitrary contours. Several alternative schemes (which are expected to yield more accurate results) are available in the literature (e.g. Mei et al. (1999) and Guo et al. (2002a)). The general idea behind such schemes is to increase the radius of the stencils, by considering neighboring nodes.[3] Importantly, these higher-order BC schemes do not significantly increase the complexity of the code, which makes the *lattice Boltzmann method* (LBM) approach a promising research tool for problems with complex geometries. A systematic investigation of the effects of such schemes on the present setup is an interesting topic, to be investigated in the future.

*Quantification of performance and of parallel scalability*

In addition to the ability of LBM to simulate flows with complex geometries, the method is also an ideal candidate for parallelization on common (CPUs, GPUs) or emerging (e.g. FPGAs) hardware-architectures. A natural continuation of the work presented in this thesis (especially the GeLB frameworks) is to study the performance, as well as the parallel-execution scaling of the LBM algorithms, in comparison to other CFD methods:

- For the performance studies, several aspects can be analyzed – for example, two common metrics are the *time-to-solution* and the *energy-to-solution*.
- For the scalability studies, it is interesting to look at both the
  a) weak scaling: i.e. how the time-to-solution changes as the number of execution-units is increased, for a problem of *fixed* size, and at the
  b) strong scaling: i.e. how the amount of work per unit time changes, as the size of the problem is increased along with the number of execution-units.

3 In contrast, the BB scheme is entirely local, which greatly simplifies the implementation – this is one of the main reasons why this scheme remains very popular).

This quantity is also becoming more-and-more relevant for ESS simulations, as the resolution of the models is increased (to increase realism).

# BIBLIOGRAPHY

Aidun, C. K. and J. R. Clausen (2010). Lattice-Boltzmann Method for Complex Flows. *Annual Review of Fluid Mechanics* 42:439–472.

Amante, C. and B. W. Eakins (2008). ETOPO 1 Arc-Minute Global Relief Model: Procedures, date Sources and analysis. *NOAA Technical Memorandum NESDIS NGDC-24*:19.

Ansumali, S. and I. V. Karlin (2000). Stabilization of the lattice Boltzmann method by the H theorem: A numerical test. *Physical Review E* 62 (6):7999–8003.

Bader, M. (2012). *Space-filling curves: an introduction with applications in scientific computing*. Springer Science & Business Media.

Bénard, H. (1900). Les tourbillons cellulaires dans une nappe liquide [The cellular vortices in a liquid layer]. *Rev Gén Sci Pure Appl* 11:1261–1271.

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

Berlekamp, E. R., J. H. Conway, and R. K. Guy (2003). *Winning ways for your mathematical plays*. Vol. 3. AK Peters Natick.

Bogoliubov, N. N. (1962). Problems of a Dynamical Theory in Statistical Physics [in Russian], Gostekhizdat, Moscow (1946). *English transl., North-Holland, Amsterdam*.

Boltzmann, L. (1872). Weitere Studien über das Wärmegleichgewicht unter Gasmolekülen. *Wiener Berichte*.

Born, M. and H. S. Green (1947). A Kinetic Theory of Liquids. *Nature* 159 (4034):251–254.

Boussinesq, J. (1877). Essai sur la théorie des eaux courantes. *Mémoires présentés par divers savants à l'Académie des Sciences* 23.

Bouzidi, M., D. D'Humières, P. Lallemand, and L.-S. Luo (2001). Lattice Boltzmann Equation on a Two-Dimensional Rectangular Grid. *Journal of Computational Physics* 172 (2):704–717.

Bryan, K. (1963). A Numerical Investigation of a Nonlinear Model of a Wind-Driven Ocean. *Journal of the Atmospheric Sciences* 20 (6):594–606.

Bryan, K. and M. D. Cox (1967). A numerical investigation of the oceanic general circulation. *Tellus* 19 (1):54–80.

Buckingham, E. (1914). On Physically Similar Systems; Illustrations of the Use of Dimensional Equations. *Physical Review* 4 (4):345–376.

Businger, S. and J. A. Businger (2001). Viscous dissipation of turbulence kinetic energy in storms. *Journal of the Atmospheric Sciences* 58 (24):3793–3796.

Busse, F. H. (1967). On the Stability of Two-Dimensional Convection in a Layer Heated from Below. *Journal of Mathematics and Physics* 46 (1-4):140–150.

— (1978). Non-linear properties of thermal convection. *Reports on Progress in Physics* 41 (12):1929–1967.

Busse, F. H. (1981). Transition to turbulence in Rayleigh-Beénard convection. Springer Berlin Heidelberg, p. 97–137.

Busse, F. H. and R. M. Clever (1979). Instabilities of convection rolls in a fluid of moderate Prandtl number. *Journal of Fluid Mechanics* 91 (02):319.

Caiazzo, A., M. Junk, and M. Rheinländer (2009). Comparison of analysis techniques for the lattice Boltzmann method. *Computers & Mathematics with Applications* 58 (5):883–897.

Canuto, C., M. Y. Hussaini, A. M. Quarteroni, and A. Thomas Jr (1988). *Spectral methods in fluid dynamics*. Springer Science & Business Media. ISBN: 978-3-540-52205-8.

Chamberlain, B. L. (2015). *Programming models for parallel computing*. Ed. by Pavan Balaji. MIT Press, p. 458. ISBN: 9780262528818.

Chandrasekhar, S. (1981). *Hydrodynamic and hydromagnetic stability*. Dover Publications, p. 652. ISBN: 048664071X.

Chapman, G. T. and M. Tobak (1985). Observations, Theoretical Ideas, and Modeling of Turbulent Flows – Past, Present, and Future. *Theoretical Approaches to Turbulence*.

Chapman, S. and T. G. Cowling (1970). *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. 3rd. Cambridge University Press. ISBN: 9780521408448.

Chen, S., G. D. Doolen, Y. Shi, and J. E. Sader (1998). Lattice boltzmann method for fluid flows. *Annual Review of Fluid Mechanics* 30 (Kadanoff 1986):329–364.

Chirila, D. B. and G. Lohmann (2015). *Introduction to Modern Fortran for the Earth System Sciences*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-37008-3.

Clever, R. M. and F. H. Busse (1974). Transition to time-dependent convection. *J. Fluid Mech* 65 (4):625–645.

Czaja, A. and J. Marshall (2006). The Partitioning of Poleward Heat Transport between the Atmosphere and Ocean. *Journal of the Atmospheric Sciences* 63 (5):1498–1511.

D'Humieres, D. (1992). Generalized Lattice-Boltzmann Equations. *Rarefied Gas Dynamics: Theory and Simulations* 159:450 –458.

D'Humieres, D., I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo (2002). 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.

Davis, S. H. (1969). On the Principle of Exchange of Stabilities. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 310 (1502):341–358.

Deardorff, J. W. (1970). Convective Velocity and Temperature Scales for the Unstable Planetary Boundary Layer and for Rayleigh Convection.

Dellar, P. J. (2013). An interpretation and derivation of the lattice Boltzmann method using Strang splitting. *Computers & Mathematics with Applications* 65 (2):129–141.

Demaine, E. D. (2002). Cache-oblivious algorithms and data structures. *Lecture Notes from the EEF Summer School on Massive Data Sets*:1–29.

Dirichlet, P. G. L. (1850). *Abh. Königlich. Preuss. Akad. Wiss.*99–116.

Dong, S. and G. E. Karniadakis (2005). DNS of flow past a stationary and oscillating cylinder at Re = 10000. *Journal of Fluids and Structures* 20 (4 SPEC. ISS.):519–531.

Durrant, J. D. and J. A. McCammon (2011). Molecular dynamics simulations and drug discovery. *BMC Biology* 9 (1):71.

E, W. (2011). *Principles of Multiscale Modeling*. English. 1 edition. Cambridge, UK ; New York: Cambridge University Press, p. 488. ISBN: 978-0-521-82544-3.

Eckhardt, Bruno (2012). Turbulence Transition in Shear Flows: Chaos in High-Dimensional Spaces. *Procedia IUTAM* 5:165–168.

Ekman, V. W. (1905). On the influence of the Earth's rotation on ocean-currents. *Arkiv för matematik, astronomi och Fysik* 2 (11).

— (1923). Über Horizontalzirkulation bei winderzeugten Meeresströmungen. *Arkiv för matematik, astronomi och Fysik* 17 (26).

Feichtinger, C., S. Donath, H. Köstler, J. Götz, and U. Rüde (2011). WaLBerla: HPC software design for computational engineering simulations. *Journal of Computational Science* 2 (2):105–112.

Filippova, O. and D. Hänel (1998a). Boundary-Fitting and Local Grid Refinement for Lattice-BGK Models. *International Journal of Modern Physics C* 09 (08):1271–1279.

— (1998b). Grid Refinement for Lattice-BGK Models. *Journal of Computational Physics* 147 (1):219–228.

Fox, R. O. (2011). Large-eddy-simulation tools for multiphase flows. 44:47–76.

Frisch, U. (1995). *Turbulence : the legacy of A.N. Kolmogorov*. Cambridge University Press, p. 296. ISBN: 0521457130.

Frisch, U., B. Hasslacher, and Y. Pomeau (1986). Lattice-Gas Automata for the Navier-Stokes Equation. *Physical Review Letters* 56 (14):1505–1508.

Geier, M., M. Schönherr, A. Pasquali, and M. Krafczyk (2015). The cumulant lattice Boltzmann equation in three dimensions: Theory and validation. *Computers and Mathematics with Applications* 70 (4):507–547.

Getling, A. V. and O. Brausch (2003). Cellular flow patterns and their evolutionary scenarios in three-dimensional Rayleigh-Bénard convection. *Physical Review E* 67 (4):046313.

Ginzburg, I. and D. D'Humieres (2003). Multireflection boundary conditions for lattice Boltzmann models. *Physical Review E* 68 (6):066614.

Grossmann, S. and D. Lohse (2000). Scaling in thermal convection: a unifying theory. *Journal of Fluid Mechanics* 407 (March):S0022112099007545.

Grossmann, Siegfried (2000). The onset of shear flow turbulence. *Reviews of Modern Physics* 72 (2):603–618.

Guo, Z. and C. Shu (2013). *Lattice Boltzmann method and its applications in engineering*. English. Singapore; Hackensack, N.J.: World Scientific Pub. Co. ISBN: 9789814508308 9814508306.

Guo, Z., C. Zheng, and B. Shi (2002a). An extrapolation method for boundary conditions in lattice Boltzmann method. *Physics of Fluids* 14 (6):2007–2010.

— (2002b). Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E* 65 (4):046308.

Hardy, J., Y. Pomeau, and O. de Pazzis (1973). Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions. *Journal of Mathematical Physics* 14 (12):1746.

Hegele Jr, L. A., K. Mattila, and P. C. Philippi (2013). Rectangular Lattice-Boltzmann Schemes with BGK-Collision Operator. *Journal of Scientific Computing* 56 (2):230–242.

Hellerman, S. and M. Rosenstein (1983). Normal Monthly Wind Stress Over the World Ocean with Error Estimates. *Journal of Physical Oceanography* 13 (7):1093–1104.

Heuveline, V. and J. Latt (2007). The OpenLB project: an open source and object oriented implementation of lattice Boltzmann methods. *International Journal of Modern Physics C* 18 (04):627–634.

Hinze, J. O. (1959). *Turbulence: An Introduction to Its Mechanism and Theory*. McGraw-Hill.

Huang, K. (1987). *Statistical Mechanics, 2nd Edition*. English. 2 edition. New York: Wiley, p. 493. ISBN: 978-0-471-81518-1.

Irving, J. H. and J. G. Kirkwood (1950). The Statistical Mechanical Theory of Transport Processes. IV. The Equations of Hydrodynamics. *The Journal of Chemical Physics* 18 (6):817–829.

Kardar, M. (2007). *Statistical Physics of Particles*. English. 1 edition. Cambridge University Press, p. 330. ISBN: 978-0-521-87342-0.

Kirkwood, J. G. (1946). The Statistical Mechanical Theory of Transport Processes I. General Theory. *The Journal of Chemical Physics* 14 (3):180–201.

Kleinstreuer, C. and Z. Zhang (2010). Airflow and Particle Transport in the Human Respiratory System. *Annual Review of Fluid Mechanics* 42 (1):301–334.

Kolmogorov, A. N. (1941a). Dissipation of energy in isotropic turbulence. *Dokl. Akad. Nauk SSSR* 32:325–327.

— (1941b). Local structure of turbulence in an incompressible viscous fluid at very large Reynolds numbers. *Dokl. Akad. Nauk SSSR* 30:299–301.

— (1941c). On the degeneration of isotropic turbulence in an incompressible viscous fluid. *Dokl. Akad. Nauk SSSR* 31:319–323.

Koschmieder, E. L. (1993). *Bénard cells and Taylor vortices*. Cambridge University Press. ISBN: 0521402042.

Krishnamurti, R. (1973). Some further studies on the transition to turbulent convection. *Journal of Fluid Mechanics* 60 (02):285.

Krüger, T., H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and Viggen E. M. (2017). *The Lattice Boltzmann Method: Principles and Practice*. Springer. ISBN: 978-3-319-44647-9.

Kumar, P. (2003). Cache oblivious algorithms. Springer, p. 193–212.

Kundu, P. K., I. M. Cohen, and D. R. Dowling (2012). *Fluid mechanics*. English. Waltham, M. A.: Academic Press. ISBN: 9780123821003 0123821002.

Kuo, L.-S. and P.-H. Chen (2009). Numerical implementation of thermal boundary conditions in the lattice Boltzmann method. *International Journal of Heat and Mass Transfer* 52 (1-2):529–532.

Lappa, M. (2010). *Thermal convection : patterns, evolution and stability*. Wiley, p. 670. ISBN: 9780470699942.

Latt, J. and B. Chopard (2006). Lattice Boltzmann method with regularized pre-collision distribution functions. *Mathematics and Computers in Simulation* 72 (2-6):165–168.

LeVeque, R. J. (1997). Wave Propagation Algorithms for Multidimensional Hyperbolic Systems. *Journal of Computational Physics* 131 (2):327–353.

— (1998). Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm. *Journal of Computational Physics* 146 (1):346–365.

Li, L., R. Mei, and J. F. Klausner (2013). Boundary conditions for thermal lattice Boltzmann equation method. *Journal of Computational Physics* 237:366–395.

Llewellin, E. W. (2010a). LBflow : An extensible lattice Boltzmann framework for the simulation of geophysical flows . Part II : usage and validation. *Computers & Geosciences* 36 (2):115–122.

— (2010b). LBflow: An extensible lattice Boltzmann framework for the simulation of geophysical flows. Part I: theory and implementation. *Computers and Geosciences* 36 (2):115–122.

Lohmann, G. (1996). Stability of the thermohaline circulation in analytical and numerical models= Stabilität der thermohalinen Zirkulation in analytischen und numerischen Modellen. *Berichte zur Polarforschung (Reports on Polar Research)* 200.

Lorenz, E. N. (1963). Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences* 20 (2):130–141.

Lumley, J. L. (1970). *Stochastic Tools in Turbulence*. Applied Mathematics and Mechanics. Dover Publications. ISBN: 9780123957726.

Luo, L.-S., M. Krafczyk, and W. Shyy (2010). Lattice Boltzmann Method for Computational Fluid Dynamics. *Encyclopedia of Aerospace Engineering* (0):651–660.

MacCormick, J. (2012). *Nine algorithms that changed the future: the ingenious ideas that drive today's computers*. Princeton University Press, p. 219. ISBN: 9780691147147.

Manneville, P. (2004). *Instabilities, chaos and turbulence : an introduction to nonlinear dynamics and complex systems*. Imperial College Press, p. 391. ISBN: 9781848163928.

Marshall, J. and R. A. Plumb (2008). *Atmosphere, ocean, and climate dynamics : an introductory text*. Elsevier Academic Press, p. 319. ISBN: 9780080556703.

Marshall, J. and F. Schott (1999). Open-ocean convection: Observations, theory, and models. en. *Reviews of Geophysics* 37 (1):1–64.

Martínez, D. O., W. H. Matthaeus, S. Chen, and D. C. Montgomery (1994). Comparison of spectral method and lattice Boltzmann simulations of two-dimensional hydrodynamics. *Physics of Fluids* 6 (3):1285–1298.

Mattila, K., J. Hyväluoma, T. Rossi, M. Aspnäs, and J. Westerholm (2007). An efficient swap algorithm for the lattice Boltzmann method. *Computer Physics Communications* 176 (3):200–210.

Mei, R., L.-S. Luo, and W. Shyy (1999). An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method. *Journal of Computational Physics* 155 (2):307–330.

Mei, R., W. Shyy, D. Yu, and L.-S. Luo (2000). Lattice Boltzmann Method for 3-D Flows with Curved Boundary. *Journal of Computational Physics* 161 (2):680–699.

Mernik, M., J. Heering, and A. M. Sloane (2005). When and how to develop domain-specific languages. *ACM computing surveys (CSUR)* 37 (4):316–344.

Miller, R. N. (2007). *Numerical Modeling of Ocean Circulation*. Cambridge: Cambridge University Press. ISBN: 9780511618512.

Moon, I.-J. (2005). Impact of a coupled ocean wave-tide-circulation system on coastal modeling. *Ocean Modelling* 8 (3):203–236.

Moore, D. R. and N. O. Weiss (1973). Two-dimensional Rayleigh-Benard convection. *Journal of Fluid Mechanics* 58 (02):289–312.

Mori, H. (1965). Transport, Collective Motion, and Brownian Motion. *Progress of Theoretical Physics* 33 (3):423–455.

Müller, P. (2006). *The equations of oceanic motions*. Cambridge University Press.

Munk, W. H. (1950). On the wind-driven ocean circulation. *Journal of Meteorology* 7 (2):80–93.

Neumann, C. (1887). *Über die Methode des arithmetischen Mittels*. Vol. 13. Hirzel.

Nitsure, A., K. Iglberger, U. Rüde, C. Feichtinger, G. Wellein, G. Hager, and F. Alexander (2006). *Optimization of Cache Oblivious Lattice Boltzmann Method in 2D and 3D*. Tech. rep.

Oberbeck, A. (1879). Über die Wärmeleitung der Flüssigkeiten bei Berücksichtigung der Strömungen infolge von Temperaturdifferenzen. *Annalen der Physik und Chemie* 243 (6):271–292.

Obrecht, C., F. Kuznik, B. Tourancheau, and J.-J. Roux (2011). The TheLMA project: Multi-GPU implementation of the lattice Boltzmann method. *International Journal of High Performance Computing Applications*:1094342011414745.

Obukhov, A. M. (1941a). On the distribution of energy in the spectrum of turbulent flow. *Dokl. Akad. Nauk SSSR* 32 (1):22–24.

— (1941b). Spectral energy distribution in a turbulent flow. *Izv. Akad. Nauk SSSR Ser. Geogr. Geofiz.* 5 (4-5):453–466.

Olbers, D. J., J. Willebrand, and C. Eden (2012). *Ocean dynamics*. Springer, p. 704. ISBN: 9783642234491.

Olla, S., S. R. S. Varadhan, and H.-T. Yau (1993). Hydrodynamical limit for a Hamiltonian system with weak noise. *Communications in Mathematical Physics* 155 (3):523–560.

Orszag, S. A. (1969). Numerical Methods for the Simulation of Turbulence. *Physics of Fluids* 12 (12):II–250.

— (1971). Numerical simulation of incompressible flows within simple boundaries: accuracy. *Journal of Fluid Mechanics* 49 (01):75–112.

Parr, T. (2013). *The definitive ANTLR 4 reference*. Pragmatic Bookshelf. ISBN: 978-1-93435-699-9.

Pedlosky, J. (1987). *Geophysical Fluid Dynamics*. New York, NY: Springer New York. ISBN: 978-0-387-96387-7.

— (1996). *Ocean circulation theory*. Springer, p. 453. ISBN: 9783662032046.

Peng, Y., W. Liao, L.-S. Luo, and L.-P. Wang (2010). Comparison of the lattice Boltzmann and pseudo-spectral methods for decaying turbulence: Low-order statistics. *Computers & Fluids* 39 (4):568–591.

Pope, S. B. (2000). *Turbulent flows*. English. Cambridge; New York: Cambridge University Press. ISBN: 0521591252 9780521591256 0521598869 9780521598866.

Qian, Y. H., D. D'Humières, and P. Lallemand (1992). Lattice BGK Models for Navier-Stokes Equation. *Europhysics Letters* 17 (6):479–484.

Rayleigh, Lord (1916). LIX. On convection currents in a horizontal layer of fluid, when the higher temperature is on the under side. *Philosophical Magazine Series 6* 32 (192):529–546.

Reid, W. H. and D. L. Harris (1958). Some Further Results on the Bénard Problem. *Physics of Fluids* 1 (2):102.

Reynolds, O. (1883). An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Proc. R. Soc. Lond.* 35:84–99.

Richardson, L. F. (1922). *Weather Prediction by Numerical Process*. Cambridge University Press.

Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 43 (2):357–372.

Ronchi, C., R. Iacono, and P. S. Paolucci (1996). The "cubed sphere": a new method for the solution of partial differential equations in spherical geometry. *Journal of Computational Physics* 124 (1):93–114.

Sakievich, P. J., Y. T. Peet, and R. J. Adrian (2016). Large-scale thermal motions of turbulent Rayleigh-Bénard convection in a wide aspect-ratio cylindrical domain. *International Journal of Heat and Fluid Flow* 61:183–196.

Salmon, R. (1999). The lattice Boltzmann method as a basis for ocean circulation modeling. *Journal of Marine Research* 57 (3):503–535.

— (2009). An Ocean Circulation Model Based on Operator-Splitting, Hamiltonian Brackets, and the Inclusion of Sound Waves. *Journal of Physical Oceanography* 39 (7):1615–1633.

Shankar, D., J. P. McCreary, W. Han, and S. R. Shetye (1996). Dynamics of the East India Coastal Current: 1. Analytic solutions forced by interior Ekman pumping and local alongshore winds. *Journal of Geophysical Research: Oceans* 101 (C6):13975–13991.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations. I: The basic eperiment. *Monthly Weather Review* 91 (3):99–164.

Stevens, B. et al. (2013). Atmospheric component of the MPI-M Earth System Model: ECHAM6. *Journal of Advances in Modeling Earth Systems* 5 (2):146–172.

Stommel, H. (1948). The westward intensification of wind-driven ocean currents. *Transactions, American Geophysical Union* 29 (2):202.

Strumpen, V. and M. Frigo (2006). *Software engineering aspects of cache oblivious stencil computations*. Tech. rep. Citeseer.

Succi, S. (2001). *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford University Press. ISBN: 9780198503989.

Sukop, M. C. and D. T. Thorne (2006). *Lattice Boltzmann modeling: an introduction for geoscientists and engineers*. English. Berlin [u.a.]: Springer. ISBN: 3540279814 9783540279815 9783642066252 3642066259.

Sverdrup, H. U. (1947). Wind-Driven Currents in a Baroclinic Ocean; with Application to the Equatorial Currents of the Eastern Pacific. *Proceedings of the National Academy of Sciences of the United States of America* 33 (11):318–26.

Tang, Y., R. A. Chowdhury, B. C. Kuszmaul, C.-K. Luk, and C. E. Leiserson (2011). The Pochoir stencil compiler, p. 117–128.

Taylor, G. I. (1935). Statistical theory of turbulence. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 151 (873):421–444.

Tian, F.-B., H. Luo, L. Zhu, J. C. Liao, and X.-Y. Lu (2011). An efficient immersed boundary-lattice Boltzmann method for the hydrodynamic interaction of elastic filaments. *Journal of Computational Physics* 230 (19):7266–7283.

Trevorrow, A. and T. Rokicki (2009). Golly: open source, cross-platform application for exploring Conways Game of Life and other cellular automata.

Veronis, G. (1966). Wind-driven ocean circulation – Part 2. Numerical solutions of the non-linear problem. *Deep Sea Research and Oceanographic Abstracts* 13 (1):31–55.

Wallace, J. M. and P. V. Vukoslavčević (2010). Measurement of the Velocity Gradient Tensor in Turbulent Flows. *Annual Review of Fluid Mechanics* 42 (1):157–181.

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

Watari, M. (2012). What is the Small Parameter $\epsilon$ in the Chapman-Enskog Expansion of the Lattice Boltzmann Method? *Journal of Fluids Engineering* 134 (1):011401–011401.

Wellein, G., G. Hager, T. Zeiser, M. Wittmann, and H. Fehske (2009). Efficient temporal blocking for stencil computations by multicore-aware wavefront parallelization. *2009 33rd Annual IEEE International Computer Software and Applications Conference*. IEEE, p. 579–586. ISBN: 978-0-7695-3726-9.

Wilcox, D. C. (2006). *Turbulence modeling for CFD*. English. DCW Industries. ISBN: 9781928729082 1928729088.

Willis, G. E. and J. W. Deardorff (1970). The oscillatory motions of Rayleigh convection. *Journal of Fluid Mechanics* 44 (04):661.

Wittmann, M., T. Zeiser, G. Hager, and G. Wellein (2013). Comparison of different propagation steps for lattice Boltzmann methods. *Computers & Mathematics with Applications* 65 (6):924–935.

Wolf-Gladrow, D. A. (2000). *Lattice-gas cellular automata and lattice Boltzmann models: an introduction*. English. New York: Springer. ISBN: 3540669736 9783540669739.

Wolfram, S. (1986). Cellular automaton fluids 1: Basic theory. *Journal of Statistical Physics* 45 (3-4):471–526.

Yoshida, H. and M. Nagaoka (2010). Multiple-relaxation-time lattice Boltzmann model for the convection and anisotropic diffusion equation. *Journal of Computational Physics* 229 (20):7774–7795.

Yvon, J. (1935). *La théorie statistique des fluides et l'équation d'état*. Vol. 203. Hermann & cie.

Zhang, J. (2011). Lattice Boltzmann method for microfluidics: Models and applications.

Zhou, J. G. (2012). MRT rectangular lattice Boltzmann method. *International Journal of Modern Physics C* 23 (05):1250040.

Zong, Y., C. Peng, Z. Guo, and L.-P. Wang (2016). Designing correct fluid hydrodynamics on a rectangular grid using MRT lattice Boltzmann approach. *Computers & Mathematics with Applications* 72 (2):288–310.

Zwanzig, R. (1960). Ensemble Method in the Theory of Irreversibility. *The Journal of Chemical Physics* 33 (5):1338.

— (1980). Problems in nonlinear transport theory. *Systems far from equilibrium*.

van Engelen, R. A. (2001). ATMOL: A domain-specific language for atmospheric modeling. *CIT. Journal of computing and information technology* 9 (4):289–303.

## ERKLÄRUNG / DECLARATION

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulsѕige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- oder Beratungsdiensten (Promotionsberaterinnen oder Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die in Zussamenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungbehörde vorgelegt.

*Bremen, 10. März, 2018*

Dragos Bogdan Chirila