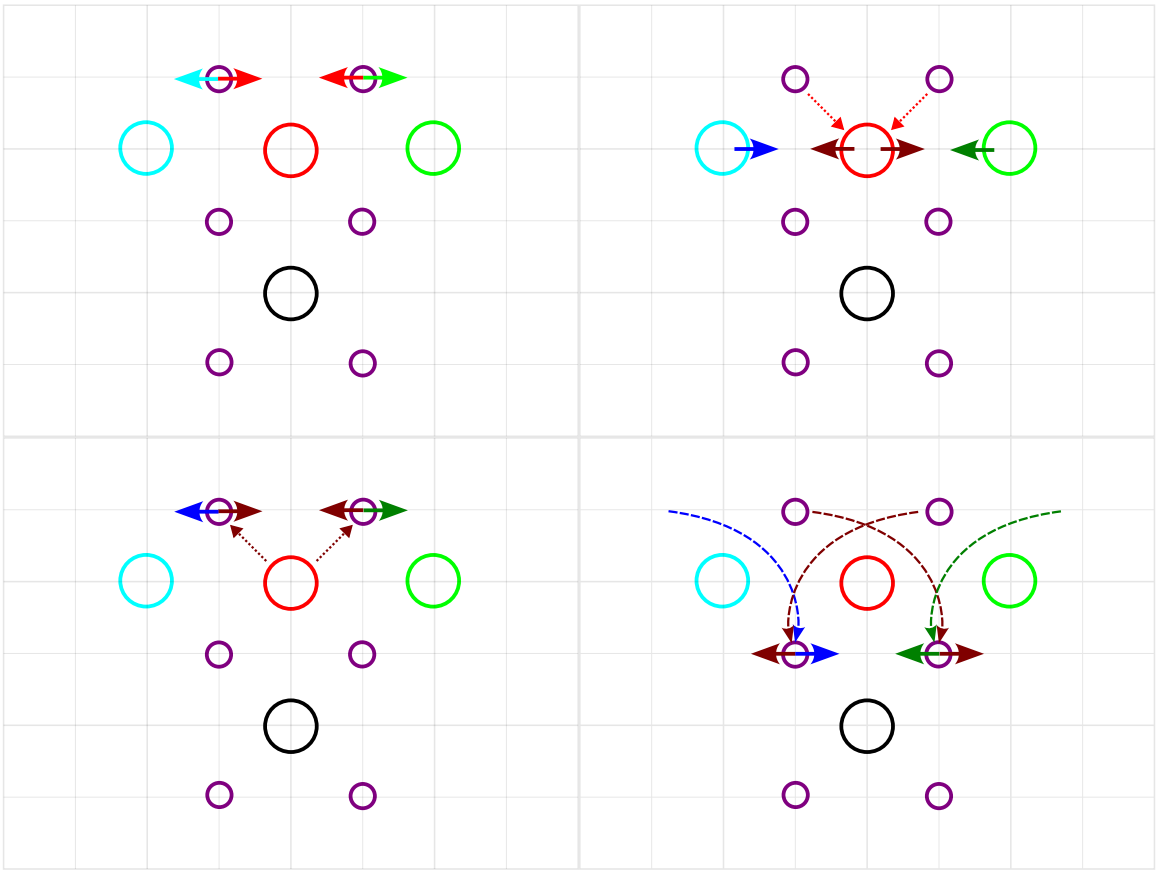

Deriving Filter Matrix Methods for Lattice Boltzmann Simulations

Efficient Simulations of the Convection Diffusion Equation and Navier Stokes Equations

Master's thesis in Applied Mechanics

Viktor Sundström

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

Master's Thesis in Applied Mechanics

Deriving Filter Matrix Methods for Lattice Boltzmann Simulations

Efficient Simulations of the Convection Diffusion Equation and Navier Stokes Equations

Viktor Sundström

CHALMERS
UNIVERSITY OF TECHNOLOGY

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

Deriving Filter Matrix Methods for Lattice Boltzmann Simulations

Efficient Simulations of the Convection Diffusion Equation and Navier Stokes Equations
Viktor Sundström

Supervisor: Martin Rohde, Department of Radiation Science and Technology, Delft University of Technology
Examiner: Lars Davidson, Department of Mechanics and Maritime Sciences, Division of Fluid Dynamics, Chalmers University of Technology

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

Cover: The stream and collide algorithm applied to a staggered grid.

Typeset in LaTeX
Gothenburg, Sweden 2025

Deriving Filter Matrix Methods for Lattice Boltzmann Simulations
Efficient Simulations of the Convection Diffusion Equation and Navier Stokes Equations

Viktor Sundström
Department of Mechanics and Maritime Sciences
Division of Division of Fluid Dynamics
Chalmers University of Technology

Abstract

The Lattice Boltzmann Method (LBM) is an increasingly popular method for simulating fluid flows, which through its mesoscopic and explicit formulation allows for effective parallelization. The choice of a so called collisional operator largely determines the stability and accuracy of LBM simulations. Linear operators such as the Bhatnagar-Gross-Krook operator (BGK) and the Multiple-Relaxation-Times (MRT) both have the fundamental problem of defining a relaxation time, which is a tradeoff between numerical stability and range of flow types that can be simulated. The Filter Matrix method, originally proposed by Somers in 1993 and extended by Zhuo et al. in 2012, is a non-linear collisional operator which removes the need tune relaxation times. However, its derivation and assumptions are not well documented. In this thesis, the Filter Matrix formulation is derived through classical Chapman–Enskog analysis with the BGK operator, although using some arguments which is hard to justify for general flow cases. Observations from this analysis leads to a much more general formulation with Filter Matrix method seen as a collisional operator defined in a moment space. The discretized Boltzmann equation is mapped to the moment space using a linear projection similar to what is used for MRT schemes. This allows for a Chapman-Enskog-like analysis in moment space leading to a Filter Matrix formulation that is very similar, although not identical to previous literature. The new formulation is derived without neglecting any higher order hydrodynamic modes appearing throughout the derivation, which was done extensively in previous derivations. Turning the Filter Matrix formulation of this thesis into a computational procedure requires some closure laws, similarly to what was used in previous derivations. How to formulate such closure laws is illustrated by analyzing how the $k - \varepsilon$ equations can be solved within the FMLB framework. The study is purely mathematical, and as such numerical validation would be a starting point for further work.

Keywords: Lattice Boltzmann Method, Filter Matrix Method, Chapman–Enskog expansion, Multiple Relaxation Time operator

Preface

This report presents the outcome of a master's thesis project carried out at the Department of Radiation Science and Technology at Delft University of Technology during the spring of 2025, with the Department of Mechanics and Maritime Sciences at Chalmers University of Technology being responsible for the examination of the project. It was made possible by the IDEA League Student Grant, a collaboration between, Chalmers University of Technology and Delft University of Technology as well as several other institutions around Europe.

Acknowledgements

I have felt really welcome at the RPNM department and would like to extend my gratitude to all my colleagues for making my time in Delft so enjoyable. I want to thank my supervisor Martin Rohde for letting me do the thesis in his group and for all the hours we have spend discussing minor assumptions made in the derivation of the filter matrix method. I have truly enjoyed every second of it. On the same theme, I am very grateful for going through my entire master program together with Pontus Nilsson, and especially for all our early mornings with "LBokklubbM", our self proclaimed book circle for discussing all things Lattice Boltzmann. I also want to thank Lars Davidson, not only for examining this project, but also for all the opportunities he has opened up for me throughout my studies. A special thanks goes to Christophe Duwig, who took me in for an internship in his group at KTH back when I was still in high school. His encouragement throughout my studies since then have meant a lot, and it was on his recommendation I applied to do my thesis in Martins group. Lastly, I want to thank everyone who have patiently helped me with all logistics surrounding this project.

Viktor Sundström, Delft, June 2025

List of Acronyms

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

BGK	Bhatnagar-Gross-Krook
DNS	Direct Numerical Simulation
DDF	Double Distribution Function
FMLB	Filter Matrix Lattice Boltzmann
LBM	Lattice Boltzmann Method
LES	Large Eddy Simulation
MSFR	Molten Salt Fast Reactor
MPB	Moment Projected Boltzmann
MRT	Multiple-Relaxation-Times
PDE	Partial Differential Equation
RANS	Reynolds Average Navier Stokes

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis. Multiple usage of similar symbols separated with commas.

Indices

$\alpha, \beta, \gamma, \delta, \alpha', \beta'$	Indices for spatial coordinates
	0 corresponds to zeroth order Hermite polynomials
	1α corresponds to first order Hermite polynomials
	$2\alpha\beta$ corresponds to second order Hermite polynomials
$\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta}, \tilde{\alpha}', \tilde{\beta}'$	Indices for spatial with symmetry
	$2\tilde{\alpha}\tilde{\beta}$ corresponds to second order symmetric Hermite polynomials
i,j	Indices for discretized velocity directions, Hermite polynomial indices
k	Index for moment vector α, hermite polynomial index
ϕ	quantity associated with a transported scalar
ρ	quantity associated with momentum
t	quantity associated with time
scale	Scale parameter to change to lattice units

Discretized space

$\mathcal{H}$	Scaled Hermite polynomial
$\mathscr{H}$	Non-scaled Hermite polynomials
$c_{i\alpha}$	Discretized velocity vectors
E	Transformation matrix / Error term in Chapman–Enskog expansion
D	Corresponding gram matrix to E_{ik}

Δt	Time discretization step
α	Moment vector
c_s	Lattice speed of sound
Ω	Collisional operator
$\mho$	Collisional operator in moment space
ω	Lattice weights
f	Distribution function
τ	Relaxation time, subgrid scale stresses
$\mathcal{T}$	Change of basis transformation
δ	Kronecker delta
Q	Discretized source term
$\mathcal{F}$	Discretized force term
Π	Tensor in Chapman–Enskog analysis
Θ	Tensor in Chapman–Enskog analysis
m_i	Velocity weights for 24-speed velocity set

Macroscopic quantities

u_α	Velocity vector
k	Turbulent kinetic energy
ε	Turbulent dissipation rate
q	Source term per unit volume
F	Force term per unit volume, fourth order term
D	Diffusion coefficient, Diffusion term
ν	Viscosity
ν_t	Turbulent viscosity
ζ	Bulk viscosity
T	Temperature, third order terms
C	Convection term, contribution to macroscopic collision operator
P	Production term
G	Gravitational term
Φ	Destruction term
$s_{\alpha\beta}$	Strain rate tensor
g	Gravity term

β	Thermal expansion coefficient
S	Magnitude of strain rate tensor, second order terms
p	Pressure, momentum
ξ	Particle velocities
σ	Subgrid scalar fluxes

Dimensionless numbers

σ_T	Turbulent Prandtl number
ϵ	Scale separation in Chapman–Enskog analysis

Contents

List of Acronyms ix

Nomenclature xi

1 Introduction 1
- 1.1 Purpose and goals . . . 2
- 1.2 Limitations . . . 2

2 Theory 3
- 2.1 Fluid mechanics preliminaries . . . 3
- 2.2 Turbulence modelling . . . 4
- 2.3 Lattice Boltzmann Method . . . 6
  - 2.3.1 Chapman–Enskog analysis . . . 7
  - 2.3.2 Velocity sets . . . 8
  - 2.3.3 Coupled equations in LBM . . . 9
  - 2.3.4 Non-dimensionalization . . . 9
- 2.4 Hermite expansions . . . 10
- 2.5 Multiple Relaxation Time collisional operators . . . 11
- 2.6 The Filter Matrix Lattice Boltzmann Method . . . 12
  - 2.6.1 Derivation of the Filter Matrix Method . . . 14
- 2.7 Turbulence modelling in LBM . . . 15

3 Chapman–Enskog analysis of convection–diffusion equation 17
- 3.1 Staggering the discretized Boltzmann equation . . . 17
- 3.2 Chapman–Enskog expansion of a staggered formulation BGK formulation of the convection diffusion equation . . . 18
  - 3.2.1 Taylor expansion and scale separation . . . 19
  - 3.2.2 Summing into macroscopic properties . . . 20
  - 3.2.3 Isolating error term . . . 22
  - 3.2.4 Momentum substitution . . . 23
- 3.3 The collisional operator for BGK in macroscopic variables . . . 25
  - 3.3.1 Time derivatives . . . 25
  - 3.3.2 Spatial derivative . . . 27
  - 3.3.3 Source discretization . . . 27
  - 3.3.4 Resubstituting in BGK . . . 27
- 3.4 Determining unneeded components . . . 28
- 3.5 Filtering the collisional operator . . . 29

3.6 Constructing the moment vector 30

4 Derivation of Filter Matrix Methods **33**
4.1 Derivation of transformation matrix for a general lattice 33
4.2 Transforming the Boltzmann equation to moment space 37
4.3 General Filter Matrix algorithm 38
4.4 Deriving FMLB for the convection–diffusion equation 40
4.4.1 Computational procedure 42
4.5 Navier Stokes equations 43
4.5.1 The continuity equation 43
4.5.2 Momentum equation 44
4.5.2.1 Constant density case 46
4.6 Filter Matrix for $\boldsymbol{k-\varepsilon}$ equations 46

5 Comparison to Other Derivations **49**
5.1 Comparison with Zhuo et al. 49
5.1.1 Comparison with Somers 52

6 Discussion and conclusion **55**
6.1 Implication for minimal lattice 55
6.2 Name of the algorithm 55
6.3 Further work 56
6.4 Conclusion 56

Bibliography **57**

A Elaboration on used syntax **I**
A.1 Second order products I
A.2 Computations in original equilibrium distribution II

B Inner products of scaled Hermite basis **V**

C Transforming spatial derivatives to moment space **IX**

D Supplied code **XIII**

1

Introduction

Climate change is a great challenge for humanity going forward. There are many causes of climate change with energy generation being a significant source of greenhouse gas emissions. The causes and consequences of climate change are extensively investigated in the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [1]. A simple solution to decrease climate sector would be to reduce the energy consumption. However, this has proven unfeasible and the International Energy Agency concludes that energy demand is increasing at an increased rate due to greater use of cooling during heat waves, electrification of the transport sector, development of new industry and a significant expansion of the amount of data centers globally [2]. Nuclear power is an energy source with low green house gas footprint and is already an important part of the energy mix. The International Energy Agency mentions new construction of nuclear power plants as one of the ways the increased energy demand is met. Through the European project ENDURANCE, the next generation of nuclear power plants are being developed. The chosen design is a Molten Salt Fast Reactor (MSFR) in which the nuclear material is dissolved in a liquid state [3]. This design is advantageous due to inherent safety features and ease of disposable of burnt out fuel. However, it creates significant challenges in fluid modelling as outlined by the International Atomic Energy Agency in the report *Challenges Related to the Use of Liquid Metal and Molten Salt Coolants in Advanced Reactors* [4].

The *Transport Phenomena in Nuclear Applications* at TU Delft has extensively experimented with the Lattice Boltzmann Method (hereon LBM) for simulating fluid flows in a reactor context and specifically for some of the flow features associated with MSFR [5, 6, 7, 8]. One of the main advantages of a LBM approach is its ease of parallelization, as investigated by reference [7] and [5]. Through a Double Distribution Function (DDF) thermodynamic properties and neutronics can implemented as shown in reference [6, 8, 9]. In many of these implementations, the Filter Matrix Lattice Boltzmann (hereon FMLB) method is used. The method has shown promising results, although due to the limited amount of studies on the method, there is some difficulty in extending the method beyond the limitations of current literature.

1.1 Purpose and goals

The purpose of this thesis is to thoroughly investigate the origins of the Filter Matrix method in order for the *Transport Phenomena in Nuclear Applications* lab to be able to deepen the development of simulation methods for Molten Salt Fast Reactors and alike. Due to the extensive use of the FMLB method by the lab, this work could potentially allow the lab to understand the numerical limitations and more easily integrate more multiphysics into the FMLB formalism.

The goal of this thesis is to theoretically derive the FMLB method presented by Zhou et al. [10] and use this derivation to find ways to make it computationally cheaper and applicable to simulate a wider range of flow types. Specifically, the following is sought

1. Find E_{ki}, E_{ik} for a general lattice Boltzmann scheme DdQq.
2. Extend the formulation to allow the bulk viscosity to be varied independently from the viscosity.
3. Formulate the FMLB for the $k - \varepsilon$ equations.

1.2 Limitations

This work is limited to investigating the theoretical and mathematical origins of the Filter Matrix Method. Previous studies have proven the efficiency and applicability of the method. However, because some the final formulation differs slightly from current literature, a numerical investigation is the logical continuation of this project.

2

Theory

In this chapter the macroscopic equations governing fluid mechanics are introduced together with some turbulence modelling. Thereafter, the mesoscopic formulation for fluid mechanics through the Boltzmann equation is shown. Lastly, simulation techniques and current gaps in literature surround lattice Boltzmann simulation are discussed.

2.1 Fluid mechanics preliminaries

Fluid mechanical phenomena are governed and modelled by several Partial Differential Equations (PDE). The central equations in fluid mechanics are the Navier Stokes equations, which are composed of the continuity equation and the momentum equations. The continuity equation is given by

$$\partial_t \rho + \partial_\alpha(\rho u_\alpha) = 0, \qquad \text{(Continuity)}$$

where ρ is the density and u_α the fluid velocity. The continuity equation represents mass conservation The momentum equations are formulated for each spatial dimension, which represent momentum conservation in each velocity direction,

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = -\partial_\alpha p + \partial_\beta\left(\rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\delta_{\alpha\beta}\partial_\gamma u_\gamma\right) + \delta_{\alpha\beta}\zeta\partial_\gamma u_\gamma\right) + F_\alpha \qquad \text{(Momentum)}$$

where p is the pressure, F_α a body force (i.e. having the units of density times acceleration), ν is the kinematic viscosity and ζ bulk viscosity [11, 12, 13][1].

The energy and temperature of the fluid needs to be modelled in other ways. A similar energy equation to the Navier Stokes equations could be formulated for energy, although a simpler way is to treat the fluid temperature as a passive scalar and model it through the convection–diffusion equation

$$\underbrace{\partial_t\phi + \partial_\alpha(\phi u_\alpha)}_{C^\phi} = \underbrace{\partial_\alpha\left(D_\phi\partial_\alpha\phi\right)}_{D^\phi} + q_\phi \qquad \text{(Convection diffusion equation)}$$

[1] There is some nuance to the factor 2/3 as it is related to a linear stress constitutive equation, which differs between two and three dimensions. However, the shape of the equation in terms of derivatives is the same but the bulk viscosity ζ would need to be redefined in terms of the Lamé parameters [14]. This thesis will only investigate the form of the differential equation and this nuance will not be taken into account in the final formulations.

where the passive scalar would be set to the temperature, i.e. $\phi = T$, and diffusion coefficient D_ϕ to the thermal diffusivity [12]. However, eq. (Convection diffusion equation) could be used to model several other physical phenomena as well, such as neutron transport [15]. This setup has a one way coupling from the momentum equation to passive scalar, meaning that the scalar will not change the flow field. This is not necessarily realistic, especially in the case of temperature, which due to density changes creates natural convection. One way to introduce the coupling from the thermal convection–diffusion equation (heat equation) to the momentum equation is to model the body force based on the temperature field. One way to do this is through the Boussinesq approximation which is defined as

$$F_T = (\rho - \rho_0) g_\alpha = -\rho_0 \beta (\bar{T} - T_0) g_\alpha \qquad \text{(Boussinesq approximation)}$$

with ρ_0 being the density at some reference temperature T_0, g_α the gravitational acceleration and β the thermal expansion coefficient [11, 12].

2.2 Turbulence modelling

The Navier Stokes equation becomes increasingly computationally expansive with increasing Reynolds numbers, defined as the ratio between inertial and viscous forces. This makes solving the discretized Navier Stokes equation, called Direct Numerical Simulation (DNS), computationally infeasible. This is due to turbulent phenomena, which could to be modelled separately as a way to decrease the computational demand. Two of the most common approaches are Large Eddy Simulations (LES) and Reynolds Averaged Navier-Stokes (RANS) approaches. RANS will be the focus in this report. The main principle is to time average the Navier-Stokes equations. In the incompressible case with the boussinesq approximation, this leads to

$$\partial_\alpha \bar{u} = 0 \qquad \text{(RANS continuity)}$$

and

$$\partial_t \left(\rho_0 \bar{u}_\alpha\right) + \partial_\beta \left(\rho_0 \bar{u}_\alpha \bar{u}_\beta\right) = -\partial_\alpha \bar{p} + \rho_0 \nu \partial_{\beta\beta} \bar{u}_\alpha - \partial_\beta \tau_{\alpha\beta} - \beta \rho_0 \left(\bar{T} - T_0\right) g_\alpha, \qquad \text{(RANS momentum)}$$

where the bar indicates an averaged property. The difficult part lies in modelling Reynolds stress that appeared $\tau_{\alpha\beta} = \rho_0 \overline{u'_\alpha u'_\beta}$, where u'_α is the instantaneous velocity. Several models exists, most commonly eddy viscosity models. [16, 11]

One of the most widely RANS models for industrial applications is the two equation $k - \varepsilon$-model. In this model, three important quantities, turbulent kinetic energy k, turbulent viscous dissipation ε and turbulent viscosity ν_t are defined as

$$k = \frac{1}{2}\overline{u'_\alpha u'_\alpha}, \qquad \varepsilon = \nu \overline{(\partial_\beta u'_\alpha)(\partial_\beta u'_\alpha)}, \qquad \nu_t = C_\mu \frac{k^2}{\varepsilon}. \tag{2.1}$$

Through the Boussinesq assumption, the stress $\partial_\beta \tau_{\alpha\beta}$ in eq. (RANS momentum) is incorporated into the viscosity, by changing the term $\rho_0 \nu \partial_{\beta\beta} \bar{u}_\alpha$ to $\rho_0 (\nu_0 + \nu_t) \partial_{\beta\beta} \bar{u}_\alpha$. By modelling k and ε, the turbulent viscosity can be calculated. The transport

equations for k and ε also needs to be averaged, with assumptions and derivation found in Davidson [11], leading to incompressible[2] modelled $k-\varepsilon$ transport equation. For k this is

$$\overbrace{\partial_t k + \bar{u}_\beta \partial_\beta k}^{C^k} = \overbrace{\nu_t \left(\partial_\beta \bar{u}_\alpha + \partial_\alpha \bar{u}_\beta\right) \partial_\beta \bar{u}_\alpha}^{P^k} + \overbrace{g_\alpha \beta \frac{\nu_t}{\sigma_T} \partial_\alpha \bar{T}}^{G^k} - \overbrace{\varepsilon}^{\Phi^k} + \underbrace{\partial_\beta \left(\left(\nu + \frac{\nu_t}{\sigma_k} \right) \partial_\beta k \right)}_{D^k} \tag{k transport}$$

which is of the general form

$$C^\phi = P^\phi + D^\phi + G^\phi - \Phi^\phi, \tag{2.2}$$

with Convective term, Production, Diffusion, Gravity induced bounacy and destruction term Φ. In this case the turbulent dissipation ε is the destruction term for turbulent kinetic energy, $\Phi^k = \varepsilon$. For the purpose of solving the equation this may be further collected to one effective source term $q_k = P^k + G^k - \Phi^k$ [11, 16]. The destruction term ε is furthered modelled with another transport equation

$$\overbrace{\frac{\partial \varepsilon}{\partial t} + \bar{u}_\beta \partial_\beta \varepsilon}^{C^\varepsilon} = \overbrace{\frac{\varepsilon}{k} c_{\varepsilon 1} \nu_t \left(\partial_\beta \bar{u}_\alpha + \partial_\alpha \bar{u}_\beta\right) \partial_\beta \bar{u}_\alpha}^{P^\varepsilon} + \underbrace{c_{\varepsilon 1} g_\alpha \frac{\varepsilon}{k} \frac{\nu_t}{\sigma_T} \partial_\alpha \bar{T}}_{G^\varepsilon} - \underbrace{c_{\varepsilon 2} \frac{\varepsilon^2}{k}}_{\Phi^\varepsilon} + \underbrace{\partial_\beta \left(\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \partial_\beta \varepsilon \right)}_{D^\varepsilon}, \tag{ε transport}$$

where it is possible to write an effective source term as $q_\varepsilon = P^\varepsilon + G^\varepsilon - \Phi^\varepsilon$. In this case there is a set of constants defined as

$$(c_\mu, c_{\varepsilon 1}, c_{\varepsilon 2}, \sigma_k, \sigma_\varepsilon) = (0.09, 1.44, 1.92, 1, 1.3), \tag{2.3}$$

and σ_T being the turbulent Prandtl number which depends on the flow situation, but is usually set a value between 0.7 and 0.9. The stress defined through the strain rate $s_{\alpha\beta}$ can be defined as

$$\nu_t |S|^2 = 2\nu_t \bar{s}_{\alpha\beta} \bar{s}_{\alpha\beta} = \nu_t \left(\partial_\beta \bar{u}_\alpha + \partial_\alpha \bar{u}_\beta\right) \partial_\beta \bar{u}_\alpha, \tag{2.4}$$

with the general strain rate tensor

$$s_{\alpha\beta} = \frac{1}{2} \left(\partial_\alpha u_\beta + \partial_\beta u_\alpha\right) \tag{2.5}$$

For the rest of this report the $k-\varepsilon$ equations will be treated as a convection diffusion equation with a source term. The terms in eqs. (k transport) and (ε transport) are are identified as the corresponding convection–diffusion equation properties as[3]

$$q_k = C_\mu \frac{k^2}{\varepsilon} |S|^2 + g_\alpha \beta C_\mu \frac{k^2}{\varepsilon \sigma_T} \partial_\alpha \overline{T} - \varepsilon \tag{q_k}$$

[2] Compressible versions of the transport equations can be found in Versteeg & Malalasekera [13] and Wilcox [16].

[3] Note that k here refers to the turbulent kinetic energy and not a tensor index

and

$$q_\varepsilon = C_\mu c_{\varepsilon 1} k |S|^2 - g_\alpha \beta C_\mu \frac{k}{\sigma_T} \partial_\alpha \overline{T} - c_{\varepsilon 2} \frac{\varepsilon^2}{k}. \tag{q_ε}$$

The diffusion coefficients can be defined similarly

$$D_k = \nu_0 + C_\mu \frac{k^2}{\varepsilon \sigma_k} \tag{D_k}$$

and

$$D_\varepsilon = \nu_0 + C_\mu \frac{k^2}{\varepsilon \sigma_\varepsilon}. \tag{D_ε}$$

2.3 Lattice Boltzmann Method

The Lattice Boltzmann method is an alternative approach to simulate the Navier Stokes equations, with its origins in statistical physics[4]. The basis for the simulation method a mesoscopic description of a moving fluid. That is, the effect of particle–particle interactions are studied. Note the difference to the Navier-Stokes equations, which are macroscopic meaning that macroscopic effects such as stresses are modelled for the flow. It is also distinct from a microscopic formulation, in which individual particle–particle interactions are studied, the mesoscopic formulation looks at the statistics of such interactions. The governing equation is the Boltzmann equation, given by

$$\partial_t f + \xi_\alpha \partial_\alpha f + \frac{F_\alpha}{\rho} \partial_{\xi_\alpha} f_i = \Omega(f) \tag{Boltzmann}$$

with particle distribution function (hereon distribution function) f, collisional operator Ω, particle velocities ξ. The distribution function is defined such that moments, i.e. integrals over different powers of ξ, equals various macroscopic properties of the flow, such as density and momentum. In the continuous case it can be shown that eq. (Boltzmann) is equivalent to solving the Navier-Stokes equations [12].

Because the particle–particle interactions are local, in a way that stresses are not, any attempts to solve eq. (Boltzmann) would be a local endeavour, making it fundamentally suitable for parallelization. This leads to the Lattice Boltzmann Method (LBM), which discretizes eq. (Boltzmann) into a regular lattice. The particle velocities are discretized into a finite set $c_{i\alpha}$, which determines the directions of possible particle–particle interactions, with the distribution function in each direction given by f_i. The particle–particle interaction can be seen as a collision, which determined by the distribution function in each direction as $\Omega(f_i)$. The collisional operator can be defined in multiple different ways, with one of the most common being the Bhatnagar-Gross-Krook operator (BGK),

$$\Omega_{\mathrm{BGK},i} = -\frac{1}{\tau}(f_i(x_\alpha) - f_i^{\mathrm{eq}}(x_\alpha)) \tag{2.6}$$

[4]The description of LBM given here is very brief, the reader is referred to Krüger et al. [12] for details.

which defines an equilibrium distribution as

$$f_{\phi,i}^{\text{eq}} = \phi\omega_i \odot \left(1 + u_\alpha \frac{c_{i\alpha}}{c_s^2} + u_\alpha u_\beta \left(\frac{1}{2c_s^4}\left(c_{i\alpha} \odot c_{i\beta} - c_s^2\delta_{\alpha\beta}\right)\right)\right), \tag{2.7}$$

where the notation is explained in appendix A. The interpretation of eq. (2.6) is that the difference between the current distribution and the equilibrium determines how large the effect of the collision will be. This is rescaled with a relaxation time τ that represent how quickly the distribution is relaxed back to its equilibrium. the equation is now solved through the *stream and collide algorithm*, with the most important steps being

Stream and collide algorithm

1. **Collide** Updated the distribution in all direction through the collisional operator

$$f_i^\star(x_\alpha, t) = f_i(x_\alpha, t) + \Delta t\Omega(f_i) \tag{2.8}$$

2. **Stream** Move the updated collisions to neighboring nodes in the direction corresponding to the component of the distribution function

$$f_i(x_\alpha + c_{i\alpha}\Delta t, t + \Delta t) = f_i^\star(x_\alpha, t). \tag{2.9}$$

The convergence of the algorithm depend heavily on the choice of the relaxation time τ, as it is related to the physical viscosity and resolution as in [12],

$$\nu = c_s^2\left(\tau - \frac{\Delta t}{2}\right) \implies \tau = \frac{\nu}{c_s^2} + \frac{\Delta t}{2}, \tag{2.10}$$

there are some alternative methods to chose the relaxation time as discussed in Wolf [17].

2.3.1 Chapman–Enskog analysis

When eq. (Boltzmann) is discretized it must be shown that the computational procedure is equivalent to solving the Navier Stokes equations. This is done through an analysis method called Chapman–Enskog analysis, originating from lattice gas treatments [18, 19]. It involves separating the derivatives in the discretized Lattice Boltzmann equation into scales, which can be related using the moments of the distribution function. One such analysis is performed in chapter 3. There is one important detail with the Chapman–Enskog analysis, the discretized Boltzmann equation is actually not exactly equivalent to the Navier Stokes equation, rather there are some extra terms which can be argued to be small, although it is not always the case. In Krüger et al. [12] a Chapman–Enskog analysis to show the equivalence of eq. (Boltzmann) and the Navier Stokes equations is presented as well as one for the convection–diffusion equation. To motivate the derivations done in this thesis, some of the nuances in the Chapman–Enskog analysis for the convection–diffusion equation will be discussed.

Krüger et al. [12] presents a derivation with a linear equilibrium and without any source term that has a corresponding error term

$$E = \partial_\alpha \left(\frac{D_\phi}{c_s^2} \partial_t \left(\phi u_\alpha \right) \right), \qquad \text{(Krüger et al. 8.35c)}$$

meaning that solving the discretized Boltzmann equation correspond to solving the convection diffusion equation equation with an additional term. Krüger et al. [12] notes that this term is not necessarily negligible and cannot accommodate large variations in the viscosity D_ϕ. Increasing to a second order equilibrium distribution instead yields

$$E = -\frac{D_\phi}{c_s^2} \partial_\alpha \left(\frac{\phi}{\rho} \partial_\alpha p \right). \qquad \text{(Krüger et al. 8.37)}$$

However, this term is still not always negligible, which is why several studies have found ways to suppress it, some of which are outlined in [20, 21]. One such method is to use the source term to absorb this error [22]. When using the convection–diffusion equation for modelling the $k-\varepsilon$-equations as in eqs. (D_k) and (D_ε), the viscosity will vary significantly, thus requiring a more advanced expansion than what is proposed by Krüger et al. [12]. The complexity in the source terms eqs. (q_k) and (q_ε) further complicates this.

2.3.2 Velocity sets

The discretization into lattice velocities is characterized with three important quantities, the lattice weights ω_i, lattice directions $c_{i\alpha}$ and lattice speed c_s [12, 23]. For the D2Q9 model the lattice speed is $c_s = c_\Delta/\sqrt{3}$ with $c_\Delta = \Delta x/\Delta t$ [5]

$$\omega_{\mathrm{D2Q9},i} = \begin{bmatrix} \frac{4}{9} & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} & \frac{1}{36} \end{bmatrix}^T \tag{2.11}$$

and

$$c_{\mathrm{D2Q9},i\alpha} = c_\Delta \overbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \\ -1 & 1 \\ -1 & -1 \\ 1 & -1 \end{bmatrix}}^{c_{ix}\downarrow \quad c_{iy}\downarrow}. \tag{2.12}$$

The lattice speed and weight factors come from the fact that the equations must satisfy the following relations [12, 17]

[5] In several commonly used models $c_s = c_{c_\Delta}/3$, there are however some models such as D3Q7 that has other lattice speed of sounds, e.g. $c_s = c_\Delta/2$ from [24]. It is also common to set $c_\Delta = 1$.

$$\sum_i \omega_i = 1 \qquad (\sum \omega)$$

$$\sum_i \omega_i c_{i\alpha} = 0 \qquad (\sum \omega c)$$

$$\sum_i \omega_i c_{i\alpha} c_{i\beta} = c_s^2 \delta_{\alpha\beta} \qquad (\sum \omega c^2)$$

$$\sum_i \omega_i c_{i\alpha} c_{i\beta} c_{i\gamma} = 0 \qquad (\sum \omega c^3)$$

$$\sum_i \omega_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} = c_s^4 \left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right). \qquad (\sum \omega c^4)$$

2.3.3 Coupled equations in LBM

As discussed in section 2.1, the energy, or temperature, of a fluid can be treated as a conserved quantity and thus can be incorporated into the moments of the distribution function. This approach is outlined in [25] and is referred to as Multi-Speed approach. However, [25] suggest using a so called Double Distribution Function (DDF) as more stable and accurate way of resolving the temperature evolution. The idea is to have one distribution function that resolves the mass and momentum conservation and another distribution function that models the temperature or energy evolution. This can be done in several ways, e.g. coupling the heat equation with the Navier Stokes equations through the Boussinesq approximation, split into two different distribution functions [25]. This makes implementing other coupled phenomena comparatively straight forward as another distribution function is simply introduced for all coupled equations, such as the neutron diffusion equation [9].

2.3.4 Non-dimensionalization

It is customary to formulate the lattice Boltzmann equations and simulation procedures in non-dimensional units. Typically the scales are defined through a velocity scale u_{scale}, length scale l_{scale} and density scale ρ_{scale}. These scales can then be combined to be used to normalize all components of the Boltzmann equation as per Krüger et al. [12]. When the non-dimensionalization is performed on the velocity $v_{i\alpha}^{\text{physical}}$ and particle velocity $\xi_{i\alpha}^{\text{physical}}$, the non-dimensionalized versions are given as

$$v_{i\alpha}^* = v_{i\alpha}^{\text{physical}} / u_{\text{scale}} \tag{2.13}$$

$$\xi_{i\alpha}^* = \xi_{i\alpha}^{\text{physical}} / u_{\text{scale}}, \tag{2.14}$$

where all other variables are rescaled likewise. Typically, as done in Krüger et al. [12], the $*$-notation is dropped and it is understood that all equations in a lattice Boltzmann context and non-dimensional. The velocities are however rescaled again with the *lattice speed of sound* c_s^*, defined in Shan et al. [23], finally resulting in

$$\xi_{i\alpha}^* = c_{i\alpha}^* / c_s^* \tag{2.15}$$

$$v_{i\alpha}^* = u_{i\alpha}^* / c_s^* \tag{2.16}$$

once again, dropping the $*$ notation for the lattice particle velocity $c^*_{i\alpha}$ and the lattice velocity $u^*_{i\alpha}$. Throughout this report all values are dimensionless unless otherwise mentioned.

2.4 Hermite expansions

In both the discrete and continuos case, the equilibrium distribution $f^{\text{eq}}_{\phi,i}$ can be expanded into a set of Hermite polynomials that correspond to the Maxwell Boltzmann distribution. Most commonly found in literature is the second order expansion eq. (2.7), as this is what is needed for a reconstruction of the Navier Stokes relation [23]. The Hermite polynomials can be derived mathematically as in [26], although the formulations from the literature must be modified to include the non-dimensionalization defined in section 2.3.4.

In the notation of Shan et al. [23], the first three Hermite polynomials are

$$\mathscr{H}^{(0)}_i(\xi) = 1, \tag{$\mathscr{H}^{(0)}$}$$
$$\mathscr{H}^{(1)}_i(\xi) = \xi_i, \tag{$\mathscr{H}^{(1)}$}$$
$$\mathscr{H}^{(2)}_{ij}(\xi) = \xi_i\xi_j - \delta_{ij}, \tag{$\mathscr{H}^{(2)}$}$$
$$\mathscr{H}^{(3)}_{ijk}(\xi) = \xi_i\xi_j\xi_j - \xi_i\delta_{jk} - \xi_j\delta_{ki} - \xi_k\delta_{ij}. \tag{$\mathscr{H}^{(3)}$}$$

which have the recursive definition[6]

$$\xi_i\mathscr{H}^{(n)}_{i_1,i_2,\ldots,i_n} = \mathscr{H}^{(n+1)}_{i,i_1,i_2,\cdots,i_n} + \sum_{k=1}^{n}\delta_{ii_k}\mathscr{H}^{(n-1)}_{i_1,i_2,i_{k-1},i_{k+1},\cdots,i_n}. \tag{2.17}$$

For the isothermal case (meaning $\theta = 1$ in the corresponding expressions in Krüger et al. [12] and Shan et al. [23]), the equilibrium moments, translated to the notation of this report, are given by

$$a^{(0)}_0 = \rho, \tag{2.18}$$
$$a^{(1)}_0 = \rho v^*_\alpha = \rho\frac{u^*_\alpha}{c_s}, \tag{2.19}$$
$$a^{(2)}_0 = \rho v^*_\alpha v^*_\beta = \rho\frac{u^*_\alpha u^*_\beta}{c_s^2}, \tag{2.20}$$
$$a^{(3)}_0 = \rho v^*_\alpha v^*_\beta v^*_\gamma = \rho\frac{u^*_\alpha u^*_\beta u^*_\gamma}{c_s^3}. \tag{2.21}$$

These expression are combined to a projection of the population density function onto the Hermite polynomials, as defined by Shan et al. [23]

$$f_{\phi,i}(x_\alpha,\xi_\alpha,t) \approx \omega(\xi_\alpha)\sum_{n=0}^{N}\frac{1}{n!}a^{(n)}(x_\alpha,t)\mathscr{H}^{(n)}_i(\xi_\alpha) \tag{2.22}$$

[6] This recursive definition is the reason calculations in appendix C simplify as it does.

Now, the notation is changed to be similar to what is used throughout Krüger et al. [12]. The indices i, j, k used by Shan et al. [23] for the Hermite polynomials are changed to $(i\alpha), (i\beta), (i\gamma)$ and replace with the rescaled lattice speeds from section 2.3.4

$$\begin{aligned}\mathcal{H}^{(3)}_{(i\alpha)(i\beta)(i\gamma)}(\xi) &= \xi_{i\alpha} \odot \xi_{i\beta} \odot \xi_{i\gamma} - \xi_{i\alpha}\delta_{\beta\gamma} - \xi_{i\beta}\delta_{\alpha\gamma} - \xi_{i\gamma}\delta_{\beta\alpha} \\ &= \frac{c_{i\alpha} \odot c_{i\beta} \odot c_{i\gamma}}{c_s^3} - \frac{c_{i\alpha}\delta_{\beta\gamma} + c_{i\beta}\delta_{\alpha\gamma} + c_{i\gamma}\delta_{\beta\alpha}}{c_s}\end{aligned} \tag{2.23}$$

Rewriting the lower ordered Hermite polynomials in a similar way and substituting appropriate values for a_0 in eq. (2.22) yields a third order equilibrium function.

$$\begin{aligned}f^{\text{eq}}_{\phi,i} = \phi\omega_i \odot \Bigg(1 + u_\alpha \frac{c_{i\alpha}}{c_s^2} + u_\alpha u_\beta \left(\frac{1}{2c_s^4}\left(c_{i\alpha} \odot c_{i\beta} - c_s^2\delta_{\alpha\beta}\right)\right) \\ + u_\alpha u_\beta u_\gamma \left(\frac{1}{6c_s^6}\left(c_{i\alpha} \odot (c_{i\beta} \odot c_{i\gamma}) - c_s^2\left(c_{i\alpha}\delta_{\beta\gamma} + c_{i\beta}\delta_{\gamma\alpha} + c_{i\gamma}\delta_{\beta\alpha}\right)\right)\right)\Bigg),\end{aligned} \tag{2.24}$$

Although, as mentioned in Krüger et al. [12], the second order distribution in eq. (2.7) is sufficient for recovering the Navier Stokes equation. The reason to write out the third order distributution is because it is needed as a part of some of the calculations in appendix C.

There is one important caveat with the change in notation from i, j, k to $(i\alpha)$, $(i\beta)$, $(i\gamma)$, and that is that the Hermite polynomials are symmetric, that is $\mathcal{H}^{(2)}_{ij} = \mathcal{H}^{(2)}_{ji}$. When trying to represent the Hermite expansion as a tensor as will be done in chapter 4,the tensor operations will not behave as could be expected. This nuance is elaborated on extensively in appendix A. The reason to use this notation is because it is the convection used by Krüger et al. [12] and because the symmetric notation used by reference [23, 26] is hard to compare to the macroscopic formulations of eq. (Convection diffusion equation), eq. (Continuity) and eq. (Momentum). The definition of tensor multiplications and generalization of the kronecker δ are especially unintuitive. Since the derivations in this thesis mostly use terms corresponding up to the second order Hermite expansion, writing the notation explicitly is still manageable, although cumbersome. If the procedures introduced in this thesis are used to formulate a similar method for the Burnett equation, the notation would become even more convoluted.

2.5 Multiple Relaxation Time collisional operators

The Multiple Relaxation Time (MRT) collisional operator was originally proposed by d'Humieres in 1992 [27] and further formalized by d'Humieres in 2002 [28]. The general idea is that instead using a single relaxation time for the entire population density like BGK, the sums leading to macroscopic moments can be relaxed individually. To perform this relaxation, the population density is transformed to a

moment vector through a transformation matrix. The following is based on Hermite polynomials and given by Krüger et al. [12]

$$M = \begin{bmatrix} 1_i, & c_{ix}, & c_{iy}, & c_{ix}c_{ix} - c_s^2, & c_{iy}c_{iy} - c_s^2, \\ c_{ix}c_{iy} & c_{ix}c_{iy}^2 - c_s^2 c_{ix}, & c_{iy}c_{ix}^2 - c_s^2 c_{iy}, & c_{ix}^2 c_{iy}^2 - c_s^2 c_{ix}^2 - c_s^2 c_{iy}^2 + c_s^4 \end{bmatrix}^T, \tag{2.25}$$

which relates the moment vector α_ϕ (denoted m in Krüger et al. [12]) to the population f_ϕ through $\alpha_\phi = M f_\phi$. For example, multiplying eq. (2.7) with M and substituting the relations the relations eqs. $(\sum \omega)$ to $(\sum \omega c^4)$ would give a vector with components in macroscopic variables. The last components of that vector does not correspond to any of the term in eq. (2.7).

Then a relaxation matrix is introduced which has different relaxation times for the different moments, two of which are related to physical quantities and two which corresponds to higher order terms. The higher order terms are useful for increasing stability and accuracy, although these improvements have shown to be minor. Choosing the relaxation rates for the MRT is difficult and has to be done on a case by case basis [12].

2.6 The Filter Matrix Lattice Boltzmann Method

Filter matrix method is an alternative approach to solve the Lattice Boltzmann Equation. To the author only six articles developing the method are known. The two original articles from Shell Research, Netherlands [29, 30] from the 1990's and four based in Northwestern Polytechnical University, China, from the 2010's [10, 31, 32, 33]. Moreover, on top of these articles the method has succefully been used by Rohde in [34] as well as several master thesis projects based at TU Delft, Netherlands such as reference [5, 6, 7, 8].

The method works on a staggered grid, where the populations $f_{\phi,i}$ at each staggered node is transformed into a vector $\alpha_{\phi,k}$ corresponding to the macroscopic moments of the distribution function at the corresponding non-staggered node. The components of the α vector can then be used to calculate the collision, which is applied to the α vector directly. Then the updated α is transformed back to the staggered distribution functions and later streamed. The formulation presented below is based on Zhou et al. [10] and [33].

Calculating the moment vector α from the staggered populations through the matrix E_{ki}

$$\alpha_{\phi,i}^{\pm} = \sum_i E_{ki} f_{\phi,i} \left(x_\alpha \pm c_{i\alpha} \frac{\Delta t}{2}, t \pm \frac{\Delta t}{2} \right), \tag{2.26}$$

where the matrix is defined based on the used velocity sets, e.g. for the D2Q9 scheme

$$E^{(\text{Zhuo})}_{ki,\text{D2Q9}} = \begin{bmatrix} 1, & c_{ix}, & c_{iy}, & 3c_{ix}^2 - 1 & 3c_{ix}c_{iy}, \\ 3c_{iy}^2 - 1, & c_{ix}(3c_{iy}^2 - 1), & c_{iy}(3c_{ix}^2 - 1), & \frac{1}{2}(3c_{ix}^2 - 1)(3c_{iy}^2 - 1) & \end{bmatrix}^T . \tag{2.27}$$

Then the collision is performed by calculating the $\pm$-terms through the description of the α-vector, given in two dimension as

$$\alpha^{\pm}_{\text{D2Q9},k} = \begin{bmatrix} \rho \\ \rho u_x \pm \Delta t F_x/2 \\ \rho u_y \pm \Delta t F_y/2 \\ 3\rho u_x u_x + \rho\left(-6\nu \pm \Delta t\right)\partial_x u_x + (3 - 3B)\,\rho\,\nu\,\text{div}(u_\alpha) \\ 3\rho u_x u_y + \rho\left(-6\nu \pm \Delta t\right)(\partial_y x + \partial_x y)/2 \\ 3\rho u_y u_y + \rho\left(-6\nu \pm \Delta t\right)\partial_y u_y + (3 - 3B)\,\rho\,\nu\,\text{div}(u_\alpha) \\ T_1^{\pm} \\ T_2^{\pm} \\ F^{\pm} \end{bmatrix} \tag{2.28}$$

where bulk viscosity is related to the viscosity through the constant B as $\nu_{\text{bulk}} = B\nu$. The T and F terms are higher order terms, which can be set to zero or treated in somewhat more complex ways as described by reference [30, 10, 32]. In three dimension for the D3Q19 scheme, the moment vector is instead given by

$$\alpha^{\pm}_{\text{D3Q19}} = \begin{bmatrix} \rho \\ \rho u_x \pm \Delta t F_x/2 \\ \rho u_y \pm \Delta t F_y/2 \\ \rho u_z \pm \Delta t F_z/2 \\ 3\rho u_x u_x + \rho\left(-6\nu \pm \Delta t\right)\partial_x u_x + (2 - 3B)\rho\,\nu\,\text{div}(u_\alpha) \\ 3\rho u_y u_y + \rho\left(-6\nu \pm \Delta t\right)\partial_y u_y + (2 - 3B)\rho\,\nu\,\text{div}(u_\alpha) \\ 3\rho u_z u_z + \rho\left(-6\nu \pm \Delta t\right)\partial_z u_z + (2 - 3B)\rho\,\nu\,\text{div}(u_\alpha) \\ 3\rho u_y u_z + \rho\left(-6\nu \pm \Delta t\right)(\partial_y z + \partial_z y)/2 \\ 3\rho u_x u_z + \rho\left(-6\nu \pm \Delta t\right)(\partial_z x + \partial_x z)/2 \\ 3\rho u_x u_y + \rho\left(-6\nu \pm \Delta t\right)(\partial_y x + \partial_x y)/2 \\ T^{\pm}_{1,2,3,4,5,6} \\ F^{\pm}_{1,2,3} \end{bmatrix} . \tag{2.29}$$

When the collision is performed, that is the α^- have been changed to the corresponding α^+, the moment vector is transformed back to the moment space as

$$f_{\phi,i}\left(x_\alpha \pm c_{i\alpha}\frac{\Delta t}{2}, t \pm \frac{\Delta t}{2}\right) = \sum_k \omega_i E_{ik}\alpha^{\pm}_{\phi,i}, \tag{2.30}$$

in this case using the matrix

$$E^{(\text{Zhuo})}_{ik,\text{D2Q9}} = \begin{bmatrix} 1, & 3c_{ix}, & 3c_{iy}, & \frac{3c_{ix}^2 - 1}{2}, & 3c_{ix}c_{iy}, \\ \frac{3c_{iy}^2 - 1}{2}, & \frac{3c_{ix}(3c_{iy}^2 - 1)}{2}, & \frac{3c_{iy}(3c_{ix}^2 - 1)}{2}, & \frac{1}{2}(3c_{ix}^2 - 1)(3c_{iy}^2 - 1) & \end{bmatrix} \tag{2.31}$$

where the two matrices are related by $\omega_i E_{ik} = E_{ki}^{-1}$.

The α vector changes when considering a convection–diffusion problem instead of the Navier Stokes equations, algorithm is the same except that α is renamed to β and given by

$$\beta^{\pm}_{\mathrm{D2Q9}} = \Big[\begin{array}{cccccc} \phi, & u_x\phi + \left(\frac{-6D_\phi \pm \Delta t}{6}\right)\partial_x\phi, & u_y\phi + \left(\frac{-6D_\phi \pm \Delta t}{6}\right)\partial_y\phi, & S_1^{\pm}, & S_2^{\pm}, \\ S_3^{\pm}, & T_1^{\pm}, & T_2^{\pm}, & F^{\pm} & \end{array} \Big]. \tag{2.32}$$

2.6.1 Derivation of the Filter Matrix Method

The derivation procedure for the original formulation of FMLB is given by Somers [29], which extensively relies on the work *Lattice gas hydrodynamics in two and three dimensions* from 1987 [19]. In it, arguments form lattice gas modelling are used together with cellular automata rules to derive hydrodynamic equations. Imporantly, the study constructs a dynamic population density N_i, which is conceptually different from the equilibrium distribution. In N_i there are several free parameters that can be chosen to match the Navier Stokes equations, c.f. equation 4.14, 5.1, 5.13, 5.14, 5.15 in [19]. This is the starting point for both the derivations of both Somers [29] and Zhou et al. [10].

In Somers [29] the N_i population is formulated for a 24-speed grid, having the property that $\sum_i m_i = 24$ and four dimension that has the desired symmetric properties when projected down to three dimensions, as described by Krüger et al. [12]. The initial motivation is in Somers [29], although the formulations from Eggels & Somers [30] are shown below. The population density is determined to be, with subgrid stress tensor τ

$$\begin{aligned} N_i = \frac{m_i\rho}{24}\Big(1 + 2c_i\cdot u + 3\left((c_i\cdot u)^2 - \frac{1}{2}u^2 + c_ic_i \,:\, \tau - \frac{1}{2}\mathrm{trace}(\tau)\right) \\ + 6\nu\left((c_i\cdot\nabla)(c_i\cdot u) - \frac{1}{2}\nabla\cdot u\right) + \mathcal{O}(u^3, u\nabla u)\Big), \end{aligned} \tag{2.33}$$

from which the challenge is to construct the collisional operator. This is done through a staggered formulation of the population density

$$N_i\left(x + \frac{1}{2}c_i, t + \frac{1}{2}\right) = N_i\left(x - \frac{1}{2}c_i, t - \frac{1}{2}\right) + \Omega(N_i) \tag{2.34}$$

and a Taylor expansion of the staggered

$$N_i\left((x + \frac{1}{2}c_i, t + \frac{1}{2}\right) = N_i(x,t) \pm \frac{1}{2}c_i\nabla N_i(x,t) + \frac{1}{2}\partial_t N_i(x,t) \tag{2.35}$$

from which so called higher order terms, of the form $\mathcal{O}(u^3, u\nabla u)$ are neglected and the expression rewritten with the continuity equation, the approximation,

$$\partial_t(\rho u) \approx -\frac{1}{2}\nabla p + F + \mathcal{O}(\nabla u^2, \nabla^2 u), \tag{2.36}$$

for some force term F and the pressure defined as

$$p = \frac{\rho}{2}\left(1 - \frac{1}{2}(u^2 + \text{trace}(\tau))\right) \tag{2.37}$$

Finally resulting in

$$\Omega_i = \frac{m_i \rho}{12}\left[(c_i \cdot \nabla)(c_i \cdot u) - \frac{1}{2}\nabla \cdot u\right]. \tag{2.38}$$

From the N_i and Ω_i, Somers [29] proceeds to find the eigenvectors corresponding to different hydrodynamical modes, such as density mode 1, momentum modes $c_{ix}, c_{iy}, c_{iz}, c_{iw}$, as well as shear modes $2c_{ix}^2 - 1, 2c_{iy}^2 - 1, 2c_{iz}^2 - 1, c_{ix}c_{iy}, c_{ix}c_{iz}, c_{iz}c_{iy}, c_{iw}c_{ix}$, $c_{iw}c_{iy}, c_{iw}c_{iz}$, c.f. Table I in Somers [29]. The projection of the population densities on these modes are then written out in Eggels & Somers [30], which gives transformation matrices and moment vectors similar to what was introduced in section 2.6.

The derivation by Zhou et al. [10] is similar, although the tuning the free parameters to match the Navier Stokes equation of the general N_i formulated in [19] is written out explicitly. However, the derivational procedure in both Somers [29] and Zhou et al. [10] is described only briefly, and is difficult to extend for arbitrary velocity sets. It is reasonable to expect that the tuning of the parameters from N_i in [19] is similar for the convection diffusion equation, although it is not written out explicitly. The inclusion of higher order hydrodynamical modes in the projection matrices is also not motivated, espceially since some higher order terms are neglected later in the derivation. Therefore, this thesis will investigate the derivational procedure in great detail.

Currently Rohde [35] tries to perform a more explicit derivation for the FMLB formulation for the momentum equation, and that methodology will largely be the basis for the derivation in chapter 3. The general approach is to do a full Chapman–Enskog expansion of the BGK and calculate the corresponding N_i and identify terms the least amount of terms needed to recover the Navier–Stokes equation. That derivation have several limitations, such relying on the matrices calculated by Zhou et al. [10] and not implementing any force term.

2.7 Turbulence modelling in LBM

Turbulence modelling through LES has been studied both for LBM in general as well as specifically for FMLB, as in reference [33, 29]. However, LES is a computationally expensive turbulence model compared to the $k - \varepsilon$ model. Despite its prevalence in finite volume method simulations, very few studies have been conducted on $k - \varepsilon$ modelling in the LBM framework. Only two such studies are known to the author, Succi et al. [36] and Sajjadi et al. [37]. The former did not numerically test its proposed schemed and uses a first order equilibrium distribution function, something Krüger et al. [12] recommends against due to the error term appearing in the Chapman–Enskog analysis. The study by Sajjadi et al. [37] uses the BGK operator and a second order equilibrium function for the k and ε populations. However, it

seemingly does not use the variable substitutions needed to recover second order accuracy when using force and source terms as described by Krüger et al. [12]. Further, there are no studies implementing the $k-\varepsilon$ model in the FMLB framework.

3

Chapman–Enskog analysis of convection–diffusion equation

In this chapter, a convectional Chapman–Enskog expansion is used to show how the Boltzmann equation with the BGK collisional operator in a staggered formulation is similar to the convection–diffusion equation. Based on this expansion, the BGK operator is rewritten in terms of macroscopic flow properties, similarly to an internal memo by Rohde [35] where this is done for the forceless momentum equations. This expansion contains several so called *higher order terms* deemed unphysical, which are removed to form a new collisional operator and effectively "filtering" the BGK operator. Note that in order to make for easier comparison to literature, the notational nuances and aids introduced in appendix A are not considered in this chapter to be more easily compared to Krüger et al. [12] and Rohde [35].

3.1 Staggering the discretized Boltzmann equation

Similarly to Zhou et al. [10], the discretized version of eq. (Boltzmann) looks like

$$\partial_t f_{\phi,i} + c_{i\alpha}\partial_\alpha f_{\phi,i} + a_\alpha \partial_{c_{i\alpha}} f_{\phi,i} = \Omega'_{\phi,i}(f_{\phi,i}) \tag{3.1}$$

where the last term of the left hand side is a source term that adds scalar, e.g. a heat source, or scalar flux, e.g. momentum. Similarily to Zhou et al. [10] this could be grouped into an effective collision term

$$\partial_t f_{\phi,i} + c_{i\alpha}\partial_\alpha f_{\phi,i} = \Omega_{\phi,i}(f_{\phi,i}) \tag{3.2}$$

or implement a discretized source contribution explicitly

$$\partial_t f_{\phi,i} + c_{i\alpha}\partial_\alpha f_{\phi,i} = \Omega'_{\phi,i}(f_{\phi,i}) - Q_{\phi,i}. \tag{3.3}$$

Second order taylor expansion of a perturbation of the equilibrium results in

$$\begin{aligned} f_{\phi,i}\left(x_\alpha \pm c_{i\alpha}\frac{\Delta t}{2}, t \pm \frac{\Delta t}{2}\right) = f_{\phi,i}(x_\alpha, t) \\ \pm c_{i\alpha}\frac{\Delta t}{2}\partial_\alpha f_{\phi,i}(x_\alpha, t) \pm \frac{\Delta t}{2}\partial_t f_{\phi,i}(x_\alpha, t) \\ + \left(\frac{\Delta t}{2}\right)^2 \Big(c_{i\alpha}\partial_t\partial_\alpha f_{\phi,i}(x_\alpha, t) \\ + \frac{1}{2}(c_{i\alpha}\odot c_{i\alpha})\partial_{\alpha\alpha} f_{\phi,i}(x_\alpha, t) + \frac{1}{2}\partial_{tt} f_{\phi,i}(x_\alpha, t)\Big) \end{aligned} \tag{3.4}$$

For the Filter Matrix method it is appropriate to discretize the Boltzmann equation in a staggered manner, a second order accurate leapfrog scheme, similarly to what has been done in the original derivation by Somers [29]

$$f_{\phi,i}\left(x_\alpha + \frac{c_{i\alpha}\Delta t}{2}, t + \frac{\Delta t}{2}\right) = f_{\phi,i}\left(x_\alpha - \frac{c_{i\alpha}\Delta t}{2}, t - \frac{\Delta t}{2}\right) + \Delta t\, \Omega_{\phi,i}(f_{\phi,i}). \tag{3.5}$$

or with the explicit source treatment $Q_{\phi,i}$-term as described by Krüger et al. [12], but here applied to the staggered discretization,

$$f_{\phi,i}\left(x_\alpha + \frac{c_{i\alpha}\Delta t}{2}, t + \frac{\Delta t}{2}\right) = f_{\phi,i}\left(x_\alpha - \frac{c_{i\alpha}\Delta t}{2}, t - \frac{\Delta t}{2}\right) + \Delta t\left(\Omega'_{\phi,i}(f_{\phi,i}) + Q_{\phi,i}\right). \tag{3.6}$$

Substituting eq. (3.4) into eq. (3.5) and simplifying results in[1]

$$\Omega_{\phi,i}(f_{\phi,i}) = \partial_t f_{\phi,i}(x_\alpha, t) + c_{i\alpha}\partial_\alpha f_{\phi,i}(x_\alpha, t) \tag{3.7}$$

and equivalently with the explicit source,

$$\Omega'_{\phi,i}(f_{\phi,i}) + Q_{\phi,i} = \partial_t f_{\phi,i}(x_\alpha, t) + c_{i\alpha}\partial_\alpha f_{\phi,i}(x_\alpha, t). \tag{3.8}$$

As will be seen, this staggering is a critical property of the Filter Matrix method and the reason for its second order accuracy. The staggered formulation is typically not used with BGK. The reason for this is likely to be that the values of $f_{\phi,i}$ on the non-staggered nodes are needed to perform the collision, this has to be extrapolated from the nodes, c.f. fig. 3.1, which is cumbersome to do. In the Filter Matrix framework presented below, it will be shown to become easier to do this extrapolation.

3.2 Chapman–Enskog expansion of a staggered formulation BGK formulation of the convection diffusion equation

The following Chapman–Enskog is based on the convection–diffusion equation expansion in Krüger et al. [12], although here a source term is added. This is done in a similar way to the force term in the expansion for the momentum equation by Krüger et al. [12].

Throughout this derivation, an equilibrium distribution which is a second order Hermite polynomial expansion of the Maxwell-Boltzmann distribution is used, the superscript (0) is used to denote the equilibrium from eq. (2.7)

$$f^{(0)}_{\phi,i}(x_\alpha, t) = \omega_i \phi\left(1 + \frac{c_{i\alpha}u_\alpha}{c_s^2} + \frac{1}{2}\frac{(c_{i\alpha}u_\alpha)^2}{c_s^4} - \frac{1}{2}\frac{u_\alpha u_\alpha}{c_s^2}\right). \tag{$f^{(0)}$}$$

Summing this equilibrium distribution and applying the the properties of the discretization in eqs. ($\sum\omega$) to ($\sum\omega c^3$) yields

$$\sum_i f^{(0)}_{\phi,i} = \phi, \qquad \sum_i c_{i\alpha} f^{(0)}_{\phi,i} = \phi u_\alpha. \tag{3.9}$$

[1] Notice that the second order derivatives disappear due to the staggered formulation.

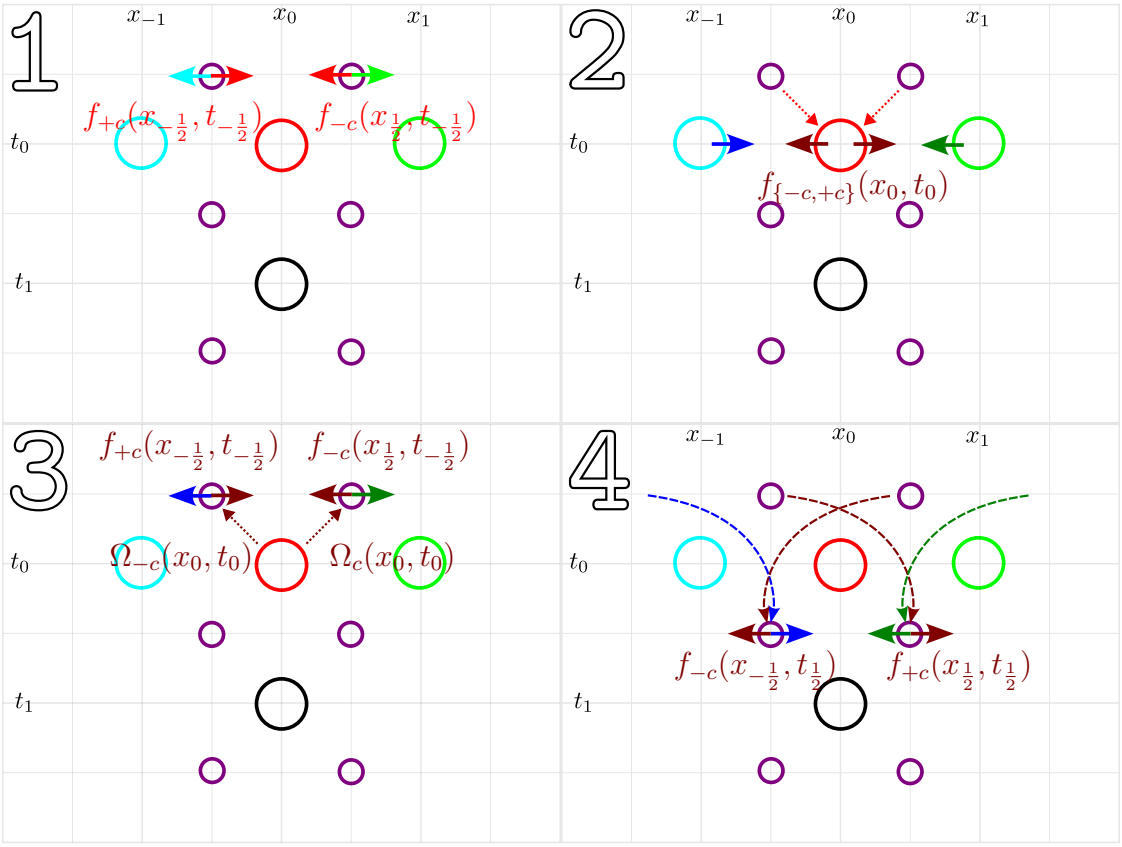

Figure 3.1: Staggering in a D1Q2 scheme. 1. Initial populations defined at half nodes. 2. Half node populations are used to estimate the integer node populations. 3. The collision values for each half node is calculated through the integer node populations. 4. The half node values are streamed and the process from 1. can be repeated toward the next time step.

3.2.1 Taylor expansion and scale separation

The key technique to the Chapman–Enskog expansion is to separate into different scales, which could be argued represents different time scales for physical processes, for details see Krüger et al. [12] and [19]. The distribution function is expanded to second order as

$$f_{\phi,i} = \underbrace{f^{(0)}_{\phi,i}}_{\text{Equilibrium}} \underbrace{+\epsilon f^{(1)}_{\phi,i} + \epsilon^2 f^{(2)}_{\phi,i}}_{\text{Non-equilibrium}}, \tag{3.10}$$

which is split into an equilibrium part and non-equilibrium part. Derivatives in space and time are also expanded as in Krüger et al. [12] and [19, 17] to

$$\partial_t = \epsilon\partial^{(1)}_t + \epsilon^2\partial^{(2)}_t \tag{3.11}$$

and

$$\partial_\alpha = \epsilon\partial^{(1)}_\alpha \tag{3.12}$$

because of the scale separation, terms multiplied by ϵ^n with $n > 2$ are not accounted for. Thus, combining the scale separations of the time derivative and the distribution

function results in

$$\begin{aligned}\partial_t f_{\phi,i}(x_\alpha, t) &= \left(\epsilon\partial_t^{(1)} + \epsilon^2\partial_t^{(2)}\right)\left(f_{\phi,i}^{(0)} + \epsilon f_{\phi,i}^{(1)} + \epsilon^2 f_{\phi,i}^{(2)}\right) \\ &= \epsilon\partial_t^{(1)} f_{\phi,i}^{(0)} + \epsilon^2\left(\partial_t^{(1)} f_{\phi,i}^{(1)} + \partial_t^{(2)} f_{\phi,i}^{(0)}\right) + \mathcal{O}(\epsilon^3).\end{aligned} \tag{3.13}$$

Now, using the scale separations from before in eq. (3.8) together with the scale separation suggested by Krüger et al. [12] $Q_{\phi,i} = \epsilon Q_{\phi,i}^{(1)}$

$$\begin{aligned}\epsilon(\partial_t^{(1)} f_{\phi,i}^{(0)} + c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(0)} - Q_{\phi,i}^{(1)}) + \epsilon^2\left(\partial_t^{(1)} f_{\phi,i}^{(1)} + \partial_t^{(2)} f_{\phi,i}^{(0)} + c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(1)}\right) \\ = \epsilon\Omega_{\phi,i}^{(1)} + \epsilon^2\Omega_{\phi,i}^{(2)} + \mathcal{O}(\epsilon^3)\end{aligned} \tag{3.14}$$

In this case the collisional operator of choice is the Bhatnagar–Gross–Krook operator (BGK), which is defined and rewritten with the scale separation as

$$\Omega_{\phi,i,\mathrm{BGK}} = -\frac{1}{\tau}\left(f_{\phi,i} - f_{\phi,i}^{(0)}\right) = -\frac{1}{\tau}\left(\epsilon f_{\phi,i}^{(1)} + \epsilon^2 f_{\phi,i}^{(2)}\right). \tag{3.15}$$

which yields the final discretized, staggered, scale separated BGK expression

$$\begin{aligned}\epsilon(\partial_t^{(1)} f_{\phi,i}^{(0)} + c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(0)} - Q_{\phi,i}^{(1)}) + \epsilon^2\left(\partial_t^{(1)} f_{\phi,i}^{(1)} + \partial_t^{(2)} f_{\phi,i}^{(0)} + c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(1)}\right) \\ = \frac{-1}{\tau}\left(\epsilon f_{\phi,i}^{(1)} + \epsilon^2 f_{\phi,i}^{(2)}\right) + \mathcal{O}(\epsilon^3).\end{aligned} \tag{3.16}$$

3.2.2 Summing into macroscopic properties

The main difference to Chapman–Enskog analysis of the momentum equations comes from the non-equilibrium sums as these represents whether certain quantities are preserved after a collision. For the momentum equations, both mass and momentum are conserved. For scalar transport, only the scalar is conserved, the scalar fluxes are not conserved. In this case, conservation of a property means that the non-equilibrium part sums to zero, i.e.

$$\sum_i f_{k,i}^{(\mathrm{neq})} = 0. \tag{3.17}$$

The so called *solvability conditions* discussed by Krüger et al. [12] is to further assume that each individual scale sums to zero, that for $n > 1$

$$\sum_i f_{k,i}^{(n)} = 0. \tag{3.18}$$

The scalar fluxes are not conserved for the convection–diffusion equations, and thus nothing is know about $\sum_i c_{i\alpha} f_{k,i}^{(\mathrm{neq})}$ Now, each scale ϵ and ϵ^2 in eq. (3.16) can be analyzed individually,

Summations in ϵ^1

The equation associated with the scale ϵ in eq. (3.16) is

$$-\frac{1}{\tau_\phi} f_{\phi,i}^{(1)} = \partial_t^{(1)} f_{\phi,i}^{(0)} + c_{i\alpha} \partial_\alpha^{(1)} f_{\phi,i}^{(0)} - Q_{\phi,i}^{(1)}, \tag{3.19}$$

summing each term in eq. (3.19) yields

$$\sum_i f_{\phi,i}^{(1)} = -\tau_\phi \left(\sum_i \partial_t^{(1)} f_{\phi,i}^{(0)} + \sum_i c_{i\alpha} \partial_\alpha^{(1)} f_{\phi,i}^{(0)} - \sum_i Q_{\phi,i}^{(1)} \right) \tag{3.20}$$

in which the previously established summations for $Q_{\phi,i}^{(1)}$, $f_{\phi,i}^{(0)}$ and $f_{\phi,i}^{(1)}$ can be substituted, resulting in

$$\partial_t^{(1)} \phi + \partial_\alpha^{(1)} (\phi u_\alpha) - q_\phi = 0. \tag{$\epsilon^1 c^0$}$$

Then all terms in eq. (3.19) are multiplied with $c_{i\alpha}$ and then summed, as

$$\sum_i c_{i\alpha} f_{\phi,i}^{(1)} = -\tau_\phi \left(\sum_i c_{i\alpha} \partial_t^{(1)} f_{\phi,i}^{(0)} + \sum_i c_{i\alpha} \left(c_{i\beta} \partial_\beta^{(1)} \right) f_{\phi,i}^{(0)} - \sum_i c_{i\alpha} Q_i^{(1)} \right) \tag{3.21}$$

and inspired by derivation in Krüger et al. [12], two new variables are introduced as

$$\Pi_{\phi,\alpha\beta}^{(0)} = \sum_i c_{i\alpha} c_{i\beta} f_{\phi,i}^{(0)} \tag{3.22}$$

and

$$\Theta_{\phi,\alpha}^{(1)} = \sum_i c_{i\alpha} f_{\phi,i}^{(1)}, \tag{3.23}$$

resulting in

$$\Theta_{\phi,\alpha}^{(1)} = -\tau_\phi \left(\partial_t^{(1)} (\phi u_\alpha) + \partial_\beta^{(1)} \Pi_{\phi,\alpha\beta}^{(0)} \right). \tag{$\epsilon^1 c^1$}$$

Summations in ϵ^2

The equation for ϵ^2 in eq. (3.16) is

$$-\frac{1}{\tau_\phi} f_{\phi,i}^{(2)} = \partial_t^{(2)} f_{\phi,i}^{(0)} + \partial_t^{(1)} f_{\phi,i}^{(1)} + c_{i\alpha} \partial_\alpha^{(1)} f_{\phi,i}^{(1)} \tag{3.24}$$

which is then summed termwise as

$$\sum_i f_{\phi,i}^{(2)} = -\tau_\phi \left(\sum_i \partial_t^{(2)} f_{\phi,i}^{(0)} + \sum_i \partial_t^{(1)} f_{\phi,i}^{(1)} + \sum_i c_{i\beta} \partial_\beta^{(1)} f_{\phi,i}^{(1)} \right) \tag{3.25}$$

where previous summations and redefinitions are substituted into, giving the relation

$$\partial_t^{(2)} \phi + \partial_\alpha^{(1)} \Theta_{\phi,\alpha}^{(1)} = 0. \tag{$\epsilon^2 c^0$}$$

3.2.3 Isolating error term

Summing[2] eq. ($\epsilon^1 c^0$) and eq. ($\epsilon^2 c^0$), using $\partial_\alpha^{(1)} = \partial_\alpha$ and substituting $(\partial_t^{(1)} + \partial_t^{(2)})\phi = \partial_t \phi$ yields[3]

$$\partial_t \phi + \partial_\alpha^{(1)}(\phi u_\alpha) - q_\phi + \partial_\alpha \Theta_{\phi,\alpha}^{(1)} = 0. \tag{3.26}$$

Finally, substituting eq. ($\epsilon^1 c^1$) into eq. (3.26) yields

$$\partial_t \phi + \partial_\alpha(\phi u_\alpha) = q_\phi + \partial_\alpha \left(\tau_\phi \left(\partial_t^{(1)}(\phi u_\alpha) + \partial_\beta \Pi_{\phi,\alpha\beta}^{(0)} \right) \right). \tag{3.27}$$

The diffusion coefficient is defined as,

$$D_\phi = c_s^2 \tau_\phi \tag{3.28}$$

and this is allowed to vary in both space and time, which is a necessary in order to incorporate turbulent fluctuations.

Now $\Pi_{\phi,\alpha\beta}^{(0)}$ can be calculated by substituting the second order equilibrium distribution function eq. ($f^{(0)}$) and using the relations eqs. ($\sum \omega$) to ($\sum \omega c^3$) $c_{i\alpha}$ to get

$$\Pi_{\phi,\alpha\beta}^{(0)} = \sum_i c_{i\alpha} c_{i\beta} f_{\phi,i}^{(0)} = c_s^2 \phi \delta_{\alpha\beta} + \phi u_\alpha u_\beta. \tag{3.29}$$

Thus, substituting D_ϕ and eq. (3.29) into the appropriate term in eq. (3.27) and performing the product rule twice allows separation of ϕ and u_α-derivatives

$$\begin{aligned} \partial_\alpha \left(\frac{D_\phi}{c_s^2} \partial_\beta \Pi_{\phi,\alpha\beta}^{(0)} \right) &= \partial_\alpha \left(\frac{D_\phi}{c_s^2} \partial_\beta \left(c_s^2 \phi \delta_{\alpha\beta} + \phi u_\alpha u_\beta \right) \right) \\ &= \partial_\alpha (D_\phi \partial_\alpha \phi) + \partial_\alpha \left(\frac{D_\phi}{c_s^2} \partial_\beta (\phi u_\alpha u_\beta) \right). \end{aligned} \tag{3.30}$$

Substituting this back into eq. (3.27) yields

$$\begin{aligned} \partial_t \phi + \partial_\alpha(\phi u_\alpha) &= \partial_\alpha (D_\phi \partial_\alpha \phi) + q_\phi \\ &\quad + \underbrace{\partial_\alpha \left(\frac{D_\phi}{c_s^2} \left(\partial_\beta (\phi u_\alpha u_\beta) + \partial_t^{(1)}(\phi u_\alpha) \right) \right)}_{E} \end{aligned} \tag{3.31}$$

Which is already has the form of the advection diffusion equation

$$C^\phi = D^\phi + q_\phi + E, \tag{3.32}$$

where E is a quite complex unwanted error term. Now, this term has to be expanded further to see whether it significant for the flow cases of interest. The interior of E can be expanded with the product rule as

$$\begin{aligned} \partial_\beta (\phi u_\alpha u_\beta) + \partial_t^{(1)}(\phi u_\alpha) &= \phi \partial_\beta (u_\alpha u_\beta) + u_\alpha u_\beta \partial_\beta \phi + \phi \partial_t^{(1)} u_\alpha + u_\alpha \partial_t^{(1)} \phi \\ &= u_\alpha \left(\partial_t^{(1)} \phi + u_\beta \partial_\beta \phi \right) + \phi \left(\partial_t^{(1)} u_\alpha + \partial_\beta (u_\alpha u_\beta) \right). \end{aligned} \tag{3.33}$$

[2] The fact that the scale separation is summed to, in some sense, remove the scale separation will serve as inspiration the derivation in chapter 4.

[3] It will later be shown that this expression is exactly the same as if the BGK was not assumed, i.e. it is also valid for a general Ω.

Now the two sums multiplied with u_α and ϕ in eq. (3.33) can be analyzed individually. The result for the term multiplied with u_α is simplified with eq. ($\epsilon^1 c^0$) as

$$\begin{aligned} \partial_t^{(1)}\phi + u_\beta \partial_\beta \phi &= q_\phi - \partial_\alpha(\phi u_\alpha) + u_\beta \partial_\beta \phi \\ &= q_\phi - \phi \partial_\alpha u_\alpha. \end{aligned} \tag{3.34}$$

3.2.4 Momentum substitution

The term multiplied with ϕ in eq. (3.33) can be expanded through terms from the Chapman–Enskog analysis of the momentum equation, as done by Rohde [35]. The derivation by Rohde [35] does not account for forces, below is a derivation that assumes a body force implemented in a the most simple way, similarly to the source term in this report. Expanding and grouping terms from the ϕ in eq. (3.33) results in

$$\partial_t^{(1)} u_\alpha + \partial_\beta \left(u_\alpha u_\beta\right) = \left(\partial_t^{(1)} u_\alpha + u_\beta \partial_\beta u_\alpha\right) + u_\alpha \partial_\beta u_\beta \tag{3.35}$$

The left hand side parentheses is rewritten using repeated application of the inverse product rule,

$$\partial_t^{(1)} u_\alpha + u_\beta \partial_\beta u_\alpha = \frac{1}{\rho}\left(\left(\partial_t^{(1)}(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta)\right) - u_\alpha \underbrace{\left(\partial_t^{(1)}\rho - \partial_\beta(\rho u_\beta)\right)}_{=0, \text{ from momentum eq. } (\epsilon^1 c^0)}\right). \tag{3.36}$$

The momentum equations contain the volumetric force $F_{\rho,\alpha}$[4] which is decomposed to the velocities as $\mathcal{F}_{\rho,i}$ that satisfies (using the first order force assumption)

$$\sum_i \mathcal{F}_{\rho,i} = 0, \qquad \sum_i c_{i\alpha}\mathcal{F}_{\rho,i} = F_{\rho,\alpha}. \tag{3.37}$$

Two important intermediate results from the momentum Chapman–Enskog analysis are

$$\Pi^{(0)}_{\rho,\alpha\beta} = c_s^2 \rho \delta_{\alpha\beta} + \rho u_\alpha u_\beta \qquad = p\delta_{\alpha\beta} + \rho u_\alpha u_\beta \tag{3.38}$$

$$\partial_t^{(1)}(\rho u_\alpha) + \partial_\beta^{(1)}\Pi^{(0)}_{\rho,\alpha\beta} = \sum_i c_{i\alpha}\mathcal{F}^{(1)}_{\rho,i} \qquad = F_{\rho,\alpha}, \tag{3.39}$$

which comes from the grid discretization and the equivalent expression to eq. ($\epsilon^1 c^1$) in Rohde [35] although extended with the force implementation above. Thus the first term on the right hand side in eq. (3.36) becomes

$$\partial_t^{(1)}(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = -\frac{1}{\rho}\partial_\alpha p + \frac{F_{\rho,\alpha}}{\rho} \tag{3.40}$$

with the entire expression

$$\partial_t^{(1)} u_\alpha + \partial_\beta \left(u_\alpha u_\beta\right) = -\frac{1}{\rho}\partial_\alpha p + \frac{F_{\rho,\alpha}}{\rho} + u_\alpha \partial_\beta u_\beta. \qquad \text{(Momentum subs.)}$$

[4] Meaning it will have units of $\mathrm{N\,m^{-3}}$

Error terms

Resubstituting the previous results into eq. (3.33)

$$\partial_t^{(1)}(\phi u_\alpha) + \partial_\beta(\phi u_\alpha u_\beta) = u_\alpha q_\phi - \frac{\phi}{\rho}\partial_\alpha p + \frac{\phi}{\rho}F_{\rho,\alpha}. \tag{3.41}$$

Rearranging using the relation $\frac{\partial_\alpha p}{p} = \frac{\partial_\alpha \rho}{\rho}$ makes it more clear when incompressibility is relevant. Substituting this and eq. (3.41) back into eq. (3.31) and using $p = c_s^2\rho$ finally results in

BGK error terms for convection diffusion equation

The sums defined in section 3.2.2 together with the BGK operator is equivalent to the modified convection diffusion equation

$$\partial_t\phi + \partial_\alpha(\phi u_\alpha) = q_\phi + \partial_\alpha(D_\phi\partial_\alpha\phi) + E.$$

where E is an unwanted error depending on parameters from the couple flow

$$\begin{aligned}
E &= \partial_\alpha\left(\frac{D_\phi}{c_s^2}\left(u_\alpha q_\phi - \frac{\phi}{\rho}\partial_\alpha p + \phi F_{\rho,\alpha}\right)\right) \\
&= \underbrace{-\partial_\alpha\left(D_\phi\frac{\phi}{p}\partial_\alpha p\right)}_{\text{Non physical}} \\
&+ \underbrace{\partial_\alpha\left(D_\phi\frac{\phi}{p}F_{\rho,\alpha}\right)}_{\text{Non physical}} \\
&+ \underbrace{\frac{u_\alpha}{c_s^2}\partial_\alpha\left(\frac{D_\phi}{c_s^2}q_\phi\right)}_{\text{Small for small Mach numbers}} \\
&+ \underbrace{\frac{q_\phi D_\phi}{c_s^2}\partial_\alpha u_\alpha}_{=0 \text{ if incompressible}} \\
&+ \mathcal{O}(\epsilon^3),
\end{aligned} \tag{3.42}$$

In eq. (3.42) some terms could reasonable be claimed to be small for incompressible, low Mach number flows. However, the first term is not necessarily negligible, as discussed by Krüger et al. [12]. The second term, first derived here, through the force substitution, is also difficult to argue to be small in general.

To conclude, using a staggered formulation and summation properties, the BGK is equivalent to the convection diffusion equation up to the error term E. N.b., compared to the derivations in Krüger et al. [12] no term of the form $\Delta t\partial_t q_\phi$ shows up in the expression. This is a direct consequence of the staggered formulation and significantly improves the accuracy and means that there are no need for complex

treatments of the source term as is otherwise required by the BGK. As noted by Krüger et al. [12], E is not always negligible.

3.3 The collisional operator for BGK in macroscopic variables

The general idea of this section is to show that the BGK expressed in macroscopic quantities is a very cumbersome collisional operator. This will be the motivation for constructing a new operator directly in macroscopic properties later.

From the scale separation in eq. (3.16), without summing, the different orders of $f_{\phi,i}$ can be written as

$$f_{\phi,i}^{(1)} = -\tau_\phi \left(\partial_t^{(1)} f_{\phi,i}^{(0)} + c_{i\alpha} \partial_\alpha^{(1)} f_{\phi,i}^{(0)} - Q_{\phi,i}^{(1)} \right) \tag{3.43}$$

and

$$f_{\phi,i}^{(2)} = -\tau_\phi \left(\partial_t^{(2)} f_{\phi,i}^{(0)} + \partial_t^{(1)} f_{\phi,i}^{(1)} + c_{i\beta} \partial_\beta^{(1)} f_{\phi,i}^{(1)} \right), \tag{3.44}$$

which together form

$$f_{\phi,i} = f_{\phi,i}^{(0)} + f_{\phi,i}^{(1)} + f_{\phi,i}^{(2)} + \mathcal{O}(\epsilon^3). \tag{3.45}$$

Then, the following approximation is inherited from Rohde [35][5],

$$f_{\phi,i} \approx f_{\phi,i}^{(0)} + f_{\phi,i}^{(1)} \tag{3.46}$$

and as a consequence the collisional operator becomes,

$$\begin{aligned} \Omega_{\phi,i}(f_{\phi,i}) &= -\frac{1}{\tau_\phi} \left(f_{\phi,i} - f_{\phi,i}^{(0)} \right) \\ &\approx -\frac{1}{\tau_\phi} f_{\phi,i}^{(1)} \\ &= \partial_t^{(1)} f_{\phi,i}^{(0)} + c_{i\alpha} \partial_\alpha^{(1)} f_{\phi,i}^{(0)} - Q_{\phi,i}^{(1)} \end{aligned} \tag{3.47}$$

where it is known that the different derivatives of the equilibrium distribution correspond to macroscopic properties from the previous Chapman–Enskog expansion.

3.3.1 Time derivatives

The goal now is to reformulate the expression below without any explicit time dependence. That is to replace all time derivatives with spatial derivatives[6]. Taking the first order time derivative of each term in the equilibrium distribution eq. ($f^{(0)}$)

[5] This approximation is however hard to justify, although the conclusions that follow all appear reasonable.

[6] This is to get results similar to Eggels & Somers [30] and Zhou et al. [10] as their final formulation does not have any time dependence.

yields

$$\begin{aligned}\frac{\partial_t^{(1)} f_{\phi,i}^{(0)}(x_\alpha,t)}{\omega_i} &= \overbrace{\partial_t^{(1)}\phi}^{\text{I}} + \frac{c_{i\alpha}}{c_s^2}\overbrace{\partial_t^{(1)}(\phi u_\alpha)}^{\text{II}} \\ &\quad + \frac{1}{2c_s^4}\underbrace{\partial_t^{(1)}\left(\phi(c_{i\alpha}u_\alpha)(c_{i\beta}u_\beta)\right)}_{\text{III}} - \frac{1}{2c_s^2}\underbrace{\partial_t^{(1)}\left(\phi u_\alpha u_\alpha\right)}_{\text{IV}}.\end{aligned} \tag{3.48}$$

in which each term is expanded.

Term I

Rewritten by eq. ($\epsilon^1 c^0$)

$$\partial_t^{(1)}\phi = q_\phi - \partial_\alpha^{(1)}\left(\phi u_\alpha\right), \tag{3.49}$$

resulting in the contribution C_{I} to the collisional operator

$$C_{\text{I}} = q_\phi - \partial_\alpha\left(\phi u_\alpha\right). \tag{3.50}$$

Term II

This is derived by rearranging eq. (3.41)

$$\partial_t^{(1)}\left(\phi u_\alpha\right) = -\partial_\beta^{(1)}(\phi u_\alpha u_\beta) + u_\alpha q_\phi - \frac{\phi}{\rho}\partial_\alpha p + \frac{\phi}{\rho}F_{\rho,\alpha}, \tag{3.51}$$

resulting in the contribution C_{II} to the collisional operator

$$C_{\text{II}} = -\frac{c_{i\alpha}}{c_s^2}\partial_\beta(\phi u_\alpha u_\beta) + \frac{c_{i\alpha}}{c_s^2}u_\alpha q_\phi - c_{i\alpha}\frac{\phi}{p}\partial_\alpha p + c_{i\alpha}\frac{\phi}{p}F_{\rho,\alpha}. \tag{3.52}$$

Term III

Rearranging eq. (Momentum subs.) and, like in Term II, substituting eq. (3.41) yields

$$\begin{aligned}\partial_t^{(1)}\left(\phi u_\alpha u_\beta\right) &= \phi u_\alpha \partial_t^{(1)} u_\beta + u_\beta \partial_t^{(1)}\left(\phi u_\alpha\right) \\ &= \phi u_\alpha\left(-\partial_\gamma^{(1)}\left(u_\beta u_\gamma\right) - \frac{1}{\rho}\partial_\beta p + \frac{F_{\rho,\beta}}{\rho} + u_\beta\partial_\gamma^{(1)}u_\gamma\right) \\ &\quad + u_\beta\left(-\partial_\gamma^{(1)}(\phi u_\alpha u_\gamma) + u_\alpha q_\phi - \frac{\phi}{\rho}\partial_\alpha p + \frac{\phi}{\rho}F_{\rho,\alpha}\right),\end{aligned} \tag{3.53}$$

resulting in the contribution C_{III} to the collisional operator

$$\begin{aligned}C_{\text{III}} &= \frac{c_{i\alpha}c_{i\beta}}{2c_s^2}\frac{\phi}{p}(u_\alpha F_{\rho,\beta} + u_\beta F_{\rho,\alpha} - u_\beta\partial_\alpha p - u_\alpha\partial_\beta p) \\ &\quad + \frac{c_{i\alpha}c_{i\beta}}{2c_s^4}(q_\phi u_\alpha u_\beta + \phi u_\alpha u_\beta\partial_\gamma u_\gamma - \phi u_\alpha\partial_\gamma(u_\beta u_\gamma) - u_\beta\partial_\gamma(\phi u_\alpha u_\gamma)).\end{aligned} \tag{3.54}$$

Term IV

Now using the product rule and Term II,

$$\begin{aligned}\partial_t^{(1)}\left(\phi u_\alpha u_\alpha\right) &= u_\alpha \partial_t^{(1)}(\phi u_\alpha) + \phi u_\alpha \partial_t^{(1)} u_\alpha \\ &= u_\alpha \left(-\partial_\gamma^{(1)}(\phi u_\alpha u_\gamma) + u_\alpha q_\phi - \frac{\phi}{\rho}\partial_\alpha p + \frac{\phi}{\rho} F_{\rho,\alpha}\right) \\ &+ \phi u_\alpha \left(-\partial_\gamma^{(1)}\left(u_\alpha u_\gamma\right) - \frac{1}{\rho}\partial_\alpha p + \frac{F_{\rho,\alpha}}{\rho} + u_\alpha \partial_\gamma^{(1)} u_\gamma\right),\end{aligned} \tag{3.55}$$

resulting in the contribution C_{IV} to the collisional operator, remembering the minus sign multiplied with the derivatives

$$\begin{aligned}C_{\mathrm{IV}} = &\frac{\phi u_\alpha}{p}(F_{\rho,\alpha} - \partial_\alpha p) \\ &- \frac{1}{2c_s^2}(q_\phi u_\alpha u_\alpha + \phi u_\alpha u_\alpha \partial_\gamma u_\gamma - \phi u_\alpha \partial_\gamma (u_\alpha u_\gamma) - u_\alpha \partial_\gamma (\phi u_\alpha u_\gamma)).\end{aligned} \tag{3.56}$$

3.3.2 Spatial derivative

Now, all the time derivatives have been formulated as spatial derivatives. The remaining term in eq. (3.47) is

$$\begin{aligned}\frac{c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(0)}(x_\alpha, t)}{\omega_i} &= c_{i\alpha}\partial_\alpha^{(1)}\phi + \frac{c_{i\alpha}c_{i\beta}}{c_s^2}\partial_\alpha^{(1)}(\phi u_\beta) \\ &+ \frac{c_{i\alpha}c_{i\beta}c_{i\gamma}}{2c_s^4}\partial_\alpha^{(1)}(\phi u_\beta u_\gamma) - \frac{c_{i\alpha}}{2c_s^2}\partial_\alpha^{(1)}(\phi u_\beta u_\beta)\end{aligned}, \tag{3.57}$$

resulting in the contribution C_{Spatial} to the collisional operator

$$\begin{aligned}C_{\mathrm{Spatial}} = &c_{i\alpha}\partial_\alpha \phi + \frac{c_{i\alpha}c_{i\beta}}{c_s^2}\partial_\alpha(\phi u_\beta) \\ &+ \frac{c_{i\alpha}c_{i\beta}c_{i\gamma}}{2c_s^4}\partial_\alpha(\phi u_\beta u_\gamma) - \frac{c_{i\alpha}}{2c_s^2}\partial_\alpha(\phi u_\beta u_\beta).\end{aligned} \tag{3.58}$$

3.3.3 Source discretization

In eq. (3.47) the expression for $Q_{\phi,i}^{(1)}$ appears and has to be discretized. Here, the simplest discretization suggested by Krüger et al. [12] is used, namely $Q_{i,\phi} = \omega_i q_\phi$. This results in the contribution C_{Spatial} to the collisional operator

$$C_{\mathrm{Source}} = -q_\phi. \tag{3.59}$$

3.3.4 Resubstituting in BGK

Combining all of these expression finally yields the sum of first order spatial and temporal derivatives of the second order equilibrium function.

The BGK expanded into macroscopic variables

$$
\begin{aligned}
\Omega_{\mathrm{BGK},\phi,i}/\omega_i \approx\ & C_{\mathrm{I}} + C_{\mathrm{II}} + C_{\mathrm{III}} + C_{\mathrm{IV}} + C_{\mathrm{Spatial}} + C_{\mathrm{Source}} \\
=\ & q_\phi - \partial_\alpha(\phi u_\alpha) \\
& - \frac{c_{i\alpha}}{c_s^2}\partial_\beta(\phi u_\alpha u_\beta) + \frac{c_{i\alpha}}{c_s^2}u_\alpha q_\phi - c_{i\alpha}\frac{\phi}{p}\partial_\alpha p + c_{i\alpha}\frac{\phi}{p}F_{\rho,\alpha} \\
& + \frac{c_{i\alpha}c_{i\beta}}{2c_s^2}\frac{\phi}{p}(u_\alpha F_{\rho,\beta} + u_\beta F_{\rho,\alpha} - u_\beta\partial_\alpha p - u_\alpha\partial_\beta p) \\
& + \frac{c_{i\alpha}c_{i\beta}}{2c_s^4}(q_\phi u_\alpha u_\beta + \phi u_\alpha u_\beta\partial_\gamma u_\gamma - \phi u_\alpha\partial_\gamma(u_\beta u_\gamma) - u_\beta\partial_\gamma(\phi u_\alpha u_\gamma)) \\
& + \frac{\phi u_\alpha}{p}(F_{\rho,\alpha} - \partial_\alpha p) \\
& - \frac{1}{2c_s^2}(q_\phi u_\alpha u_\alpha + \phi u_\alpha u_\alpha\partial_\gamma u_\gamma - \phi u_\alpha\partial_\gamma(u_\alpha u_\gamma) - u_\alpha\partial_\gamma(\phi u_\alpha u_\gamma)) \\
& + c_{i\alpha}\partial_\alpha\phi + \frac{c_{i\alpha}c_{i\beta}}{c_s^2}\partial_\alpha(\phi u_\beta) \\
& + \frac{c_{i\alpha}c_{i\beta}c_{i\gamma}}{2c_s^4}\partial_\alpha(\phi u_\beta u_\gamma) - \frac{c_{i\alpha}}{2c_s^2}\partial_\alpha(\phi u_\beta u_\beta) \\
& - q_\phi \\
& + \mathcal{O}(\epsilon^2).
\end{aligned}
\tag{3.60}
$$

In eq. (3.60) the only terms that cancels are the source contribution q_ϕ, which is logical since the discretized source $Q_{\phi,i}$ accounts for that effect. Some terms can be combined using derivative properties. However, this does not significantly simplify the expression. The physical intuition for inidviual terms is also limited, some could be neglected for different flow types, such as a sourceless, forceless incompressible regime, but still so called *higher order terms* such as $\phi u_\alpha u_\beta \partial_\gamma u_\gamma$ cannot easily be argued to be insignificant.

It should also be noted that this is only an approximation as terms corresponding to eq. (3.44) have been neglected. This approximation is hard to justify from a physical or numerical point of view. Despite this simplification, the BGK collisional operator expands into something very complicated, which will motivate the filtering done for the rest of this chapter.

3.4 Determining unneeded components

Now returning to the scale separation for a staggered solution in eq. (3.14), i.e. the collisional operator has not yet been chosen. Summing $\sum_i$ in both ϵ and ϵ^2 gives the equations, remembering $\sum_i \Omega^{(1)}_{\phi,i} = \sum_i \Omega^{(2)}_{\phi,i} = 0$

$$
\partial_t^{(1)}\phi + \partial_\alpha^{(1)}(\phi u_\alpha) - q_\phi = \sum_i \Omega^{(1)}_{\phi,i} = 0 \qquad (\epsilon^1 c^0 - \Omega)
$$

$$\partial_t^{(2)}\phi + \sum_i c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(1)} = \sum_i \Omega_{\phi,i}^{(2)} = 0. \qquad (\epsilon^2 c^0 - \Omega)$$

Similarly to the BGK calculations, these expressions are summed and $\partial_\alpha^{(1)} = \partial_\alpha$ is substituted[7]. Then the expression becomes

$$\partial_t\phi + \partial_\alpha(\phi u_\alpha) = q_\phi - \partial_\alpha \sum_i c_{i\alpha} f_{\phi,i}^{(1)}, \qquad (3.61)$$

considering the exact convection diffusion equation eq. (Convection diffusion equation), here rewritten for convenience,

$$\partial_t\phi + \partial_\alpha(\phi u_\alpha) = q_\phi + \partial_\alpha(D_\phi \partial_\alpha \phi). \qquad \text{(Convection diffusion equation)}$$

it is clear that the equations match perfectly if and only if

$$\sum_i c_{i\alpha} f_{\phi,i}^{(1)} = -D_\phi \partial_\alpha \phi, \qquad \text{(ansatz)}$$

Assuming that eq. ($\sum \omega c^2$) is satisfied by the velocity sets, then one solution is $f_{\phi,i}^{(1)} = -\omega_i c_{i\beta} \dfrac{D_\phi}{c_s^2} \partial_\beta \phi$ since

$$\begin{aligned}
\sum_i c_{i\alpha} \left(-\omega_i c_{i\beta} \frac{D_\phi}{c_s^2} \partial_\beta \phi \right) &= -\frac{D_\phi}{c_s^2} \partial_\beta \phi \sum_i \omega_i c_{i\alpha} c_{i\beta} \\
&= -c_s^2 \delta_{\alpha\beta} \frac{D_\phi}{c_s^2} \partial_\beta \phi \\
&= -D_\phi \partial_\alpha \phi,
\end{aligned}$$

it is also clear that any homogenous solution, i.e. $\sum_i c_{i\alpha} f_{\phi,i}^{(1)} = 0$ could be included in $f_{\phi,i}^{(1)}$. From eqs. ($\sum \omega c$) and ($\sum \omega c^3$) it is clear that any expression of the form

$$f_{\text{Homogenous},\phi,i}^{(1)} = \omega_i A_0 + \omega_i c_{i\alpha} c_{i\beta} B_{\alpha\beta} \qquad (3.62)$$

satisfies this property. This insight gives some information of which terms in eq. (3.60) are actually corresponding to the similarity between the BGK and the convection–diffusion equation.

3.5 Filtering the collisional operator

The assumption in this section are all similar to those by Rohde [35], and it is clear that the result is the same as that of Zhou et al. [10] although with the added source term. However, the assumptions are made more out of convenience that based on physical insights.

[7] Non of the operations have required any more knowledge about the collisional operator than the sums.

From section 3.4 it is clear what is needed from the non-equilibrium distribution. The idea is to construct the collisional operator just like how it is done for the BGK. The scale separation in eq. (3.14) gives the collisional operator based on the non-equilibrium distributions. Writing them explicitly here

$$\Omega_{\phi,i}^{(1)} = -\frac{1}{\tau_\phi} f_{\phi,i}^{(1)} = \partial_t^{(1)} f_{\phi,i}^{(0)} + c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(0)} - Q_{\phi,i}^{(1)} \tag{3.63}$$

and

$$\Omega_{\phi,i}^{(2)} = -\frac{1}{\tau_\phi} f_{\phi,i}^{(2)} = \partial_t^{(2)} f_{\phi,i}^{(0)} + \partial_t^{(1)} f_{\phi,i}^{(1)} + c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(1)}. \tag{3.64}$$

However, once again the approximation $\Omega_{\phi,i} \approx \Omega_{\phi,i}^{(1)}$ is used similarly to Rohde [35]. As seen previously one of the terms in $\partial_t^{(1)} f_{\phi,i}^{(0)}$ is $\omega_i q_\phi$ and in $c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(0)}$ the term $\omega_i c_{i\alpha}\partial_\alpha\phi$ appears. Looking back at the definition for the approximated BGK collisional operator eq. (3.47) it is clear that only using these terms and the source discretization gives a collisional operator

$$\begin{aligned} \Omega_{\phi,i} &= \omega_i c_{i\alpha}\partial_\alpha\phi \\ Q_{\phi,i} &= \omega_i q_\phi \\ f_{\phi,i}^{(1)} &= -\omega_i c_{i\beta}\frac{D_\phi}{c_s^2}\partial_\beta\phi \end{aligned} \qquad (\Omega, Q, f^{(1)})$$

and because the conserved and non-conserved properties, the properties $\sum_i \Omega_{\phi,i} = 0$, $\sum_i Q_{\phi,i} = q_\phi$ and $\sum_i c_{i\alpha} Q_{\phi,i} = 0$ must be satisfied, which they clearly are because of eqs. $(\sum\omega)$ to $(\sum\omega c^2)$. Because only a few terms were kept from the full expansion of the BGK operator, whist having the same relation between non-equilibrium distribution and it could be argued that this is a form of filtering of unneeded terms.

3.6 Constructing the moment vector

To calculate how the population density function changes through the non-equilibrium contributions and collisional operator, the population is extrapolated with a first order Taylor expansion as

$$f_{\phi,i}^{\pm} = f_{\phi,i}(x_\alpha, t) \pm \frac{\Delta t}{2}\left(\Omega_{\phi,i} + Q_{\phi,i}\right) \tag{3.65}$$

where the following notation was used

$$f_{\phi,i}^{\pm} = f_{\phi,i}\left(x_\alpha \pm c_{i\alpha}\frac{\Delta t}{2}, t \pm \frac{\Delta t}{2}\right). \tag{3.66}$$

Substituting the approximation that only the first order non-equilibrium distribution matters yields

$$\begin{aligned} f_{\phi,i}^{\pm} &\approx \left(f_{\phi,i}^{(0)}(x_\alpha, t) + f_{\phi,i}^{(1)}(x_\alpha, t)\right) \pm \frac{\Delta t}{2}\left(-\frac{1}{\tau_\phi} f_{\phi,i}^{(1)} + Q_{\phi,i}\right) \\ &= f_{\phi,i}^{(0)}(x_\alpha, t) - \tau_\phi\left(\partial_t^{(1)} f_{\phi,i}^{(0)} + c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(0)} - Q_{\phi,i}^{(1)}\right) \pm \frac{\Delta t}{2}\left(\partial_t^{(1)} f_{\phi,i}^{(0)} + c_{i\alpha}\partial_\alpha^{(1)} f_{\phi,i}^{(0)}\right), \end{aligned}$$

and substituting the properties in eq. $(\Omega, Q, f^{(1)})$

$$f^{\pm}_{\phi,i} = \left(\omega_i \phi \left(1 + \frac{c_{i\alpha}u_\alpha}{c_s^2} + \frac{1}{2}\frac{(c_{i\alpha}u_\alpha)^2}{c_s^4} - \frac{1}{2}\frac{u_\alpha u_\alpha}{c_s^2}\right) - \frac{D_\phi}{c_s^2}\left(\omega_i c_{i\alpha}\partial_\alpha \phi\right)\right) \pm \frac{\Delta t}{2}\left(\omega_i c_{i\alpha}\partial_\alpha \phi + \omega_i q_\phi\right) \tag{3.67}$$

grouping terms results in

$$\begin{aligned} f^{\pm}_{\phi,i} = {} & \omega_i \left(\phi \pm q_\phi \frac{\Delta t}{2}\right) \\ & + \omega_i \frac{c_{i\alpha}}{c_s^2}\left(\phi u_\alpha + \partial_\alpha \phi \left(-D_\phi \pm \frac{c_s^2 \Delta t}{2}\right)\right) \\ & + \omega_i \frac{1}{2c_s^4}\left(c_{i\alpha}c_{i\beta} - c_s^2 \delta_{\alpha\beta}\right)\phi u_\alpha u_\beta \end{aligned}$$

which conveniently can be written as a product

$$f^{\pm}_{\phi,i} = \omega_i \left(\begin{bmatrix} 1_i \\ \frac{c_{i\alpha}}{c_s^2} \\ \frac{1}{2c_s^4}\left(c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta}\right) \end{bmatrix}^T \cdot \begin{bmatrix} \phi \pm q_\phi \frac{\Delta t}{2} \\ \phi u_\alpha + \partial_\alpha \phi \left(-D_\phi \pm \frac{c_s^2 \Delta t}{2}\right) \\ \phi u_\alpha u_\beta \end{bmatrix}\right). \tag{3.68}$$

although since the last term is unaffected by the collisional operator it will be discarded[8]

$$f^{\pm}_{\phi,i} = \omega_i \underbrace{\begin{bmatrix} 1_i \\ \frac{c_{i\alpha}}{c_s^2} \end{bmatrix}^T}_{E_{ik}} \cdot \underbrace{\begin{bmatrix} \phi \pm q_\phi \frac{\Delta t}{2} \\ \phi u_\alpha + \partial_\alpha \phi \left(-D_\phi \pm \frac{c_s^2 \Delta t}{2}\right) \end{bmatrix}}_{\alpha^{\pm}_k}. \tag{3.69}$$

This in the same form as that used by Zhou et al. [10], although the matrix E_{ik} here is not square, since it does not contain any higher order Hermite basis vectors. There are several important things to note at this point,

1. Neglecting $f^{(2)}_{\phi,i}$ was not justified physically or mathematically
2. There was no justification for reusing the BGK relation between collisional operator.
3. Although it is clear which terms were needed from section 3.4, it was not clear how these relate neglecting terms in the derivative of the equilibrium function.

For the reasons listed above, it is unclear whether this expression still corresponds to the staggered discretization. I.e. in some sense the E-term in eq. (3.42) from the Chapman–Enskog expansion was moved from calculating the wrong equation to calculating the right equation on the wrong discretization. This effect have not been quantified.

[8] This will not be the case for the momentum equations later.

The discussion in section 3.4 gives hope though, that reasoning was *constructive*, the desired partial differential equation was used as an input to derive the non-equilibrium distribution. Traditional Chapman–Enskog analysis of the BGK reason in the opposite way, a collisional operator is chosen, which is then shown to be equivalent to the partial differential equation.

The identification of the basis vector and α representation is also an important realization. Maybe this change of basis could be done in an earlier stage of the derivation, which would allow for a reasoning similar to section 3.4, but also for the collisional operator? This will be explored in the following chapter of this thesis.

4

Derivation of Filter Matrix Methods

In this chapter, a modified version of Filter Matrix method for Lattice Boltzmann simulations is derived by directly projecting the discretized Boltzmann equation onto a Hermite polynomial base. This allows makes it possible to use the convection–diffusion equation and Navier Stokes equations as an "input" to determine a collisional operator that gives an exact match for the Chapman–Enskog expansion.

4.1 Derivation of transformation matrix for a general lattice

The factorization form section 3.6 leads to the realization that the equilibrium distribution, as written in eq. (2.7), can be written as

$$f_{\phi,i}^{(0)} = \omega_i \odot \left(\begin{bmatrix} 1_i \\ \frac{c_{i\alpha}}{c_s^2} \\ \frac{1}{2c_s^4}\left(c_{i\alpha} \odot c_{i\beta} - c_s^2 \delta_{\alpha\beta} 1_i\right) \end{bmatrix}^T \cdot \begin{bmatrix} \phi \\ \phi u_\alpha \\ \phi u_\alpha u_\beta \end{bmatrix} \right). \tag{4.1}$$

which is the dot product between two vectors and 1_i indicates a vector of ones. Now, a generic population vector will be denoted f_i and a vector of macroscopic quantities as α_k. Further, the right most vector is similar to the Hermite polynomials from Shan et al. [23] in eqs. ($\mathscr{H}^{(0)}$) to ($\mathscr{H}^{(2)}$), only rescaled with different constant multiples due to the full expansion[1] in eq. (2.22).

$$\mathcal{H}_i^{(0)} = 1_i \tag{4.2}$$

$$\mathcal{H}_{\alpha,i}^{(1)} = \frac{c_{i\alpha}}{c_s^2} \tag{4.3}$$

$$\mathcal{H}_{\alpha\beta,i}^{(2)} = \frac{1}{2c_s^4}\left(c_{i\alpha} \odot c_{i\beta} - c_s^2 \delta_{\alpha\beta} 1_i\right) \tag{4.4}$$

as seen in eq. (2.24), the third order Hermite polynomial is given by

$$\mathcal{H}_{\alpha\beta\gamma,i}^{(3)} = \frac{1}{6c_s^6}\left(c_{i\alpha} \odot (c_{i\beta} \odot c_{i\gamma}) - c_s^2 \left(c_{i\alpha}\delta_{\beta\gamma} + c_{i\beta}\delta_{\alpha\gamma} + c_{i\gamma}\delta_{\beta\alpha}\right)\right). \tag{4.5}$$

[1] Note that a different scaling could be used, and indeed Zhou et al. [10] uses a slightly different scaling, but this does not change the derivation procedure.

For the second and third Hermite polynomials there are some important caveats. The main one is symmetry, e.g. $\mathcal{H}^{(2)}_{xy,i} = \mathcal{H}^{(2)}_{yx,i}$ which means that these two polynomials are not orthogonal. This creates some syntactical inconveniences as discussed in appendix A. For a grid with $c_s^2 = 1/3$ a polynomial like $\mathcal{H}^{(3)}_{xxx,i}$ is not linearly independent to the others, although $\mathcal{H}^{(3)}_{xxy,i}$ is. This is probably related to the specific of choice some of the higher order terms in Somers [29], Eggels & Somers [30] and Zhou et al. [10].

Extending this reasoning to a more generic distribution, eq. (4.1) can be further generalized as

$$f_i = \omega_i \odot \underbrace{\begin{bmatrix} | & | & | \\ \mathcal{H}^{(0)}_i & \mathcal{H}^{(1)}_{\alpha,i} & \mathcal{H}^{(2)}_{\alpha\beta,i} \\ | & | & | \end{bmatrix}}_{E_{ik}} \cdot \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_{1\alpha} \\ \alpha_{2\alpha\beta} \end{bmatrix}}_{\alpha_k}, \tag{4.6}$$

although here the tensors for the second order terms are all a major issue. The matrix E_{ik} must be orthogonal in order to be inverted, but removing one of the terms such as $\mathcal{H}^{(2)}_{xy,i} = \mathcal{H}^{(2)}_{yx,i}$ leads to the wrong computation. However, because of the symmetry, this missing computation just leads to doubling the contribution in the double contraction from the other term, c.f. appendix A. Thus the following notation is introduced

$$\mathcal{H}^{(2)}_{\tilde{\alpha}\tilde{\beta},i} = \frac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4}\left(c_{i\tilde{\alpha}} \odot c_{i\tilde{\beta}} - c_s^2 \delta_{\tilde{\alpha}\tilde{\beta}} 1_i\right) \tag{4.7}$$

which can be thought of as the unordered set, e.g. for two dimensions $\{\tilde{\alpha}\tilde{\beta}\} = \{\{x,x\},\{x,y\},\{y,y\}\}$. It can be thought to behave identical except for two dimensional double contractions. In order to preserve the computation in eq. (4.1) when reducing eq. (4.6), the "missing" computation for the symmetrical elements are would be added in $\mathcal{H}^{(2)}_{\tilde{\alpha}\tilde{\beta},i}$

$$f_i = \omega_i \odot \underbrace{\begin{bmatrix} | & | & | \\ \mathcal{H}^{(0)}_i & \mathcal{H}^{(1)}_{\alpha,i} & \mathcal{H}^{(2)}_{\tilde{\alpha}\tilde{\beta},i} \\ | & | & | \end{bmatrix}}_{E_{ik}} \cdot \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_{1\alpha} \\ \alpha_{2\tilde{\alpha}\tilde{\beta}} \end{bmatrix}}_{\alpha_k}, \tag{4.8}$$

where the moment vector α_k can be written explicitly in two and three dimensions as

$$\alpha_{2\mathrm{D},k} = \begin{bmatrix} \alpha_0 & \alpha_{1x} & \alpha_{1y} & \alpha_{2xx} & \alpha_{2xy} & \alpha_{2yy} \end{bmatrix}^T, \tag{4.9}$$

$$\alpha_{3\mathrm{D},k} = \begin{bmatrix} \alpha_0 & \alpha_{1x} & \alpha_{1y} & \alpha_{1z} & \alpha_{2xx} & \alpha_{2xy} & \alpha_{2xz} & \alpha_{2yy} & \alpha_{2yz} & \alpha_{2zz} \end{bmatrix}^T, \tag{4.10}$$

where once again, there symmetry in the Hermite polynomials means that the same symmetry must be present in the moment vectors components, e.g. $\alpha_{xy} = \alpha_{yx}$.

It is clear that in the case of a equilibrium distribution, equivalently to the definition of $f^{(0)}_{\phi,i}$, an equilibrium moment $\alpha^{(0)}_{\phi,i}$ can be defined as

$$\alpha^{(0)}_{\phi,i} = \begin{bmatrix} \phi \\ \phi u_\alpha \\ \phi u_{\tilde{\alpha}} u_{\tilde{\beta}} \end{bmatrix} \tag{4.11}$$

where the $\tilde{\alpha}, \tilde{\beta}$-notation is purely a reminder that the symmetric permutations of α and β are only considered once. However, it will later be seen that the concept of an equilibrium moment will only be used as an inspiration to guess solutions to a differential equation for the actual moment vectors that will be used for the Filter Matrix method.

Based on the discussion of Hermite polynomials and their properties in Shan et al. [23], the inner product of the space spanned by Hermite polynomials is defined through the following inner product

$$\langle A_i, B_i \rangle = (\omega_i \odot A_i) \cdot B_i = A_i \cdot (\omega_i \odot B_i) = \sum_i \omega_i A_i B_i. \tag{4.12}$$

This product is here extended for matrices as well. There are two ways to do this, either by doing the elementwise multiplication on the left hand matrix or simultaneously on both matrices. When it is done simultaneously, the multiplication dimension must match that of ω_i, i.e. below the $\max i$ must correspond to the $\max i$ for ω_i

$$\begin{aligned} \left\langle \left\langle \underbrace{A_{ji}}_{\max j \times \max i}, \underbrace{B_{ik}}_{\max i \times \max k} \right\rangle \right\rangle &= \underbrace{\begin{bmatrix} \langle A_{0i}, B_{i0} \rangle & \cdots & \langle A_{0i}, B_{i\max(k)} \rangle \\ \vdots & \ddots & \vdots \\ \langle A_{\max(j)i}, B_{i0} \rangle & \cdots & \langle A_{\max(j)i}, B_{i\max(k)} \rangle \end{bmatrix}}_{\max j \times \max k} \\ &= (\omega_i \odot A_{ji}) \cdot B_{ik} \\ &= A_{ji} \cdot (\omega_i \odot B_{ik}) \end{aligned} \tag{4.13}$$

The other way of performing the inner product on matrices are defined as below. Here only the first matrix has to match the dimension of ω_i,

$$\begin{aligned} \left\langle \underbrace{A_{ij}}_{\max i \times \max j}, \underbrace{B_{jk}}_{\max j \times \max k} \right\rangle &= \begin{bmatrix} \omega_0(A_{0j} \cdot B_{j0}) & \cdots & \omega_0(A_{0j} \cdot B_{j\max(k)}) \\ \vdots & \ddots & \vdots \\ \omega_{\max(i)}(A_{\max(i)j} \cdot B_{j0}) & \cdots & \omega_{\max(i)}(A_{\max(i)j} \cdot B_{j\max(k)}) \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} | & & | \\ (\omega_i \odot A_{ij}) \cdot B_{j0} & \cdots & (\omega_i \odot A_{ij}) \cdot B_{j\max(k)} \\ | & & | \end{bmatrix}}_{\max i \times \max k} \\ &= (\omega_i \odot A_{ij}) \cdot B_{jk} \end{aligned} \tag{4.14}$$

From this it is also clear that the Hadamard product between a vector and a matrix, $\omega_i \odot A_{ij}$, is defined as performing the product along the j-index like

$$\omega_i \odot A_{ij} = \begin{bmatrix} \omega_i \odot A_{i0}, & \cdots, & \omega_i \odot A_{i\max(j)} \end{bmatrix}. \tag{4.15}$$

Carrying on this notation, because the Hermite polynomials form an orthogonal basis, any two *different* Hermite polynomials $\mathcal{H}_{A,i}$ and $\mathcal{H}_{B,i}$ are orthogonal, meaning

$$\langle \mathcal{H}_{A,i}, \mathcal{H}_{B,i} \rangle = 0, \tag{4.16}$$

however since the chosen Hermitian basis vectors are not normed, their magnitude is not necessarily unity, i.e. $\langle \mathcal{H}_{A,i}, \mathcal{H}_{A,i} \rangle \neq 1$. These magnitudes are important for inverting and as such calculated up to a second order expansion in appendix B. This means that the product of the basis vector-matrix transpose E_{ki} and itself E_{ik}, a so called Gram matrix, is given by

$$\begin{aligned} D &= \langle\langle E_{ki}, E_{ik} \rangle\rangle \\ &= \text{diagonal}\left[1, \frac{1_\alpha}{c_s^2}, \frac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4}\right] \end{aligned} \tag{4.17}$$

and because it is a diagonal matrix it is trivially invertible,

$$D^{-1} = \text{diagonal}\left[1, 1_\alpha c_s^2, (1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4\right] \tag{4.18}$$

Now, using this notation it is possible to define transforms between the population densities f_i and the (macroscopic) moment vectors α_k. The eq. (4.8) can already be seen as such a transform, changing the notation yields

$$\mathcal{T}_{\alpha \to f}(\alpha_k) = (\omega_i \odot E_{ik}) \cdot \alpha_k = \langle E_{ik}, \alpha_k \rangle = f_i. \tag{4.19}$$

and then the inverse transform is given by, where conventional matrix multiplication is written out explicitly with $\cdot$ for clarity

$$\begin{aligned} D^{-1}E_{ki} \cdot f_i &= D^{-1} \cdot E_{ki} \cdot ((\omega_i \odot E_{ik}) \cdot \alpha_k) \\ &= D^{-1} \cdot (E_{ki} \cdot (\omega_i \odot E_{ik})) \cdot \alpha_k \\ &= D^{-1} \cdot \langle\langle E_{ki}, E_{ik} \rangle\rangle \cdot \alpha_k \\ &= D^{-1} \cdot D \cdot \alpha_k \\ &= \alpha_k \end{aligned}$$

meaning that the inverse transform looks like

$$\mathcal{T}_{f \to \alpha}(f_i) = D^{-1} \cdot E_{ki} \cdot f_i = \alpha_k, \tag{4.20}$$

which has the transformation matrix

$$\begin{aligned} D^{-1} \cdot E_{ki} &= \begin{bmatrix} - & \mathcal{H}_i^{(0)} & - \\ - & c_s^2 \mathcal{H}_{\alpha,i}^{(1)} & - \\ - & (1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4 \mathcal{H}_{\tilde{\alpha}\tilde{\beta},i}^{(2)} & - \end{bmatrix} \\ &= \begin{bmatrix} - & 1_i & - \\ - & c_{i\alpha} & - \\ - & \left(c_{i\tilde{\alpha}} \odot c_{i\tilde{\beta}} - c_s^2 \delta_{\tilde{\alpha}\tilde{\beta}} 1_i\right) & - \end{bmatrix} \end{aligned} \tag{4.21}$$

and similarly the other transform can be written as

$$\begin{aligned}\omega_i E_{ik} &= \omega_i \odot \begin{bmatrix} | & | & | \\ \mathcal{H}_i^{(0)} & \mathcal{H}_{\alpha,i}^{(1)} & \mathcal{H}_{\tilde{\alpha}\tilde{\beta},i}^{(2)} \\ | & | & | \end{bmatrix} \\ &= \omega_i \odot \begin{bmatrix} | & | & | \\ 1_i & \dfrac{c_{i\alpha}}{c_s^2} & \dfrac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4}\left(c_{i\tilde{\alpha}} \odot c_{i\tilde{\beta}} - c_s^2 \delta_{\tilde{\alpha}\tilde{\beta}} 1_i\right) \\ | & | & | \end{bmatrix} \\ &= \begin{bmatrix} | & | & | \\ \omega_i & \dfrac{\omega_i \odot c_{i\alpha}}{c_s^2} & \dfrac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4}\left(\omega_i \odot c_{i\tilde{\alpha}} \odot c_{i\tilde{\beta}} - c_s^2 \delta_{\tilde{\alpha}\tilde{\beta}} \omega_i\right) \\ | & | & | \end{bmatrix}\end{aligned} \tag{4.22}$$

These transformations between a lattice speed representation and a moment representation is similar to what is done for MRT operators, c.f. section 2.5, and how the E_{ki} and E_{ik} matrices are defined in Zhou et al. [10]. However, in both those cases, the matrices are related by "$\omega_i E_{ik} = E_{ki}^{-1}$", which means that an inverse has to exist and thus for the matrix to be square, in this thesis, $E_{ik} = E_{ki}^T$ and the inverse can be calculated using the D-matrix. The most important consequence of this is that the number of used modes does not have to correspond to the amount of lattice directions, i.e. $\max i \neq \max k$, meaning that the number of computations can be reduced significantly.

Using this method a term like $(c_{i\alpha}\partial_\alpha) \odot f_i$ can be represented in moment space. The lengthy calculation can be found in appendix C. The conclusion is

$$\mathcal{T}_{f\to\alpha}\left(c_{i\alpha}\partial_\alpha f_i\right) = \begin{bmatrix} \partial_\alpha \alpha_{1\alpha} \\ c_s^2 \partial_\alpha \alpha_0 + \partial_\beta \alpha_{2\alpha\beta} \\ c_s^2\left(\partial_\alpha \alpha_{1\beta} + \partial_\beta \alpha_{1\alpha}\right) \end{bmatrix}, \qquad \left(\mathcal{T}_{f\to\alpha}\left(c_{i\alpha}\partial_\alpha f_i\right)\right)$$

in which the symmetrical notation has been be dropped since there are no double contractions being performed.

4.2 Transforming the Boltzmann equation to moment space

In this section it will be shown that the entire Enskog analysis can be performed in terms of α_k, meaning that no sums with dubious physical meaning has to be defined, everything will be defined in terms of macroscopic variables, which could then be transformed back into lattice velocities.

The starting point is the staggered formulation of the Boltzmann equation, i.e. eq. (3.7). This time the the population density is simply split into a non-equilibrium and equilibrium part as $f_{\phi,i} = f_{\phi,i}^{\text{eq}} + f_{\phi,i}^{\text{neq}}$ with the equilibrium distribution derived as previously. Due to the linearity of the transform, this is not an assumption, it

is simply a rewriting that happens to make it easy to find solutions to some of the differential equations that appears as intermediate steps. This separation looks like

$$\begin{aligned}\Omega_{\phi,i}(f_{\phi,i}) = \partial_t f^{\mathrm{eq}}_{\phi,i}(x_\alpha,t) + (c_{i\alpha}\partial_\alpha) \odot f^{\mathrm{eq}}_{\phi,i}(x_\alpha,t) \\ + \partial_t f^{\mathrm{neq}}_{\phi,i}(x_\alpha,t) + (c_{i\alpha}\partial_\alpha) \odot f^{\mathrm{neq}}_{\phi,i}(x_\alpha,t).\end{aligned} \tag{4.23}$$

Now, it is simply transformed into the moment space, i.e. to α_k-vectors, using eq. (4.20) which is a linear operator applied termwise. In this case, the decomposition of $f_{\phi,i}$ corresponds to the moment decomposition $\alpha_{\phi,k} = \alpha^{\mathrm{eq}}_{\phi,k} + \alpha^{\mathrm{neq}}_{\phi,k}$. The collisional operator in the moment space is named $\mho$. Importantly, eq. ($\mathcal{T}_{f\to\alpha}\left(c_{i\alpha}\partial_\alpha f_i\right)$) is used to transform the spatial derivatives. The termwise transform looks like

$$\begin{aligned}\mathcal{T}_{f\to\alpha}(\Omega_{\phi,i}(f_{\phi,i})) = \mathcal{T}_{f\to\alpha}(\partial_t f^{\mathrm{eq}}_{\phi,i}(x_\alpha,t)) + \mathcal{T}_{f\to\alpha}((c_{i\alpha}\partial_\alpha) \odot f^{\mathrm{eq}}_{\phi,i}(x_\alpha,t)) \\ + \mathcal{T}_{f\to\alpha}(\partial_t f^{\mathrm{neq}}_{\phi,i}(x_\alpha,t)) + \mathcal{T}_{f\to\alpha}((c_{i\alpha}\partial_\alpha) \odot f^{\mathrm{neq}}_{\phi,i}(x_\alpha,t)).\end{aligned} \tag{4.24}$$

From Shan et al. [23] and Krüger et al. [12], it is expected that a second order Hermite expansion is sufficient to recover the Navier-Stokes and the convection–diffusion equation. The time derivatives are simple to transform due to the linearity of the operator and the spatial derivatives in eq. ($\mathcal{T}_{f\to\alpha}\left(c_{i\alpha}\partial_\alpha f_i\right)$)

The discretized Boltzmann equation in moment space

$$\begin{aligned}\begin{bmatrix}\mho_{\phi,0}\\ \mho_{\phi,1\alpha}\\ \mho_{\phi,2\alpha\beta}\end{bmatrix} &= \partial_t \begin{bmatrix}\alpha^{\mathrm{eq}}_{\phi,0}\\ \alpha^{\mathrm{eq}}_{\phi,1\alpha}\\ \alpha^{\mathrm{eq}}_{\phi,2\alpha\beta}\end{bmatrix} + \begin{bmatrix}\partial_\alpha \alpha^{\mathrm{eq}}_{1\alpha}\\ c_s^2\partial_\alpha\alpha^{\mathrm{eq}}_0 + \partial_\beta\alpha^{\mathrm{eq}}_{2\alpha\beta}\\ c_s^2\left(\partial_\alpha\alpha^{\mathrm{eq}}_{1\beta} + \partial_\beta\alpha^{\mathrm{eq}}_{1\alpha}\right)\end{bmatrix} \\ &\quad + \partial_t \begin{bmatrix}\alpha^{\mathrm{neq}}_{\phi,0}\\ \alpha^{\mathrm{neq}}_{\phi,1\alpha}\\ \alpha^{\mathrm{neq}}_{\phi,2\alpha\beta}\end{bmatrix} + \begin{bmatrix}\partial_\alpha \alpha^{\mathrm{neq}}_{1\alpha}\\ c_s^2\partial_\alpha\alpha^{\mathrm{neq}}_0 + \partial_\beta\alpha^{\mathrm{neq}}_{2\alpha\beta}\\ c_s^2\left(\partial_\alpha\alpha^{\mathrm{neq}}_{1\beta} + \partial_\beta\alpha^{\mathrm{neq}}_{1\alpha}\right)\end{bmatrix}.\end{aligned} \tag{$\mathcal{T}_{f\to\alpha}$(B.E.))}$$

Now, eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)) is simply eq. (3.1) in a moment basis. In this case, $\alpha^{\mathrm{eq}}_\phi, \alpha^{\mathrm{neq}}_\phi, \mho_\phi$ and their components are free parameters that can be chosen to recover different PDEs. Although from the form of eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)), it is clear that only a limited subset of PDEs will be suitable to be represented in this way.

4.3 General Filter Matrix algorithm

The fact that the computational procedures introduced by Eggels & Somers [30] and Zhou et al. [10] are completely local is a key part of the Filter Matrix method. These serve as a basis for the computational procedure presented here.

In order to compute the collision eq. (3.5) is transformed as

$$\begin{aligned}\mathcal{T}_{f\to\alpha}\left(f_{\phi,i}\left(x_\alpha + \frac{c_i\Delta t}{2}, t + \frac{c_i\Delta t}{2}\right)\right) = \mathcal{T}_{f\to\alpha}\left(f_{\phi,i}\left(x_\alpha - \frac{c_i\Delta t}{2}, t - \frac{c_i\Delta t}{2}\right)\right), \\ + \mathcal{T}_{f\to\alpha}\left(\Delta t\, \Omega_{\phi,i}(f_{\phi,i})\right).\end{aligned} \tag{4.25}$$

which using moment vectors and collision is written as

$$\alpha^+_{\phi,k} = \alpha^-_{\phi,k} + \Delta t\, \mho_{\phi,k}(\alpha_{\phi,k}). \tag{4.26}$$

Initially the α^-_k is know, but because of the staggered formulation, the center value is needed to compute the collision. Similarly to Zhou et al. [10], this can be approximated by extrapolating the moment vector with a first order Taylor expansion

$$\alpha^-_{\phi,k} \approx \alpha_{\phi,k} - \frac{\Delta t}{2}\, \mho_{\phi,k}(\alpha_{\phi,k}). \tag{4.27}$$

although these expressions may be combined to

$$\alpha^+_{\phi,k} = \alpha^-_{\phi,k} + \Delta t \mho_{\phi,k} = \alpha^-_{\phi,k} + 2(\alpha_{\phi,k} - \alpha^-_{\phi,k}) = 2\alpha_{\phi,k} - \alpha^-_{\phi,k} \tag{4.28}$$

now, all these results allow for formulating a computational algorithm. It is similar to that of previous literature, although here it is made very explicit what step correspond to a collision.

The Filter Matrix algorithm

1. f^-_ϕ is defined
2. Change to moment basis $\alpha^-_\phi = \mathcal{T}_{f\to\alpha}(f^-_\phi)$
3. Solve (potentially non-linear) system of equation for α_ϕ with the system coming from the different vector components of $\alpha^-_\phi = \alpha_\phi - \Delta t \mho(\alpha_\phi)/2$
4. Perform collision as $\alpha^+_\phi = 2\alpha_\phi - \alpha^-_\phi$
5. Change back to population basis $f^+_\phi = \mathcal{T}_{\alpha\to f}(\alpha^+_\phi)$
6. Stream f^+_ϕ
7. Apply boundary conditions
8. GOTO 1.

The bulk of the computational procedure described by Zhou et al. [10] corresponds to computing step 3 in a local manner. Solving this system for given $\alpha^{\text{eq}}_\phi, \alpha^{\text{neq}}_\phi, \mho_\phi$ may or may not be possible depending on the exact values. Various techniques could be used to form approximations, e.g. solving for some variables implicitly. If too many variables are introduced, gradients and time derivatives could be calculated with finite difference methods to close the system, although this will sacrifice the local nature of the algorithm. Complementing the system of equation with additional empirical correlations is also a possibility.

All of these steps may be optimized in various ways, for instance the transforms can be saved directly as matrices as $P^+ = D^{-1}E_{ki}$ and $P^- = \omega_i \odot E_{ki}$. On a related note, the name "Filter Matrix" is hardly an appropriate name for the algorithm in this formulation. The matrix is a change of basis and although it can be argued through the derivation procedure in chapter 3 that it is a filtered version of BGK, this is more of a coincidence than a fundamental part of algorithm. This is elaborated further on in section 6.2.

4.4 Deriving FMLB for the convection–diffusion equation

Without loss of generality, the first two equilibrium distributions are chosen based on the equilibrium decomposition $\alpha^{\text{eq}}_{\phi,0} = \phi$, $\alpha^{\text{eq}}_{\phi,1\alpha} = \phi u_\alpha$ in eq. (4.1) for a generic scalar ϕ in a velocity field u_α that is solved in a coupled way. Considering the first row in eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)) and substituting yields

$$\mho_{\phi,0} = \partial_t\phi + \partial_\alpha(\phi u_\alpha) + \partial_t\alpha^{\text{neq}}_{\phi,0} + \partial_\alpha\alpha^{\text{neq}}_{\phi,1\alpha}. \tag{4.29}$$

Now, considering the exact convection diffusion equation eq. (Convection diffusion equation), here rewritten for convenience,

$$\partial_t\phi + \partial_\alpha(\phi u_\alpha) = q_\phi + \partial_\alpha(D_\phi\partial_\alpha\phi). \qquad \text{(Convection diffusion equation)}$$

it is clear that the following must be satisfied to reconstruct the convection–diffusion equation

$$\partial_t\alpha^{\text{neq}}_{\phi,0} + \partial_\alpha\alpha^{\text{neq}}_{\phi,1\alpha} - \mho_{\phi,0} = -\partial_\alpha(D_\phi\partial_\alpha\phi) - q_\phi \tag{4.30}$$

which has several solutions, with the arguably simplest being

$$\alpha^{\text{neq}}_{\phi,0} = 0 \quad , \quad \alpha^{\text{neq}}_{\phi,1\alpha} = -(D_\phi\partial_\alpha\phi) \quad , \quad \mho_{\phi,0} = q_\phi. \tag{4.31}$$

This was the reason to split the equation into the equilibrium and non-equilibrium parts, it is very easy to see a solution, although it could have been derived in any other way as well. The solution is not necessarily unique, meaning there is some flexibility in choosing a combination of $\mho_{\phi,k}$ and $\alpha^{\text{neq}}_{\phi,k}$ that results in the most efficient collision computation depending on use case.

By deciding on these moments, a relation between the macroscopic variables are enforced, with that relation being the convection–diffusion equation. This means that the rest of the moments must give "no information", that is they should be reducible to something like $0 = 0$. Thus, the collisional operator should be set equal to expression formed by the moments. Now, to get as simple $\Omega_{\phi,i}$ as possible, the $\mho_{\phi,k}$ should preferably be kept simple as well. The remaining free parameters are $\alpha^{\text{neq}}_{\phi,2\beta\gamma}$, which appear in the second moment, i.e. second component in eq. ($\mathcal{T}_{f\to\alpha}$(B.E.))

$$\begin{aligned}\mho_{\phi,1\alpha} = {}& \partial_t(\phi u_\alpha) + c_s^2\partial_\alpha(\phi) + \partial_\beta(\phi u_\alpha u_\beta) \\ & + \partial_t\alpha^{\text{neq}}_{\phi,1\alpha} + c_s^2\partial_\alpha\alpha^{\text{neq}}_{\phi,0} + \partial_\beta\alpha^{\text{neq}}_{\phi,2\alpha\beta}\end{aligned} \tag{4.32}$$

and the last moment will also be a part of the collision

$$\begin{aligned}\mho_{\phi,2\alpha\beta} = {}& \partial_t(\phi u_\alpha u_\beta) + c_s^2\left(\partial_\alpha(\phi u_\beta) + \partial_\beta(\phi u_\alpha)\right) \\ & + \partial_t\alpha^{\text{neq}}_{\phi,2\alpha\beta} + c_s^2\left(\partial_\alpha\alpha^{\text{neq}}_{1\beta} + \partial_\beta\alpha^{\text{neq}}_{1\alpha}\right).\end{aligned} \tag{4.33}$$

The two logical choices for $\alpha^{\text{neq}}_{\phi,2\alpha\beta}$ would be $\alpha^{\text{neq}}_{\phi,2\alpha\beta} = 0_{\tilde{\alpha}\tilde{\beta}}$ or $\alpha^{\text{neq}}_{\phi,2\alpha\beta} = -\phi u_\alpha u_\beta$. The latter being more logical as it will cancel a significant amount of terms, and since it does not affect the correspondence to the convection diffusion equation, it also

shows that the second order term does not need to be simulated.

At this point, for the sake of simpler implementation and reduced amount of computations, it can investigated which terms would correspond to physical small quantities. I.e. no longer consider *exactly* what equations correspond to, but approximately. If need be, the time derivatives can be replace, mostly, with spatial derivatives from the momentum equation relation.

To conclude, the most simple collisional operators would be

Components of second order convection–diffusion equation

The moment vector

$$\alpha_\phi = \begin{bmatrix} \phi \\ \phi u_\alpha - D_\phi \partial_\alpha \phi \\ 0_{\alpha\beta} \end{bmatrix}, \tag{4.34}$$

and corresponding collisional operator

$$\mho_\phi = \begin{bmatrix} q_\phi \\ \partial_t(\phi u_\alpha) + c_s^2 \partial_\alpha \phi - \partial_t(D_\phi \partial_\alpha \phi) \\ c_s^2 \left(\partial_\alpha(\phi u_\beta) + \partial_\beta(\phi u_\alpha) - \partial_\alpha(D_\phi \partial_\beta \phi) - \partial_\beta(D_\phi \partial_\alpha \phi)\right) \end{bmatrix}. \tag{4.35}$$

It is clear that the last component of eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)) is not needed, as it contains no information about the moment itself. Krüger et al. [12] argues that the second order expansion of the convection–diffusion equation helps with numerical stability, but it is not necessarily clear from this expression why it would be the case. However, assuming a first order Hermite expansion and simply removing terms corresponding to the second order Hermite polynomial in the derivation appendix C it is clear that eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)) would be

$$\begin{bmatrix} \mho_{\phi,0} \\ \mho_{\phi,1\alpha} \end{bmatrix} = \partial_t \begin{bmatrix} \alpha^{\text{eq}}_{\phi,0} \\ \alpha^{\text{eq}}_{\phi,1\alpha} \end{bmatrix} + \begin{bmatrix} \partial_\alpha \alpha^{\text{eq}}_{1\alpha} \\ c_s^2 \partial_\alpha \alpha^{\text{eq}}_0 \end{bmatrix} + \partial_t \begin{bmatrix} \alpha^{\text{neq}}_{\phi,0} \\ \alpha^{\text{neq}}_{\phi,1\alpha} \end{bmatrix} + \begin{bmatrix} \partial_\alpha \alpha^{\text{neq}}_{1\alpha} \\ c_s^2 \partial_\alpha \alpha^{\text{neq}}_0 \end{bmatrix}$$

which clearly has the same exact solution as previously, but in this case, there is no need to modify the last element. Thus,

Components of first order convection–diffusion equation

The moment vector

$$\alpha_\phi = \begin{bmatrix} \phi \\ \phi u_\alpha - D_\phi \partial_\alpha \phi \end{bmatrix}, \tag{4.36}$$

and corresponding collisional operator

$$\mho_\phi = \begin{bmatrix} q_\phi \\ \partial_t(\phi u_\alpha) + c_s^2 \partial_\alpha \phi - \partial_t(D_\phi \partial_\alpha \phi) \end{bmatrix} \tag{4.37}$$

4.4.1 Computational procedure

The general algorithm is described by section 4.3, and the step that needs to be elaborated on for the convection diffusion equation specifically is step 3, solving the non-linear system of equations. Solving the system depends on the complexity of the free parameters in the convection diffusion equation. For instance, in the $k-\varepsilon$ transport equations, the source term q_ϕ and diffusion coefficient D_ϕ are functions of other macroscopic moments.

Initially considering the first order expansion, then the system of equations becomes

$$\begin{cases} q_\phi(\phi, \partial_\alpha \phi) \\ \alpha_0^- = \phi - \dfrac{\Delta t}{2} q_\phi \\ D_\phi(\phi, \partial_\alpha \phi) \\ \alpha_{1\alpha}^- = \phi u_\alpha - D_\phi \partial_\alpha \phi - \dfrac{\Delta t}{2}(\partial_t(\phi u_\alpha) + c_s^2 \partial_\alpha \phi - \partial_t(D_\phi \partial_\alpha \phi)) \end{cases} \tag{4.38}$$

which should be solved for $\phi, \partial_\phi, q_\phi, D_\phi$. The source term and diffusion coefficient could naturally be arbitrarily complex, although for the following analysis it will be assumed to only vary with the scalar value and scalar gradients, similarly to the $k-\varepsilon$ transport equations. Solving the full system is clearly not possible due to there being more variables than equations.

The unknowns in eq. (4.38) are $q_\phi, D_\phi, \phi, \partial_t\phi, \partial_\alpha\phi, \partial_t\partial_\alpha\phi$. In two dimension the would be $1+1+1+1+2+2=8$ unknowns although there is only $1+1+1+2=5$ equations. This means that some terms must be approximated using closure laws or finite differencing techniques. If $q_\phi(\phi, \partial_\alpha\phi)$ and $D_\phi(\phi, \partial_\alpha\phi)$ are both linear functions, then the entire system is linear, meaning that it can be solved with Gaussian elimination or similar techniques. One of the simplest closure relations would be to assume a steady state, meaning all time derivatives are zero. This would reduce the system enough to be solvable.

Another approach to form more equations is to use the full second order expansion, then the system of equation becomes

$$\begin{cases} q_\phi(\phi, \partial_\alpha \phi) \\ \alpha_0^- = \phi - \dfrac{\Delta t}{2} q_\phi \\ D_\phi(\phi, \partial_\alpha \phi) \\ \alpha_{1\alpha}^- = \phi u_\alpha - D_\phi \partial_\alpha \phi - \dfrac{\Delta t}{2}(\partial_t(\phi u_\alpha) + c_s^2 \partial_\alpha \phi - \partial_t(D_\phi \partial_\alpha \phi)) \\ \alpha_{2\alpha\beta}^- = -\dfrac{c_s^2 \Delta t}{2}(\partial_\alpha(\phi u_\beta) + \partial_\beta(\phi u_\alpha) - \partial_\alpha(D_\phi \partial_\beta \phi) - \partial_\beta(D_\phi \partial_\alpha \phi)) \end{cases} \tag{4.39}$$

further introduces $\partial_\beta(\partial_\alpha\phi)$ that needs to be solved for, and the system is still no closed, however it opens up the possibility to formulate closure laws in terms of $\partial_\beta(\partial_\alpha\phi)$ which was not possible with the first order system of equations.

Although close laws are needed for the system of equations to have a solution, several tricks can be used to solve the system more efficiently than using Gaussian elimination. One trick that could be employed is to sum the second order terms over the diagonal, e.g. in three dimensions $\alpha^-_{2xx} + \alpha^-_{2yy} + \alpha^-_{2zz}$, allowing for back substitution of the convection diffusion equation. Now using tensor notation,

$$\begin{aligned}\alpha^-_{2\alpha\alpha} &= -c_s^2\Delta t\left(\partial_\alpha(\phi u_\alpha) - \partial_\alpha(D_\phi\partial_\alpha)\right)\\ &= -c_s^2\Delta t\left(q_\phi - \partial_t\phi\right),\end{aligned} \tag{4.40}$$

which directly relates the source term to a time derivative. Another interesting, but not necessarily useful trick, is to think of the system of equation as a differential equation since

$$\alpha^-_{1\alpha} = \alpha_{1\alpha} - \frac{\Delta t}{2}\partial_t\alpha_{1\alpha} - \frac{\Delta t}{2}c_s^2\partial_\alpha\phi \tag{4.41}$$

can be factored into

$$-e^{\frac{-2t}{\Delta t}}\left(\frac{2}{\Delta t}\alpha^-_{1\alpha} + c_s^2\partial_\alpha\phi\right) = e^{\frac{-2t}{\Delta t}}\partial_t\alpha_{1\alpha} - e^{\frac{-2t}{\Delta t}}\frac{2}{\Delta t}\alpha_{1\alpha} \tag{4.42}$$

$$= \partial_t\left(e^{\frac{-2t}{\Delta t}}\alpha_{1\alpha}\right) \tag{4.43}$$

resulting in

$$\alpha_{1\alpha} = -e^{\frac{2t}{\Delta t}}\int e^{\frac{-2t}{\Delta t}}\left(\frac{2}{\Delta t}\alpha^-_{1\alpha} + c_s^2\partial_\alpha\phi\right)\,\mathrm{d}t. \tag{4.44}$$

This does still not solve the closure problem, but formulating closure laws may or may not be more convenient in this form.

4.5 Navier Stokes equations

The key difference between to Navier Stokes equation and the convection diffusion equation is that it has two parts, the continuity equation and the momentum equation. In this section it is shown that the vector components eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)) is a convenient way to embed both of these equations.

4.5.1 The continuity equation

Reusing eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)), but now using the density instead of a generic scalar i.e. $\phi = \rho$. The equilibrium moments are once again used without loss of generality as defined in eq. (4.11)

$$\alpha^{\mathrm{eq}}_{\rho,i} = \begin{bmatrix}\rho\\ \rho u_\alpha\\ \rho u_\alpha u_\beta\end{bmatrix}, \tag{4.45}$$

which is substituted in the first row of eq. ($\mathcal{T}_{f\to\alpha}$(B.E.))

$$\partial_t\rho + \partial_\alpha(\rho u_\alpha) = \partial_t\alpha^{\mathrm{neq}}_{\rho,0} + \partial_\alpha\alpha^{\mathrm{neq}}_{\rho,1\alpha} - \mho_{\rho,0}. \tag{4.46}$$

Comparing this with the continuity equation eq. (Continuity)

$$\partial_t \rho + \partial_\alpha(\rho u_\alpha) = 0 \tag{Continuity}$$

gives the equation

$$\partial_t \alpha^{\text{neq}}_{\rho,0} + \partial_\alpha \alpha^{\text{neq}}_{\rho,1\alpha} - \mho_{\rho,0} = 0 \tag{4.47}$$

which has several solutions, with a trivial solution being.

$$\alpha^{\text{neq}}_{\rho,0} = 0 \quad , \quad \alpha^{\text{neq}}_{\rho,1\alpha} = 0_\alpha \quad , \quad \mho_{\rho,0} = 0. \tag{4.48}$$

4.5.2 Momentum equation

Although the continuity equation had very simple non-equilibrium moments, the main complexity comes in the moments corresponding to the second order expansion through the momentum equation.

The momentum equation has a pressure term which must be modelled with a constitutive relation. In this case, it is customary in lattice Boltzmann[2] to use the relation $p = c_s^2 \rho$. Substituting the previous moments, notably $\alpha^{\text{neq}}_{\rho,0} = 0, \alpha^{\text{neq}}_{\rho,1\alpha} = 0_\alpha$ into the second row of eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)), yields

$$\mho_{\rho,1\alpha} = \partial_t(\rho u_\alpha) + \partial_\alpha p + \partial_\alpha(\rho u_\alpha u_\beta) + \partial_\beta \alpha^{\text{neq}}_{\rho,2\alpha\beta}, \tag{4.49}$$

which is compared to eq. (Momentum)

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = -\partial_\alpha p + \partial_\beta\left(\rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\delta_{\alpha\beta}\partial_\gamma u_\gamma\right) + \delta_{\alpha\beta}\zeta\partial_\gamma u_\gamma\right) + F_\alpha \tag{Momentum}$$

which is replicated whenever

$$\mho_{\rho,1\alpha} - \partial_\beta \alpha^{\text{neq}}_{\rho,2\alpha\beta} = F_\alpha + \partial_\beta\left(\rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\delta_{\alpha\beta}\partial_\gamma u_\gamma\right) + \delta_{\alpha\beta}\zeta\partial_\gamma u_\gamma\right). \tag{4.50}$$

with the simplest solution having

$$\mho_{\rho,1\alpha} = F_\alpha \tag{4.51}$$

and

$$\alpha^{\text{neq}}_{\rho,2\alpha\beta} = -\left(\rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\partial_\gamma u_\gamma \delta_{\alpha\beta}\right) + \zeta\partial_\gamma u_\gamma \delta_{\alpha\beta}\right). \tag{4.52}$$

Just like for the convection diffusion equation, the last component in eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)) must be chosen in a way that does not enforce any new relations between the macroscopic variables, i.e. being reducible to $0 = 0$. Thus $\mho_{\rho,2\alpha\beta}$ is equal to the rest of the terms, that is

$$\begin{aligned}\mho_{\rho,2\alpha\beta} = &\left(\partial_t(\rho u_\alpha u_\beta) + c_s^2\left(\partial_\alpha(\rho u_\beta) + \partial_\beta(\rho u_\alpha)\right)\right.\\ &\left. - \partial_t\left(\rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\partial_\gamma u_\gamma\delta_{\alpha\beta}\right) + \zeta\partial_\gamma u_\gamma\delta_{\alpha\beta}\right)\right)\end{aligned} \tag{4.53}$$

[2] This is essentially substituting the equation of state, in Somers [29] and Eggels & Somers [30] a different equation of state is used as in eq. (2.37).

Components of second order Navier Stokes equation

The moment vector

$$\alpha_\rho = \begin{bmatrix} \rho \\ \rho u_\alpha \\ \rho u_\alpha u_\beta - \rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\partial_\gamma u_\gamma \delta_{\alpha\beta}\right) - \zeta \partial_\gamma u_\gamma \delta_{\alpha\beta} \end{bmatrix}, \tag{4.54}$$

and corresponding collisional operator

$$\mho_\phi = \begin{bmatrix} 0 \\ F_\alpha \\ \partial_t(\rho u_\alpha u_\beta) + c_s^2\left(\partial_\alpha(\rho u_\beta) + \partial_\beta(\rho u_\alpha)\right) \hookleftarrow \\ -\partial_t\left(\rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\partial_\gamma u_\gamma \delta_{\alpha\beta}\right) + \zeta \partial_\gamma u_\gamma \delta_{\alpha\beta}\right) \end{bmatrix} \tag{4.55}$$

The strain rate tensor as in eq. (2.5) can be introduced, where it is important to keep track of the extra factor 2 that appears, resulting in the substitution

$$\partial_\alpha(\rho u_\beta) + \partial_\beta(\rho u_\alpha) = 2\rho s_{\alpha\beta} + u_\beta \partial_\alpha \rho + u_\alpha \partial_\beta \rho \tag{4.56}$$

Using $\alpha_{\rho,k}$ and $\mho_{\rho,k}$, it is again possible to formulate a system of equations like

$$\begin{cases} \alpha_0^- = \rho \\ F_\alpha(\rho, u_\alpha, s_{\alpha\beta}) \\ \alpha_{1\alpha}^- = \rho u_\alpha - \dfrac{\Delta t}{2} F_\alpha \\ \nu(\rho, u_\alpha, s_{\alpha\beta}) \\ \zeta(\rho, u_\alpha, s_{\alpha\beta}) \\ \alpha_{2\alpha\beta}^- = \rho u_\alpha u_\beta - \rho\nu(2s_{\alpha\beta} - \dfrac{2}{3}\delta_{\alpha\beta}\mathrm{div}(u_\alpha)) - \zeta\delta_{\alpha\beta}\mathrm{div}(u_\alpha) \hookleftarrow \\ -\dfrac{\Delta t}{2}\Big(\partial_t(\rho u_\alpha u_\beta) + c_s^2(2s_{\alpha\beta} + u_\beta\partial_\alpha\rho + u_\alpha\partial_\beta\rho) \hookleftarrow \\ -\partial_t\left(\rho\nu\left(2s_{\alpha\beta} - \dfrac{2}{3}\mathrm{div}(u_\alpha)\delta_{\alpha\beta}\right) + \zeta\mathrm{div}(u_\alpha)\delta_{\alpha\beta}\right)\Big) \end{cases} \tag{4.57}$$

Just like in the first order expansion of the convection–diffusion equation, this system of equations, regardless of relations for force and viscosity, will not be closed. The total amount of unknowns, remembering that the velocity is unknown in this case, will be $\rho, u_\alpha, F_\alpha, \nu, \zeta, \partial_t\rho$, $\partial_\alpha\rho$, $s_{\alpha\beta}$, $\partial_t u_\alpha$, $\partial_t s_{\alpha\beta}$, which in 3D is $1+3+3+1+1+1+3+6+3+6 = 28$ unknown with $1+3+3+1+1+6 = 15$ equations. Once again, the derivatives could be calculated by discretizing the field and calculating the values with finite differences although this is potentially an expensive operation to perform. To get an efficient computational procedure the continuity equation $\partial_t\rho = -\partial_\alpha(\rho u_\alpha)$ could be resubstituted. The relation $\mathrm{div}(u_\alpha) = \mathrm{trace}(s_{\alpha\beta})$ gives one more equation, but it is not enough to close the system. At this point Eggels & Somers [30] makes approximation in eq. (2.36), which could be applicable in this case as well.

4.5.2.1 Constant density case

If assuming constant density[3] the expression can be simplified significantly with $\partial_t \rho = 0$, $\partial_\alpha \rho = \vec{0}$ and $\mathrm{div}(u_\alpha) = 0$ (from continuity equation). This allows the simplification

Components of second order constant density Navier Stokes equation

The moment vector

$$\alpha_{\rho,k} = \begin{bmatrix} \rho \\ \rho u_\alpha \\ \rho u_\alpha u_\beta - 2\rho\nu s_{\alpha\beta} \end{bmatrix}, \tag{4.58}$$

and corresponding collisional operator

$$\mho_{\phi,k} = \begin{bmatrix} 0 \\ F_\alpha \\ \rho\partial_t(u_\alpha u_\beta) + 2c_s^2 s_{\alpha\beta} - 2\rho\partial_t\left(\nu s_{\alpha\beta}\right) \end{bmatrix} \tag{4.59}$$

Using $\alpha_{\rho,k}$ and $\mho_{\rho,k}$, it is again possible to formulate a system of equation like

$$\begin{cases} \bar{\alpha_0} = \rho \\ F_\alpha(\rho, u_\alpha, s_{\alpha\beta}) \\ \bar{\alpha_{1\alpha}} = \rho u_\alpha - \dfrac{\Delta t}{2} F_\alpha \\ \nu(\rho, u_\alpha, s_{\alpha\beta}) \\ \zeta(\rho, u_\alpha, s_{\alpha\beta}) \\ \bar{\alpha_{2\alpha\beta}} = \rho u_\alpha u_\beta - 2\rho\nu s_{\alpha\beta} - \dfrac{\Delta t}{2}\left(\partial_t(\rho u_\alpha u_\beta) + 2c_s^2 s_{\alpha\beta} - 2\rho\partial_t\left(\nu s_{\alpha\beta}\right)\right) \end{cases} \tag{4.60}$$

4.6 Filter Matrix for $k - \varepsilon$ equations

The $k-\varepsilon$ equations was the initial motivation to investigate the origins of the FMLB method. Here, the k implementation will be used as an example of some of the challenges and simplifications associated with formulating a computational procedure when the source term and diffusion coefficient becomes more complex.

The formulation of eq. (k transport) is almost the same as eq. (Convection diffusion equation), if the term $\bar{u}_\beta \partial_\beta k$ is rewritten as $\bar{u}_\beta \partial_\beta k = \partial_\alpha(k u_\alpha) - k\partial_\alpha u_\alpha$, which is valid in the incompressible limit, the equations match perfectly. Thus, using the corresponding source term eq. (D_k) and diffusion coefficient eq. (q_k) in the eqs. (4.34) and (4.35) yields

[3] This is not necessarily the same thing as the flow being incompressible, although result when using the full definition of incompressibility should be similar.

Components of second order k transport equation

The moment vector

$$\alpha_{k,k} = \begin{bmatrix} k \\ ku_\alpha - \left(\nu + C_\mu \frac{k^2}{\varepsilon\sigma_k}\right)\partial_\alpha k \\ 0_{\alpha\beta} \end{bmatrix}, \tag{4.61}$$

and corresponding collisional operator

$$\mho_{k,k} = \begin{bmatrix} C_\mu \frac{k^2}{\varepsilon}|S|^2 + g_\alpha\beta C_\mu \frac{k^2}{\varepsilon\sigma_\theta}\partial_\alpha \overline{T} - \varepsilon \\ \partial_t(ku_\alpha) + c_s^2\partial_\alpha k - \partial_t((\nu + C_\mu \frac{k^2}{\varepsilon\sigma_k})\partial_\alpha k) \\ c_s^2\left(\partial_\alpha(ku_\beta) + \partial_\beta(ku_\alpha) - \partial_\alpha((\nu + C_\mu \frac{k^2}{\varepsilon\sigma_k})\partial_\beta k) - \partial_\beta((\nu + C_\mu \frac{k^2}{\varepsilon\sigma_k})\partial_\alpha k)\right) \end{bmatrix}. \tag{4.62}$$

in which the derivatives can be expanded

$$\mho_{k,k} = \begin{bmatrix} C_\mu \frac{k^2}{\varepsilon}|S|^2 + g_\alpha\beta C_\mu \frac{k^2}{\varepsilon\sigma_\theta}\partial_\alpha \overline{T} - \varepsilon \\ k\partial_t u_\alpha + u_\alpha\partial_t k + c_s^2\partial_\alpha k - \nu\partial_t\partial_\alpha k - C_\mu \frac{k^2}{\varepsilon\sigma_k}\partial_t\partial_\alpha k \hookleftarrow \\ + \frac{k^2 C_\mu}{\varepsilon^2\sigma_k}(\partial_\alpha k)\partial_t\varepsilon - \frac{2kC_\mu}{\sigma_k\varepsilon}(\partial_\alpha k)\partial_t k \\ c_s^2\big(2ks_{\alpha\beta} + u_\beta\partial_\alpha k + u_\alpha\partial_\beta k \hookleftarrow \\ -2\nu\partial_\beta\partial_\alpha k - 2C_\mu \frac{k^2}{\varepsilon\sigma_k}\partial_\beta\partial_\alpha k \hookleftarrow \\ + \frac{k^2}{\sigma_k}(\partial_\alpha k)C_\mu \frac{\partial_\beta\varepsilon}{\varepsilon^2} + \frac{k^2}{\sigma_k}(\partial_\beta k)C_\mu \frac{\partial_\alpha\varepsilon}{\varepsilon^2} - \frac{4C_\mu k}{\sigma_k\varepsilon}(\partial_\alpha k)\partial_\beta k\big) \end{bmatrix}. \tag{4.63}$$

Similarly to before, if it is solved in a coupled way this yields a system of equations. The unknowns would be $k, \partial_\alpha k, \partial_t k, \partial_t\partial_\alpha k, \partial_\beta\partial_\alpha k$, which in three dimensions totals $1 + 3 + 1 + 3 + 6 = 14$. However, in the computational procedure, most of these term are not needed when computing the collision, only $k, \partial_\alpha k$ are found in the moment vector, and as such needed to perform the collision. However, some way must be found to close the system of equations. The simplest being to neglect all time derivatives.

Just considering the first component of the vector, its contribution to the system of equation would be

$$\alpha_k^- = k - \frac{\Delta t}{2}\left(C_\mu \frac{k^2}{\varepsilon}|S|^2 + g_\alpha\beta C_\mu \frac{k^2}{\varepsilon\sigma_\theta}\partial_\alpha \overline{T} - \varepsilon\right), \tag{4.64}$$

which is a quadratic equation in k. Everything in the equation is known except k. It would be solved explicitly using e.g. the quadratic formula, or solved implicitly,

to increase stability and most likely reach a sufficient numerical accuracy at a lower computational cost.

Here the second order terms are included, although to model the full collision, this is not needed and does not help close the system of equations. However, gives the opportunity to formulate the closure laws in terms of $\partial_\beta \partial_\alpha k$, with the first order expansion, the closure laws had to be formulated for $\partial_t k$ and $\partial_t \partial_\alpha k$, which depending on the flow case may or may not be more applicable.

5

Comparison to Other Derivations

The method to derive the FMLB method presented in chapter 4 differs significantly from previous literature, the final formulation is compared to other studies in this chapter.

5.1 Comparison with Zhuo et al.

The main difference between the matrix and α formulations derived in this report to that of Zhou et al. [10] is the exclusion of higher order terms. However, these higher order terms are fixed through the constants γ_1, γ_2. Zhuo et al. further tests the influence of these factors in [32], where only marginal improvements are found when tuning them and it is mentioned that $\gamma_1 = \gamma_2 = 0$ is sufficient. Hence, already from the results of Zhuo et al. the higher order terms can be set to zero. This effectively means rows and columns responsible for projecting the α vector on these higher order modes can be ignored and E_{ik} and E_{ki} can be reduced to the non-square matrices shown in this report.

With this mind, the higher order terms will be ignored from now on, and the comparison is between the 0th, 1st and 2nd order terms. Comparing the two and three dimensional formulations by Zhuo et al. [10, 33] shows they can both be written more compactly using the notation of this report. In the case of the matrices, the initial terms are similar to the scaled Hermite polynomials introduced in this thesis, both for the two and three dimensional formulations. The remaining components are similar to the higher order Hermite polynomials shown by Shan et al. [23] and described for the MRT scheme by Krüger et al. [12]. Taking eqs. (2.27) and (2.31) from Zhou et al. [10] and rewriting in the compact notation of this report yields

$$E^{(\text{Zhuo})}_{ki,\text{D2Q9}} = \left[\quad 1_i, \quad c_{i\alpha}, \quad \frac{c_{i\alpha} \odot c_{i\beta}}{c_s^2} - \delta_{\alpha\beta}, \quad \text{higher order terms} \quad \right]^T, \tag{5.1}$$

and similarly for $E^{(\text{Zhuo})}_{ik,\text{D2Q9}}$. Condensing the α in eq. (2.29)[1] with the notation used

[1] Note that the two dimensional eq. (2.28) and three dimensional moment vectors eq. (2.29) differs in a way very similar to the discussion about two and three dimensional stress tensor at eq. (Momentum). With the same reasoning, considering the three dimensional version is equivalent to the two dimensional if the bulk viscosity is appropriately scaled.

in this report and ignoring higher order terms gives

$$\alpha^{\pm} = \begin{bmatrix} \rho \\ \rho u_\alpha \pm \Delta t F_\alpha/2 \\ 3\rho u_\alpha u_\beta + 3\rho\left(-\nu \pm \frac{\Delta t}{3\cdot 2}\right)(\partial_\alpha u_\beta + \partial_\beta u_\alpha) + 3\left(\frac{2}{3} - B\right)\delta_{\alpha\beta}\rho\nu\partial_\gamma u_\gamma \end{bmatrix} \tag{5.2}$$

$$= \begin{bmatrix} \rho \\ \rho u_\alpha \pm \Delta t F_\alpha/2 \\ \frac{1}{c_s^2}\left(\rho u_\alpha u_\beta - \rho\nu(\partial_\alpha u_\beta + \partial_\beta u_\alpha) + \left(\frac{2}{3} - B\right)\delta_{\alpha\beta}\rho\nu\partial_\gamma u_\gamma\right) \pm \frac{\Delta t}{2}(\partial_\alpha u_\beta + \partial_\beta u_\alpha) \end{bmatrix}, \tag{5.3}$$

where the terms have been carefully been factored and replaced with $c_s^2 = 1/3$ where applicable. The same procedure for the heat equation from Zhou et al. [10] yields

$$\beta^{\pm} = \begin{bmatrix} \phi \\ u_\alpha\phi - D_\phi\partial_\alpha\phi \pm \frac{1}{3}\cdot\frac{\Delta t}{2}\partial_\alpha \end{bmatrix} \tag{5.4}$$

$$= \begin{bmatrix} \phi \\ u_\alpha\phi - D_\phi\partial_\alpha\phi \pm \frac{\Delta t}{2}c_s^2\partial_\alpha\phi \end{bmatrix}, \tag{5.5}$$

now, moment vectors can be split into a moment part and a moment collision part. This has been done on the next page and gives a direct comparison between this work and that of Zhou et al. [10].

Zhuo [10, 33] formulation of FMLB for the Navier Stokes equations

$$E_{ik}^{(\text{Zhuo})} = \left[\begin{array}{ccc} 1_i, & \frac{c_{i\alpha}}{c_s^2}, & \frac{1}{1+\delta_{\alpha\beta}}\left(\frac{c_{i\alpha}\odot c_{i\beta}}{c_s^2} - \delta_{\alpha\beta}\right) \end{array} \right] \tag{5.6}$$

$$E_{ki}^{(\text{Zhuo})} = \left[\begin{array}{ccc} 1_i, & c_{i\alpha}, & \frac{c_{i\alpha}\odot c_{i\beta}}{c_s^2} - \delta_{\alpha\beta} \end{array} \right]^T \tag{5.7}$$

$$\alpha_\phi^{(\text{Zhuo})} = \begin{bmatrix} \phi \\ \phi u_\alpha - D_\phi \partial_\alpha \phi \end{bmatrix} \tag{5.8}$$

$$\mho_\phi^{(\text{Zhuo})} = \begin{bmatrix} 0 \\ c_s^2 \partial_\alpha \phi \end{bmatrix} \tag{5.9}$$

$$\alpha_\rho^{(\text{Zhuo})} = \begin{bmatrix} \rho \\ \rho u_\alpha \\ \frac{1}{c_s^2}\left(\rho u_\alpha u_\beta - \rho\nu\left(\partial_\alpha u_\beta + \partial_\beta u_\alpha - \frac{2}{3}\partial_\gamma u_\gamma \delta_{\alpha\beta}\right) - B\rho\nu\partial_\gamma u_\gamma \delta_{\alpha\beta}\right) \end{bmatrix} \tag{5.10}$$

$$\mho_\rho^{(\text{Zhuo})} = \begin{bmatrix} 0 \\ F_\alpha \\ \rho(\partial_\alpha u_\beta + \partial_\beta u_\alpha) \end{bmatrix} \tag{5.11}$$

Present formulation of FMLB for the Navier Stokes equations

$$E_{ik}^{(\text{Present})} = \left[\begin{array}{ccc} 1_i, & \frac{c_{i\alpha}}{c_s^2}, & \frac{1}{c_s^2}\frac{1}{1+\delta_{\alpha\beta}}\left(\frac{c_{i\alpha}\odot c_{i\beta}}{c_s^2} - \delta_{\alpha\beta}\right) \end{array} \right] \tag{5.12}$$

$$D^{-1}E_{ki}^{(\text{Present})} = \left[\begin{array}{ccc} 1_i, & c_{i\alpha}, & c_s^2\left(\frac{c_{i\alpha}\odot c_{i\beta}}{c_s^2} - \delta_{\alpha\beta}\right) \end{array} \right]^T \tag{5.13}$$

$$\alpha_\phi^{(\text{Present})} = \begin{bmatrix} \phi \\ \phi u_\alpha - D_\phi \partial_\alpha \phi \end{bmatrix} \tag{5.14}$$

$$\mho_\phi^{(\text{Present})} = \begin{bmatrix} q_\phi \\ c_s^2 \partial_\alpha \phi + \partial_t(\phi u_\alpha) - \partial_t(D_\phi \partial_\alpha \phi) \end{bmatrix} \tag{5.15}$$

$$\alpha_\rho^{(\text{Present})} = \begin{bmatrix} \rho \\ \rho u_\alpha \\ \rho u_\alpha u_\beta - \rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\partial_\gamma u_\gamma \delta_{\alpha\beta}\right) - \zeta\partial_\gamma u_\gamma \delta_{\alpha\beta} \end{bmatrix},$$

$$\mho_\rho^{(\text{Present})} = \begin{bmatrix} 0 \\ F_\alpha \\ \partial_t(\rho u_\alpha u_\beta) + c_s^2\left(\rho\partial_\alpha u_\beta + \rho\partial_\beta u_\alpha + u_\beta\partial_\alpha\rho + u_\alpha\partial_\beta\rho\right) \hookleftarrow \\ -\partial_t\left(\rho\nu\left(\partial_\beta u_\alpha + \partial_\alpha u_\beta - \frac{2}{3}\partial_\gamma u_\gamma\delta_{\alpha\beta}\right) + \zeta\partial_\gamma u_\gamma \delta_{\alpha\beta}\right) \end{bmatrix}$$

The blue terms are simply due to different convention when constructing the initial E_{ik}, if a factor of c_s^2 is put into the α vector in eq. (4.8) and propagated into the corresponding eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)), the final formulation would not differ with these terms. The computations being done will also be identical since the resulting system of equation can be rescaled with c_s^2 to yield an identical system. Further, the bulk viscosity can be replaced with $B\rho\nu = \zeta$ to get the same equation.

The red terms differs between the present derivation and Zhou et al. [10]. For the convection–diffusion equation the equations of Zhou et al. [10] corresponds to the special case of when the present derivation is sourceless and time derivatives are zero. Note that this is not necessarily the same as assuming a steady state solution because the entire derivate is neglected. In the derivation of Somers [29], some of the higher order terms are neglecting when calculating the collisional operator which is thought to have a similar effect to this. As Zhou et al. [10] refers to Somers [29] and [19] for the details of calculating the collisional operator, that derivation has most likely neglected higher order terms and applied some closure law similar to eq. (2.36).

Throughout the derivation of [19], part of the reasoning why some higher order terms can be neglected are related to the incompressible limit. When decreasing the fluid velocity, the terms $u_\beta\partial_\alpha\rho + u_\alpha\partial_\beta\rho$ are argued deviate from the second order equilibrium distribution, and as such are removed by Zhou et al. [10]. In the present derivation, there is no physically meaningful equilibrium distribution, it is a purely mathematical trick to substitute the "equilibrium values" to more easily find the definitions of α and $\mho$. As such, these terms are kept in this derivation. There is no study that investigates strong density gradients with FMLB, and as such it is hard to judge whether any error might be due to these effects.

5.1.1 Comparison with Somers

As mentioned in section 2.6.1, the lattice used in Somers [29] and Eggels & Somers [30] is a four dimensional 24-speed, as well as using a different pressure definition compared to this study. Further the time step is set to $\Delta t = 1$, as such the matrices cannot simply be factored into an easily comparable form as was done for Zhou et al. [10]. However, it is still possible to identify the α and $\mho$ for the momentum equation. The moment vector in Eggels & Somers [30] is given by

$$\alpha^{\pm} = \begin{bmatrix} \rho \\ \rho u_x \pm F_x/2 \\ \rho u_y \pm F_y/2 \\ \rho(u_x u_x + \tau_{xx}) + \rho\left(-6\nu \pm 1\right) 2\partial_x u_x/6 \\ \rho(u_x u_y + \tau_{xy}) + \rho\left(-6\nu \pm 1\right)\left(\partial_y x + \partial_x y\right)/6 \\ \rho(u_y u_y + \tau_{yy}) + \rho\left(-6\nu \pm 1\right) 2\partial_y u_y/6 \\ \text{higher order terms} \end{bmatrix} \tag{5.16}$$

which is separated into

$$\alpha_\rho^{(\text{Eggels \& Somers})} = \begin{bmatrix} \rho \\ \rho u_\alpha \\ \rho(u_\alpha u_\beta + \tau_{\alpha\beta} - \nu(\partial_\beta u_\alpha + \partial_\alpha u_\beta)) \end{bmatrix} \tag{5.17}$$

and

$$\mathcal{U}_\rho^{(\text{Eggels \& Somers})} = \begin{bmatrix} 0 \\ F_\alpha \\ (\partial_\beta u_\alpha + \partial_\alpha u_\beta)/3 \end{bmatrix}. \tag{5.18}$$

Further, Eggels & Somers [30] presents a moment vector for the convection–diffusion with subgrid scale scalar fluxes σ

$$\beta^\pm = \begin{bmatrix} \rho\phi \\ \rho(\phi u_x + \sigma_x - D_\phi \partial_x \phi) \pm \rho \partial_x \phi/4 \\ \rho(\phi u_y + \sigma_y - D_\phi \partial_y \phi) \pm \rho \partial_y \phi/4 \\ \text{higher order terms} \end{bmatrix} \tag{5.19}$$

once again corresponding to

$$\alpha_\phi^{(\text{Eggels \& Somers})} = \begin{bmatrix} \rho\phi \\ \rho(\phi u_\alpha + \sigma_\alpha - D_\phi \partial_\alpha \phi) \end{bmatrix} \tag{5.20}$$

$$\mathcal{U}_\phi^{(\text{Eggels \& Somers})} = \begin{bmatrix} 0 \\ \rho \partial_\alpha \phi/2 \end{bmatrix} \tag{5.21}$$

It is worth noting that there is no constant factor in front $\rho u_\alpha u_\beta$-term in this case, which is more similar to the separation done in this study than in Zhou et al. [10]. The subgrid fluxes τ and σ are not incorporated in this report. However, in Eggels & Somers [30], τ is formulated as an extra term $\partial_\alpha(\rho\tau_{\alpha\beta})$ in the Navier Stokes equation. In eq. ($\mathcal{T}_{f\to\alpha}$(B.E.)), this is precisely the type of derivative that is modelled through the $\alpha_{2\alpha\beta}$ term and as such, the present formulation can easily be extended to include subgrid stresses. The convection–diffusion equation from Eggels & Somers [30] also differs through the inclusion of the density. This can also be included in the present derivation by redefining the scalar as a product of the density and density normalized scalar, i.e. $\phi = \rho\phi'$. When expanding the moment vector, the derivative product rule lead to a dependence on the density gradients.

6

Discussion and conclusion

Initially in this section brief remarks about implications of the results from this thesis are given, thereafter the focus areas of future work relating to this present formulation of FMLB are given. Lastly, the thesis is summarized with a conclusion.

6.1 Implication for minimal lattice

The minimal amount of hydrodynamic modes are discussed in Shan et al. [23] and [38, 39]. This depends on how the temperature is treated, although if only the mass and momentum conservation are of interest, only the hydrodynamic modes corresponding to second order and below need to be resolved. From the perspective of the derivation in this report, the important constraints are that the lattice velocities satisfies eqs. $(\sum \omega)$ to $(\sum \omega c^4)$, or eqs. $(\sum \omega)$ to $(\sum \omega c^2)$ if the convection–diffusion equation is considered, as these are needed when constructing the gram matrix D. This a if there is more modes than lattice directions, D can instead not be inverted. Theoretically not much changes when using cheaper schemes as long as the previously listed constraints are med, but the implementation of streaming and boundary conditions becomes significantly more cumbersome. A hexagonal lattice similar to [39] could most likely be used, and as that study shown significant reductions in computations per collision, pairing it with FMLB is a promising simulation method.

6.2 Name of the algorithm

As mentioned in section 4.3, ”The Filter Matrix Method” as presented in this report is a misleading name. Fistly, the word ”Matrix”. The FMLB was proposed before much of the work had been done on the MRT collisional operator, which shares a similar matrix transform, which is why it might have been a distinguishing name at the time, but with the current prevalence of MRT, it is not a unique feature for FMLB. Moreover,the word ”Filter” is also misleading. Higher terms are removed throughout the derivation procedure in the works of Somers [29] and Zhou et al. [10], leading to the derivation procedure by Rohde [35] and further developed in this thesis, but as mentioned in chapter 3 neglecting higher order terms moves the error from the Chapman–Enskog expansion to a mismatch in the discretization. The derivation procedure in chapter 4 does not require neglecting terms throughout the process, and the simplifications to reduce the final form to the formulation by Zhou et al. [10] are based on physical understanding of the problem by small time derivatives or density gradients. Instead, FMLB should preferably be named after its

fundamental components. These could be argued to be the transformations $\mathcal{T}_{f\to\alpha}$, and $\mathcal{T}_{\alpha\to f}$. These have a matrix representation, but since the transforms are better thought of as linear projections, especially since ω_i appears in $\mathcal{T}_{f\to\alpha}$, the word "Matrix" is also misleading. Further keywords for the method are "non-linear collisional operator", "projection", "moment space", "closure laws" and also the constructive nature of mapping the discretized Boltzmann equation to moment vectors to regain the PDEs of interest. Since the entire computational algorithm is a crucial part of what makes the operator local, it also does not make sense to name just the collisional operator, but the entire method. It is up to the LBM community at large to decide on a suitable name, whether to keep the FMLB convection, otherwise the name "Moment Projected Boltzmann" (MPB) method is suggested.

6.3 Further work

The proposed algorithm needs to be numerically tested and evaluated. The evaluation should focus on two things, accuracy of simulated results and computational efficiency. The accuracy is how close to an analytic solution the algorithm can calculate a specific simulation case. As discussed in section 2.3.1, there will be errors of higher order macroscopic properties between what is actually simulated with LBM, and the simulation of interest. Since the proposed method is shown to be exact in the moment space, it should be investigated whether this actually the case numerically. Moreover, due to the limited amount of studies on FMLB in general, a investigation of how it handles transients and compressibility effects is unknown.

Another focus should be to quantify the computational cost of the algorithm, as a collision with FMLB is more computationally expensive to compute, it has to improve accuracy enough for it to not be better to just use the e.g. BGK with an increase resolution instead. In Zhou et al. [10] it is shown that the computational grid can be reduced significantly with FMLB for comparable results to BGK and MRT, c.f. figure 5 in [10], which would make up for the added computational cost per iteration. Using the modifications from this thesis and running the same tests would make for a valuable comparison.

6.4 Conclusion

This thesis has shown a new way of deriving the Filter Matrix Lattice Boltzmann method. The assumption for the methods are the same as those leading up to the projection of the Boltzmann equation onto a Hermite polynomial basis. From that point, the derivation is strictly mathematical, and physics based approximations can intuitively be applied when formulating an exact computational procedure. Current literature can mostly be seen as a special case of the proposed algorithm, although there are some cases where they differ and those cases should be the starting point for further investigations.

Bibliography

[1] IPCC, "Summary for policymakers," in *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change* (C. W. Team, H. Lee, and J. Romero, eds.), pp. 1–34, Geneva, Switzerland: IPCC, 2023.

[2] International Energy Agency, "Global energy review 2025." https://www.iea.org/reports/global-energy-review-2025, 2025. Paris [Online; 2025-06-12].

[3] European Commission, "Eu knowledge hub for enabling molten salt reactor safety development and deployment." Grant agreement ID: 101165896, 2024. Euratom Research and Training Programme (EURATOM).

[4] *Challenges Related to the Use of Liquid Metal and Molten Salt Coolants in Advanced Reactors*. No. 1696 in TECDOC Series, Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 2013.

[5] P. van der Spek, "Gpu-accelerated large eddy simulation of non-eutectic msfr salt freezing in turbulent channel flow," Master's thesis, Delft University of Technology, February 2024.

[6] C. Bus, "Simulating the transient freezing in cooled non-eutectic molten salt channel flow," Master's thesis, Delft University of Technology, 2022.

[7] M. Wortelboer, "Investigating gpu-accelerated double distribution function lattice boltzmann schemes for heat transfer and phase change in turbulent flows," Master's thesis, Delft University of Technology, July 2023.

[8] L. Borstlap, "Developing a filter-matrix lattice-boltzmann boundary method for conjugate heat transfer and phase change," Master's thesis, Delft University of Technology, February 2024.

[9] C. Polderman, "Developing a multi-physics tool for the neutronics[1]thermal-hydraulics in a molten salt reactor using the lattice boltzmann method," Master's thesis, Delft University of Technology, February 2024.

[10] C. Zhuo, C.-w. Zhong, and J. Cao, "Filter-matrix lattice boltzmann model for incompressible thermal flows," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 85, no. 4, p. 046703, 2012.

[11] L. Davidson, "Fluid mechanics, turbulent flow and turbulence modeling." https://www.tfd.chalmers.se/~lada/postscript_files/solids-and-fluids_turbulent-flow_turbulence-modelling.pdf, 2024. [Online; accessed 2025-03-17].

[12] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, "The lattice boltzmann method," *Springer International Publishing*, vol. 10, no. 978-3, pp. 4–15, 2017.

[13] H. Versteeg and W. Malalasekera, "Computational fluid dynamics," *The finite volume method*, pp. 1–26, 1995.

[14] Wikipedia contributors, "Navier–stokes equations — Wikipedia, the free encyclopedia." https://en.wikipedia.org/w/index.php?title=Navier%E2%80%93Stokes_equations&oldid=1292228922, 2025. [Online; 2025-05-30].

[15] J. J. Duderstadt and L. J. Hamilton, *Nuclear reactor analysis*. Wiley, 1976.

[16] D. C. Wilcox *et al.*, *Turbulence modeling for CFD*, vol. 2. DCW industries La Canada, CA, 1998.

[17] D. A. Wolf-Gladrow, *Lattice-gas cellular automata and lattice Boltzmann models: an introduction*. Springer, 2004.

[18] D. Enskog, *Kinetische Theorie der Vorgänge in mässig verdünnten Gasen*, vol. 1. Almquist & Wiksell, 1917.

[19] U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet, "Lattice gas hydrodynamics in two and three dimensions," in *Modern Approaches to Large Nonlinear Systems*, Los Alamos National Laboratory, 1987.

[20] B. Chopard, J. L. Falcone, and J. Latt, "The lattice boltzmann advection-diffusion model revisited," *The European Physical Journal Special Topics*, vol. 171, no. 1, pp. 245–249, 2009.

[21] J. Lätt, *Hydrodynamic limit of lattice Boltzmann equations*. PhD thesis, University of Geneva, 2007. doi: 10.13097/archive-ouverte/unige:464.

[22] B. Chopard, J. L. Falcone, and J. Lätt, "The lattice boltzmann advection-diffusion model revisited," *The European Physical Journal Special Topics*, vol. 171, no. 1, pp. 245–249, 2009.

[23] X. Shan, X.-F. Yuan, and H. Chen, "Kinetic theory representation of hydrodynamics: a way beyond the navier–stokes equation," *Journal of Fluid Mechanics*, vol. 550, pp. 413–441, 2006.

[24] T.-M. Liou and C.-S. Wang, "Large eddy simulation of rotating turbulent flows and heat transfer by the lattice boltzmann method," *Physics of Fluids*, vol. 30, no. 1, 2018.

[25] K. V. Sharma, R. Straka, and F. W. Tavares, "Current status of lattice boltzmann methods applied to aerodynamic, aeroacoustic, and thermal flows," *Progress in Aerospace Sciences*, vol. 115, p. 100616, 2020.

[26] J. W. Shim and R. Gatignol, "How to obtain higher-order multivariate hermite expansion of maxwell–boltzmann distribution by using taylor expansion?," *Zeitschrift für angewandte Mathematik und Physik*, vol. 64, pp. 473–482, 2013.

[27] D. d'Humieres, "Generalized lattice-boltzmann equations," *Rarefied gas dynamics*, 1992.

[28] D. d'Humières, "Multiple–relaxation–time lattice boltzmann models in three dimensions," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 360, no. 1792, pp. 437–451, 2002.

[29] J. Somers, "Direct simulation of fluid flow with cellular automata and the lattice-boltzmann equation," *Applied Scientific Research*, vol. 51, pp. 127–133, 1993.

[30] J. Eggels and J. Somers, "Numerical simulation of free convective flow using the lattice-boltzmann scheme," *International Journal of Heat and Fluid Flow*, vol. 16, no. 5, pp. 357–364, 1995.

[31] Z. Cheng-Wen, X. Jian-Fei, Z. Cong-Shan, X. Sheng-Wei, and Y. Da-Chuan, "Simulation of natural convection under high magnetic field by means of the thermal lattice boltzmann method," *Chinese Physics B*, vol. 18, no. 10, p. 4083, 2009.

[32] C. Zhuo and C.-w. Zhong, "Les-based filter-matrix lattice boltzmann model for simulating turbulent natural convection in a square cavity," *International Journal of Heat and Fluid Flow*, vol. 42, pp. 10–22, 2013.

[33] C. Zhuo and C.-w. Zhong, "Les-based filter-matrix lattice boltzmann model for simulating fully developed turbulent channel flow," *International Journal of Computational Fluid Dynamics*, vol. 30, no. 7-10, pp. 543–553, 2016.

[34] M. Rohde, D. Kandhai, J. J. Derksen, and H. E. Van den Akker, "A generic, mass conservative local grid refinement technique for lattice-boltzmann schemes," *International journal for numerical methods in fluids*, vol. 51, no. 4, pp. 439–468, 2006.

[35] M. Rohde, "The filter-matrix lattice-boltzmann method," *Unpublished, work in progress*, 2025.

[36] S. Succi, G. Amati, and R. Benzi, "Challenges in lattice boltzmann computing," *Journal of statistical physics*, vol. 81, pp. 5–16, 1995.

[37] H. Sajjadi, M. Salmanzadeh, G. Ahmadi, and S. Jafari, "Lattice boltzmann method and rans approach for simulation of turbulent flows and particle transport and deposition," *Particuology*, vol. 30, pp. 62–72, 2017.

[38] D. d'Humieres, M. Bouzidi, and P. Lallemand, "Thirteen-velocity three-dimensional lattice boltzmann model," *Physical Review E*, vol. 63, no. 6, p. 066702, 2001.

[39] K. Petkov, F. Qiu, Z. Fan, A. E. Kaufman, and K. Mueller, "Efficient lbm visual simulation on face-centered cubic lattices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 5, pp. 802–814, 2009.

[40] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, "Sympy: symbolic computing in python," *PeerJ Computer Science*, vol. 3, p. e103, Jan. 2017.

A

Elaboration on used syntax

The extensive use of tensor notation in this thesis may be somewhat difficult to follow. The notation for the tensors in lattice Boltzmann derivation changes somewhat between different articles, and this report has tried to use the notation in Krüger et al. [12] as a basis. However, to avoid confusion and clarify tensor algebra operators some new notation is introduced. First, some initial clarifications.

1. Spatial indices $(\alpha, \beta, \cdots)$ are treated with tensor notation
2. Indices corresponding to the second order polynomials are written as $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta}$ and means that symmetric combinations only appear once in all tensor algebra
3. Lattice direction indices i are *not* subject to tensor notation rules. Hadamard products $\odot$ always applies over this dimension

A.1 Second order products

The stresses in the used formulation of the Navier Stokes equation and the Hermite polynomials are symmetric, meaning $\tau_{\alpha\beta} = \tau^T_{\alpha\beta}$ and $\mathcal{H}_{2\alpha\beta} = \mathcal{H}^T_{2\alpha\beta}$, thus special care has to be taken when doing tensor algebra. To illustrate the issue an example of a double contraction calculation subject to symmetry $\alpha_{2\delta\gamma} = \alpha_{2\gamma\delta}$ is shown. It can be treated in different ways, initially following standard rules for tensor notation, ($\delta_{\alpha\beta}$ substituting α with β indices and vice versa)

$$(\delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma})\alpha_{2\alpha\beta} = \delta_{\alpha\gamma}\delta_{\beta\delta}\alpha_{2\alpha\beta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\alpha_{2\alpha\beta} \tag{A.1}$$

$$= \alpha_{2\gamma\delta} + \alpha_{2\delta\gamma} \tag{A.2}$$

$$= 2\alpha_{2\gamma\delta} = \begin{bmatrix} 2\alpha_{xx} \\ 2\alpha_{xy} \\ 2\alpha_{yy} \end{bmatrix} \tag{A.3}$$

This could also be expanded into matrices

$$(\delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma})\alpha_{2\alpha\beta} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} \alpha_{yy} \\ \alpha_{xy} \\ \alpha_{yx} \\ \alpha_{yy} \end{bmatrix} = \begin{bmatrix} 2\alpha_{yy} \\ \alpha_{xy} + \alpha_{yx} \\ \alpha_{yx} + \alpha_{xy} \\ 2\alpha_{yy} \end{bmatrix} \Leftrightarrow \begin{bmatrix} 2\alpha_{yy} \\ 2\alpha_{xy} \\ 2\alpha_{yy} \end{bmatrix} \tag{A.4}$$

alternatively treating the symmetric parts implicitly results in

$$(\delta_{\tilde{\alpha}\tilde{\gamma}}\delta_{\tilde{\beta}\tilde{\delta}} + \delta_{\tilde{\alpha}\tilde{\delta}}\delta_{\tilde{\beta}\tilde{\gamma}})\alpha_{2\tilde{\alpha}\tilde{\beta}} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} \alpha_{xx} \\ \alpha_{xy} \\ \alpha_{yy} \end{bmatrix} = \begin{bmatrix} 2\alpha_{xx} \\ \alpha_{xy} \\ 2\alpha_{yy} \end{bmatrix} = (1 + \delta_{\tilde{\gamma}\tilde{\delta}})\alpha_{2\tilde{\gamma}\tilde{\delta}}. \tag{A.5}$$

It is the last case that is of interest since the Hermite polynomials are not properly represented for symmetric indices. Thus $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta}$ are introduced. That is the implicit treatment of the indices above is what is actually computed in numerical implementations. The notation is that a tensor of the for $M_{\tilde{\alpha}\tilde{\beta}\tilde{\gamma}\tilde{\delta}}$ can be thought to be expanded as

$$\begin{bmatrix} & \tilde{\gamma}=x,\tilde{\delta}=x & \tilde{\gamma}=x,\tilde{\delta}=y & \tilde{\gamma}=y,\tilde{\delta}=y \\ & \downarrow & \downarrow & \downarrow \\ \tilde{\alpha}=x,\tilde{\beta}=x \;\rightarrow & \delta_{xx}\delta_{xx}+\delta_{xx}\delta_{xx} & \delta_{xx}\delta_{xy}+\delta_{xy}\delta_{xx} & \delta_{xy}\delta_{xy}+\delta_{xy}\delta_{xy} \\ \tilde{\alpha}=x,\tilde{\beta}=y \;\rightarrow & \delta_{xx}\delta_{yx}+\delta_{xx}\delta_{yx} & \delta_{xx}\delta_{yy}+\delta_{xy}\delta_{yx} & \delta_{xy}\delta_{yy}+\delta_{xy}\delta_{yy} \\ \tilde{\alpha}=y,\tilde{\beta}=y \;\rightarrow & \delta_{yx}\delta_{yx}+\delta_{yx}\delta_{yx} & \delta_{yx}\delta_{yy}+\delta_{yy}\delta_{yx} & \delta_{yy}\delta_{yy}+\delta_{yy}\delta_{yy} \end{bmatrix}. \tag{A.6}$$

A.2 Computations in original equilibrium distribution

In the second order equilibrium distribution eq. ($f^{(0)}$) the product $u_\alpha u_\beta(c_{i\alpha} \odot c_{i\beta})$ appears. Which for a given i straight forward to compute. Take the case of D2Q9. The velocity corresponds to the matrix

$$u_\alpha u_\beta = \begin{bmatrix} u_x u_x & u_x u_y \\ u_y u_x & u_y u_y \end{bmatrix} \tag{A.7}$$

and considering the general lattice speed in direction i as well as the specific examples $c_{1\alpha} = [1, 0]$ and $c_{5\alpha} = [1, 1]$ results in the matrices

$$c_{i\alpha}c_{i\beta} = \begin{bmatrix} c_{ix}c_{ix} & c_{iy}c_{ix} \\ c_{ix}c_{iy} & c_{iy}c_{iy} \end{bmatrix} \quad , \quad c_{1\alpha}c_{1\beta} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad , \quad c_{5\alpha}c_{5\beta} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{A.8}$$

then performing the double contraction in the first case is obviously

$$u_\alpha u_\beta c_{1\alpha}c_{1\beta} = u_x^2 \tag{A.9}$$

and in the second case

$$u_\alpha u_\beta c_{5\alpha}c_{5\beta} = u_x^2 + u_x u_y + u_y u_x + u_y^2 = u_x^2 + 2u_x u_y + u_y^2, \tag{A.10}$$

which is the same the expansion in example 3.6, eq. (3.65) in Krüger et al. [12]. However, when switching to the $\tilde{\alpha}, \tilde{\beta}$ notation, only part of these matrices are actually represented, in some sense it is thought to be

$$u_{\tilde{\alpha}} u_{\tilde{\beta}} = \begin{bmatrix} u_x u_x & u_x u_y \\ \times & u_y u_y \end{bmatrix} \tag{A.11}$$

and likewise

$$c_{i\tilde{\alpha}}c_{i\tilde{\beta}} = \begin{bmatrix} c_{ix}c_{ix} & c_{iy}c_{ix} \\ \times & c_{iy}c_{iy} \end{bmatrix} \quad , \quad c_{1\tilde{\alpha}}c_{1\tilde{\beta}} = \begin{bmatrix} 1 & 0 \\ \times & 0 \end{bmatrix} \quad , \quad c_{5\tilde{\alpha}}c_{5\tilde{\beta}} = \begin{bmatrix} 1 & 1 \\ \times & 1 \end{bmatrix} \tag{A.12}$$

thus, to preserve the initial calculation, the lattice velocities must be adjusted to

$$c_{i\alpha}c_{i\beta} = \begin{bmatrix} c_{ix}c_{ix} & 2c_{iy}c_{ix} \\ \times & c_{iy}c_{iy} \end{bmatrix} \tag{A.13}$$

which can be though of as

$$c_{i\alpha}c_{i\beta} = \frac{2}{1+\delta_{\tilde{\alpha}\tilde{\beta}}} c_{i\tilde{\alpha}}c_{i\tilde{\beta}}. \tag{A.14}$$

These matrices can of course be flattened for a representation more similar to the expansion in eq. (A.6), although computationally it should be thought of as the above when performing double contractions.

B

Inner products of scaled Hermite basis

These components all appear on the diagonal of $\langle\langle E_{ki}, E_{ik}\rangle\rangle$. For the zeroth order Hermitian polynomial $\mathcal{H}_0$

$$\begin{aligned}\langle \mathcal{H}_{0,i}, \mathcal{H}_{0,i}\rangle &= \langle 1_i, 1_i\rangle \\ &= \sum_i \omega_i 1_i 1_i \\ &= 1,\end{aligned}$$

and first order polynomials

$$\begin{aligned}\langle \mathcal{H}_{1\alpha,i}, \mathcal{H}_{1\beta,i}\rangle &= \left\langle \frac{c_{i\alpha}}{c_s^2}, \frac{c_{i\beta}}{c_s^2} \right\rangle \\ &= \sum_i \omega_i \frac{c_{i\alpha}}{c_s^2}\frac{c_{i\beta}}{c_s^2} \\ &= \frac{1}{c_s^4}\sum_i \omega_i c_{i\alpha} c_{i\beta} \\ &= \frac{\delta_{\alpha\beta}}{c_s^2},\end{aligned}$$

and finally second order polynomials

$$
\begin{aligned}
\langle \mathcal{H}_{2\alpha\beta,i}, \mathcal{H}_{2\gamma\delta,i} \rangle &= \left\langle \frac{1}{2c_s^4} \left(c_{i\alpha} \odot c_{i\beta} - c_s^2 \delta_{\alpha\beta} 1_i \right), \frac{1}{2c_s^4} \left(c_{i\gamma} \odot c_{i\delta} - c_s^2 \delta_{\gamma\delta} 1_i \right) \right\rangle \\
&= \omega_i \odot \frac{1}{2c_s^4} \left(c_{i\alpha} \odot c_{i\beta} - c_s^2 \delta_{\alpha\beta} 1_i \right) \odot \frac{1}{2c_s^4} \left(c_{i\gamma} \odot c_{i\delta} - c_s^2 \delta_{\gamma\delta} 1_i \right) \\
&= \frac{\sum_i \omega_i}{4c_s^8} \odot \Big(c_{i\alpha} \odot c_{i\beta} \odot c_{i\gamma} \odot c_{i\delta} - c_s^2 \delta_{\gamma\delta} c_{i\alpha} \odot c_{i\beta} \odot 1_i \\
&- c_s^2 \delta_{\alpha\beta} c_{i\gamma} \odot c_{i\delta} \odot 1_i + c_s^4 \delta_{\alpha\beta} \delta_{\gamma\delta} 1_i \odot 1_i \Big) \\
&= \frac{1}{4c_s^8} \Big(\sum_i \omega_i \odot c_{i\alpha} \odot c_{i\beta} \odot c_{i\gamma} \odot c_{i\delta} - c_s^2 \delta_{\gamma\delta} \sum_i \omega_i \odot c_{i\alpha} \odot c_{i\beta} \\
&- c_s^2 \delta_{\alpha\beta} \sum_i \omega_i \odot c_{i\gamma} \odot c_{i\delta} + c_s^4 \delta_{\alpha\beta} \delta_{\gamma\delta} \sum_i \omega_i \Big) \\
&= \frac{1}{4c_s^8} \left(\sum_i \omega_i \odot c_{i\alpha} \odot c_{i\beta} \odot c_{i\gamma} \odot c_{i\delta} - c_s^2 \delta_{\gamma\delta} c_s^2 \delta_{\alpha\beta} - c_s^2 \delta_{\alpha\beta} c_s^2 \delta_{\gamma\delta} + c_s^4 \delta_{\alpha\beta} \delta_{\gamma\delta} \right) \\
&= \frac{1}{4c_s^4} \left(\delta_{\alpha\beta} \delta_{\gamma\delta} + \delta_{\alpha\gamma} \delta_{\beta\delta} + \delta_{\alpha\delta} \delta_{\beta\gamma} - \delta_{\alpha\beta} \delta_{\gamma\delta} - \delta_{\alpha\beta} \delta_{\gamma\delta} + \delta_{\alpha\beta} \delta_{\gamma\delta} \right) \\
&= \frac{\delta_{\alpha\gamma} \delta_{\beta\delta} + \delta_{\alpha\delta} \delta_{\beta\gamma}}{4c_s^4}.
\end{aligned}
$$

where definitions of higher order sums have been used from eqs. ($\sum \omega$) to ($\sum \omega c^4$). This means that the values of those sums are required for a valid velocity scheme. This is then used for the rescaled polynomials as

$$
\begin{aligned}
\left\langle \mathcal{H}_{2\tilde{\alpha}\tilde{\beta},i}, \mathcal{H}_{2\tilde{\gamma}\tilde{\delta},i} \right\rangle &= \left\langle \frac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4} \left(c_{i\tilde{\alpha}} c_{i\tilde{\beta}} - c_s^2 \delta_{\tilde{\alpha}\tilde{\beta}} 1_i \right), \frac{1}{(1+\delta_{\tilde{\gamma}\tilde{\delta}})c_s^4} \left(c_{i\tilde{\gamma}} c_{i\tilde{\delta}} - c_s^2 \delta_{\tilde{\gamma}\tilde{\delta}} 1_i \right) \right\rangle \\
&= \frac{\delta_{\tilde{\alpha}\tilde{\gamma}} \delta_{\tilde{\beta}\tilde{\delta}} + \delta_{\tilde{\alpha}\tilde{\delta}} \delta_{\tilde{\beta}\tilde{\gamma}}}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})(1+\delta_{\tilde{\gamma}\tilde{\delta}})c_s^4}
\end{aligned}
$$

which gives the magnitudes

$$
\begin{aligned}
\left\langle \mathcal{H}_{2\tilde{\alpha}\tilde{\beta},i}, \mathcal{H}_{2\tilde{\alpha}\tilde{\beta},i} \right\rangle &= \frac{\delta_{\tilde{\alpha}\tilde{\alpha}} \delta_{\tilde{\beta}\tilde{\beta}} + \delta_{\tilde{\alpha}\tilde{\beta}} \delta_{\tilde{\beta}\tilde{\alpha}}}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})^2 c_s^4} \\
&= \begin{cases} \dfrac{1}{2c_s^4} & \tilde{\alpha} = \tilde{\beta} \\ \dfrac{1}{c_s^4} & \tilde{\alpha} \neq \tilde{\beta} \end{cases} \\
&= \frac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4},
\end{aligned}
\tag{B.1}
$$

here it is clear that the notation $\delta_{\tilde{\alpha}\tilde{\beta}}$ cannot be thought to represent a matrix, as the dimensions for the final expression would not make sense. It should instead be thought of as a indices, similarly to how i is used in Krüger et al. [12].

For appendix C the product between the rescaled and non-rescaled second order is also needed

$$\begin{aligned}\left\langle \mathcal{H}_{2\alpha\beta,i}, \mathcal{H}_{2\tilde{\gamma}\tilde{\delta},i} \right\rangle &= \left\langle \frac{1}{2c_s^4}\left(c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta}1_i\right), \frac{1}{(1+\delta_{\tilde{\gamma}\tilde{\delta}})c_s^4}\left(c_{i\tilde{\gamma}}c_{i\tilde{\delta}} - c_s^2\delta_{\tilde{\gamma}\tilde{\delta}}1_i\right)\right\rangle \\ &= \frac{\delta_{\alpha\tilde{\gamma}}\delta_{\beta\tilde{\delta}} + \delta_{\alpha\tilde{\delta}}\delta_{\beta\tilde{\gamma}}}{2(1+\delta_{\tilde{\gamma}\tilde{\delta}})c_s^4}.\end{aligned}$$

It is also easy to show the following, which are direct consequences of the orthogonal basis

$$\begin{aligned}\langle \mathcal{H}_{0,i}, \mathcal{H}_{1\alpha,i} \rangle &= \left\langle 1_i, \frac{c_{i\alpha}}{c_s^2} \right\rangle &&= 0 \\ \left\langle \mathcal{H}_{1\alpha,i}, \mathcal{H}_{2\tilde{\alpha}\tilde{\beta},i} \right\rangle &= \left\langle \frac{c_{i\alpha}}{c_s^2}, \frac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4}\left(c_{i\tilde{\alpha}}c_{i\tilde{\beta}} - c_s^2\delta_{\tilde{\alpha}\tilde{\beta}}\right)\right\rangle &&= 0 \\ \left\langle \mathcal{H}_{2\tilde{\alpha}\tilde{\beta},i}, \mathcal{H}_{0,i} \right\rangle &= \left\langle \frac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4}\left(c_{i\tilde{\alpha}}c_{i\tilde{\beta}} - c_s^2\delta_{\tilde{\alpha}\tilde{\beta}}\right), 1_i \right\rangle &&= 0,\end{aligned}$$

and for the derivation in appendix C it is also important to note that the orthogonality property also holds for $\mathcal{H}_{3\alpha\beta\gamma,i}$ as well as any redefined set $\mathcal{H}_{3\tilde{\alpha}\tilde{\beta}\tilde{\gamma},i}$ if it is only rescaled by constants.

Based on the inner products calculated above, the gram matrix D to second order is represented as

$$\begin{aligned}D &= \langle\langle E_{ki}, E_{ik} \rangle\rangle \\ &= \begin{bmatrix} \left\langle \mathcal{H}_i^{(0)}, \mathcal{H}_i^{(0)} \right\rangle & \left\langle \mathcal{H}_i^{(0)}, \mathcal{H}_{\alpha,i}^{(1)} \right\rangle & \left\langle \mathcal{H}_i^{(0)}, \mathcal{H}_{\tilde{\alpha}\tilde{\beta},i}^{(2)} \right\rangle \\ \left\langle \mathcal{H}_{\alpha,i}^{(1)}, \mathcal{H}_i^{(0)} \right\rangle & \left\langle \mathcal{H}_{\alpha,i}^{(1)}, \mathcal{H}_{\beta,i}^{(1)} \right\rangle & \left\langle \mathcal{H}_{\alpha,i}^{(1)}, \mathcal{H}_{\tilde{\alpha}\tilde{\beta},i}^{(2)} \right\rangle \\ \left\langle \mathcal{H}_{\tilde{\alpha}\tilde{\beta},i}^{(2)}, \mathcal{H}_i^{(0)} \right\rangle & \left\langle \mathcal{H}_{\tilde{\alpha}\tilde{\beta},i}^{(2)}, \mathcal{H}_{\alpha,i}^{(1)} \right\rangle & \left\langle \mathcal{H}_{\tilde{\alpha}\tilde{\beta},i}^{(2)}, \mathcal{H}_{\tilde{\gamma}\tilde{\delta},i}^{(2)} \right\rangle \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \dfrac{\delta_{\alpha\beta}}{c_s^2} & 0 \\ 0 & 0 & \dfrac{\delta_{\tilde{\alpha}\tilde{\gamma}}\delta_{\tilde{\beta}\tilde{\delta}} + \delta_{\tilde{\alpha}\tilde{\delta}}\delta_{\tilde{\beta}\tilde{\gamma}}}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})(1+\delta_{\tilde{\gamma}\tilde{\delta}})c_s^4}, \end{bmatrix}\end{aligned} \tag{B.2}$$

which is composed of three blocks on the diagonal, corresponding to the zeroth, first and second order Hermite polynomials respectively, with all entries not defined by these three blocks being zeros.

Due to the definition of $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta}$ it is not immediately obvious that D is diagonal, although when expanding like eq. (A.6) or numerically computing it as in appendix D shows that this is the case. To simplify the notation, the diagonals are represented as below,

$$D = \text{diagonal}\left[1, \frac{1_\alpha}{c_s^2}, \frac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4}\right] \tag{B.3}$$

which can be expanded depending on the dimension, e.g. in two dimensions it would be $D = \text{diagonal}\left[1, 1/c_s^2, 1/c_s^2, 1/(2c_s^4), 1/c_s^4, 1/(2c_s^4)\right]$. In this representation, it is clear that the inverse of the matrix is given by

$$D^{-1} = \text{diagonal}\left[1, 1_\alpha c_s^2, (1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4\right]. \tag{B.4}$$

The inverse transformation $\mathcal{T}_{f\to\alpha}$ has the matrix representation $D^{-1}E_{ki}$, which is calculated with the transpose of E_{ik}, which is just Hermite basis vectors times the corresponding entry in D^{-1}. To zeroth and first order this computation is straight forward, for the second order it is given by

$$\begin{aligned}(D^{-1}_{\tilde{\alpha}\tilde{\beta}}1_i) \odot E_{2(\tilde{\alpha}\tilde{\beta})\times i} &= ((1+\delta_{\tilde{\alpha}\tilde{\beta}})1_i c_s^4) \odot \frac{1}{(1+\delta_{\tilde{\alpha}\tilde{\beta}})c_s^4}\left(c_{i\tilde{\alpha}} \odot c_{i\tilde{\beta}} - c_s^2\delta_{\tilde{\alpha}\tilde{\beta}}1_i\right) \\ &= c_{i\tilde{\alpha}} \odot c_{i\tilde{\beta}} - c_s^2\delta_{\tilde{\alpha}\tilde{\beta}}1_i.\end{aligned} \tag{B.5}$$

C

Transforming spatial derivatives to moment space

As will be seen for the Enskog analysis, a really important special case of transforming between moment and lattice speed representations is when applying the inverse transform to $(c_{i\alpha}\partial_\alpha) \odot f_i$ (up to a second order expansion). This computation is shown below,

$$
\begin{aligned}
\mathcal{T}_{f\to\alpha}((c_{i\alpha}\partial_\alpha)\odot f_i) &= D^{-1}E_{ki}((c_{i\alpha}\partial_\alpha)\odot f_i) \\
&= (c_{i\alpha}\partial_\alpha)(D^{-1}E_{ki})f_i \\
&= \begin{bmatrix} (c_{i\alpha}\partial_\alpha)\odot 1_i \\ (c_{i\alpha}\partial_\alpha)\odot c_{i\beta} \\ (c_{i\alpha}\partial_\alpha)\odot\left(c_{i\tilde{\beta}}\odot c_{i\tilde{\gamma}} - c_s^2\delta_{\tilde{\beta}\tilde{\gamma}}1_i\right) \end{bmatrix}\cdot f_i \\
&= \begin{bmatrix} c_{i\alpha}\partial_\alpha \\ (c_{i\alpha}\odot c_{i\beta})\,\partial_\alpha \\ \left(c_{i\alpha}\odot(c_{i\tilde{\beta}}\odot c_{i\tilde{\gamma}} - c_s^2\delta_{\tilde{\beta}\tilde{\gamma}}1_i)\right)\partial_\alpha \end{bmatrix}\cdot f_i \\
&= \begin{bmatrix} c_s^2\mathcal{H}^{(1)}_{\alpha,i}\cdot\partial_\alpha \\ \left(2c_s^4\mathcal{H}^{(2)}_{\alpha\beta,i} + c_s^2\delta_{\alpha\beta}\mathcal{H}^{(0)}_i\right)\cdot\partial_\alpha \\ \left(6c_s^6\mathcal{H}^{(3)}_{\alpha\tilde{\beta}\tilde{\gamma},i} + c_s^2\left(c_{i\alpha}\delta_{\tilde{\beta}\tilde{\gamma}} + c_{i\tilde{\beta}}\delta_{\alpha\tilde{\gamma}} + c_{i\tilde{\gamma}}\delta_{\tilde{\beta}\alpha}\right) - c_s^2 c_{i\alpha}\delta_{\tilde{\beta}\tilde{\gamma}}\right)\cdot\partial_\alpha \end{bmatrix}\cdot f_i \\
&= \begin{bmatrix} c_s^2\mathcal{H}^{(1)}_{\alpha,i}\cdot\partial_\alpha \\ \left(2c_s^4\mathcal{H}^{(2)}_{\alpha\beta,i} + c_s^2\delta_{\alpha\beta}\mathcal{H}^{(0)}_i\right)\cdot\partial_\alpha \\ \left(6c_s^6\mathcal{H}^{(3)}_{\alpha\tilde{\beta}\tilde{\gamma},i} + c_s^4\left(\delta_{\alpha\tilde{\gamma}}\mathcal{H}^{(1)}_{\tilde{\beta},i} + \delta_{\tilde{\beta}\alpha}\mathcal{H}^{(1)}_{\tilde{\gamma},i}\right)\right)\cdot\partial_\alpha \end{bmatrix}\cdot f_i
\end{aligned}
$$

which is reminiscent of the recursive relations defining the Hermite expansion [26] in eq. (2.17). The third order polynomial has components from the symmetric $\tilde{\beta},\tilde{\gamma}$ representation as well as non-symmetric α, this is not an issue since the orthogonality properties remain the same. Now, using the definition of f_i in terms of α_k for the transform above

$$
\begin{aligned}
&\mathcal{T}_{f\to\alpha}((c_{i\alpha}\partial_\alpha)\odot f_i)\\
&= \mathcal{T}_{f\to\alpha}((c_{i\alpha}\partial_\alpha)\odot\langle E_{ik},\alpha_k\rangle)\\
&= D^{-1}\cdot E_{ki}\cdot((c_{i\alpha}\partial_\alpha)\odot(\omega_i\odot E_{ik}\alpha_k))\\
&= \omega_i\odot\Big([D^{-1}\cdot((c_{i\alpha}\partial_\alpha)\odot E_{ki})]\cdot E_{ik}\Big)\,\alpha_k\\
&= \omega_i\odot\begin{bmatrix} c_s^2\mathcal{H}^{(1)}_{\alpha,i}\cdot\partial_\alpha\\ (2c_s^4\mathcal{H}^{(2)}_{\alpha\beta,i}+c_s^2\delta_{\alpha\beta}\mathcal{H}^{(0)}_i)\cdot\partial_\alpha\\ \Big(6c_s^6\mathcal{H}^{(3)}_{\alpha\tilde\beta\tilde\gamma,i}+c_s^4\Big(\delta_{\alpha\tilde\gamma}\mathcal{H}^{(1)}_{\tilde\beta,i}+\delta_{\tilde\beta\alpha}\mathcal{H}^{(1)}_{\tilde\gamma,i}\Big)\Big)\cdot\partial_\alpha\end{bmatrix}\begin{bmatrix} | & | & |\\ \mathcal{H}^{(0)}_i & \mathcal{H}^{(1)}_{\alpha',i} & \mathcal{H}^{(2)}_{\tilde\alpha'\tilde\beta',i}\\ | & | & |\end{bmatrix}\cdot\alpha_k\\
&= \left\langle\begin{bmatrix} c_s^2\mathcal{H}^{(1)}_{\alpha,i}\cdot\partial_\alpha\\ (2c_s^4\mathcal{H}^{(2)}_{\alpha\beta,i}+c_s^2\delta_{\alpha\beta}\mathcal{H}^{(0)}_i)\cdot\partial_\alpha\\ \Big(6c_s^6\mathcal{H}^{(3)}_{\alpha\tilde\beta\tilde\gamma,i}+c_s^4\Big(\delta_{\alpha\tilde\gamma}\mathcal{H}^{(1)}_{\tilde\beta,i}+\delta_{\tilde\beta\alpha}\mathcal{H}^{(1)}_{\tilde\gamma,i}\Big)\Big)\cdot\partial_\alpha\end{bmatrix},\begin{bmatrix} | & | & |\\ \mathcal{H}^{(0)}_i & \mathcal{H}^{(1)}_{\alpha',i} & \mathcal{H}^{(2)}_{\tilde\alpha'\tilde\beta',i}\\ | & | & |\end{bmatrix}\right\rangle\cdot\alpha_k\\
&= \left\langle\begin{bmatrix} c_s^2\mathcal{H}^{(1)}_{\alpha,i}\cdot\partial_\alpha\\ 2c_s^4\mathcal{H}^{(2)}_{\alpha\beta,i}\cdot\partial_\alpha\\ 6c_s^6\mathcal{H}^{(3)}_{\alpha\tilde\beta\tilde\gamma,i}\cdot\partial_\alpha\end{bmatrix},\begin{bmatrix} | & | & |\\ \mathcal{H}^{(0)}_i & \mathcal{H}^{(1)}_{\alpha',i} & \mathcal{H}^{(2)}_{\tilde\alpha'\tilde\beta',i}\\ | & | & |\end{bmatrix}\right\rangle\cdot\alpha_k\\
&+\left\langle\begin{bmatrix} 0_i\\ \delta_{\alpha\beta}\mathcal{H}^{(0)}_i c_s^2\cdot\partial_\alpha\\ c_s^4\left(\delta_{\alpha\tilde\gamma}\mathcal{H}^{(1)}_{\tilde\beta,i}+\delta_{\tilde\beta\alpha}\mathcal{H}^{(1)}_{\tilde\gamma,i}\right)\cdot\partial_\alpha\end{bmatrix},\begin{bmatrix} | & | & |\\ \mathcal{H}^{(0)}_i & \mathcal{H}^{(1)}_{\alpha',i} & \mathcal{H}^{(2)}_{\tilde\alpha'\tilde\beta',i}\\ | & | & |\end{bmatrix}\right\rangle\cdot\alpha_k\\
&= \begin{bmatrix} 0 & c_s^2\langle\mathcal{H}^{(1)}_{\alpha,i},\mathcal{H}^{(1)}_{\alpha',i}\rangle\partial_\alpha & 0\\ c_s^2\delta_{\alpha\beta}\langle\mathcal{H}^{(0)}_i,\mathcal{H}^{(0)}_i\rangle\partial_\alpha & 0 & 2c_s^4\langle\mathcal{H}^{(2)}_{\alpha\beta,i},\mathcal{H}^{(2)}_{\tilde\alpha'\tilde\beta',i}\rangle\partial_\alpha\\ 0 & c_s^4\Big(\delta_{\alpha\tilde\gamma}\langle\mathcal{H}^{(1)}_{\tilde\beta,i},\mathcal{H}^{(1)}_{\alpha',i}\rangle & 0\\ & +\delta_{\tilde\beta\alpha}\langle\mathcal{H}^{(1)}_{\tilde\gamma,i},\mathcal{H}^{(1)}_{\alpha',i}\rangle\Big)\partial_\alpha & \end{bmatrix}\cdot\alpha_k\\
&= \begin{bmatrix} 0 & c_s^2\dfrac{\delta_{\alpha\alpha'}}{c_s^2}\partial_\alpha & 0\\ c_s^2\delta_{\alpha\beta}\partial_\alpha & 0 & 2c_s^4\dfrac{\delta_{\alpha\tilde\alpha'}\delta_{\beta\tilde\beta'}+\delta_{\alpha\tilde\beta'}\delta_{\beta\tilde\alpha'}}{2c_s^4(1+\delta_{\tilde\alpha'\tilde\beta'})}\partial_\alpha\\ 0 & c_s^4\dfrac{\delta_{\alpha\tilde\gamma}\delta_{\tilde\beta\tilde\alpha'}+\delta_{\tilde\beta\alpha}\delta_{\tilde\gamma\tilde\alpha'}}{c_s^2}\partial_\alpha & 0\end{bmatrix}\cdot\alpha_k\\
&= \begin{bmatrix} \partial_\alpha\delta_{\alpha\alpha'}\alpha_{1\alpha'}\\ c_s^2\partial_\beta\alpha_0+\partial_\alpha\left(\dfrac{\delta_{\alpha\tilde\alpha'}\delta_{\beta\tilde\beta'}+\delta_{\alpha\tilde\beta'}\delta_{\beta\tilde\alpha'}}{1+\delta_{\tilde\alpha'\tilde\beta'}}\alpha_{2\tilde\alpha'\tilde\beta'}\right)\\ c_s^2(\delta_{\alpha\tilde\gamma}\delta_{\tilde\beta\tilde\alpha'}+\delta_{\tilde\beta\alpha}\delta_{\tilde\gamma\tilde\alpha'})\partial_\alpha\alpha_{1\alpha'}\end{bmatrix}
\end{aligned}
$$

Now, the two terms involving second order terms must be handled with care to not do any implicit summations. The "second order contraction"-part of the first problematic product is similar to the discussion in section A.1, and by the same

reasoning, the product must be

$$\begin{aligned}\frac{(\delta_{\alpha\tilde{\alpha}'}\delta_{\beta\tilde{\beta}'}+\delta_{\alpha\tilde{\beta}'}\delta_{\beta\tilde{\alpha}'})\alpha_{2\tilde{\alpha}'\tilde{\beta}'}}{1+\delta_{\tilde{\alpha}'\tilde{\beta}'}} &= \frac{\delta_{\alpha\tilde{\alpha}'}\delta_{\beta\tilde{\beta}'}\alpha_{2\tilde{\alpha}'\tilde{\beta}'}+\delta_{\alpha\tilde{\beta}'}\delta_{\beta\tilde{\alpha}'}\alpha_{2\tilde{\alpha}'\tilde{\beta}'}}{1+\delta_{\tilde{\alpha}'\tilde{\beta}'}} \\ &= \frac{(1+\delta_{\tilde{\alpha}'\tilde{\beta}'})\alpha_{2\alpha\beta}}{1+\delta_{\tilde{\alpha}'\tilde{\beta}'}} \\ &= \alpha_{2\alpha\beta}\end{aligned}$$

it can thought of as when the variable substitutions are done through the kronecker deltas, it must be compensated for the symmetric element not existing with the extra factor, similarly to the last part of appendix A. The second component involving a second order terms there is actually no potential confusion since there is no cross substitutions between the kronecker deltas,

$$\begin{aligned}(\delta_{\alpha\tilde{\gamma}}\delta_{\tilde{\beta}\alpha'}+\delta_{\tilde{\beta}\alpha}\delta_{\tilde{\gamma}\alpha'})(\partial_\alpha\alpha_{1\alpha'}) &= (\delta_{\alpha\tilde{\gamma}}\partial_\alpha)(\delta_{\tilde{\beta}\alpha'}\alpha_{1\alpha'})+(\delta_{\tilde{\beta}\alpha}\partial_\alpha)(\delta_{\tilde{\gamma}\alpha'}\alpha_{1\alpha'}) \\ &= \partial_{\tilde{\gamma}}\alpha_{1\tilde{\beta}}+\partial_{\tilde{\beta}}\alpha_{1\tilde{\gamma}}\end{aligned} \tag{C.1}$$

The indices can now be renamed and the symmetrical notation dropped, this is because there will not be any more second order products performed on the terms, and as such the symmetric notation is equivalent to normal tensor notation. The conclusion from this is that the spatial derivatives are mapped to the moment-space as follows

$$\mathcal{T}_{f\to\alpha}\left(c_{i\alpha}\partial_\alpha f_i\right)=\begin{bmatrix}\partial_\alpha\alpha_{1\alpha}\\ c_s^2\partial_\alpha\alpha_0+\partial_\beta\alpha_{2\alpha\beta}\\ c_s^2\left(\partial_\alpha\alpha_{1\beta}+\partial_\beta\alpha_{1\alpha}\right)\end{bmatrix}. \qquad (\mathcal{T}_{f\to\alpha}\left(c_{i\alpha}\partial_\alpha f_i\right))$$

D

Supplied code

In this thesis, a somewhat convoluted syntax has been introduced. Although the math is consistent, the numerical calculations are accompanied by a numerical Python Notebook using the package `Sympy` [40]. In the code the exact computations done with the abstract syntax in the appendices are performed using an explicit chose of lattice velocities, such as D2Q9 and D3Q15 defined by Krüger et al. [12] and D3Q7 from [24].

The code is found on the following repository
https://github.com/vrogly/filter-matrix-lattice-boltzmann-matrices

As an example for the D3Q15 lattice,

$$c_{i\alpha} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix}^T$$

labeled from 0 to 14. The weights $\omega_0 = 2/9, \omega_{1,2,3,4,5,6} = 1/9, \omega_{7,8,9,10,11,12,13,14} = 1/72$ and lattice speed of sound $c_s^2 = 1/3$. This results in

$$\omega_i E_{ik} = \begin{bmatrix} 2/9 & 0 & 0 & 0 & -1/3 & 0 & 0 & -1/3 & 0 & -1/3 \\ 1/9 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & -1/6 & 0 & -1/6 \\ 1/9 & -1/3 & 0 & 0 & 1/3 & 0 & 0 & -1/6 & 0 & -1/6 \\ 1/9 & 0 & 1/3 & 0 & -1/6 & 0 & 0 & 1/3 & 0 & -1/6 \\ 1/9 & 0 & -1/3 & 0 & -1/6 & 0 & 0 & 1/3 & 0 & -1/6 \\ 1/9 & 0 & 0 & 1/3 & -1/6 & 0 & 0 & -1/6 & 0 & 1/3 \\ 1/9 & 0 & 0 & -1/3 & -1/6 & 0 & 0 & -1/6 & 0 & 1/3 \\ 1/72 & 1/24 & 1/24 & 1/24 & 1/24 & 1/8 & 1/8 & 1/24 & 1/8 & 1/24 \\ 1/72 & -1/24 & -1/24 & -1/24 & 1/24 & 1/8 & 1/8 & 1/24 & 1/8 & 1/24 \\ 1/72 & 1/24 & 1/24 & -1/24 & 1/24 & 1/8 & -1/8 & 1/24 & -1/8 & 1/24 \\ 1/72 & -1/24 & -1/24 & 1/24 & 1/24 & 1/8 & -1/8 & 1/24 & -1/8 & 1/24 \\ 1/72 & 1/24 & -1/24 & 1/24 & 1/24 & -1/8 & 1/8 & 1/24 & -1/8 & 1/24 \\ 1/72 & -1/24 & 1/24 & -1/24 & 1/24 & -1/8 & 1/8 & 1/24 & -1/8 & 1/24 \\ 1/72 & -1/24 & 1/24 & 1/24 & 1/24 & -1/8 & -1/8 & 1/24 & 1/8 & 1/24 \\ 1/72 & 1/24 & -1/24 & -1/24 & 1/24 & -1/8 & -1/8 & 1/24 & 1/8 & 1/24 \end{bmatrix}$$

and

$$D^{-1}E_{ki} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1/3 & 0 & 0 & -1/3 & 0 & -1/3 \\ 1 & 1 & 0 & 0 & 2/3 & 0 & 0 & -1/3 & 0 & -1/3 \\ 1 & -1 & 0 & 0 & 2/3 & 0 & 0 & -1/3 & 0 & -1/3 \\ 1 & 0 & 1 & 0 & -1/3 & 0 & 0 & 2/3 & 0 & -1/3 \\ 1 & 0 & -1 & 0 & -1/3 & 0 & 0 & 2/3 & 0 & -1/3 \\ 1 & 0 & 0 & 1 & -1/3 & 0 & 0 & -1/3 & 0 & 2/3 \\ 1 & 0 & 0 & -1 & -1/3 & 0 & 0 & -1/3 & 0 & 2/3 \\ 1 & 1 & 1 & 1 & 2/3 & 1 & 1 & 2/3 & 1 & 2/3 \\ 1 & -1 & -1 & -1 & 2/3 & 1 & 1 & 2/3 & 1 & 2/3 \\ 1 & 1 & 1 & -1 & 2/3 & 1 & -1 & 2/3 & -1 & 2/3 \\ 1 & -1 & -1 & 1 & 2/3 & 1 & -1 & 2/3 & -1 & 2/3 \\ 1 & 1 & -1 & 1 & 2/3 & -1 & 1 & 2/3 & -1 & 2/3 \\ 1 & -1 & 1 & -1 & 2/3 & -1 & 1 & 2/3 & -1 & 2/3 \\ 1 & -1 & 1 & 1 & 2/3 & -1 & -1 & 2/3 & 1 & 2/3 \\ 1 & 1 & -1 & -1 & 2/3 & -1 & -1 & 2/3 & 1 & 2/3 \end{bmatrix}^T$$

which results in the following transformation of the spatial derivatives

$$\mathcal{T}_{f\to\alpha}((c_{i\alpha}\partial_\alpha)\odot f_i) = \begin{bmatrix} \partial_x\alpha_{\phi,1x} + \partial_y\alpha_{\phi,1y} + \partial_z\alpha_{\phi,1z} \\ \frac{\partial_x\alpha_{\phi,0}}{3} + \partial_x\alpha_{\phi,2xx} + \partial_y\alpha_{\phi,2xy} + \partial_z\alpha_{\phi,2xz} \\ \frac{\partial_y\alpha_{\phi,0}}{3} + \partial_x\alpha_{\phi,2xy} + \partial_y\alpha_{\phi,2yy} + \partial_z\alpha_{\phi,2yz} \\ \frac{\partial_z\alpha_{\phi,0}}{3} + \partial_x\alpha_{\phi,2xz} + \partial_y\alpha_{\phi,2yz} + \partial_z\alpha_{\phi,2zz} \\ \frac{2\partial_x\alpha_{\phi,1x}}{3} \\ \frac{\partial_y\alpha_{\phi,1x}}{3} + \frac{\partial_x\alpha_{\phi,1y}}{3} \\ \frac{\partial_z\alpha_{\phi,1x}}{3} + \frac{\partial_x\alpha_{\phi,1z}}{3} \\ \frac{2\partial_y\alpha_{\phi,1y}}{3} \\ \frac{\partial_z\alpha_{\phi,1y}}{3} + \frac{\partial_y\alpha_{\phi,1z}}{3} \\ \frac{2\partial_z\alpha_{\phi,1z}}{3} \end{bmatrix}.$$

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

CHALMERS
UNIVERSITY OF TECHNOLOGY