Solving Kinetic Equations Using Entropic Quadrature and the Discontinuous Galerkin Method

Die Lösung von kinetischen Gleichungen mit entropischer Quadratur und der diskontinuierlichen Galerkin Methode

Bachelor thesis
Computer Science

July 2018

Vorgelegt von Niclas Böhmer
Presented by student number: 355902
niclas.boehmer@rwth-aachen.de

Erstprüfer Prof. Dr. Uwe Naumann
First examiner Lehrstuhl für Informatik 12
RWTH Aachen University

Zweitprüfer Prof. Dr. Manuel Torrilhon
Second examiner Lehrstuhl für Mathematik (MathCCES)
RWTH Aachen University
Eigenständigkeitserklärung


Aachen, im Juli 2018

Niclas Böhmer
## Contents

List of Figures V  
List of Tables VI  

1. Introduction 1  
   1.1. Organisation of the thesis ...................................... 2  

2. Kinetic theory 3  

3. Moment equations 5  
   3.1. Central and conserved moments ................................. 7  
   3.2. The method of moments and different governing equations .... 8  
   3.3. Realisability of moments ...................................... 9  

4. The closure problem 9  
   4.1. Problem of moments ........................................... 10  
   4.2. Desirable properties ........................................... 11  
   4.3. Grad’s closure .................................................. 12  
   4.4. Maximum-entropy closure ..................................... 13  
      4.4.1. Some general notes on information entropy ............... 13  
      4.4.2. Maximising entropy ....................................... 14  
      4.4.3. Entropy in statistical mechanics and thermodynamics ...... 15  
      4.4.4. H-theorem ................................................. 16  
      4.4.5. Thermodynamic and information entropy .................... 16  
      4.4.6. Maximum-entropy closure in kinetic theory ............... 17  
   4.5. Quadrature method of moments .................................. 19  
      4.5.1. A two-node closure for kinetic equations ................. 20  
      4.5.2. Higher order methods for the QMOM ....................... 20  
      4.5.3. Properties of the QMOM ................................... 23  
      4.5.4. Relationship to other methods and extensions ............. 24  

5. Entropic Quadrature 26  
   5.1. Overview of Entropic Quadrature ................................ 27  
      5.1.1. One-dimensional case ...................................... 31  
      5.1.2. Extension to multi-dimensions ............................. 33  
   5.2. Calculating entropic quadratures ................................ 34  
      5.2.1. Infeasible Newton method for entropic quadrature ........ 34  
      5.2.2. Calculating entropic quadratures using Lagrange duality ... 36  
      5.2.3. Properties of optimising via the Lagrange dual ........... 39  
   5.3. Properties of Entropic Quadrature ............................. 42  
      5.3.1. Relationship to the QMOM and maximum-entropy closure .... 42  
   5.4. Dealing with non-realisable moments and problems of EQ in multiple dimensions 44  

6. Numerical scheme 48  
   6.1. Finite Volume Method ......................................... 48  
      6.1.1. Numerical flux functions .................................. 50  
   6.2. Discontinuous Galerkin method ................................ 52  
      6.2.1. Timestepping schemes ..................................... 55
7. Numerical results
  7.1. Entropic Quadrature for predicting higher order moments
  7.1.1. Maximum-entropy distribution in one dimension
  7.1.2. Sums of two maximum-entropy distributions
  7.1.3. Maximum entropy distribution in two dimensions
  7.2. Kinetic equations
  7.2.1. Smooth perturbation
  7.2.2. Two-beam problem
  7.2.3. Stationary shock
  7.2.4. Smooth perturbation in two dimensions

8. Conclusion

A. Visualisation of the numerical results
  A.1. Two-beam problem
  A.2. Stationary shock

B. Infeasible Newton method
  B.1. Minimising a quadratic function under linear constraints
  B.2. Infeasible Newton method

C. Further on Finite volume methods
  C.1. Reconstruction techniques: Higher order finite volume methods
  C.2. Realisability
  C.3. Further remarks on theoretical properties

D. Further on the discontinuous Galerkin method
  D.1. Detailed derivation of the Discontinuous Galerkin method in two dimensions
  D.2. Relationship between the nodal and the modal form
  D.3. Limiter and Filtering
  D.4. Time step size and order of convergence

E. Validation of numerical scheme
  E.1. Advection equation
  E.2. Euler equations

F. Riemann problem

References
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example output of algorithm 2 and algorithm 3 on bimodal Gaussian</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>Overview of all steps involved in solving the Boltzmann equation</td>
<td>57</td>
</tr>
<tr>
<td>3</td>
<td>Average relative error of different versions of EQ in the maximum-entropy test case with $M = 8$</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>Average relative error of different versions of EQ in the sum of maximum-entropy distribution test case with $M = 8$</td>
<td>62</td>
</tr>
<tr>
<td>5</td>
<td>Average error of different versions of EQ in the maximum-entropy test case in two dimensions with $M = 6$</td>
<td>63</td>
</tr>
<tr>
<td>6</td>
<td>Plot of the initial and final density of the smooth perturbation test case and various solutions of the implemented solver</td>
<td>65</td>
</tr>
<tr>
<td>7</td>
<td>Error of the density in the infinity norm calculated using EQ and DG with various degrees in two dimensions with $M = 21$</td>
<td>71</td>
</tr>
<tr>
<td>8</td>
<td>Solutions of the two-beam problem with $v_0 = 0.5$ calculated using the discontinuous Galerkin method</td>
<td>75</td>
</tr>
<tr>
<td>9</td>
<td>Solutions of the two-beam problem with $v_0 = 0.5$ calculated using the finite volume method</td>
<td>76</td>
</tr>
<tr>
<td>10</td>
<td>Comparison of different finite volume schemes for solving the two-beam problem with $v_0 = 0.5$</td>
<td>77</td>
</tr>
<tr>
<td>11</td>
<td>Solutions of the two-beam problem with $v_0 = 2$ calculated using the finite volume method</td>
<td>77</td>
</tr>
<tr>
<td>12</td>
<td>Stationary shock profile for $Ma=2$; FV, $D=1$, LLF, $N_x = 200$</td>
<td>78</td>
</tr>
<tr>
<td>13</td>
<td>Stationary shock profile for $Ma=4$; FV, $D=1$, LLF, $N_x = 200$</td>
<td>78</td>
</tr>
<tr>
<td>14</td>
<td>Stationary shock profile for $Ma=8$; FV, $D=1$, $SF$, $N_x = 400$</td>
<td>79</td>
</tr>
<tr>
<td>15</td>
<td>Riemann problem</td>
<td>97</td>
</tr>
<tr>
<td>16</td>
<td>Riemann problem for different Knudsen numbers solved using the discontinuous Galerkin method</td>
<td>97</td>
</tr>
</tbody>
</table>
List of Tables

1. Relative error of the predicted higher order moment of a set of maximum-entropy distributions ............................................. 59
2. $e(\rho)$ of the solution calculated by different closures for the smooth perturbation test case for different numbers of moments with DG, $N_x = 100$, D=1, LLF . . . . 66
3. $e(\rho)$ and order of convergence of the DG method for various degrees with LLF, $T = 5$ ................................................................. 66
4. $e(\rho)$ and order of convergence of the FV method for various degrees with KF, $T = 5$ ................................................................. 67
5. Error of solution of the discontinuous Galerkin for the advection equation . . . . 95
6. Error of solution of the discontinuous Galerkin for a steady manufactured solution of the Euler equations ........................................ 96
7. Error of solution of the discontinuous Galerkin for a transient manufactured solution of the Euler equations ........................................ 96
1. Introduction

The ultimate goal of the kinetic theory of gases is to model a gas from a microscopic point of view. Unfortunately, this task is a very challenging problem since to solve it exactly one needs to keep track of every single particle. Due to the fact that the number of particles in a gas is extraordinary high, this strategy is usually not possible.\(^1\)

Instead of the position of every particle, the distribution of particles is usually considered as the quantity of interest because this distribution entails all necessary information [90]. The evolution of this distribution in the context of describing a monoatomic ideal gas is governed by the Boltzmann equation. For arbitrary conditions and especially for systems far away from equilibrium, solving this equation has proven to be challenging because no fast and accurate solver of this equation is known [90].

Nevertheless, there exist several techniques to solve a wide range of problems of interest. The behaviour of a gas, and as a result, the methods for the simulation to consider are highly dependent on the average number of collisions of a particle in this gas. To measure this number, the expected distance that a random particle travels between two collisions, called the mean free path \(\lambda\), is related to the macroscopic length scale \(l_0\). The resulting quantity is called Knudsen number and is defined as: \(Kn = \frac{\lambda}{l_0}\). Using this quantity, three regimes with a very typical behaviour of the gas can be identified.

Particles which collide very often with each other usually behave like a continuum. Such behaviour occurs when the gas is in the hydrodynamic regime at \(Kn \lesssim 0.01\). In said regime, the gas-flow can be described using a small set of partial differential equations. For a very low Knudsen number, the gas is basically at a local equilibrium and can be described using the Euler equations. Up to a Knudsen number of around 0.01 the extended system of Navier-Stokes and Fourier equations (NFS) can be used to accurately model the behaviour of a gas. Beyond this boundary, for so-called rarefied gases, NFS usually lose their accurateness and reliability [90].

In contrast to this regime, for very high Knudsen numbers, the effects of collision become negligible, and the particles can be considered as being in free-flight. Additionally, for \(Kn \gtrsim 10\), the number of collisions is so low that the Boltzmann equation in this regime can either be solved directly or by using the direct simulation Monte Carlo (DSMC) method introduced by Bird [11].

The regime in between those two (\(0.01 \lesssim Kn \lesssim 10\)) is called transition regime in which the modelling of the flow of gases is still a challenging problem in fluid dynamics [90, 94]. Under such conditions where the mean free path is comparable to the length scale, which is the case in the outer atmosphere for example, the behaviour of a gas can no longer be accurately described by the Navier-Stokes and Fourier equations, since very strong nonequilibrium effects can take place. Instead, the flow needs to be described in greater detail using the Boltzmann equation. Methods used for moderately large Knudsen numbers are often still valid in this regime but become very computationally expensive.

The modelling of gases in the transition regime has been extensively studied over the last decades [94]. One crucial first step was made in Grad [37] by introducing the method of moments to kinetic gas theory. By using a Hilbert expansion of the distribution function in Hermite polynomials, Grad developed an extended set of field equations describing the flow of the gas by modelling the evolution of an extended set of 13 variables. This closure has been extensively studied and there exist multiple extensions to this system of partial differential equations [90, 91]. Since then, the moment method has become increasingly popular because the exact velocity distribution at each point in the phase space is often

---

\(^1\)A ideal gas under standard conditions contains around \(10^{25}\) particles per cubic meter [90].
not of interest; instead, it is usually sufficient to know the macroscopic quantities of the
gas (e.g. density, velocity, temperature, and pressure) which can be calculated using
the distribution’s physical meaningful moments [90]. This method has proven to be well
suited for solving kinetic equations and especially for non-equilibrium micro-scale flows in
the transition regime [94]. Unfortunately, the set of governing equations of the moments
of the distribution of particles, which can be obtained using the Boltzmann equation, is
not closed because higher-order moments are needed to calculate the spatial flux.

Therefore, apart from Grad’s closure, various methods to close this system have been
introduced, and the behaviour of the solution highly depends on the form of closure
chosen [90, 94]. To calculate the required higher-order moments, the underlying velocity
distribution needs to be reconstructed using the given set of moments. The explicitness
of this procedure varies among the available closures. One closure that has witnessed
increased popularity is the quadrature method of moments (QMOM), which assumes
that the underlying number density function (NDF) can be represented by a weighted
combination of delta functions [28, 32, 33]. In contrast to discrete velocity methods
with fixed abscissas, this method uses a product-difference (PD) algorithm to calculate
a Gaussian quadrature of the NDF including both abscissas and weights from the given
set of moments. Within this method, no additional degree of freedom is introduced.
Consequently, the weights and abscissas are uniquely determined by the given set of
moments and the ultimate aim of the method is to construct a fitting quadrature.

Another popular closure is the maximum-entropy closure which is based on the principle
of maximum entropy and was introduced by Levermore [56]. It has been shown that
this closure yields symmetric hyperbolic moment equations and can provide accurate
descriptions of non-equilibrium gas flows [56, 64]. However, one serious problem of the
maximum-entropy closure is that its computational cost can be very high when moments
of order higher than two are used because a complex optimisation algorithm needs to be
performed to reconstruct the underlying velocity distribution.

In this thesis, said problem is overcome by reconstructing the unknown velocity distri-
bution as a sum of weighted delta functions as it is done in the QMOM. The proposed
method first uses a PD-algorithm to obtain a set of abscissas as well as weights and then
introduces a degree of freedom by inserting additional abscissas. After that, new weights
fitting the given set of moments need to be calculated. In contrast to the QMOM, by in-
roducing additional abscissas, these weights are no longer unique. Following the principle
of maximum entropy, the set of weights maximising the entropy while simultaneously fit-
ting the given moments is chosen. To solve the resulting constrained convex optimisation
problem, different optimisation algorithms will be used. The presented closure, which we
name “Entropic Quadrature” (EQ), can either be seen as an extension to the QMOM or
as a new way to efficiently tackle the maximum entropy problem.

To make use of the improved speed of the proposed closure a Runge-Kutta discontinuous
Galerkin (RKDG) method is used to discretise the equation in space and time, since this
method is stable, high-order accurate, hp-adaptive, and can easily handle complicated
geometries.

### 1.1. Organisation of the thesis

The remainder of this thesis is organised as follows: First, in chapter 2, the essentials of
the kinetic theory of gases are reviewed. After that, the moment method enabling us to
simplify the Boltzmann equation introduced in chapter 2 will be presented in chapter 3.
To solve the resulting system of governing partial differential equations, the problem of
the unclosed spatial flux presented in chapter 3 needs to be overcome. Therefore, closing this equation requires the usage of a closure theory. In chapter 4, different closures will be presented and discussed. The focus is going to be on the maximum-entropy closure and the QMOM because we will combine them in chapter 5 into the entropic quadrature closure. Being the main original contribution of this thesis, this method will be extensively motivated and described in some computational detail. Then, we will present the numerical methods used in chapter 6. First, the very intuitive finite volume scheme will be presented, and its fundamental properties concerning stability and precision will be discussed. After that, the higher order accurate discontinuous Galerkin method will be described. To complete this chapter, the Runge-Kutta time-stepping scheme will be presented. In the final chapter 7 we will test the capability of the proposed closure to accurately model flows from different regimes and discuss its relationship to the maximum-entropy closure.

2. Kinetic theory

In the following, the kinetic theory of a classical monoatomic ideal gas composed of identical particles is considered (see [90] for a more detailed introduction). As mentioned in the previous section, to exactly describe the state of such a gas on a micro level, one needs to describe the physical state of every particle. Otherwise, one is unable to, for example, capture the collisions between two particles correctly. A particle is fully described by its spatial location \( x \in \Omega_x \subseteq \mathbb{R}^d \) and microscopic velocity \( c \in \Omega_c \subseteq \mathbb{R}^d \). The resulting space spanned by \( x \) and \( c \) is called phase space. To accurately describe the flow of a gas by considering each particle, a set of coupled equations which lie in the phase space and contain at least one equation per particle needs to be solved [90]. Since this theoretical approach is not feasible, in kinetic theory, the micro-state of a gas is usually described by the distribution of particles

\[
f : \Omega_x \times \Omega_c \times \Omega_t \rightarrow \mathbb{R}_{\geq 0}, (x, c, t) \mapsto f(x, c, t),
\]

where \( x \) is again the spatial location, \( c \) the microscopic velocity and \( t \in \Omega_t \subseteq \mathbb{R}_{\geq 0} \) the time. The resulting distribution \( f(x, c, t) \) lies in the phase space and is defined such that by integrating:

\[
\int_{x_1}^{x_2} \int_{y_1}^{y_2} \int_{z_1}^{z_2} \int_{c_{x_1}}^{c_{x_2}} \int_{c_{y_1}}^{c_{y_2}} \int_{c_{z_1}}^{c_{z_2}} f(x, c, t) dc \ dx
\]

the number of particles at time \( t \) with a microscopic velocity lying in the cube defined by \( c_{x_1} \leq c_x \leq c_{x_2} \), \( c_{y_1} \leq c_y \leq c_{y_2} \), and \( c_{z_1} \leq c_z \leq c_{z_2} \) and a spatial position lying in the cube defined by \( x_1 \leq x \leq x_2 \), \( y_1 \leq y \leq y_2 \), and \( z_1 \leq z \leq z_2 \) can be obtained. Consequently, \( f \) describes the phase density of a gas and by integrating over the whole velocity space the number density of the gas at each point in time and space can be obtained:

\[
n(x, t) = \int_{\Omega_c} f(x, c, t) dc.
\]

Since it will often be necessary to integrate over the whole velocity domain, let us define:

\[
\langle f \rangle(x, t) := \int_{\Omega_c} f(x, c, t) dc.
\]

Hence, it is possible to define the probability density function of the velocity of all particles at position \( x \) at time \( t \) as:

\[
\tilde{f}(x, c, t) := \frac{f(x, c, t)}{n(x, t)}.
\]
Multiplying $f$ with a velocity depended weight, different macroscopic quantities of the gas can be calculated. Let $m$ be the mass of one particle, $k_B$ the Boltzmann constant, $\mathbf{v}$ the average velocity, and $\mathbf{C} = \mathbf{c} - \mathbf{v}$ the central or peculiar velocity of the gas. Then, the mass density $\rho$, the mean velocity $\mathbf{v}$, the pressure $p$, the temperature $T$ and the temperature in energy units $\theta$ can be related to $f$:

$$
\rho = m \langle f \rangle, \quad \mathbf{v}(x,t) = \frac{1}{n} \langle \mathbf{c} f \rangle, \quad \theta = \frac{1}{d c n} \langle \mathbf{C}^2 f \rangle \quad \text{and} \quad T = \frac{m}{d c n k_B} \langle \mathbf{C}^2 f \rangle.
$$

In order to simplify calculations, in the following chapters, it will be assumed that $m = 1$.

Additional physical meaningful moments of $f$ are the heat flux and stress tensor which are given by:

$$
p_{ij} = m \langle C_i C_j f \rangle, \quad q_i = \frac{m}{2} \langle C_i^2 f \rangle.
$$

Considering the evolution of ideal monoatomic gases, the temporal evolution of $f$ is governed by the Boltzmann equation:

$$
\frac{\partial f}{\partial t} + c_i \frac{\partial f}{\partial x_i} + \frac{g_i}{m} \frac{\partial f}{\partial c_i} = S(f, f)
$$

where $g_i$ represents acceleration due to any external forces, for example gravity or electromagnetic fields, and the collision integral $S(f, f)$ describes the change of the phase density due to interaction between particles.

Assuming that only collisions of two identical particles take place, it is possible to calculate $S(f, f)$. Boltzmann proved that the following expression mimics the effect of such binary collisions successfully:

$$
S(f, f) = \int_{\Omega_c} \int_0^{2\pi} \int_0^{\pi} \left( f(x, c', t) f(x, c'_1, t) - f(x, c, t) f(x, c_1, t) \right) g \sigma \sin(\theta) d\theta d\epsilon d c_1,
$$

where $\mathbf{c}$ and $\mathbf{c}_1$ denote the velocity of two particles before a collision with collision angle $\theta$, while $\mathbf{c}'$ and $\mathbf{c}'_1$ denote the respective velocities after the collision. Additionally, $g$ denotes the absolute pre-collisional relative velocity between both particles, $\epsilon$ the orientation of the collision plane, and $\sigma$ the differential cross section taking into account the physical properties of the particles.

Adding no fundamental difficulty to the problem of solving the Boltzmann equation any force due to external sources will be omitted in the rest of this thesis. The main challenge of kinetic theory is to solve (2.1) subjected to some initial constraints $f(x, c, 0) = f(x, c)$ on $f$ and on the boundaries of the domain. In general, solving (2.1) is not analytically possible because $S(f, f)$ is a high dimensional integral and very computationally expensive. Since the Boltzmann equation is a non-linear integro-differential equation, numerical solutions for this equation are needed and as a first step the collision operator needs to be simplified.

Bhatnagar et al. [10] introduced a very popular simplified collision model. The reduced BGK operator, which satisfies the most important properties of the Boltzmann collision operator, is given by:

$$
S_{BGK}(f, f) = - \frac{f(x, c, t) - f_M(x, c, t)}{\tau},
$$

where $f_M$ denotes the Maxwellian function.

---

2For a detailed derivation of the Boltzmann equation see Struchtrup [90], for example.

3These properties are the conservation of mass, momentum, and energy, the always positive production of entropy by the collision operator, and the fact that the collision term is zero in case the distribution function is the Maxwellian function.
where \( \tau \) denotes the collision frequency and \( f_M(x, c, t) \) the local equilibrium Maxwell distribution:

\[
f_M(x, c, t) = f_M(x, c; \rho(x, t), v(x, t), \theta(x, t)) = \frac{\rho(x, t)}{m\sqrt{2\pi\theta(x, t)}} \exp\left[-\frac{(c - v(x, t))^2}{2\theta(x, t)}\right].
\]

Apart from its important role in the BGK operator, the Maxwellian distribution also plays a crucial role in fluid dynamics in general. It can be proven that in any situation a system is always driven towards a local equilibrium by inter-particle collisions. Once the system has reached this distribution, no particle collision will take place and the system will be locally in a thermodynamic equilibrium. Additionally, it can be proven that every system is driven to a global equilibrium which is a local equilibrium with a uniform mass density, average velocity and temperature by the interplay between the local velocity distribution. Said distribution is a stationary solution of the Boltzmann equation and, without any external forces, the gas will not leave this global thermodynamic equilibrium [90].

To standardise computations, the non-dimensional version of the Boltzmann equation is considered in the remainder of the thesis. In the non-dimensional version of the Boltzmann equation, we divide every quantity by a scale which depends on the considered problem and the domain of interest. Depending on the application domain, the different reference scales can be related to each other. Since we are interested in the effects of spatial transport, the reference scales can be related as follows: First, a macroscopic length scale \( l_0 \), time scale \( t_0 \), reference number density \( n_0 \), and reference temperature \( \theta_0 \) needs to be chosen. This selection should be made as to ensure that the resulting quantities are consistent with the spatial domain and timespan of interest and the chosen initial distribution. Based on this, the characteristic velocity \( c_0 \) can be defined as: \( c_0 = \frac{v_0}{n_0} \). The resulting dimensionless quantities involved in the Boltzmann equation are:

\[
\hat{t} = \frac{t}{t_0}, \quad \hat{x} = \frac{x}{l_0}, \quad \hat{\theta} = \frac{\theta}{\theta_0}, \quad \hat{c}_i = \frac{c_i}{c_0}, \quad \hat{v}_i = \frac{v_i}{c_0}, \quad \hat{n} = \frac{n}{n_0}, \quad \hat{f} = \frac{c_0}{n_0} f.
\]

In the remainder of this thesis, only non-dimensional quantities will be used, and to simplify readability, all heads will be dropped. Inserting the non-dimensional quantities into the BGK-Boltzmann equation, a set of governing equations for the evolution of the non-dimensional version of the distribution function can be obtained [90]:

\[
\frac{\partial f}{\partial \hat{t}} + c_i \frac{\partial f}{\partial \hat{x}_i} + \frac{g_i}{m} \frac{\partial f}{\partial \hat{c}_i} = -\frac{f - f_M}{Kn}.
\]

In the following, we will refer to this non-dimensional version of the BGK-Boltzmann equation as the Boltzmann equation.

### 3. Moment equations

Despite the previous simplifications, it is not possible to solve the Boltzmann equation directly because the BGK-collision term is still a high dimensional integral. Fortunately, the exact solution is often not of interest; instead, knowing a relatively small set of its velocity moments is often sufficient [37, 90]. Using this set, the macroscopic quantities mentioned in section 2 can be recovered. Those quantities have a real physical meaning and provide important information about the gas. In many situations, a set of moments

---

\[\text{4}\text{see 4.4.4 for a sketch of a proof.}\]
up to the fourth order is sufficient to describe the state of a gas quite accurately [90]. Another argument for considering the evolution of moments instead of the evolution of the distribution of particles is that using this method any desired level of accuracy can be achieved because a distribution is fully defined by all of its moments. Consequently, using the method of moments a trade-off between speed and accuracy can be chosen. Additionally, being able to calculate the evolution of a flexible set of moments is very desirable because such a method can face the different requirements of different regimes. Thereby, the method of moments provides a systematic framework for the derivation of approximative models of the Boltzmann equation.

Following this reasoning, we would like to derive governing equations for a chosen set of moments of the distribution of particles. Let \( \Phi \subset \mathbb{R}[c]^N \) be a set of polynomial basis functions in the velocity domain \( \Omega_c \) and

\[
\mathbf{u}(x,t) := \langle \Phi f(x,c,t) \rangle, \quad \mathbf{F}_i(\mathbf{u}(x,t)) := \langle c_i \Phi f(x,c,t) \rangle, \quad \mathbf{E}(\mathbf{u}(x,t)) := \langle \Phi f_M(x,c,t) \rangle \in \mathbb{R}^N.
\]

Using this notation, after multiplying the Boltzmann equation with the set of polynomial basis functions \( \Phi \) and subsequent integration over \( \Omega_c \), a set of governing equations for the chosen set of moments can be obtained [90]:

\[
\frac{\partial \mathbf{u}(x,t)}{\partial t} + \frac{\partial \mathbf{F}_i(\mathbf{u}(x,t))}{\partial x_i} = \frac{\mathbf{u}(x,t) - \mathbf{E}(\mathbf{u}(x,t))}{Kn}, \quad (3.1)
\]

By projecting (2.2) onto the subspace spanned by \( \Phi \) a system of \( N \) coupled governing equations for the considered macroscopic quantises is derived. Moreover, the dimension of (2.2) has been reduced. (3.1) does no longer lie in the \( d_c + d_x \) dimensional phase space; instead, the dimension has been reduced to \( d_x \).

For reasons that will become apparent later, we note that (3.1) is in the form:

\[
\partial_t \mathbf{u}(x,t) + \partial_x_i \mathbf{F}_i(\mathbf{u}(x,t)) = \mathbf{P}(\mathbf{u}(x,t)). \quad (3.2)
\]

In the remainder of this thesis, we are mainly interested in full sets of monomial basis functions because they have various desirable properties such as symmetry and Galilean invariance. Let us therefore define a concise notion of the involved quantities. Firstly, let \( \Phi_{x,y,z}^i \) denote the set of monomial basis functions of \( \mathbb{R}[c_x, c_y, c_z] \) up to order \( i \), for example, \( \Phi_{x,y}^3 = (1, c_x, c_y, c_x^2, c_x c_y, c_y^2, c_x^3, c_x^2 c_y, c_x c_y^2, c_y^3)^T \). Secondly, let us define a concise notion of a particular moment of the particle distribution function:

\[
u_{\lambda_{i,j,k}}(x,t) := \langle c_x^i c_y^j c_z^k f(x,c,t) \rangle,
\]

where \( \lambda := i + j + k \) denotes the order of the moment. Analogously, a notation for a moment of a local Maxwellian distribution is introduced:

\[
\mathcal{E}_{\lambda_{i,j,k}}(\mathbf{u}(x,t)) := \langle c_x^i c_y^j c_z^k f(x,c,t) \rangle.
\]

For the sake of readability, we will drop the functional dependencies of \( u_{\lambda_{i,j,k}}(x,t) \) and \( \mathcal{E}_{\lambda_{i,j,k}}(\mathbf{u}(x,t)) \) in the main part of the thesis. However, it is important to keep them in mind.

\[\text{At lower Knudsen numbers, for example, the knowledge of a small number of moments is sufficient to describe the behaviour of the gas accurately because due to the high influence of the collision term the system is drawn very quickly to a local equilibrium. After that, the gas is fully described by the mass, momentum and energy which are governed by the Euler equations.}\]
and they will be included from time to time. Using the introduced notation, (3.1) can be rewritten to:

$$\frac{\partial u^\lambda_{i,j,k}}{\partial t} + \frac{\partial u^\lambda_{i+1,j,k}}{\partial x} + \frac{\partial u^\lambda_{i,j+1,k}}{\partial y} + \frac{\partial u^\lambda_{i,j,k+1}}{\partial z} = -\frac{u^\lambda_{i,j,k}}{Kn}, \forall c_i^i, c_j^j, c_k^k \in \Phi$$  \hspace{1cm} (3.3)

(3.3) forms a system of coupled partial differential equations governing the evolution of the moments of the distribution function. However, due to the need for higher order moments in the spatial transport terms of the equation, this system is only closed when infinitely many moments are considered. In the case that the cardinality of \( \Phi \) is finite, some moments of higher order which are not contained in \( \mathbf{u} \) will be needed to calculate the spatial derivatives. As a result, the system of partial differential equations will become unclosed. The number of missing moments depends on the spatial dimension and the chosen set \( \Phi \). For instance, in one spatial dimension using monomial basis up to a certain order only one moment needs to be reconstructed to close the equation. Generally speaking, for \( \Phi_{x,y,z} \), to close (3.3) all moments of order \( i+1 \) are needed. The reconstruction of the missing moments is a challenging and non-trivial task. To reconstruct them, several closures have been introduced trying to overcome this problem. In Chapter 4, several of the already proposed closures will be discussed and in Chapter 5 a new closure will be introduced.

### 3.1. Central and conserved moments

There are two different standard ways to express the moments of the distribution function of particles. The first one, which is called the conservative form of moments, has already been introduced in the section above. The second one is called central form of moments and relies on the fact that the velocity of each particle \( c \) can be decomposed into two parts: \( c_i = v_i + C_i \), where \( C_i \) is the peculiar velocity which can be understood as the velocity of the particle observed from a point moving at the average speed of the particles. Using this, a centralised version of moments can be defined as:

$$w^\lambda_{i,j,k}(x,t) := \langle C_1^i C_2^j C_3^k f(x,c,t) \rangle,$$

because, as argued in [90], integrating over \( dc \) instead of \( dC \) makes no difference at all.

Those moments are called primitive or centralised moments and are relevant for two reasons. First of all, some of the most important macroscopic quantities with a meaningful physical interpretation, for instance the temperature, contain centralised moments. Fortunately, it is possible to convert the primitive form of moments into the conservative form and vice versa using the following formula:

$$w^\lambda_{i,j,k}(x,t) = \sum_{l \leq i, m \leq j, n \leq k} (-1)^{l+m+n} \binom{i}{l} \binom{j}{m} \binom{k}{n} u^\lambda_{i-l,j-m,k-n} v_1^l v_2^m v_3^n \hspace{1cm} (3.4)$$

with \( \mu = l + m + n \). This formula can be derived using the binomial theorem and the relationship \( w^\lambda_{i,j,k}(x,t) = \langle (c_1 - v_1)^i (c_2 - v_2)^j (c_3 - v_3)^k f(x,c,t) \rangle \). Since we will consider the method of conserved moments, and the results as well as most of the initial conditions are usually given using macroscopic quantities, it is important to note that (3.4) can be used to recover \( \mathbf{u} \) from the lower order macroscopic quantities.

Secondly, to make the algorithm calculate the moments needed to close (3.3) in a more reliable and robust manner, it is always beneficial to standardise the input as much as possible. To do so, the set of moments can be divided by \( u^0_{0,0,0} \) and subsequently
centralised using (3.4). After that, the zero-order moment is always one and the first-order moments are always zero. As an additional step, the system of moments can be rotated leading to a rotated version of the distribution function with a diagonalised covariance matrix. See [32] for the corresponding formulas.

3.2. The method of moments and different governing equations

It is worth noticing that several important systems of equations are entailed in the hierarchy of approximations provided by the method of moments. For example, for every $\Phi^i_{x,y,z}$ with $i \geq 1$ the laws of conservation of mass and momentum are included in the resulting system of equations. In two dimensions, for example, the first three equations of said system are:

$$
\frac{\partial u^{0}_{0,0}}{\partial t} + \frac{\partial u^{1}_{1,0}}{\partial x} + \frac{\partial u^{2}_{1,1}}{\partial y} = 0
$$

$$
\frac{\partial u^{1}_{1,0}}{\partial t} + \frac{\partial u^{2}_{2,0}}{\partial x} + \frac{\partial u^{2}_{2,1}}{\partial y} = 0
$$

$$
\frac{\partial u^{0}_{1,1}}{\partial t} + \frac{\partial u^{1}_{1,1}}{\partial x} + \frac{\partial u^{2}_{2,2}}{\partial y} = 0.
$$

The right hand side is always zero because the BGK-operator satisfies the conservation of mass and momentum. This holds due to the fact that the Maxwellian distribution is collision invariant. Using (3.4) these equations can be rewritten to the conservation of mass and momentum:

$$
\frac{\partial \rho}{\partial t} + \frac{\partial \rho v_x}{\partial x} + \frac{\partial \rho v_y}{\partial y} = 0
$$

$$
\frac{\partial \rho v_x}{\partial t} + \frac{\partial \rho v_x^2 + p}{\partial x} + \frac{\partial \rho v_x v_y + p}{\partial y} = 0
$$

$$
\frac{\partial \rho v_y}{\partial t} + \frac{\partial \rho v_x v_y + p}{\partial x} + \frac{\partial \rho v_y^2 + p}{\partial y} = 0.
$$

Additionally, by including $C^2$ into $\Phi$, the conservation law of energy is included in the resulting set of governing equations. The resulting system of four equations is well known as the Euler equations. Consequently, the Euler equations are obviously a low-order part of the hierarchy of approximations of the Boltzmann equation offered by the method of moments. By not only considering monomial basis functions it is also possible to add higher order macroscopic quantities such as the heat flux and stress tensor to $\mathbf{u}$ and therefore also to include governing equations for these higher order macroscopic quantities. Adding such higher order moments to the set of moments considered, it is, in principle, possible to accurately capture effects occurring in the transition regime and when general non-equilibrium flows occur. A set of governing equations for these higher order moments is usually able to model the flow of a gas in the transition regime accurately. Unfortunately, as mentioned earlier, when such higher order moments are considered, the resulting system of governing equations can no longer be closed using some well known relations of lower order moments such as the equation of state. As a result, it is unclear if an approximation of the higher order moments needed can guarantee that the evolution of the moments remains accurate. Fortunately, one promising closure was proposed by Grad [37]. He derived a set of closed governing equations for the density, average velocity, heat flux and stress tensor. Struchtrup and Torrilhon [91] used the same set of basis functions as a starting point to derive an improved system of governing equations called R13. As argued
in Torrilhon [94], also those methods relying on an approximation of the higher order moments needed are able to model non-equilibrium flows for different Knudsen numbers below 1.

Having argued that some powerful systems of governing equations are part of the hierarchy of approximations, offered by the method of moments, supports our decision to consider this method as a flexible framework to approximate the solution of the Boltzmann equation.

3.3. Realisability of moments

In this section, some general remarks on a vector $\mathbf{u}$ of monomial moments of a distribution function are made. Such insights are essential for judging possible closures of (3.3), as they should be well defined for all such sets of moments.

A vector $\mathbf{u}$ is called realisable if there exists a corresponding distribution function. We will start our explanation with the one dimensional case. The set of all realisable sets of moments up to order $n$ is defined as:

$$\mathcal{R}^n = \{ \mathbf{u} \in \mathbb{R}^{n+1} \mid \int_{\mathbb{R}} x^i g(x) dx = u_i, \forall i \in [0, n] \},$$

where $g(x)$ is a non-negative distribution function. Hamburger [39] proved that $\mathbf{u} \in \mathcal{R}^n$ if and only if the Hankel matrix $(H_u)_{ij} = u_{i+j}, i+j \leq n$ is positive-definite. In the case that the number of moments is even, $(H_u)_{ij} = u_{i_1+i_2+j}, 1 + i + j \leq n$ needs to have a positive determinant as well. For more details see Shohat and Tamarkin [86] and Levermore et al. [57]. A necessary and sufficient condition for positive-definiteness is the Sylvester criterion. Said criterion states that the determinants of all upper-left submatrices have to be positive to ensure that the matrix is positive-definite. As a result, a set of moments $\mathbf{u}$ is realisable if and only if the determinants of all upper left submatrices of $H_u$ are positive.\(^6\) For example, for $n = 2$, the resulting conditions are: $u_0 > 0$ and $u_2 > u_2^2/u_0$. The extension to two dimensions is somehow natural. In two dimensions the set of realisable moments up to n-th order is defined as:

$$\mathcal{R}^n = \{ \mathbf{u} \in \mathbb{R}^{0.5(n+1)(n+2)} \mid \int_{\mathbb{R}} \int_{\mathbb{R}} \Phi_{x,y}^n g(x, y) dxdy = \mathbf{u} \},$$

where $g(x, y)$ is a non-negative two dimensional distribution function. In two dimensions, the criterion for realisability generalises to: A set of monomial moments is realisable if and only if the matrix $(\Phi_{x,y}^{\lfloor n/2 \rfloor} (\Phi_{x,y}^{\lfloor n/2 \rfloor})^T g(x, y))$ is positive-definite. In the case that $n$ is odd, the matrix $(\Phi_{x,y}^{\lfloor n/2 \rfloor+1} (\Phi_{x,y}^{\lfloor n/2 \rfloor})^T g(x, y))$ needs to have a positive determinate, too.

4. The closure problem

To close the Boltzmann equation formulated in terms of the velocity moments of the distribution of particles (3.1), higher order moments need to be reconstructed. In the past, many closures have been proposed using a variety of different techniques to calculate these higher order moments. Usually, using the given set of moments a reconstruction $f^*(\mathbf{c})$ of the unknown underlying distribution function is obtained which fits the given set of moments ($\langle \mathbf{f} \rangle = \mathbf{u}$) while being optimal concerning the chosen closure. $f^*$ can

\(^6\)In fact, it is sufficient that the determinants are non-negative if one allows for functions with a finite support.
then be used to calculate the higher order moments missing \( F_i = \langle c_i \Phi f^* \rangle = u \). In the following chapter, the general problem of moments and possible solution strategies will be presented. After that, some closures for the problem of moments in the context of solving the Boltzmann equation will be introduced and analysed. I will extensively deal with the maximum-entropy closure and the QMOM as they are both closely related to the method of EQ, which will be proposed in the following chapter. EQ uses the same justification as the maximum-entropy closure, while using the underlying reconstruction technique of the QMOM.

### 4.1. Problem of moments

The problem of finding a positive function \( g : \mathbb{R} \mapsto \mathbb{R} \) with a given sequence of numbers \( \rho_0, ..., \rho_N \) as moments \( \int_{\Omega} x^i g(x) dx = \rho_i, \ i = 0, ..., N \) is known as the problem of moments and has been extensively studied independent from its application in solving the Boltzmann equation [39, 49, 67, 86, 89]. Different choices of \( \Omega \) result in different properties of the problem. Originally Stieltjes [89] introduced the so-called Stieltjes moment problems considering \( \Omega = [0, \infty) \). In the following, the problem was extended by Hamburger [39] to the more general case with \( \Omega = \mathbb{R} \), which is known as the Hamburger moment problem. The problem of moments is not only studied because of its inherent mathematical interest but also because it is faced in many different areas (e.g. physics, economics, and chemical engineering) [47].

Unfortunately, there usually exist infinitely many positive density functions fitting the given set of moments provided that the given set of moments is realisable. Consequently, the system is underdetermined and choosing one hopefully plausible approximation of the original underlying function requires some criterion. Three different philosophies to reconstruct a distribution function from its moments are usually considered.

One possibility is that a particular form of the underlying distribution function is assumed. Thereby, the reconstructed \( f^* \) is restricted to being in the form of the chosen model distribution, for instance in the form of a sum of multiple Gaussian distributions. In this approach, the model of the distribution usually depends on \( N \) coefficients which can be determined using the known moments \( \rho_0, ..., \rho_N \) to construct a function fitting this set. Introducing a higher number of coefficients, it is quite natural for this type of closure that a higher number of moments known improves the accuracy of the solution, since a wider variety of distribution functions can be captured. Naturally, the properties of closures of this type depend crucially on the chosen model distribution.

Another possibility is that an optimisation criterion is used. In this case, the solution of the problem of moments is the function \( g(x) \) maximising the chosen criterion while fitting the given set of moments. The most famous closure following this line of reasoning is the maximum-entropy closure where the function \( g(x) \) with the highest entropy is chosen. One can argue that the function with the highest entropy should be chosen because from the perspective of information theory this distribution is the most probable one in the absence of any additional information [43, 49, 67]. Despite looking unrelated at the first glance, the two mentioned approaches are often closely related.

Considering a quadrature-based reconstruction of the unknown distribution function is a third method which can be used to solve the problem of moments. To calculate this reconstruction, the problem of moments can be reformulated in terms of a Gaussian quadrature as: Given the moments \( \rho_0, ..., \rho_N \) of an unknown probability density \( g(x) \) find a sequence of pairs \( (\gamma_1, w_1), (\gamma_2, w_2), ..., (\gamma_n, w_n) \) of abscissas \( \gamma_i \) and weights \( w_i \) such that \( \sum_{i=1}^{n} w_i \gamma_i^k = \rho_k, \forall k = 0, ..., N \). This problem can be solved using a product-difference
algorithm [73, 103]. Using this, the next higher order moments can be calculated using: \( \rho_{N+1} = \sum_{i=1}^{n} w_i \gamma_i^{N+1} \). It is possible to relate this approach to the popular Pade approximation which was frequently used to solve the problem of moments [103] [67] [7].

Aiming to reconstruct a probability density function in several variables, it is also possible to extend the problem of moments to multiple dimensions. In fact, the multi-dimensional problem is often the one of practical interest. Fortunately, the presented approaches can also be extended to the multi-dimensional case. Moreover, the introduced philosophies are also applicable to close (3.1), and in the context of fluid dynamics, a whole theory around such closures has been established. In the following, exemplifications of all three different approaches in the context of closing the Boltzmann equation will be presented. Nevertheless, they are also applicable to solving the Hamburger problem of moments.

4.2. Desirable properties

To judge the quality of a closure of (3.1) and the resulting system of governing equations, practical and theoretical considerations need to be taken into account. In the following, the most important desirable properties of a closure are presented:

- **non-negativity**: The reconstructed function \( f^* \) should be physical realisable. This implies that the distribution has to be non-negative everywhere on \( \Omega \), because the number of particles within a subset of the phase space cannot fall below zero. Furthermore, with a partly negative reconstructed distribution function \( f^* \), a negative density can be produced.

- **realisability**: The closure should be able to calculate a reconstruction \( f^* \) from all sets of realisable moments \( \textbf{u} \in \mathcal{R} \). Furthermore, it is desirable that the reconstruction \( f^* \) always fits all moments given.

- **flexibility**: Following the philosophy that we want to construct a flexible framework to approximate the Boltzmann equation to any desired degree of precision, it is of great interest that the closure can be used for diverse \( \Phi \) having different cardinalities.

- **Galilean invariance**: Out of physical considerations, the space spanned by the chosen \( \Phi \) should be Galilean invariant because otherwise non-physical behaviour will occur, which, for strong violations, will govern the system [68] [96]. Whether or not this property is fulfilled depends on the chosen set of basis functions \( \Phi \) and not on the closure technique itself. For example, for \( \Phi_{x,y,z} \) this property is always fulfilled [96]. To guarantee the conservation of mass, momentum and energy, another important constraint that should be placed on \( \Phi \) is that \( 1, \textbf{c}, \) and \( |\textbf{c}|^2 \) should lie in the span of \( \Phi \). Since the conservation laws are always included, this property of \( \Phi \) will be assumed as fulfilled in the following.

- **hyperbolicity**: One necessary condition for a robust and stable simulation is that the closed system should be hyperbolic. A moment system is called hyperbolic in \( \mathcal{D} \) if and only if for all \( \textbf{u} \) the matrix \( J(\textbf{u}) = \frac{\partial F(\textbf{u})}{\partial \textbf{u}} \) is diagonalisable to \( J = V \Lambda V^{-1} \) with real eigenvalues. To understand the implications of hyperbolicity, we will expand a little bit on this topic in the following. First, note that one important quantity in computational fluid dynamics associated with \( J(\textbf{u}) \) is the characteristic speed of an equation which is defined as the maximal absolute eigenvalue \( \lambda \) of \( J(\textbf{u}) \). \( \lambda \) quantifies the maximal speed at which information travels. For example, in the simple
advection equation $u_t + au_x = 0$ the characteristic speed and the eigenvalue of $J$ are $a$.

The connection between the eigenvalues of $J(u)$ and the speed at which information travels through the system is also the reason why hyperbolicity is crucial. Hyperbolicity ensures that the speed at which any information travels through the system is always finite which is necessary to prevent a breakdown of the simulation. To get a better understanding of the eigenvalues of $J$, we follow an argument presented in Schärer [82]: Let us first write the conservative form of (3.2) in a quasi-linear form: $\partial_t u + J(u)\partial_x u = 0$. Furthermore, consider a linearised version of $u$: $u = u_0 + \epsilon u_1$ where $u_0$ describes a homogeneous equilibrium state and $u_1$ some sort of perturbation. Since $J_0$ is always the same, we can rewrite an approximation of the quasi-linear form as:

$$\partial_t u_1 + J_0 \partial_x u_1 = 0 \iff \partial_t w_1 + \Lambda_0 \partial_x w_1 = 0$$

with $w_1 = V_0^{-1}u_1$ and $J_0 = \frac{\partial F}{\partial u}(u_0)$. The constant eigenvalues $\Lambda$ of $J_0$ can be understood as the propagation velocities of the different components of the system. Consequently, the maximal absolute eigenvalue is the maximal speed at which any information in the system travels (characteristic speed).

Unfortunately, proving hyperbolicity is very difficult because $J$ needs to be known. For non-linear equations, this can sometimes be achieved through rewriting the system in the quasi-linear form. For the Euler equations, for instance, we can write $J$ explicitly in terms of the elements of $u$. Consequently, the eigenvalues and, especially, the characteristic speed of the equation depend on the current solution. Since $F(u)$ in the Boltzmann equation (3.1) depends on the chosen closure, $J(u)$ also does. Consequently, the hyperbolicity is an inherent property of the chosen closure. For explicit closures it is often possible to write them in a quasi-linear form and to check for hyperbolicity and $\lambda$ directly (see [79] for an extensive example). For closures involving non-linear processes such as EQ and QMOM, however, this is not possible, and we have to use numerical methods to approximate $J(u)$ for specific values of $u$.

- **capability**: The closure should be able to provide a robust simulation of equilibrium and non-equilibrium flows in the transition regime with reasonable precision.

### 4.3. Grad’s closure

In the following, the system of governing equations introduced by Grad [37] will be briefly presented as one of the most common closure of (3.1). By assuming a model distribution of $f^*$, Grad’s closure is an exemplification of the first philosophy presented. Here, a Hermite expansion of the local equilibrium distribution serves as a model of the unknown distribution function:

$$f^{(Grad)}(x, c, t) = f_M(x, c, t; \rho, v, \theta) \sum_{n=0}^{N-1} \alpha_n(x, t) He_n(c),$$

where $He_n$ denotes the Hermite polynomial of order $n$ and $\alpha(x, t)$ a set of coefficients determined by the given set of moments $u$ in a way that the given moments are fitted. One significant advantage of Grad’s closure is that the coefficients $\alpha(x, t)$ can be calculated directly from $u$. Hence, it is possible to write the higher order moments needed to
close (3.1) as an explicit function of $u$ and it is not necessary to perform any additional calculations to reconstruct the distribution of particles.

Furthermore, Grad’s closure is quite flexible and not restricted to a particular set of moments thereby fitting in the hierarchy of approximation of the Boltzmann equation offered by the method of moments [90]. Initially, in [37] a set of governing equations for the 13-moment $u = \{\rho, v_i, p_{ij}, q_i\}$ and for the 20-moment case with the full heat flux tensor instead of only the heat-flux vector was introduced.

Unfortunately, this closure has the undesirable property that the reconstructed distribution function $f^{(\text{Grad})}$ can become negative for some values in $\Omega \times \Omega_c$ and therefore non-physical. Additionally, despite being locally hyperbolic, the proposed system of partial differential equations may lose hyperbolicity even for moderate derivations from equilibrium as shown in [14, 92, 93]. Despite being computationally very efficient, Grad’s closure has many undesirable properties and is therefore only capable of simulating flows from a very restricted class of problems correctly. Consequently, Grad’s closure is generally not suitable to accurately describe micro-scale flows.

4.4. Maximum-entropy closure

One of the most important methods to solve the Hamburger moment equation is the maximum-entropy closure. In the following section, first, a justification of the maximum-entropy closure for said problem will be presented. Subsequently, the role of entropy in statistical mechanics and the closely connected maximum-entropy closure for the Boltzmann equation will be presented and justified. The following trenchant quote of Claude Shannon dealing with the question of how to name his starting point in information theory can provide guidance to follow the course of this section:

My greatest concern was what to call it. I thought of calling it ‘information,’ but the word was overly used, so I decided to call it ‘uncertainty.’ When I discussed it with John von Neumann, he had a better idea. Von Neumann told me, ‘You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, no one really knows what entropy really is, so in a debate you will always have the advantage.’

4.4.1. Some general notes on information entropy

Independently from its application in statistical mechanics, the concept of entropy has been introduced in the field of information theory by Shannon [85]. In our presentation of the concept of information entropy, we start by introducing a finite random variable $X$ with different values $X_1, X_2, ..., X_n$ and corresponding probabilities $p_1, p_2, ..., p_n$. Shannon [85] defined the information entropy of this random variable as:

$$H(p_1, ..., p_n) = -K \sum_i p_i \ln p_i$$ (4.1)

for some arbitrary positive constant $K$. This definition can be somehow naturally extended to the definition of the entropy of a continuous probability density function $g(x)$ a

\footnote{Please note that we will adapt the convention to set $\log 0 := 0$ following the values of the limit at this point.}
Taking a step back, one general interpretation of information entropy is that it can be understood as the average information gained by observing an outcome of X. This intuition served as Shannon’s ansatz for deriving (4.1). Shannon [85] proved that (4.1) is the only possible positive quantity satisfying all reasonable properties of such a measure of information gain. Some of his assumptions are that the measure is continuous, that a rare event covers more information than a frequent one and that a single-valued random variable has no information gain. Closely related to the understanding of information entropy as information gain, information entropy is frequently interpreted as the amount of uncertainty introduced by a probability function. For example, considering the probability distribution of some finite-valued random variable X, a uniform distribution has the highest possible information entropy and also introduces the highest amount of uncertainty concerning a possible outcome of X. Another possible interpretation of information entropy is that it is a measure for disarray. Observing a sequence of outcomes of X, we will see a high variation if the information entropy is high and a low variation if the information entropy is low.

4.4.2. Maximising entropy

Using the introduced meanings of information entropy, it is usually argued that in a situation where some, yet insufficient, information about a function, for instance, its moments, is known, the function with the highest information entropy should be chosen among all possible solutions [43, 45, 49, 67]. A possible line of reasoning justifying this claim is presented in the following [45]: In a situation with some information given and other information being uncertain, since conclusions about the distribution should only be based on the given data, and no additional information should be introduced, the function taking into account the given information while making the least possible additional assumptions should be the function of choice. Or to put it in other words, we should acknowledge our ignorance about parts of the problem and should allow for all possibilities allowed by the given data. Consequently, the function with the highest level of uncertainty concerning all missing information thereby introducing no additional assumptions should be chosen. Fortunately, we already know a quantity measuring the amount of uncertainty introduced by a probability function: the information entropy. As a result, in an underdetermined problem the function with the highest information entropy which is thereby as uniform as possible should be chosen because any other distribution introduces some arbitrary assumptions such as a model distribution which cannot be justified by the given information. Jaynes [43] summarised the justification of this approach by claiming that the maximum entropy solution “is the least biased estimate possible on the given information; i.e., it is maximally noncommittal with regard to missing information”.

Following this line of reasoning, Junk [49] and Mead and Papanicolaou [67] introduced the maximum-entropy closure to the problem of moments and proved several interesting and desirable properties of this closure. The first important property is that the maximum entropy solution to the Hamburger moment problem is always in the form: \( \exp(\sum_{i=0}^{N} \alpha_i x^i) \). From this, the essential property that the solution to the problem is always a positive function can be directly concluded. Following the reasoning that no unjustified conclusions should be drawn, another desirable property of the maximum entropy solution is that every possibility is assigned a non-zero probability.
An alternative optimisation criterion, which is used to solve the problem of moments and also tries to maximise the uniformity of the distribution as much as it is allowed by the given information, is $-\sum p_i^2$. This measure has many of the properties required by Shannon for a measure for uncertainty and often leads to the same results. However, it does not guarantee a positive distribution as the constrained maximum and is therefore computationally much harder to handle in general [43] [45].

4.4.3. Entropy in statistical mechanics and thermodynamics

In physics, the concept of entropy was first introduced by Clausius in the context of the Second Law of Thermodynamics as the thermodynamic quantity $S$. Subsequently, entropy was introduced to statistical mechanics by Boltzmann. He derived a famous formula for the macroscopic quantity of thermodynamic entropy as: $S = k_b \ln W$, where $W$ stands for the number of possible microstates giving rise to the current observable macrostate of the system which is defined by its distribution of particles. Boltzmann thereby established a useful link between the field of statistical mechanics dealing with the microstate of a gas and the field of thermodynamics dealing with macroscopic quantities.

Entropy can, in this context, be understood as the amount of uncertainty concerning the microscopic state of the gas remaining after observing its macroscopic state. From this definition of entropy, another interpretation of entropy as a measure of extraordinariness and disorder can be derived. A system with low entropy has only a limited number of microscopic counterparts and can therefore be conceived as ordered and quite extraordinary [69].

Moreover, Boltzmann’s formula for the entropy links entropy to probability theory and is, therefore, also interesting for justifying the maximum entropy principle. Assuming that all microscopic states occur with equal probability, the entropy of a macroscopic state can be related to its probability. Let $A_n$ be a macroscopic state with $W_{A_n}$ microscopic counterparts and $\Psi$ the set of all macroscopic states fitting the given information, then the probability of said system can be expressed as:

$$P(A_n) = \frac{W_{A_n}}{\sum_{A \in \Psi} W_A},$$

which is proportional to its entropy.

In the context of kinetic theory, Boltzmann has derived another notion of entropy as the so-called $H$-functional. The $H$-functional is calculated by using the macroscopic description of the gas, the distribution of particles, and can be interpreted as a measure of the total thermodynamic entropy of a system:

$$S(f) = -H(f) := - \int \int f(x, c, t) \ln f(x, c, t) dc \, dx \quad (4.3)$$

This notion of entropy can be derived as a special form of $S = k_b \ln W$, which is computationally much easier to handle. For the sake of the following explanation linking the $H$-functional to thermodynamic entropy, we will assume that the world is discrete. Consequently, the six-dimensional phase space spanned by space and velocity can be divided into $k$ non-overlapping and minimal boxes. Let $N_i$ be the number of particles in box $i$ and $f_i = \frac{N_i}{N}$ the fraction of all particles in box $i$. Assume that we consider a situation with a large number of particles $N$. Assume further that the macroscopic description of the gas, respectively, its distribution of particles is given by the corresponding relative frequencies $f_1, ..., f_k$. Since we aim to simplify Boltzmann’s famous entropy formula, it
is our goal to find the number of microstates $W_N(f)$ fitting the considered macrostate. A fitting microstate is characterised by $\frac{N_i}{N} = f_i$. Assuming that $N_1, \ldots, N_k$ is known, $W_N(f)$ can be determined using the multinomial coefficient: $W_N(f) = \frac{N!}{N_1! \cdots N_k!}$. As we are interested in systems with many particles, we let $N \to \infty$ and use Sterling’s formula to conclude: $W_N(f) \sim e^{-N \sum f_i \ln f_i}$ and subsequently $S = k \ln W_N(f) \approx \sum N_j \ln N_j$ [90]. By expanding the discrete case to the continuous one, we obtain (4.3).

### 4.4.4. H-theorem

In the famous H-theorem, using the already mentioned H-functional, Boltzmann proved that the entropy of a system is always non-decreasing over time. He even proved that the production of entropy is always strictly positive until the system has reached a local thermodynamic equilibrium. To prove this law, he first stated an alternative non-simplified version of (3.1), the equation of transfer:

$$\frac{\partial \langle \psi f \rangle}{\partial t} + \frac{\partial \langle c_i \psi f \rangle}{\partial x_i} = \langle \psi S(f, f) \rangle,$$

(4.4)

where $\psi$ is an arbitrary polynomial basis function. Setting $\psi = -\ln f$ leads to:

$$\frac{\partial \langle -f \ln f \rangle}{\partial t} + \frac{\partial \langle -c_i f \ln f \rangle}{\partial x_i} = \Sigma,$$

with

$$\Sigma = \langle \frac{1}{4} \int_{\Omega} \int_{0}^{2\pi} \int_{0}^{\frac{\pi}{2}} \ln \frac{f f_1'}{f_1 f'} (f f_1' - f f_1) g \sigma \sin \theta d\theta d\epsilon dc \rangle,$$

where $f = f(x, c, t), f' = f(x, c', t), f_1 = f(x, c_1, t)$ and $f_1' = f(x, c_1', t)$. Looking closer at $\Sigma$ leads to the insight that $\Sigma \geq 0$ since $g \geq 0, \sigma \geq 0$, and $\sin \theta \geq 0$ for $\theta \in [0, \pi/2]$ (see [90] for a detailed proof). Consequently, considering an isolated system where no flux over the surface is present, integrating over the whole space and using $\Sigma \geq 0$ and the definition from (4.3), we obtain:

$$\frac{\partial S(f)}{\partial t} \geq 0.$$

In summary, Boltzmann’s H-theorem proves that the production of entropy in an isolated system is always non-negative. Boltzmann’s H-theorem can be interpreted as an instance of the Second Law of Thermodynamics. It is, therefore, remarkable due to the fact that, in contrast to the second law, the theorem is supported by a proof. Furthermore, it is interesting to note that the production of entropy is only zero when the system is in a local equilibrium. Consequently, every system always evolves in the direction of a local equilibrium where the entropy of the system is at maximum while the conservation of mass, momentum and energy is satisfied.

### 4.4.5. Thermodynamic and information entropy

Jaynes [43] established a link between the thermodynamic and information entropy by introducing a new viewpoint on statistical mechanics as a form of statistical inference. Furthermore, he reasoned that the method of maximising the information entropy is a valuable independent method with a rigorous information theoretical justification which is also applicable to different problems in statistical mechanics [43, 44]. For example, Jaynes [43] derived the properties of a thermodynamic equilibrium from an interpretation of statistical mechanics as statistical inference using the maximum information entropy closure.
As outlined above, to conclude, there is a rigorous information theoretical justification for the usage of a maximum-entropy closure for solving the Hamburger problem of moments. Additionally, the maximum-entropy closure is especially well suited for kinetic theory because here the maximum-entropy closure is not only justified utilising information theory but also by the H-theorem and the link between probability and entropy.

4.4.6. Maximum-entropy closure in kinetic theory

Levermore [56] introduced the very popular maximum-entropy closure to the field of kinetic gas theory. The justification and basis of this closure have been extensively expounded in the previous section. In Levenmore’s maximum-entropy closure applied to close (3.1), among all possible distribution functions \( f^* \) fitting the given set of moments, the distribution function that maximises the entropy of the system is chosen. Levermore [56] used a slightly different notion of entropy known as the entropy of a classical gas:

\[
H(f) = -\langle f \log f - f \rangle.
\]

Nevertheless, this definition is equivalent to the previously presented definitions, since \( \langle f \rangle \) is given and therefore fixed. Consequently, the reconstructed distribution function \( f^{(ME)} \) is determined using:

\[
f^{(ME)} = \arg \max_f H(f) \quad \text{s.t.} \quad \langle \Phi f \rangle = u.
\]

To simplify the optimisation, the problem is converted into a different form. As shown in [56], the solution of (4.5) is always in the form of the following model distribution: \(^8\)

\[
f^{(ME)}(x, c, t) = \exp[\alpha \Phi(c)].
\]

Instead of using (4.5) to find \( f^{(ME)} \), (4.6) will be usually used. Thereby, the maximum-entropy closure, which initially is a closure of the second type, is converted into the form of a type one closure.

Unfortunately, it is not possible to obtain a closed form solution of the coefficients of (4.5) when moments of order higher than two are considered. Consequently, to obtain \( f^{(ME)} \) a constrained optimisation algorithm needs to be employed. Using (4.6) and Lagrange multipliers the set of parameters \( \alpha \) can be found using:

\[
\alpha(x, t) = \arg \min_\alpha \left( \langle \exp[\alpha \Phi(c)] \rangle - \hat{\alpha} u(x, t) \right)
\]

Apart from computational aspects, the maximum-entropy closure has various desirable properties: First of all, the reconstructed function \( f^{(ME)} \) is always in the form (4.6) and consequently satisfies the non-negativity condition. Generally speaking, one advantage of the maximum-entropy closure is that the flexibility of this approach is very high. Levermore [56] introduced the maximum-entropy closure as a hierarchy of moments systems with increasing precision.\(^9\) Nevertheless, the order of the highest polynomial in \( \Phi \) always has to be even because otherwise (4.6) will become unbounded and the moments are no longer finite. The maximum-entropy closure is especially compatible with all \( \Phi_{x,y,z}^i \) with even \( i \). Unfortunately, considering realisability, not all sets of moments in \( \mathcal{R} \) are realisable by the maximum-entropy closure. Let \( \mathcal{D} \) be the set of all \( u \) which can be realised using a maximum-entropy closure. It is interesting to note and intuitively plausible

\(^8\)The derivation is analogous to the derivation of the model distribution of the weights in EQ (see section 5.2.2).

\(^9\)Levermore [56] introduced closures involving 5, 10, 14, 21, 26 and 35 moments.
that this set $D$ is equivalent to the set of sequences of moments generated by a function in the form (4.7). Now, we are interested in the properties of $R \setminus D$. Considering an unbounded velocity domain, it can be shown that $R \setminus D$ is always non-empty and has some problematic properties. For example, Junk [48] developed an explicit characterization of $R \setminus D$ for the one-dimensional case using five moments. After normalising the given set of moments to $(1, 0, 1, \tilde{u}_3, \tilde{u}_4)$, using the results from 3.3, he first proved that $R = \{(1, 0, 1, \tilde{u}_3, \tilde{u}_4) \mid \tilde{u}_4 > 1 + \tilde{u}_3^2\}$. After that, in an extensive proof, he derived that: $D = R \setminus \{(1, 0, 1, 0, \tilde{u}_4) \mid \tilde{u}_4 > 3\}$ by making some observations on the structure of $\alpha$, for example, that the highest significant exponent has to be negative. By these restrictions on $D$, the convexity of $D$ is destroyed, and the closing flux near the boundaries of $D$ can become singular. Moreover, Junk and Unterreiter [50] proved that for functions lying in $\Phi$ which grow super-quadratic at infinity $R \setminus D$ is always non-empty and has the undesirable property that equilibrium distributions lie at the boundary of $D$ leaving some problems with near-equilibrium conditions ill-posed. It is, however, possible that $R = D$ for some bounded velocity domains as shown in [49].

Unfortunately, for such bounded velocity domains, the resulting system of moments is no longer Galilean invariant. In contrast to this, for an unbounded velocity domain, accepting some problems with realisability, Galilean invariance is not a problem as all systems introduced by [56] are Galilean invariant.

Furthermore, as proved in [56] the maximum-entropy closure leads to a hyperbolic system of moment equations in $D$ assuming that the velocity moments always remain realisable by the maximum-entropy closure.

All in all, the maximum-entropy closure is a very favourable closure because it can be proven that this closure satisfies many desirable properties, such as hyperbolicity. Additionally, as shown in many test cases conducted in [8, 17, 38, 63, 79, 80, 81, 83], this closure is capable of simulating non-equilibrium flows and even shocks in different regimes.

Despite problems with some near-equilibrium conditions, the major drawback of this approach is its computational cost. Unfortunately, in the presented maximum-entropy closure for sets of moments with order higher than two, the solution of (4.7) has to be calculated numerically and consequently takes quite a long time. This problem is intensified by the insight that this computation needs to be done for every grid-point at every timestep. Due to its computational and numerical complexity, several attempts to optimise the solution of (4.7) have been made by optimising the numerical solution algorithm in terms of speed and stability as well as examining the usage of GPUs to solve the problem more quickly [1, 2, 3, 81]. Unfortunately, none of these approaches achieved a substantial decrease of computational complexity and the problem of solving (4.7) remains.

Aiming to develop an efficient version of the maximum-entropy closure in the next chapter it is interesting to note that it is possible to calculate the maximum-entropy closure efficiently for systems where moments only up to order two are involved. In fact, the first two members of the hierarchy introduced by Levermore [56] are of such a form where no optimisation process is needed: For five moments $(1, c_i, c_i^2/2)$ the maximum entropy distribution reduces to a Maxwellian distribution $f_M$. Additionally, for 10 moments $(1, c_i, c_{ij})$ the maximum entropy distribution is a multivariate Gaussian. The resulting system is well known as the Gaussian closure. Offering a hyperbolic, always realisable, non-negative and efficient closure technique the Gaussian closure has also been extensively applied to the Boltzmann equation. In comparison to the QMOM, which will be introduced in the next section, and higher order maximum-entropy closures, this closure offers a closed form.
expression for \( f^* \) in terms of the given moments. Despite capturing only moments up to order two and thereby ignoring, for instance, the heat flux completely, in [8, 16, 38, 63] several test cases have been conducted showing that this closure can successfully capture flows in the transition regime where the heat flux is not of importance.

### 4.5. Quadrature method of moments

The quadrature method of moments (QMOM) is an exemplification of the third philosophy introduced in the beginning of this chapter. This method which relies on a reconstruction of the unknown functions in the form of a Gaussian quadrature was introduced by McGraw [65]. Since then, the closure has been applied to solving the problem of moments in different domains, for example for solving different kinetic equations [32, 33, 60, 62]. As the underlying idea of the QMOM is to obtain a quadrature-based reconstruction of \( f \), this approach also has a model distribution and can be converted into a type one approach. The QMOM assumes that the number density function can be reconstructed as a sum of weighted delta function [34]:

\[
f^{(QMOM)}(x, c, t) = \sum_{i=1}^{n} w_i(x, t) \delta(c, \gamma_i(x, t)),
\]

s.t. \( u_{i,j,k}(x, t) = \sum_{l=1}^{n} \gamma^l_i(x, t) \gamma^j_{\ell_1}(x, t) \gamma^k_{\ell_2}(x, t) w_l(x, t), \forall u_{i,j,k}(x, t) \in u(x, t) \) (4.9)

where \( \delta \) denotes the Kronecker delta which is one if both arguments are equal and zero otherwise. Consequently, the main challenge of this approach is to find a set of abscissas \( \gamma_1, ..., \gamma_n \in \mathbb{R}^d \) and weights \( w_1, ..., w_n \in \mathbb{R}_{\geq 0} \) so that the reconstructed distribution \( f^{(QMOM)} \) fits the given moments. The moments missing to close (3.1) can then be easily and fast computed using \( f^{(QMOM)} \) using the integration formula from (4.9).

One important characteristic of \( f^{(QMOM)} \) is the number of abscissas \( n \). The number of quadrature points which can be found crucially depends on the number of moments given. Usually, to ensure that the set of moments given uniquely determines the quadrature, the number of variables in the quadrature \((n + dn)\) is equal to the number of moments given. Consequently, in one dimension, the number of abscissas computed is usually half the number of moments given. As a result, the precision of this approach depends on the number of moments given and the method has the desirable property that its precision can easily be increased by increasing said number. Nevertheless, calculating the abscissas and weights is not a trivial problem. Concerning the problem in one dimension, assuming that \( 2N \) moments are given, the calculation of a quadrature requires solving the non-linear system:

\[
\sum_{i=0}^{N-1} w_i \gamma_i^k = u_k, \forall k = 0, ..., 2N - 1.
\]

(4.10)

In principle, this problem is highly non-linear and notoriously ill-conditioned [86]. Therefore, it is important that a special algorithm is used to solve this problem. In one dimension, this problem is well studied and can be solved by a product-difference (PD) algorithm [36, 73, 103] which transforms the problem into a well-conditioned eigenvalue-eigenvector problem which can be solved both efficiently and stable even for large \( n \). One of the several desirable properties of such PD-algorithms is that for a set of realisable moments the computed weights are always guaranteed to be non-negative. Consequently, such PD-algorithms can be used to calculate a physical realisable quadrature-based reconstruction \( f^{(QMOM)} \) in one dimension for all \( \Phi_1^{2N-1} \).
However, it is not possible to extend such PD-algorithms to multi-dimensions. Until now, no algorithm has been found which is able to directly, reliably and quickly calculate a set of abscissas and positive weights from a set of moments of a multidimensional distribution function. Nevertheless, in [28, 32, 33] methods aiming to extend the QMOM approach to multi-dimensions have been presented.

4.5.1. A two-node closure for kinetic equations

The simplest approach has been proposed in [28]. In this paper, assuming that the velocity distribution in d dimensions can be reconstructed using the sum of the resulting two weighted delta functions, the authors aimed to construct a set of two abscissas and corresponding weights \( V = \{(w_1, \gamma_1), (w_2, \gamma_2)\} \in \mathbb{R}^{1+d} \times \mathbb{R}^{1+d} \). To present the main results of [28] in a readable way, it is necessary to adjust our notation:

To construct a two point quadrature, the following set of moments is used:

\[
\tilde{u}^{0} = (\tilde{u}^{0}, \tilde{u}^{1}, ..., \tilde{u}^{d}, \tilde{u}^{2}) \in \mathbb{R}^{2+2d},
\]

where \( \tilde{u}^{0} = \sum_{i=1}^{d} \tilde{u}_{i,i}^{3} \). It is interesting to note that this set reduces to the first four monomial moments in one dimension. Desjardins et al. [28] have successfully proposed a method to calculate \( V \) directly from \( u \) and even proved that the reconstruction is unique and the resulting weights are always positive provided that the given moments are realisable. The key part of the method is presented in Proposition 2.1 of [28], which can be summarised as follows: Given a realisable set of moments \( u \)

\[
w_1 = \left( \frac{1}{2} + \alpha \right) \tilde{u}^{0}, \quad w_2 = \left( \frac{1}{2} - \alpha \right) \tilde{u}^{0}, \quad \gamma_{1,i} = \gamma_{p,i} - \left( \frac{w_2}{w_1} \right)^{0.5} \sigma_{p,i}, \quad \gamma_{2,i} = \gamma_{p,i} + \left( \frac{w_2}{w_1} \right)^{0.5} \sigma_{p,i},
\]

where the double subscripts are necessary because \( \gamma \in \mathbb{R}^{d} \) and the additional quantities are defined as:

\[
\alpha = \frac{p_{b}/2}{\left( p_{b}^{2} + 4 \left( \sum_{i=1}^{d} \sigma_{p,i}^{2} \right)^{2} \right)^{0.5}}, \quad \gamma_{p,i} = \frac{\tilde{u}_{i,i}^{1}}{\tilde{u}^{0}}, \quad \sigma_{p,i} = \left( \frac{\tilde{u}_{i,i}^{2} - (\tilde{u}_{i,i}^{1})^{2}}{(\tilde{u}^{0})^{2}} \right)^{0.5},
\]

\[
p_{b} = \frac{1}{\tilde{u}^{0}} \left( q - \tilde{u}^{0} \sum_{i=1}^{d} \gamma_{p,i}^{3} - 3 \tilde{u}^{0} \sum_{i=1}^{d} \sigma_{p,i}^{2} \gamma_{p,i} \right).
\]

At first glance it might look odd that the number of calculated nodes does not increase in multi dimensions, whereas the number of moments used for the calculation increases up to eight in three dimensions. However, note that an abscissa in three dimensions is determined by three variables (one for each dimension). Therefore, the two node quadrature contains eight variables and, consequently, also eight moments are necessary to make the solution unique and close the system.

4.5.2. Higher order methods for the QMOM

To be able to achieve a higher accuracy and to model complex distributions, in [32, 33] the multidimensional QMOM has been extended to a higher number of nodes. The general
approach of such higher order methods is the following:

**Algorithm 1: Steps of the general QMOM scheme**

1: Use $2N$ pure moments (e.g. $u_{0,0,0}^{0}, \ldots, u_{2N-1,0,0}^{2N-1}$) in each spatial direction to compute a $N$ point quadrature fitting the pure moments in this direction.

2: Using the sets of calculated abscissas $\gamma_1, \ldots, \gamma_d \in \mathbb{R}^N$, calculate the abscissas in $d$ dimensions as the tensor product of the $d$ vectors of one dimensional abscissas. (The set of abscissas $\Gamma \in \mathbb{R}^{N^d \times d}$ is defined as $\Gamma = \gamma_1 \times \ldots \times \gamma_d$.)

3: Calculate the $N^d$ weights of the abscissas $\Gamma$ by solving a linear system of constraints on such weights. The constraints are determined using the calculated one-dimensional weights and some mixed moments with a maximal order of $N-1$ in each direction. The constraints on the weights have to be chosen in a way that they are linear independent to ensure that the weights can be uniquely determined by solving the resulting linear system.

As mentioned earlier, step 1 of algorithm 1 is done using a PD-algorithm. The calculation of step 2 is straightforward. Unfortunately, algorithm 1 is still challenging because considering moments involving multi-dimensions in step 3 makes the output of the algorithm unclear, since the solution of a linear system of constraints (the weights of $\Gamma$) is neither guaranteed to be existent nor non-negative nor unique. To ensure that the linear system has a unique solution, the number of linear independent constraints on the weights has to be equal to the number of abscissas $N^d$. This poses a problem because the number of moments up to order $2N-1$ (those are used in step 1) is much higher than $N^d$. Consequently, only a subset of the full moment tensor up to order $2N-1$ can be used as constraints on the weights in 3, and the quadrature only needs to fit these moments. As a result, usually, only this subset of the full moment tensor is transported through the Boltzmann equation and moments of different order need to be reconstructed to close the spatial fluxes in (3.1). The reconstruction of moments of different order is a considerable drawback of the QMOM because $f^{(QMOM)}$ has to be accurate up to the highest order moment which needs to be reconstructed while not all possible information to obtain such a reconstruction are given and could not even be used. Considering only a subset of moments that needs to be fitted is one fundamental difference between the properties of algorithm 1 and a PD-algorithm in one dimension.

However, the only way to recover the property of being able to fit full moment tensors is to solve the system of constraints on the quadrature (4.9) directly instead of using the presented technique of splitting the calculation of the abscissas and the weights. Unfortunately, (4.9) is non-linear and usually ill-conditioned. Fox [32] and Yoon and McGraw [106] conducted some experiments using this numerical approach and stated that the usage of such a numerical solver of the non-linear problem is problematic because the process is usually very slow and sometimes does not even converge. Both [32, 106] suggested that instead of solving the system directly solvers building upon the result of one of the well studied and reliable PD-algorithm (such as algorithm 1) should be used.

Unfortunately, despite guaranteed uniqueness, the calculated weights from algorithm 1 cannot be guaranteed to be non-negative.

Introducing some freedom in step 3, the most fundamental difference between different versions following the general approach of algorithm 1 is the set of moments considered in step 3 and the form of moments used to calculate the quadrature. In the following, one example will be presented. In [32] a $2^d$ quadrature is computed with $u$ being the full tensor of original moments up to third order. As mentioned above, not all of these moments can be set as constraints in step 3). Therefore, the method proceeds as follows:
First, the given set of moments is normalised and centred. After that, the covariance matrix is diagonalised. A subset of the obtained tensor containing one mixed moment in the two-dimensional case and four moments in the three-dimensional case is then used to construct a quadrature of the transformed moments using algorithm 1. Note that since the rotated system of moments is a linear combination of all moments, all moments influence the final result. However, not all rotated moments up to third order are fitted by the quadrature. The calculated quadrature is afterwards converted into a quadrature of the original set of moments by a linear transformation. The resulting quadrature is then used to calculate the full moment tensor of the distribution function up to fourth order.

The third step of the described method is studied extensively in [32] and it has been proven that the resulting weights calculated from the considered subset of the transformed set of moments are always non-negative \(^{11}\) and that the method is well-defined for all realisable moments. Unfortunately, since only a subset of the rotated moment tensor is considered in step 3, the method is unable to reconstruct all realisable moments correctly. Consequently, the resulting quadrature can only capture a 14-dimensional subspace of \(\mathcal{R}\) in 3D. By using the calculated quadrature lying in said subspace of realisable moments to reconstruct the function and to recalculate and subsequently replace the current moment tensor up to third order as well as the fluxes during each timestep, the moments are always reprojected into this subspace of \(\mathcal{R}\).

One extraordinary remark concerning the construction of the linear system in the third step of algorithm 1 is that by forcing the multi-dimensional quadrature to agree with the one dimensional quadrature (e.g. the sum of all weights of abscissas with equal x-coordinate has to be equal to the weight of the corresponding abscissas in x-direction) and the fact that only linear independent constraints can be considered, a fit with 14 moments in 3D can be achieved.\(^{12}\)

The purpose of centring and rotating within this method is to standardise the input thereby simplifying proofs, making full use of the tensor product built in step 2 of algorithm 1, and to combine the third order moments in the mixed moments considered in step 3. Furthermore, through rotation, the chance of staying non-negative for higher order methods can be improved significantly [33]. As stated in [33], it is also possible to reformulate the Boltzmann equation in terms of the transformed moments and, consequently, to transport the reduced set of transformed moments considered in step 3 without considering the original set. The benefit of rewriting the Boltzmann equation is that the transported set is much smaller than the full tensor. In case that one aims to transport the original non-transformed moments, at the beginning of each step, the moments need to be transformed and for this process, the full moment tensor up to the highest order of transformed moments is required. Moreover, this process can introduce some considerable "round-off" errors.

In [33], the described method was extended to higher orders to calculate a \(N^3\) quadrature of arbitrary three-dimensional distribution functions. They considered the Boltzmann equation in terms of the transformed moments and presented the transported sets of rotated moments for \(N = 3\) and \(N = 4\) of size \((N^2 + 3)N\) involving moments up to order \(2N\) that can be fitted by the calculated quadrature. Thereby, only around half of the

\(^{11}\)In fact, they proved non-negativity for the 1D and 2D case. For the 3D case, they proved that negative weights can occur but can be avoided by setting one weight to zero thereby reducing the number of moments considered and fitted in step 3 by one. It is, however, remarkable that Fox [32] reported that this procedure was never needed in all conducted experiments.

\(^{12}\)Intuitively, the number of moments used in step 3 and thereby fitted by the quadrature would be equal to the number of weights (8).
full moment tensor is used. For such higher-order methods in multi-dimensions, however, the non-negativity of the weights computed from a realisable set of moments cannot be guaranteed. Nevertheless, experiments indicated that the computed weights are usually non-negative [33]. One possible treatment of a situation in which negative weights occur is that the negative weights are set to zero and the linear system and the set of moments fitted are reduced.

4.5.3. Properties of the QMOM

As discussed above, there exist different versions of the QMOM closure. The properties of the QMOM closure depend on the particular method chosen. However, in the following, a few general remarks will be made. First of all, as mentioned earlier non-negativity can be, at least in one dimension and for lower order methods, guaranteed. For higher order methods, this is no longer true, and one has to rely on some exception handling. For example, the set of moments that needs to be fitted is reduced while one weight is set to zero. Unfortunately, the non-negativity of the weights is very important for every simulation conducted using the QMOM because usually a kinetic flux as described in section 6.1.1 is used, and the non-negativity of weights is necessary to ensure that the numerical scheme remains stable [33].

Also the realisability property of this closure depends on the method chosen. For the one dimensional and the two-node case, the calculation of a fitting reconstruction for all sets of realisable moments can be guaranteed. As discussed extensively earlier, for higher order methods, the full moment tensor given cannot be fitted exactly. However, the method remains well-defined. The influences of the error introduced due to the recalculation of all moments from the only partly fitting reconstruction need further examination.

The flexibility of this approach is usually quite high. In one dimension, the only restriction is that solely full moment vectors up to an odd degree can be used. In the multidimensional case, the order of the method could be restricted by the fact that only parts of the full moment tensor can be fitted, the ill-conditionedness of the linear systems and the treatment of negative weights. However, the restrictions made on the transported set of moments $u$ are often quite strong [31, 33].

The Galilean invariance of the system also depends on the set and types of moments considered but can be, generally speaking, often guaranteed. However, sometimes the system of moments considered is not symmetric, for example, only one mixed moment of different orders in different dimensions is considered sometimes causing problems for the calculation of fluxes.

Unfortunately, since the QMOM is highly non-linear, it cannot be rigorously proven that this closure is hyperbolic. However, if we assume that the abscissas are fixed in advance, the hyperbolicity of the system would follow immediately. Furthermore, intuition suggests that the eigenvalues of the system are the calculated abscissas because they determine the speed at which information travels. This intuition can be confirmed by calculating $J(u)$ in the one dimensional case for some $u$ using numerical differentiation. The eigenvalues of this matrix are approximately the calculated abscissas with only sometimes very small imaginary parts (due to round off errors) indicating that our intuition was correct and that the closure is hyperbolic. Consequently, the abscissas with the maximal absolute value can be interpreted as the characteristic speed $\lambda$ of the system.

Moreover, the QMOM is, in principle, capable of simulating flow problems for any Knudsen number. However, the precision of this method crucially depends on the chosen initial conditions and Knudsen number. For non-collisional systems with uni-modal ve-
locity distribution (e.g. two non-collisional crossing jets or jets impinging on a wall), the QMOM is well suited to achieve high order accuracy depending on the chosen numerical scheme [28]. However, one of the main shortcomings of the QMOM is to capture the presence of particles with various velocities at one point. In such systems, the behaviour and accuracy of the method depend entirely on the number of moments given [33].

In [32, 33] several experiments aiming to examine the capability of the QMOM to capture different non-equilibrium flows have been conducted. Since the Riemann test case is a standard test case to exhibit the behaviour of a closure under non-equilibrium conditions, the behaviour of the 2nd-node closure described above in said test case is extensively examined. For low Knudsen numbers and therefore for flows where the velocity distribution is near to Maxwellian, the proposed third order method has proven to be able to produce simulations capturing the main properties of the solution of a Riemann problem. For the transition regime, however, different forms of oscillations and delta-shocks occur indicating that for larger Knudsen numbers (strong non-equilibrium flows) the number of moments given is insufficient. However, the authors of [32] highlighted that despite producing non-physical behaviour the closure always produces realisable moments and non-negative weights. Another interesting remark that has been made is that concerning higher Knudsen numbers, since the effects of collisions are no longer dominant, it can be observed that the abscessas remain near to their initial value during the simulation. The reason for this is that without collisions the particles with one speed represented by an absissa travels unimpeded at their speed from left to right and thereby propagate the shock wave represented by their weights through the system.

Generally speaking, the accuracy of the QMOM decreases with increasing Knudsen number where non-equilibrium and especially non-uniform velocity distribution occur [33].

To exhibit the limits of the QMOM Fox [33] examined the solution of the Riemann problem at different Knudsen and Mach numbers for higher order moment closures. Consistent with the findings of [32], it has been shown that the QMOM encounters serious problems when the Knudsen and Mach number increases. However, using a higher number of quadrature points (e.g. 27 or 64 in 3 dimensions) and thereby a higher number of moments, for moderately Mach and Knudsen number, these problems can be overcome, and a reasonable precision can be achieved. One of the general findings of [33] is therefore that capturing the non-equilibrium behaviour created by higher Knudsen numbers requires a higher number of nodes and can be done efficiently up to a certain order of both quantities. In general, the usage of a higher number of moments increases the accuracy and reliability of the method.

Additionally, because of the nature of \( f^{(QMOM)} \), the QMOM is also able to take different effects such as fluid drag, gravity and complex models of the collision process into account. This method is therefore able to close different forms of non-linear terms in kinetic equations in an efficient manner. In [28, 32] several test cases have been conducted indicating that the QMOM is also able to model the effect of those terms (e.g. a Taylor-green flow for different Stokes numbers) correctly. In [42] even the Boltzmann hard-sphere collision term has been considered.

4.5.4. Relationship to other methods and extensions

To relate the QMOM method to other approaches and to thereby somehow justify this approach and also emphasize possible applications, links to different methods will be presented in this section.
First of all, a method which is similar to the QMOM is used in geometrical optics. The method is used in the context of solving the linear wave equation which can be transformed into a kinetic equation corresponding to the collisionless Boltzmann equation. The method of moments is here also used to simplify the obtained kinetic equation. In contrast to fluid dynamics, in the geometrical optics setting, those moment systems are exact assuming that the number of crossing rays is restricted by the number of moments given [78]. In geometrical optics, the solution of this unclosed system is done using a "multiphase" solution relying on a reconstruction of the distribution in question in terms of a weighted sum of delta functions similar to the QMOM. However, one crucial difference is that usually a non-linear solver in contrast to a PD-algorithm is used to obtain the abscissas and weights thereby restricting the closure to a limited number of moments given [77, 78] due to stability issues.

Furthermore, the QMOM can be related to two types of methods aiming to find very accurate solutions of kinetic equations: Lagrangian methods [29] and direct solvers using a discretised velocity space [15]. On the one hand, a relationship between the QMOM and Lagrangian methods can be found, since (4.8) can be reinterpreted in a Lagrangian way as a discretisation of the velocity distribution into weighted parcels. Furthermore, it is also possible to interpret the QMOM in a DSMC-way by seeing the finite set of nodes at each spatial coordinate as a set of representative particles with the corresponding velocities at each place. Using this interpretation, the behaviour of the QMOM in the collisionless case can also be plausibly explained.

On the other hand, the QMOM technique can also be related to direct solvers relying on a discretisation of the velocity phase space. Those solvers usually use a fixed discretisation of the velocity phase space [15]. $f^{QMOM}$, however, can be seen as an adaptive discretisation of the velocity phase space providing a very sparse discretisation which is consistent with the given moments of the velocity distribution.

Concerning the solution of the Boltzmann equation, the most closely related method is possibly the lattice Boltzmann method and especially the multi-speed off-lattice Boltzmann method [51]. Said methods also rely on a quadrature based reconstruction of the velocity distribution aiming to fit the given moments. In contrast to QMOM, however, the abscissas are always fixed (e.g. as the roots of the Hermite polynomials) and only the weights are unknown and constructed in a way to be consistent with the given moments. Since the flexibility of this approach and the degrees of freedom are much lower in comparison to QMOM, the capability of such methods is restricted to simulating flows for low Knudsen number [51].

In the recent past, the presented quadrature method of moments has been extended in many different ways. First, the direct quadrature method of moments (DQ MOM) was introduced by [60]. In this closure (4.8) has been plugged into the system of partial differential equations in question, and the system has been reformulated in terms of the weights and the abscissas instead of the moments. Since the transport equations are no longer formulated in terms of the moments of the distribution function, no reconstruction of the underlying function is needed. As a result, this closure is very efficient. However, the applications of the DQ MOM are quite restricted. For example, aiming to model polydisperse gas-solid flows it has to be assumed that the velocity distribution is near to equilibrium and the velocity distribution is fully characterised by its mean [30].

After that, the conditional method of moments (CQ MOM) was proposed in Yuan and Fox [107]. In this closure, the weights and abscissas are determined by decomposing $f$ into a product of d-1 conditional probability distributions and one unconditional probability distribution: $f(c) = f(c_2 \mid c_y, c_z)f(c_y \mid c_x)f(c_x)$. First, a reconstruction of $f(c_1)$ is calcu-
lated. Using this, the unknown conditional moments of \( f(c_y | c_i) \) are obtained by solving a linear system. The obtained information are then used to calculate the quadrature interpolation of \( f(c_x, c_y) \). This procedure can be applied again to extend this quadrature to the 3D case.

Finally, in Yuan et al. [108], the extended quadrature based method of moments (EQ-MOM) was proposed. This method uses a slightly different model distribution (4.8). Instead of using delta functions, this closure assumes an underlying kernel density function. Consequently, the reconstructed distribution consists of a sum of weighted kernel densities, for instance Gaussians, centred at different points in the velocity space.

5. Entropic Quadrature

In section 4.1 the problem of moments has been introduced, and possible applications have been discussed shortly. Among all described and available closures for this problem, the maximum-entropy closure has proven to be one of the most promising solutions to this problem. Due to the H-theorem, this is especially true in the context of solving the Boltzmann equation. The reasons for this are its strong theoretical justification, its various desirable properties and especially its ability to capture non-equilibrium micro-flows. Unfortunately, when it comes to solving a higher number of problems of moments, the computational cost of this closure is barely affordable. In contrast to this, one of the other promising approaches to close kinetic equations is the QMOM, which has somehow diametric properties because said closure is very efficient and fast but lacks a convincing physical justification.

In the past, to the best of our knowledge, both approaches have never been combined. However, some studies have been conducted aiming to compare both approaches [52, 70]. As one would intuitively expect, Müller et al. [70] showed that, in the context of solving the population balance equation, the maximum-entropy closure usually leads to a smaller error compared to the QMOM, while having a significantly higher computationally cost. Furthermore, Laplante and Groth [52] conducted some experiments on the ability of both closures to capture stationary shock profiles and concluded that the maximum-entropy closure should be chosen instead of the QMOM when aiming to accurately capture strong non-equilibrium flows.

In the following, a novel approach combining both techniques is introduced. The proposed approach, which we call entropic quadrature (EQ), tries to find a quadrature based representation of \( f^* \) that maximises the entropy while fitting the given set of moments. This method aims to combine the power and physical meaningfulness of the maximum-entropy closure with the efficiency of the QMOM. The ultimate goal is to obtain an efficient closure which is still capable of describing flows for an arbitrary Knudsen number. By combining both approaches, we hope that the problem of the computational cost of the maximum-entropy closure and the lack of ability of the QMOM to capture flows with non-equilibrium velocity distributions can be overcome. Generally speaking, depending on the perspective, entropic quadrature can either be seen as a very efficient and sparse version of the maximum-entropy closure or as an extension of the QMOM aiming to obtain a quadrature that not only fits the given moments but also maximises the entropy of the reconstructed distribution.

The method of entropic quadrature, which will be proposed in this section, can be applied to the Hamburger problem of moments and all of its applications. However, because of its computational efficiency, using EQ is especially beneficial when a higher number of problems of moments needs to be solved. In this thesis, however, we choose
to introduce this closure in the context of solving the Boltzmann equation because the
maximum-entropy closure for this problem has some extraordinary physical and theoret-
cal justifications. Furthermore, efficient closure methods are desperately needed in this
field, and both the maximum-entropy closure and the QMOM have already been applied
to the Boltzmann equation.

5.1. Overview of Entropic Quadrature

The goal of this section is to introduce the method of entropic quadrature in some detail.
As mentioned earlier, this method aims to construct a quadrature based representation
of $f^*$ which maximises the entropy while fitting the given set of moments. It is, therefore,
similar to the QMOM, assumed that the unknown distribution can be represented as a
sum of weighted delta function. The model distribution is identical to $f^{(\text{QMOM})}$:

$$f^{(\text{EQ})}(x, c, t) = \sum_{i=1}^{n} w_i(x, t)\delta(c, \gamma_i(x, t)). \quad (5.1)$$

Also, similar to the QMOM, the model distribution is subjected to the constraint that
the given moments are fitted:

$$\text{s.t. } u_{i,j,k} = \sum_{l=1}^{n} \gamma_{l_1}^{j_1} \gamma_{l_2}^{j_2} \gamma_{l_3}^{j_3} w_l, \forall u_{i,j,k} \in u \quad (5.2)$$

However, in contrast to the QMOM, an additional constraint is placed on the recon-
structed distribution:

$$f^{(\text{EQ})} = \arg \max_{f^{(\text{EQ})}} -\langle f^{(\text{EQ})} \log f^{(\text{EQ})} - f^{(\text{EQ})} \rangle. \quad (5.3)$$

As a result, we search for the distribution with the highest entropy that fits the given set
of moments and is in the form of $f^{(\text{EQ})}$. To proceed, it is important to note that using
(5.2), (5.1) and the fact that we assume the $u_{0,0,0} \in u$, (5.3) can be rewritten to:

$$f^{(\text{EQ})} = \arg \min_{f^{(\text{EQ})}} \sum_{i=1}^{n} w_i \log w_i \quad (5.4)$$

Considering (5.4), it becomes obvious that the entropy of the reconstructed function and
therefore $f^{(\text{EQ})}$ highly depend on the number of abscissas $n$ and that this parameter has to
be fixed in advance to expand on this approach further. Otherwise, since $f^{(\text{EQ})}$ is discrete,
calculating $f^{(\text{EQ})}$ will become intractable because the number of abscissas is not limited.
The reason for this is that a higher number of nodes always introduces the possibility of a
higher amount of uncertainty. This is not only intuitively plausible but also supported by
the fact that the entropy of a $n$-valued random variable is at maximum $\log(n)$. Moreover,
it is possible to see the entropy of a distribution function as the entropy of such a $n$-valued
random variable with corresponding probabilities $w_1, ..., w_n$ for the events in question. To
make the consequences of a non-fixed $n$ clear, we consider an example in one dimension:
Let us assume that the domain is restricted to $[-1, 1]$ and only the first moment $\rho = 1$
and the second moment $\rho v = 0$ are given. Then, there exists always a function that
has a higher entropy than any function with a lower number of abscissas. For example,
h(x) = $\sum_{i=1}^{n} \frac{1}{n} \delta(x, 1 - 2i\frac{1}{n-1})$ fits always the given set of moments, while the entropy of
this family of functions $-\sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log(n)$ is strictly monotonically increasing and
is therefore maximal for \( \lim_{n \to \infty} \). As a result of these considerations, in the context of EQ, to ensure that the entropy of the distributions in question remains comparable and that the distribution remains sparse, we will assume that \( n \) is always fixed as part of the input of the problem and can be considered as a parameter for the density of the reconstruction.\(^{13}\)

By using one of the sets of moments considered by the QMOM as input and setting \( n \) to the number of nodes calculated by the QMOM, one can immediately recover the relationship between the QMOM and EQ. For the quadrature calculated by the QMOM, the maximum entropy property (5.4) is always fulfilled because in the QMOM the given set of moments uniquely determines the weights and abscissas. This is due to the fact that the QMOM and its input is constructed in a way that no degree of freedom is introduced, and the only goal is to fit the given set of moments. Consequently, in the case that a set of moments \( \mathbf{u} \) fulfilling the properties of an input for the QMOM is given, and the number of nodes is fixed to the size of the QMOM-quadrature, the QMOM-quadrature is always the unique solution of (5.1) and (5.2) and therefore also the solution of (5.3).

In contrast to the QMOM, EQ aims to introduce some freedom into the calculation of \( f^{(EQ)} \) to allow for multiple quadratures fitting the given set of moments which enables us to choose the reconstruction maximising the entropy. Considering the ideas from the previous paragraphs gives us a strong hint on a way how additional freedom can be introduced: To prohibit that \( f^{(EQ)} \) is uniquely determined by the set of moments given and to make use of the maximum-entropy principle, the number of unknowns in the quadrature should be higher than the number of moments given. In other words, to allow for some freedom, \( n \) needs to be increased. For example, in one dimension, using the QMOM, \( 2N \) moments are given to calculate an \( N \)-point quadrature which is thereby uniquely determined by the given moments. In EQ, to introduce some freedom, we can, for example, set \( n = 2N \) to ensure that there exist different fitting quadratures from which we can choose the one with the highest entropy. To summarise, EQ inserts new nodes to introduce an additional degree of freedom into the calculation of the quadrature and to make use of the maximum entropy principle. Otherwise, the solution of (5.1) and (5.2) is unique and the maximum entropy principle useless.

Given a realisable set of moments and a desired number of abscissas \( n \), the general idea of EQ is to find a quadrature of \( n \) nodes \( \gamma_1, \ldots, \gamma_n \in \mathbb{R}^d \) and \( n \) weights \( w_1, \ldots, w_n \) so that (5.2) is fulfilled and the entropy of \( \mathbf{w} \) and of \( f^{(EQ)} \) are maximised. Unfortunately, the resulting optimisation problem (5.4) with the non-linear constraints (5.2) is ill-conditioned and very hard to solve. Solving this problem numerically is even harder than solving the more general problem of determining a quadrature-based reconstruction (4.9) that is uniquely determined by a given set of moments numerically because additional degrees of freedom and an optimisation criterion are introduced. As mentioned above, Fox [32] suggested not to solve (4.9) directly because the computational cost is very high and convergence is not guaranteed. In the situation considered here, both problems get even intensified by the added optimisation criterion. Thus, we follow the ideas of [32, 65] and instead of solving (5.3) directly we will develop an algorithm building upon a well-studied and stable PD-algorithm [73, 103].

Hence, we will try to develop an algorithm which can calculate a quadrature without solving the original constrained optimisation problem directly. To achieve this, we need to simplify or restructure the problem. Similar work was also necessary for the QMOM in multiple dimensions [32, 33]. To justify the way in which EQ proceeds, first observe

\(^{13}\)Please note that a fixed \( n \) only provides an upper bound on the number of significant nodes because it is always possible to set weights to zero.
that the entropy of the function (5.4) depends solely on the weights and not on the chosen abscissas. Therefore, similar to [32, 33] for the multidimensional QMOM, we will split the process into two parts: First, a set of \( n \) abscissas is calculated. Subsequently, (5.4) is solved to obtain a set of \( n \) weights so that the resulting distribution function fits the given set of moments and simultaneously maximises the entropy of the distribution. However, as we will argue in the following, it will not always be possible to find a suitable realisable set of weights fitting the given set of moments.

It remains to specify how these two steps should be addressed. Concerning the first step, there is much freedom in choosing the abscissas. Since we aim to calculate a sparse representation of an arbitrary distribution function on an unbounded (or at least very large) domain, we would like to use an adaptive set of abscissas depending on the moments given instead of a fixed set of quadrature points like the Gauss-Lobatto quadrature. In the latter case, to ensure that there exists a realisable vector of weights fitting the given moments and achieve an acceptable precision, the number of notes has to be extraordinarily high. Furthermore, as noted in [33], a quadrature whose abscissas and weights are variable and adapt to the underlying moments usually has a higher order precision in comparison to the same number of fixed quadrature points. Additional two criteria that can be used to judge a method calculating the abscissas are that there, firstly, should exist a set of non-negative weights for the calculated nodes so that the resulting distribution function fits the given set of moments, and, secondly, that it should be possible to extend the method to calculate a range of possible numbers of abscissas \( n \). Concerning the first criterion, there is little hope to guarantee this property in multi-dimensions because not even the QMOM can guarantee the existence of such non-negative weights for moment tensors including moments of order higher than two [33]. However, in one dimension the PD-algorithm guarantees to find such a set of abscissas for all realisable moments [103]. Since this is the best we can hope for and PD-algorithms are extensively studied and reliably calculate an adaptive quadrature lying in the significant domain of the distribution function, we will always use the output of the PD-algorithm for the construction of the EQ-nodes. Considering the one dimensional, since using only the nodes generated from the PD-algorithm would result in the quadrature calculated by the QMOM, additional freedom in terms of additional nodes needs to be introduced. Relying on the fact that the nodes calculated by the PD-algorithm are a reliable reconstruction of the unknown distribution function and therefore also an indicator of its significant domain [34], we will insert nodes between the nodes calculated by the PD-algorithm. We will describe the procedure of calculating a set of abscissas in greater detail in the following section.

Concerning the second step, first note that the constraints when solving (5.4) are no longer non-linear because the abscissas are fixed in advance and the constraints in (5.2) are only linear in \( \mathbf{w} \). Since the resulting optimisation problem is convex, we use an iterative procedure to determine a solution. In section 5.2 different methods to tackle this problem will be introduced.

Turning back to the big picture, we will now discuss possible obstacles and justifications of EQ. Since the abscissas calculated by the QMOM are usually a subset of those used in EQ, one might argue that through inserting those additional nodes and recalculating the weights only the optimal quadrature calculated by the QMOM is perturbed and some noise is introduced. To judge this claim, the ways in which the QMOM is optimal should be discussed. One of the advantages of the QMOM is that the structure of the given set of moments is chosen in a way that the obtained quadrature is always uniquely determined by the given moments. This property has two advantages. Firstly, since there are no alternatives to the calculated quadrature with the same size, no decision between
possible reconstruction has to be made. Secondly, the calculated quadrature is minimal and following Occam’s Razor one can argue that such a minimal reconstruction always generalises best. Admittedly, concerning these two criteria, the QMOM is optimal.

However, introducing said noise, also some uncertainty concerning the resulting quadrature is introduced. A realisable solution of (5.2) is possibly no longer unique. These multiple solutions enable the usage of the maximum-entropy criterion. The introduced freedom allows us to make use of the knowledge that from all distributions fitting a given set of moments, the maximum-entropy distribution is, as discussed in 4.4, the most probable and reasonable one. Consequently, in this closure not only the knowledge of the given moments is used but also our knowledge about the increasing entropy of a system. Therefore, we think that introducing degrees of freedom and making use of our knowledge about entropy could increase the precision of the QMOM significantly. Moreover, EQ has two additional benefits: First of all, the density of the quadrature can be increased without increasing the number of moments given. This is very beneficial because it allows for a more flexible reconstruction which is especially crucial to model complex distributions far away from equilibrium. Secondly, by introducing additional nodes, we can hope that we can fit full moment tensors of multi-dimensional distributions of order higher than two. This is one of the things that the QMOM cannot achieve because of its inherit procedure.

All in all, by introducing some degrees of freedom and filling them by using the maximum entropy principle, we hope that we can create reconstructions that are more probable and physically meaningful than the quadratures calculated by the QMOM. One interesting question is whether the introduction of entropy into the quadrature provides even better results than a QMOM-quadrature constructed with a higher number of moments considered. To reformulate the question, is it possible that, in the context of solving kinetic equations, the introduction of a higher amount of entropy into a system is more beneficial than the knowledge of a higher number of moments of the distribution? To address this question, we consider the following example in section 7: In one dimension, we compare the success of $f^{(EQ)}$ with $n = 2N + 1$ calculated from $2N$ moments with the success of the QMOM calculated from $4N$ moments. Both approaches produce nearly the same number of nodes. In the QMOM, the necessary information is given to determine all their weights uniquely, while EQ has to rely on the principle of maximum entropy. From a QMOM perspective, their quadrature has to perform much better because they have much more information available\textsuperscript{14} and do not have to make any speculations about the weights.

From the opposite perspective, the success of EQ as a sparse interpolation of the maximum entropy distribution will also be discussed. In principle, there is much hope that by using EQ a meaningful quadrature of the maximum entropy distribution can be obtained because the quadrature is adaptive and the amount of uncertainty that can be produced can be increased by increasing the ratio of newly inserted nodes to original nodes. In section 7.1.1, we will conduct some experiments aiming to show the efficiency of EQ as a sparse interpolation of a maximum entropy distribution. Furthermore, we will examine the behaviour of EQ when the number of nodes gradually increases.

The remainder of this chapter is organised as follows: First, a method for calculating the abscissas of a one-dimensional distribution function is introduced. After that, the extension to multidimensional distributions will be presented. Next, to make practical use out of this approach, two different efficient ways to solve the constrained optimisation problem (5.4) to calculate the weights will be discussed. Finally, the theoretical properties

\textsuperscript{14} As we will show, an increased number of given moments significantly increases the precision of the simulation.
of EQ will be reviewed.

5.1.1. One-dimensional case

Having outlined the general structure of the EQ-algorithm, it remains to specify how the abscissas and subsequently the weights are chosen. Postponing the second issue for a while, we will first address the former problem in one and multiple dimensions.

As discussed earlier, similar to the QMOM, we want to build upon the abscissas and weights calculated by a PD-algorithm. Following the work of John and Thein [46] and Upadhyay [97], due to its high stability and precision, we will use the Wheeler algorithm as an advanced PD-algorithm to calculate the initial set of abscissas [61, 103]. Note that the usage of a PD-algorithm limits our considered set of moments for now to full sets of moments up to an odd degree. Aiming to introduce some degree of freedom concerning the weights, we insert one node between every two nodes resulting in the following algorithm for calculating the abscissas:

**Algorithm 2: Calculating \( \gamma^{(EQ)} \) in one dimension**

**Data:** Realisable set of 2N moments \( \mathbf{u} \)

1. Use the Wheeler algorithm to create a set of N weights \( \mathbf{w} \) and abscissas \( \gamma \);
2. \( \gamma^{(EQ)} := \gamma \);
3. **for** \( i \in [1, N - 1] \) **do**
4. \( \gamma^{(EQ)} = \gamma^{(EQ)} \cup \{0.5\gamma_i + 0.5\gamma_{i+1}\}; \)
5. **end**
6. \( \gamma^{(EQ)} = \gamma^{(EQ)} \cup \{1.5\gamma_1 - 0.5\gamma_2\}; \)
7. \( \gamma^{(EQ)} = \gamma^{(EQ)} \cup \{1.5\gamma_N - 0.5\gamma_{N-1}\}; \)

The last two steps of the algorithm correspond to inserting one node between each external node and an imaginary node created by the projection of the adjacent node onto the outside. This is done to extend the range of the quadrature.

It is also possible to increase the density of the constructed quadrature by inserting additional nodes thereby increasing its precision at the expense of an increased computational cost. This can be, for example, done by generalising algorithm 2. In this generalised version, instead of inserting only one node in the middle between two PD-nodes (and the imaginary one on the outside of each external node ) \( M - 1 \) nodes are uniformly inserted. In this way, an algorithm with a multiplicative factor of \( M \) instead of 2 can be constructed which calculates a set of \( M \ast N + M - 1 \) abscissas. One of the considerable advantages of this method is that the PD-nodes are kept guaranteeing the existence of a non-negative set of weights fitting the given set of moments (see 5.2.3). Furthermore, using algorithm 2, it is easy to calculate an initial guess for the weights of the abscissas by using piecewise linear interpolations between the PD-weights. However, this algorithm has also some drawbacks because independent from the multiplicative factor chosen the abscissas always stay in the same interval \( [2\gamma_1 - \gamma_2, 2\gamma_N - \gamma_{N-1}] \) thereby restricting the convergence (see 5.2.3). Furthermore, the nodes are not uniform resulting in some problems for the structure of the numerical integration of \( f^{(EQ)} \) (see 5.2.3).

Using the PD-nodes only for a justified restriction of the significant domain of the unknown distribution function, however, it is possible to overcome both issues on the cost of losing the strong connection to the QMOM method and the guarantee of the existence of a non-negative solution by using the following uniform quadrature:
Algorithm 3: Calculating $\gamma^{(EQ)}$ uniformly distributed in one dimension

Data: Realisable set of $2N$ moments $u$, multiplicative factor $M$

1. Use the Wheeler algorithm to create a set of $N$ weights $w$ and abscissas $\gamma$;
2. $a := \gamma_1 - \left(\frac{\log(4^4)}{4^4} + 1\right)(\gamma_2 - \gamma_1)$;
3. $b := \gamma_N + \left(\frac{\log(4^4)}{4^4} + 1\right)(\gamma_N - \gamma_{N-1})$;
4. Create a set of $M \cdot N + M - 1$ uniformly distributed nodes on $[a, b]$;

The variables $a$ and $b$ in this algorithm are very flexible. The choice made in this algorithm is somehow arbitrary and reflects only the motivation to extend the considered domain slightly with an increasing multiplicative factor to make sure that the solution converges to the maximum-entropy distribution (see section 5.2.3) accurate solution.

This uniform quadrature algorithm has the considerable downside that the density of the quadrature has to be much higher to ensure a high probability for the existence of a non-negative set of weights fitting the given set of moments. The reason for this is that the uniform quadrature partly ignores the information about the unknown distribution function entailed in the quadrature calculated by the PD-algorithm. As a result, the density of the non-uniform quadrature in the domain of interest of the function is usually much higher than in the uniform case. Consider, for example, a strongly bimodal Gaussian distribution. On eight moments of such a distribution, the output of both algorithms is visualised in figure 1. The nodes calculated by algorithm 2 are plotted as filled points and consist of the nodes calculated by the Wheeler algorithm and the newly inserted nodes, whereas the nodes calculated by algorithm 3 are plotted as non-filled points. One can clearly see that the nodes from algorithm 2 fit the distribution function considerably better than the one from 3. As a result, there exists a set of non-negative weights for the nodes of algorithm 2, whereas there is no set of positive weights fitting the given set of moments found for the nodes from algorithm 3.

For the sake of comparability of the uniform quadrature with algorithm 2, it is also possible to restrict the uniform quadrature to the same domain as the non-uniform quadrature. Usually, since it is the fastest of all three methods and always has a non-negative solution, we will use algorithm 2.

Figure 1: Example output of algorithm 2 and algorithm 3 on bimodal Gaussian
5.1.2. Extension to multi-dimensions

As there exists no stable algorithm as the PD-algorithm to solve the non-linear problem of calculating a quadrature in multiple dimensions (4.10), we will use the same technique as the QMOM and use the pure moments of the distribution function in each spatial direction as input parameters of the PD-algorithm. This procedure produces a one-dimensional quadrature for each direction which fits the given set of pure moments in the corresponding direction. It is possible to calculate a set of abscissas in d-dimensions from these one-dimensional quadratures by placing the abscissas on the corresponding axis (only one component of the created nodes is non-zero). The resulting quadrature is then able to fit the pure moments of the given set of moments. To enable the quadrature algorithm to fit also mixed moments, it is necessary to insert additional abscissas. To do so, in the QMOM the tensor product of the abscissas on each axis is calculated [33]. Being a reasonable interpolation of the significant domain of the unknown distribution function, we will also make use of this technique. However, the resulting quadrature is usually only able to fit a subset of moments [33]. Aiming to fit all moments up to a given order and to additionally insert some degree of freedom enabling the method to choose the solution with the highest entropy, we will again insert additional nodes. We will do so by using one of the algorithms above to multiply the univariate abscissas and then calculate the tensor product of those univariate quadratures as the multidimensional quadrature. By multiplying the nodes in this way, the quadrature is, in principle, able to fit the full moment tensor because the number of abscissas is higher than the number of nodes.\textsuperscript{15} Note, however, that in multiple dimensions there is no longer any guarantee that a non-negative set of weights exists fitting the given full moment tensor. Nevertheless, in practice as discussed in section 7.1.3, in the great majority of cases, the algorithm will find a realisable and fitting quadrature even if only one node between each node in one dimension is inserted. In the case that no realisable solution exists, which is only rarely the case, different strategies that will be discussed in section 5.4 can be chosen. All in all, the algorithm which calculates the set of abscissas in multiple dimensions is:

\begin{algorithm}
\textbf{Algorithm 4:} Calculating $\gamma^{(EQ)}$ in multiple dimension

\textbf{Data:} Tensor of moments $\textbf{u}$ up to degree $2N$ of a d dimensional distribution function

1. Use the Wheeler algorithm to create d sets of N weights $\textbf{w}^{(1)}$, ..., $\textbf{w}^{(d)}$ and abscissas $\gamma^{(1)}$, ..., $\gamma^{(d)}$ using the pure moments in each spatial direction;

2. Use either algorithm 2, its extension containing a multiplicative factor $M$ or algorithm 3 to insert additional nodes into the one-dimensional quadratures;

3. The set of abscissas in multiple dimensions $\Gamma^{(EQ)}$ is defined as:

$$\Gamma^{(EQ)} := \gamma_{1}^{(EQ)} \times ... \times \gamma_{2}^{(EQ)}$$

An alternative, however, until now untested method, which could also be used to find the abscissas in multiple dimensions, is to rotate the system of moments multiple times using different rotation angles and calculate the abscissas on the rotated axis and then rotate the abscissas back. The calculated abscissas are then a result of different linear combinations of the whole system of moments and lie on the axis of different angles in the original system. This method can be, therefore, seen as beneficial compared to the above one because the quadrature is influenced by all moments and not only by the pure ones. However, this method is computationally much more expensive and unreliable.

\textsuperscript{15}In two dimensions, for example, if we use the full moment tensor up to order $2N$ and algorithm 2 to multiply the univariate weights, the number of moments is $2N^2 + 3N + 1$ and the number of abscissas is $4N^2 + 4N + 1$
5.2. Calculating entropic quadratures

The main task that remains is to calculate the weights of the entropic quadrature. In other words, given a set of abscissas \( \gamma_1, ..., \gamma_n \in \mathbb{R}^d \), which has been already calculated using one of the algorithms described above, find a set of weights \( w \in \mathbb{R}^n \) minimising \( \sum_{k=1}^{n} w_k \log w_k \) under the constraint \((5.2)\). First note that \((5.2)\) can be written as \( Aw = b \) with:

\[
A = \begin{pmatrix}
\gamma_{11}^{i_1} \gamma_{12}^{i_2} \gamma_{13}^{i_3} & \cdots & \gamma_{n1}^{i_1} \gamma_{n2}^{i_2} \gamma_{n3}^{i_3} \\
\vdots & & \vdots \\
\gamma_{1N}^{i_1} \gamma_{12}^{i_2} \gamma_{13}^{i_3} & \cdots & \gamma_{N1}^{i_1} \gamma_{N2}^{i_2} \gamma_{N3}^{i_3}
\end{pmatrix}, \quad b = u
\]

where \( N \) denotes the number of moments given and \( i, j, k \in \mathbb{N} \) the sets of exponents of \( c_x, c_y \) and \( c_z \) in \( \Phi \), e.g. \( (c_x^j, c_y^k, c_z^m) = u \). Note that in one dimension \( A \) reduces to a transposed Vandermonde matrix. As a result, the following section aims to solve:

\[
w = \arg\min_{\tilde{w}} \sum_{i=1}^{n} \tilde{w}_i \log \tilde{w}_i \text{ s.t. } Aw = b \quad (5.5)
\]

Fortunately, the resulting non-linear optimisation problem with linear constraints has the very desirable property of being convex. To proof this, first note that the space of feasible solutions \( S \) of this problem which is defined as the set of all non-negative solutions \( w \) of \( Aw = b \) is convex, since for all \( x, y \in S \) and \( \alpha \in [0, 1] \): \( \alpha x + (1 - \alpha) y \geq 0 \) and

\[
A(\alpha x + (1 - \alpha) y) = \alpha Ax + (1 - \alpha) Ay = \alpha b + (1 - \alpha) b = b
\]

and consequently \( (\alpha x + (1 - \alpha) y) \in S \). Furthermore, \( f : \mathbb{R}^n_{\geq 0} \rightarrow \mathbb{R}, x \mapsto \sum_{i=1}^{n} x_i \log x_i \) is also strictly convex on its domain and especially convex on \( S \) because

\[
f(\alpha x + (1 - \alpha) y) = \alpha \sum_{i=1}^{n} x_i \log(\alpha x_i + (1 - \alpha) y_i) + (1 - \alpha) \sum_{i=1}^{n} y_i \log(\alpha x_i + (1 - \alpha) y_i)
\]

\[
< \alpha \sum_{i=1}^{n} x_i \log(\alpha x_i) + (1 - \alpha) \sum_{i=1}^{n} y_i \log((1 - \alpha) y_i) < \alpha f(x) + (1 - \alpha) f(y)
\]

where the last two step holds, since \( \log(x) \) is monotonically increasing and \( x, y \geq 0, \alpha \in [0, 1] \). Since \( f \) and \( S \) are convex, \((5.5)\) is convex. As a result, every local minimum of \( f \) in \( S \) is also a global minimum [40]. Consequently, to solve \((5.5)\), it is our goal to find a local minimum of \( f \) in \( S \). To find this minimum, it is sufficient to use an iterative method because the local optimality of this point is sufficient. As a result, we will use an iterative method to solve this non-linear optimisation problem with linear constraints. In the following, two methods aiming to find the solution of \((5.5)\) by finding a local minimum of \( f \) in \( S \) will be discussed.

5.2.1. Infeasible Newton method for entropic quadrature

In this section, a first very intuitive method to solve \((5.5)\) will be shortly discussed. Since there exists no analytical solution to this problem, we will use an iterative Newton method to calculate the solution of \((5.5)\). An iterative Newton algorithm calculates a sequence of points \( w^0, ..., w^n \in \text{dom} f_0 \) with the aim that this sequence always convergences to the solution of \((5.5)\) provided its existence. In principle, to solve such a constrained optimisation problem, there are two different versions of the Newton method available:
the infeasible and feasible Newton method. Both methods differ in the constraints put on each \( w^{(i)} \). In the feasible method, we will always assume that \( A w^{(i)} = b \), which posses a challenging problem because we need to find a start point where this is also true. In general, for functions where the domain is not restricted, this can be quickly done by solving \( A w = b \) directly. Unfortunately, in our case, the start point has to satisfy additionally \( w^{(i)} \in \text{dom} f_0 = \mathbb{R}_{>0} \), and a priori we neither know a solution of this system nor whether it exists. One possibility to overcome this problem is to use Linear Programming to calculate a feasible starting point, but since we want to have a very fast method and have a promising initial guess from the PD-weights, we will not consider this possibility here. Hence, we choose to use the infeasible Newton method which does not enforce \( A w^{(i)} = b \) and reduces to the feasible Newton method after \( A w^{(i)} = b \) for some \( i \) [13].

The infeasible Newton method solves (5.5) by replacing the entropy function by its second order Taylor series at the current point of evaluation \( w^{(i)} \) and subsequently minimizing this function under the constraint \( A w^{(i+1)} = b \). At each step, we want to find a \( x \) so that \( w^{(i)} + x \) is optimal in the sense that it is the solution of the problem when the entropy is replaced with the second-order Taylor series [13]. We can hope that using the second order Taylor expansion as a local approximation is quite effective because \( x \log x \) can be locally quite precisely mimicked by said function.

Aiming to calculate \( w_k^{(i+1)} = w_k^{(i)} + x_k \) following the above reasoning, we simplify the entropy functional by calculating the value at \( w^{(i+1)} \) of the second order Taylor expansion at \( w^{(i)} \):

\[
\sum_{k=1}^{N} w_k^{(i+1)} \log(w_k^{(i+1)}) = \sum_{k=1}^{N} (w_k^{(i)} + x_k) \log(w_k^{(i)} + x_k)
\]

\[
\approx \sum_{k=1}^{N} (w_k^{(i)} \log(w_k^{(i)})) + (1 + \log(w_k^{(i)}))x_k + \frac{1}{2w_k^{(i)}} x_k^2
\]

Consequently, we replace (5.5) by

\[
\min_{c} \sum_{k=1}^{N} (w_k^{(i)} \log(w_k^{(i)})) + \sum_{k=1}^{N} (1 + \log(w_k^{(i)}))x_k + \sum_{k=1}^{N} \frac{1}{2w_k^{(i)}} x_k^2 \quad \text{s.t.} \quad A w^{(i+1)} = A(w^{(i)} + x) = b
\]

(5.6)

where \( c \in \mathbb{R} \), \( x \in \mathbb{R}^n \), \( p \in \mathbb{R}^n \), \( b \in \mathbb{R}^m \), \( A \in \mathbb{R}^{m \times n} \), \( P \in \mathbb{R}^{n \times n} \) with \( m < n \) and \( \text{rank}(A) = m \). In appendix B a detailed explanation of how to solve (5.6) will be given, and the whole method will be analysed in further detail. However, it turns out that this very intuitive method has some considerable drawbacks in practice. Therefore, for now, only a summary of the essential properties of this method will be given.

First of all, assuming that the problem is feasible, e.g. there exists a positive \( w \) with \( A w = b \), it can be proven that the method outlined in B.2 always converges to the solution of (5.5) [13]. Unfortunately, despite having this desirable property, the method also has three considerable drawbacks: The first one is due to the fact that \( w^{(i)} \) always has to be positive because the second derivative becomes infinity otherwise or \( \log x \) is no longer defined, and the method breaks down. In principle, however, we would like to allow for all non-negative \( w \) including zero weights. This is vital for the reliability and flexibility of the infeasible Newton method. Furthermore, this is, in general, possible by setting \( 0 \log 0 := 0 \) which is equal to its right limit to ensure that \( \text{dom} f = \mathbb{R}_{\geq 0}^n \).
Another problem of the infeasible Newton method is also due to the restricted domain of \( f \). At each step, \( w^{(i+1)} \) has to be calculated in a way that it is always positive. Consequently, it is sometimes not possible to set \( w^{(i+1)} = w^{(i)} + x \) because \( w^{(i)} + x \) is not positive. As a result, as outlined in section B an update factor has to be calculated. This restriction on the updates of \( w^{(i)} \) causes the problem that the method can sometimes converge only very slowly because only very small steps can be taken to ensure that \( w \) remains positive. This causes, for example, big problems if the initial weights are very small. In this case, the method sometimes progresses only very slowly because if \( x \) is negative for the already small weights, only tiny steps can be taken. For some cases, one can observe the informative pattern that the solution is stuck with one very small component. In said cases, many tiny steps are taken because the update direction for this component is negative before the update direction becomes positive and a big step is taken usually leading to nearly immediate convergence. Aiming to find a fast and reliable method to calculate the EQ of a set of moments, the slow convergence is a serious disadvantage of this method.

The third problem of this method is that the behaviour of the method in the case that (5.5) is infeasible cannot be predicted. Consequently, during the calculation of a solution, it is not clear whether the problem has no solution or the method converges only very slowly to the local minimum.

Out of these three reasons, we will present an alternative way of solving (5.5) in the next section, which is in practice much more reliable and stable.

5.2.2. Calculating entropic quadratures using Lagrange duality

In the following, an alternative way to solve (5.5) using the Lagrange dual function will be introduced. To do so, some theory about convex optimisation problems and duality is necessary and will be presented following [13]. The general problem of convex optimisation is that given a convex function \( f_0(x) \) and \( A \in \mathbb{R}^{m \times n} \), \( b \in \mathbb{R}^n \) and \( x \in \mathbb{R}^n \) we want to solve the following problem:

\[
\min f_0(x) \text{ s.t. } A x = b. \tag{5.7}
\]

(5.7) is called feasible if there exists a \( x \in \text{dom} f_0 \) with \( A x = b \). The Lagrange function of (5.7) is:

\[
\mathcal{L}(\lambda, x) = f_0(x) + \lambda (A x - b)
\]

with \( \lambda \in \mathbb{R}^m \). In this approach, we will not use this function directly because the logarithm function included in the entropy functional leads to problems in the optimisation process as discussed in the previous section. Instead, we will use the Lagrange dual function:

\[
g(\lambda) = \inf_x \mathcal{L}(\lambda, x) = \inf_x f_0(x) + \lambda (A x - b) = -b^T \lambda - f_0^*(-A^T \lambda),
\]

where \( f_0^*(y) = \sup_x (y^T x - f_0(x)) \) denotes the conjugate of \( f_0(x) \). This function gives a on \( \lambda \) dependent parametrised lower bound on the Lagrange function. Furthermore, with \( p^* \) being the solution of (5.7), it is obvious that:

\[
g(\lambda) \leq p^*, \quad \forall \lambda
\]

From this start point, it is natural to search for the best lower bound that we can obtain using this technique. This bound can be obtained by solving the Lagrange dual problem:

\[
d^* = \max_{\lambda} g(\lambda) \tag{5.8}
\]
We will refer to $\lambda^*$ as the argmax of $g(\lambda)$ in (5.8) and to $p^* - d^*$ as the optimal duality gap which is always non-negative and describes the difference between the best lower bound we can obtain by using the Lagrange dual function and the solution of (5.7). One important special case whose importance will become apparent in the following is when the duality gap is zero. In this case, the bound given by $d^*$ is tight and we say that strong duality holds. Since this property is very important, different sufficient conditions called constraint qualifications for it have been established. One of these conditions is called Slater’s conditions which applied to (5.7) says that strong duality holds in the case that a feasible solution of the problem exists. As a result, if we assume that the problem is feasible and that we have already calculated the solution $\lambda^*$ of (5.8), we will know that $g(\lambda^*)$ is equal to the solution of 5.7. From the definition of the Lagrange dual function $g$ we know further that there has to exist a $x^*$ with $g(\lambda^*) = \mathcal{L}(\lambda^*, x^*)$. It can be shown that under these conditions $x^*$ is always feasible and consequently that $\mathcal{L}(\lambda^*, x^*) = f_0(x^*)$ holds [13]. It follows that $x^*$ is the solution of (5.7).

In summary, we can solve (5.7) in two steps: First by calculating $\lambda^*$ from (5.8) and subsequently by solving

$$x^* = \arg\min_x \mathcal{L}(\lambda^*, x).$$

(5.9)

In the case that the solution of this equation $x^*$ is feasible, we know that the problem is feasible and as a result that the duality gap is zero and that $x^*$ is the minimiser of (5.7). However, in the case that $x^*$ is not feasible, we know that the whole problem is infeasible.

We will now apply the presented theory to (5.5) to calculate the entropic weights. The Lagrange function of this problem is:

$$\mathcal{L}(\lambda, w) = \sum_{i=1}^{n} w_i \log w_i + \lambda(Aw - b).$$

(5.10)

An additional important insight is that the conjugate of the entropy function is: $f_0^*(y) = \sum_{i=1}^{n} e^{y_i - 1}$. Consequently, the Lagrange dual problem of (5.10) put in a slightly different form than (5.8) is:

$$\lambda^* = \arg\min_\lambda b^T \lambda + \sum_{i=1}^{n} e^{-a_i^T \lambda - 1}$$

(5.11)

where $a_i$ denotes the ith column of $A$. Fortunately, the function which needs to be optimised in (5.11) is concave. Therefore, it is sufficient to search for a local minimum of this function to calculate $\lambda^*$. Assuming that we have found $\lambda^*$, the remaining part is to determine the $w^*$ for which $g(\lambda^*) = \mathcal{L}(\lambda^*, w^*)$ by solving (5.9):

$$w = \min_w \mathcal{L}(\lambda^*, w)$$

Fortunately, $\mathcal{L}(\lambda^*, w)$ is again a convex function, and it is sufficient to solve $\frac{\partial \mathcal{L}(\lambda^*, w)}{\partial w_i} = 0$ which can be done analytically and results in:

$$w_i = e^{-1 - a_i^T \lambda^*}$$

(5.12)

Consequently, to solve (5.5), the only part that cannot be calculated analytically is solving the unconstrained convex optimisation problem (5.8). To solve this problem, we use the multidimensional Newton method, since this method has nice theoretical properties, e.g. its convergence is locally quadratic [13]. Furthermore, the multidimensional Newton method can be justified in different ways, e.g. as minimising the second order Taylor
series at the current point or as taking a step into the descent direction regarding the Hessian norm. The described method which calculates the entropic weights and thereby solves (5.5) can be therefore summarised as:

**Algorithm 5: Solving the primal problem (5.5) via the Lagrange dual**

**Data:** Start point $\lambda^{(0)}$, tolerance $\epsilon > 0$, $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$

1. while True $\triangleright$ multidimensional Newton problem
2. do
3. Compute the newton step as: $\Delta \lambda = -H_g(\lambda^{(n)})^{-1}\nabla g(\lambda^{(n)})$ and the decrementation as $\mu^2 = \nabla g(\lambda^{(n)})^T \Delta \lambda = \nabla g(\lambda^{(n)})^T H_g(\lambda^{(n)})^{-1}\nabla g(\lambda^{(n)})$;
4. if $\frac{\mu^2}{2} \leq \epsilon$ then
5. break;
6. end
7. Determine $\eta$ using backtracking line search as in algorithm 6;
8. $\lambda^{(n+1)} = \lambda^{(n)} + \eta \Delta \lambda$;
9. end
10. $w_i = e^{-1 - a_i^T \lambda}$;
11. if $Aw \approx b$ then
12. return $w$
13. else
14. return "Problem not feasible"
15. end

Note that we will not enforce that the equality $Aw = b$ holds exactly out of reasons that will become apparent later and because doing so would require a very costly exact convergence to the solution. To obtain the initial guess $\lambda^{(0)}$, we will use our initial guesses of $w$ by solving: $\log w + 1 = -\tilde{A}^T \lambda$ where $\tilde{A}$ consists of a suitable subset of columns of $A$. The required derivatives used in algorithm 5 to solve (5.11) are:

$$
\nabla g(\lambda^{(n)})_k = b_k - \sum_{i=1}^{n} a_{ki} e^{-1 - a_i^T \lambda^{(n)}} = b_k - \sum_{i=1}^{n} a_{ki} w_i^{(n)},
$$

$$
\frac{\partial^2 g(\lambda^{(n)})}{\partial \lambda_k \partial \lambda_j} = \sum_{i=1}^{n} a_{ji} a_{ki} e^{-1 - a_i^T \lambda^{(n)}} = \sum_{i=1}^{n} a_{ji} w_i^{(n)} a_{ki}
$$

Consequently, we can write:

$$
H_g(\lambda^{(n)}) = AWA^T, \text{ where } W = \text{diag}(w_1, ..., w_n)
$$

with $A \in \mathbb{R}^{m \times n}$ and $W \in \mathbb{R}^{n \times n}$ with $n > m$. Since $H_g(\lambda^{(n)})$ is symmetric positive definite, each Newton step requires only the inversion of one matrix $H_g(\lambda^{(n)})$ using a Cholesky decomposition.

One of the advantages of the outlined method is that its behaviour in the case that the problem is infeasible is well-defined. Since the weights returned by the algorithm are always in the form of (5.12), $\nabla g$ is zero and therefore a local minimum is reached if and only if the corresponding set of weights in the form of (5.12) satisfies $Aw = b$. Consequently, in the case that the algorithm returns a $w$ which does not satisfy $Aw = b$ we know that no local minimum is and can be reached. As no local minimum exists, the convex function has to be minimal when $|\lambda|$ goes to infinity. As a result, the returned minimum cannot be feasible, and therefore the whole problem has to be infeasible because
strong duality cannot hold. This reasoning is the justification for the lines 11-15 of algorithm 5. Using the explained properties of algorithm 5, the ultimate goal of this algorithm is to find a set of weights in the form of (5.12) satisfying $Aw = b$.

5.2.3. Properties of optimising via the Lagrange dual

Comparison between both methods  The presented method has several advantages over the infeasible Newton method outlined earlier. First of all, instead of a constrained optimisation problem only the unconstrained optimisation problem (5.11) needs to be solved numerically. The theoretical properties of such an unconstrained optimisation problem are much more desirable than in the constrained case because it can be proven that at each step the value of $g(\lambda^{(n)})$ decreases, whereas in the constrained case only an improvement of $r(w, \lambda^{(n)})$ (see section B.2 for an explanation) can be shown. Furthermore, the unconstrained optimisation problem is formulated in only $m$ variables, whereas the constrained optimisation problem is formulated in $n$ variables. Additionally, by replacing $\log(x)$ with its conjugate function $e^x$ the problem of the restricted domain has been overcome and the solution may no longer get stuck eventually.

Moreover, in the case that the problem is infeasible the behaviour of the method is well-defined, and from its output one can directly conclude the infeasibility of the problem. Concerning the computational efficiency of both approaches, it is interesting to note that one step of one of the two algorithms presented requires the solution of a positive definite matrix of the same size and structure.

Moreover, the solution via the Lagrange dual function has the desirable property that there exists a model distribution of the weights which enables us to show a few properties of the presented method. However, since the model distribution is still always positive, zero weights are still not allowed. Nevertheless, small weights do not longer pose problems, such as a limited step size or a very large second derivative as in the infeasible Newton method, to the algorithm, since the domain is no longer restricted and the algorithm is formulated in terms of the Lagrange multipliers instead of the weights.

Convergence of algorithm 5  In addition, the algorithm has the following very desirable property:

**Theorem 1.** Provided that there exists a set of positive weights $w$ with $Aw = b$, algorithm 5 will always converge to a set of positive weights that maximises the entropy while fitting all of the moments given.

**Proof.** The first assumption guarantees that (5.5) is feasible. Consequently, Slater’s condition holds and the duality gap is zero. It follows that the convex function (5.11) always has a local minimum where the gradient is zero. At this point, the weights which are in the form of (5.12) fit the given set of moments while maximising the entropy function. Therefore, they can be interpreted as the maximum-entropy weights. The only part that remains is to show that the first part of algorithm 5 always finds said local minimum. Fortunately, the convergence behaviour of the used multidimensional Newton method is well-studied. From the positive definiteness of $H_g(\lambda)$ it follows that $\nabla g(\lambda)^T \eta \Delta \lambda = \eta \nabla g(\lambda)^T H_g(\lambda)^{-1} \nabla g(\lambda) > 0$ unless the algorithm reaches the local minima where $\nabla g(\lambda) = 0$. Therefore, since (5.11) always have a local minimum corresponding to its global minimum and at each Newton step the value of $g(\lambda)$ decreases, the algorithm always converges to the local minimum. The used Newton method has additionally the very desirable property that it is locally of quadratic convergence [13]. All in all, algorithm 5 always returns the maximum-entropy quadrature provided its existence.

□
Dealing with the realisability of algorithm 5 in one dimension

One of the main problems of entropic quadrature is that it is unclear in which cases the feasibility assumption made in theorem 1 holds. In one dimension, however, due to the fact that it can be shown that for a realisable set of moments the Wheeler algorithm always produces a set of abscissas $\gamma$ and weights $w$ so that the $A_w = b$ [103] [61] there is hope that the feasibility assumption is always fulfilled. In principle, since we extended the entropy function to zero and usually $\gamma \subset \gamma^{EQ}$ and therefore $A_{\gamma^{EQ}} w = b$ with

$$\tilde{w}_i = \begin{cases} w_i, & \exists j : \gamma_i^{EQ} = \gamma_j \\ 0, & \text{otherwise,} \end{cases}$$

the problem (5.5) is in one dimension always feasible. Unfortunately, to prove strong duality, Slater’s condition requires that the feasible solution to the convex optimisation problem in question has to lie in the insight of its domain [13]. Unfortunately, $\tilde{w}$ lies on the boundary of the domain of $f$, since $w$ contains some zero entries. Furthermore, $w$ is also not representable by the model distribution of the weights as $e^x \neq 0$, $\forall x$. Nevertheless, this restriction is of a very theoretical nature. In practice, since we do not enforce that $A w = b$ has to hold exactly and require instead that $\| A w - b \| \leq \epsilon$ for some $\epsilon$, we actually consider a perturbed system. For this system, the following set of weights is always a feasible solution:

$$\tilde{w}_i = \begin{cases} w_i, & \exists j : \gamma_i^{EQ} = \gamma_j \\ \frac{\epsilon}{n^2 \alpha}, & \text{otherwise} \end{cases}$$

with $\alpha = \max |A_{ij}|$. Consequently, in this slightly perturbed system, Slater’s condition and strong duality holds. Continuing practical considerations, to proof Slater’s condition in practice and ensure that the resulting weights agree with the model distribution, $\frac{\epsilon}{n^2 \alpha}$ had to be above zero. Therefore, we have to choose $\epsilon$ so that $\frac{\epsilon}{n^2 \alpha}$ is at a minimum the smallest double above 0 representable by the system. Assuming that we can consider a 12-moment system with the highest abscissas below 100 and a standard system with the smallest representable double around $1 \times 10^{-300}$, $\epsilon$ has to be above $12^2 100 111 \times 10^{-300} \approx 1 \times 10^{-276}$, which is usually the smallest double over $1 \times 10^{-300}$ and still a far higher than necessary.

The hint of the above reasoning towards the insight that in practice Slater’s condition holds and that algorithm 5 consequently always returns a fitting EQ quadrature in one dimension provided that the considered set of moments is realisable is supported by the fact that, in our experiments, 5 has always returned a fitting quadratures in the more than 5000 considered sets of realisable moments. Note further that the returned quadrature was in all cases not in the form of (5.13) and usually fits the given set of moments very precisely. As a result, keeping in mind that the calculated solution may fit the given moments not up to machine precision, in the following, we will use the fact that, in practice, EQ always returns a fitting quadrature on a realisable set of moments. In the case that all these strong hints in practice are not sufficient, it is also possible to enforce

\[ \text{Note that such theoretical statements are always limited by the numerics used in the process. However,} \]

\[ \text{the Wheeler algorithm has proven to be stable and reliable for arbitrary realisable moments up to a} \]

\[ \text{quite high order.} \]

\[ \text{Due to the finite precision of every computer, this is usually not even possible.} \]

\[ \text{This would usually also not be the case if a perfectly fitting set of weights exist because then the} \]

\[ \text{termination criterion in algorithm 5 has to be very restrictive seriously increasing the computation} \]

\[ \text{time while gaining nothing for our purposes.} \]
this condition theoretically by modifying 5 in the following way: In one dimension, in the

case that the calculated quadrature does not fit the given set of moments in the desired

way return:

\[ \tilde{w}_i = \begin{cases} w_i, & \exists j : \gamma_i^{EQ} = \gamma_j \\ 0, & \text{otherwise} \end{cases} \]

Out of all these reasons, keeping the remarks made in mind, we will consider EQ as

fulfilling the full realizability condition in one dimension.

**Convergence of algorithm 5 to the maximum-entropy distributions**

The model distribution of the weights (5.12) establishes a strong relationship between the calculation of

the maximum-entropy closure and entropic quadrature. Note that the model distribution

for the weights can be written as

\[ w_i = e^{-1 - \Phi(\gamma_i)^T \lambda} \]

which is equivalent to \( f^{(ME)} \) since \( \Phi_1(\gamma_i) = 1 \) and \( \lambda_1 \) can be combined with the \(-1\) as it is done in the derivation of \( f^{(ME)} \).

Not only are the model distributions of both approaches similar but also the derivation of

both using the Lagrange duality function and the resulting minimisation problem which

needs to be solved ((5.11) and (4.7)). The only difference between the two is that in (5.11)

the numerical integration is explicitly written out at the quadrature points, while in (4.7)

a costly numerical integration usually performed using a fixed quadrature, for example, a

blockwise Gauss-Legendre quadrature, needs to be performed [81]. Consequently, entropic

quadrature can indeed be seen as a sparse way of the maximum-entropy closure where

instead of trying to calculate a continuous distribution, a quadrature-based representation

of the function at some points is calculated. To note the strong connection between the

two and the fact that, in the limit, EQ reduces to ME, first note that the optimisation

criterion to find \( f^{(ME)} \) (4.7) can be rewritten to:

\[ \lambda^* = \arg \min_{\lambda} b^T \lambda + \frac{1}{n} \sum_{i=1}^{n} e^{-\Phi(c_i)^T \lambda} \]  \hspace{1cm} (5.14)

where the integration of the model distribution is replaced by the Riemann sum which is

in the limit, for example for an infinite number of equidistant quadrature points \( c_i \), exact.

In practice, however, the number of quadrature point is always finite. Note further that

it is possible to rewrite the second part of (5.14) to:

\[ \frac{1}{n} \sum_{i=1}^{n} e^{-\Phi(c_i)^T \lambda} = \sum_{i=1}^{n} e^{-\Phi(c_i)^T \lambda + \log \frac{1}{n}} = \sum_{i=1}^{n} e^{-\Phi(c_i)^T \tilde{\lambda}} \]  \hspace{1cm} (5.15)

where we used the fact that \( \Phi_1(c_i) = 1 \) to summarise \( \log \frac{1}{n} \) and \( \lambda_1 \) into \( \tilde{\lambda}_1 \), while the

other components of \( \lambda \) remain unchanged. From this form of the numerical integration

and the maximum-entropy distribution, one can immediately recover the close relationship

to (5.11). As a result, if we assume that the EQ-abscissas are uniformly distributed in the

domain of interest of the function, then the distribution for the EQ-weights will converge,

up to a constant scaling factor, to the original maximum entropy distribution as the

number of nodes increases. The constant difference between the two is due to the fact

that in the numerical quadrature the points are weighted, whereas in entropic quadrature

this is no longer the case.

As a result of the insight above, it is also possible to use EQ as a dense and exact

reconstruction technique for maximum entropy distributions. However, to achieve this

convergence of EQ to ME if the number of abscissas is increased, a uniform distribution

of the abscissas is necessary. To understand this assumption, it is important to note that
the integration of \( f^{(EQ)} \) can only be interpreted as an unweighted numerical integration using the midpoint rule if the abscissas are uniformly distributed. Only in this case, the entropy of the weights is maximised if they correspond to a scaled version of the values of the maximum entropy distribution at this point. With non-uniform weights, this technique does not lead to a fit with the given moments. Furthermore, to achieve this convergence property it is not only vital to ensure that the abscissas are uniformly distributed but also that they are distributed in a large domain and not only closely around the PD-quadrature.

5.3. Properties of Entropic Quadrature

The properties of EQ depend crucially on the dimension and the specific algorithms considered. In the following, we will assume that the weights are found using the Lagrange dual function, and the abscissas are determined using the default algorithm 2.

First of all, the non-negativity of the reconstruction (e.g. all weights have to be non-negative) can always be guaranteed. This can be directly concluded from the model distribution of the weights.

Concerning the realisability property of this closure, the situation is rather unclear. As argued above, in one dimension, it can be assumed that EQ produces a fitting quadrature on all realisable moments. Concerning multiple dimensions, however, the calculation of a fitting quadrature cannot always be guaranteed because there does not always exist a set of non-negative weights of the predetermined abscissas fitting the given set of moments. However, as discussed above, in the case that a non-negative set of weights for the calculated abscissas exists, EQ will always find a fitting set of weights. The general problems of EQ in multiple dimensions are not surprising because the QMOM for multiple dimensions does not even aim to fit the full moment tensor as it is done in EQ and even for the reduced moment tensor cannot guarantee a non-negative solution. In contrast to the QMOM, however, there exist several ideas on how to deal with this problem for EQ which will be presented in section 5.4.

Since EQ considers all full moment tensors in the form of \( \Phi_{i,j,k}^{2N-1} \), Galilean invariance is always guaranteed, and the method offers a flexible framework of approximations of the Boltzmann equation. As introduced in the context of the QMOM, it is also possible to consider tensors of an even degree. This can be realised by always considering the centred moments and using only the first 2\( N \) moments as an input of the PD-algorithm and subsequently insert an additional node at \( x = 0 \) as part of the PD-quadrature. EQ is fully capable of following this strategy, since the weights are not found using a linear system of constraints.

As in the QMOM, the hyperbolicity of the method cannot be rigorously proven. However, similar to the QMOM, our intuition and practical experiences suggest that the chances that EQ is, in practice, hyperbolic are very high. This insight is additionally supported by the fact that the simulations conducted using EQ behave all quite stable. As in the QMOM, we will use the maximal absolute value of the abscissas as the characteristic speed \( \lambda \) of the system.

For now, we will postpone the capabilities of this closure for a while and devote section 7 to examine the ability of this closure to describe flows in different regimes.

5.3.1. Relationship to the QMOM and maximum-entropy closure

The method of EQ has already been extensively justified as a combination of the QMOM and the maximum-entropy closure. In addition, in the following, the properties of EQ
and these closures will be shortly compared.

**EQ and maximum-entropy closure**  EQ and the maximum entropy-closure are quite closely related. First of all, they both utilise the same justification and aim to maximise the entropy of the reconstruction under some constraints. As a result of this close relation, as argued in section 5.2.3, a direct connection using the model distribution of the weights can be established between both approaches. This connection gives rise to the interpretation of entropic quadrature as a sparse version of the maximum-entropy closure. However, one of the fundamental differences between both methods is that in EQ instead of the coefficients of the maximum-entropy distribution only the weights of the abscissas are calculated. The formal properties of both methods are sometimes similar but also due to the sparseness and adaptiveness of EQ sometimes very different. Since both approaches share the same model distribution, both methods share the non-negativity of the solution. Concerning realisability, the picture is somehow different. In one dimension, as mentioned earlier, EQ usually produces a fitting quadrature on all realisable sets of moments, whereas the maximum-entropy closure is realisable for all sets of realisable moments despite a small subset of moments as discussed in section 4.4.6. In multiple dimensions, the general realisability property of EQ is lost, while it can be proven that the maximum-entropy closure is still able to construct a fitting reconstruction for nearly all sets of realisable moments. Moreover, the maximum-entropy closure has the beneficial property that the hyperbolicity of this closure can be proven, whereas this is impossible for EQ due to the adaptive abscissas. Nevertheless, EQ has the crucial advantage over the maximum-entropy closure that the optimisation process is much faster, since the quadrature is very sparse and no computationally expensive numerical integration technique for the calculation of parameters of the model distribution or the calculation of the higher order moments needed needs to be performed.

Last but not least, in practice, it is also possible to interpret most of the implementations of the maximum entropy closure as a form of entropic quadrature. In said implementations, to integrate the model distribution usually a fixed quadrature is predefined. Let us, for the sake of simplicity, consider the one-dimensional case and assume that the predefined numerical quadrature uses uniform weights. The quadrature-nodes $v_1,\ldots,v_n$ can than be interpreted as the calculated abscissas. Since the numerical integration of the maximum-entropy function can be rewritten to a sum of values of the scaled model distribution of the weights as it is done in (5.15), the calculated maximum-entropy distribution can be interpreted as the model distribution of the weights of the predefined nodes. This interpretation is supported by the fact that usually the same abscissas are used to integrate the maximum-entropy distribution for the reconstruction of the higher order flux. Hence, in practice, the usage of the maximum entropy distribution reduces to:

$$
\langle c^j e^{\Phi(c)^T\lambda} \rangle = \frac{1}{n} \sum_{i=1}^{n} v_i^j e^{\Phi(v_i)^T\lambda} = \sum_{i=1}^{n} v_i^j e^{-\Phi(v_i)^T\lambda}
$$

which is equivalent to using the predefined abscissas as EQ-nodes and the reconstructed maximum-entropy distribution as the model distribution of the weights.

The only difference between EQ and this numerical version of ME is that the underlying model distributions are different. In EQ an unweighed quadrature is chosen, whereas the numerical quadrature used in the maximum-entropy closure is weighted aiming to interpolate the values of the maximum-entropy distribution exactly. Following the above reasoning, EQ, as it was introduced in this thesis, can be seen as advantageous over the numerical maximum entropy closure because they both follow the same scheme. However,
EQ uses an adaptive method to determine the quadrature points based on the available knowledge about the distribution function (its moments), whereas ME usually uses a predefined quadrature. As a result, because of its adaptiveness EQ can achieve a considerable higher precision than ME at equivalent computational cost.

**EQ and the QMOM**  
EQ and the QMOM have already been compared in section 5.1 to some extent. Aiming to simulate flows at different regimes, the EQ can be seen as somehow beneficial because this closure introduces some degree of freedom in the QMOM which allows to increase the entropy of the solution and adds a strong justification concerning the optimality of the closure to the QMOM. Apart from the application, the capabilities, and the justification, the properties of both methods are sometimes quite similar because the general procedure of both methods is to some extend closely related. Both closures share the quadrature-based reconstruction and aim to calculate an adaptive quadrature of the underlying distribution function based on the moments given. In the one dimensional case, EQ can sometimes nearly reduce to the QMOM in the absence of any other solution with a higher entropy. Concerning formal properties, both methods share that hyperbolicity cannot be proven and that the abscissas can be interpreted as the eigenvalues of the system.

One of the desirable properties of EQ is that this closure is much more flexible than the QMOM in higher dimensions because general full moment tensors can be considered. Thereby, EQ overcomes the problem of considering non-symmetric and reduced systems as it is done in the QMOM. As a result, the reconstruction obtained from EQ may be seen as advantageous because the set of moments considered has no longer to be restricted in such an artificial way as it is done in the QMOM. Concerning the realizability property of the closures as they both share the property of not being able to guarantee a non-negative reconstruction fitting the considered set of moments in the general multi-dimensional case (e.g. the full moment tensor for EQ and the subset of the full moment tensor for the QMOM), nothing is achieved by these strong restrictions of the QMOM. However, there is a difference between both methods concerning their realizability problems in multiple dimensions. For the QMOM, the computed weights can become sometimes negative and due to the fact that the weights are obtained by solving a simple linear system of constraints with a unique solution the only possibility to deal with a negative weight is to set it to zero and solve the linear system of a subset of constraints in the next step. In comparison, EQ always calculates a non-negative solution. However, in the case that no set of non-negative weights for the abscissas exists no solution at all is obtained by the EQ algorithm. Nevertheless, because of its adaptivity and degree of freedom, the abilities of EQ to deal with the problem of lacking realizability are manifold as discussed in the next section.

Finally, a comparison of the capabilities of both closures to model flow problems will be conducted in 7.

**5.4. Dealing with non-realisable moments and problems of EQ in multiple dimensions**

As mentioned above, one of the critical problems of EQ is that, in multiple dimensions, the existence of a vector of non-negative weights for the previously calculated abscissas fitting the given set of moments and therefore a solution of the EQ-algorithm can no longer be guaranteed. Despite appearing only seldom in most of the conducted experiments, a method to address this problem is necessary. In the following section, different opportunities on how to deal with this problem are presented.
Interestingly, most of the strategies presented can also be used to address a general problem of most of the closures of the Boltzmann equation: Usually, all of these closures fail in the case that the given set of moments is no longer realisable. Since the aim of all closures should be to construct a non-negative distribution function fitting the given moments, this property is, generally speaking, not a problem because a non-realisable set of moments can obviously not be fitted by a distribution function. However, we often cannot assume that all moments always remain realisable. Due to noise in the data, for example, inserted by higher order spatial discretisation for which a rigorous proof of realisability-preserving is no longer possible or small errors in the computation of the numerical flux, the moments can become unrealisable. To understand how easily this can happen, consider the moments of the normal distribution with $\mu = 2$ and $\sigma^2 = 0.1$ up to order 8:

$$
\rho_0 = 1, \; \rho_1 = 2, \; \rho_2 = 4.1, \; \rho_3 = 8.6, \; \rho_4 = 18.43, \\
\rho_5 = 40.3, \; \rho_6 = 89.815, \; \rho_7 = 203.81, \; \rho_8 = 470.491.
$$

The determinate of the corresponding Hankel matrix is $2.87914 \times 10^{-8}$ proofing realisability of the calculated set of moments. However, if we perturb the system by adding $1 \times 10^{-5}$ to $\rho_3$, the determinant becomes $-1.77052 \times 10^{-7}$ and consequently, the perturbed set of moments is no longer realisable. As a relative change of $1.16 \times 10^{-6}$ of only one component can make the considered system of moments non-realisable, the problem of non-realisable moments should be taken seriously and needs to be addressed. Otherwise, simulations conducted using a closure that cannot handle non-realisable moments will immediately fail once a non-realisable set of moments is produced because the flux is no longer well-defined. Despite being examined by a few authors (see [4] and reference therein), the work conducted on this problem is rather limited.

In the following, different techniques aiming to tackle one of the two or both problems presented above will be introduced.

**Inserting additional nodes** A first very intuitive method aiming to address the absence of a positive set of weights fitting the given moments in multi-dimensions is to increase the number of nodes in order to increase the chance of its existence. Due to the adaptivity of the quadrature, the number of nodes can be easily increased by increasing the multiplicative factor $M$ in the construction of the abscissas. In some cases, it is additionally useful to use the uniform quadrature algorithm 3 instead of algorithm 2 to calculate the nodes. This is sometimes necessary because the range of the PD-quadrature can be rather limited thereby preventing the existence of a fitting set of weights. The strategy of considering a higher number of nodes calculated with algorithm 3 is, in theory, always successful because as discussed in section 5.2.3, in the limit, the maximum entropy distribution is a solution to this problem. However, inserting a very high number of nodes is not very desirable because it increases the computational cost of the method thereby losing the main advantage over the maximum-entropy closure. In practice, inserting additional nodes produces very good results even for moderate $M$ and can be, therefore, considered as a suitable first approach to tackle the problem of infeasibility for the multidimensional the EQ-algorithm (5.5).

**Replacing the maximum entropy objective function** The following three ideas can be used to address either of this the problems presented above: One interesting opportunity which can be used to overcome both problems is to replace the maximum entropy objective function by an objective function also allowing for negative weights. Thereby, unless the number of moments is above the number of weights, there always exists a set of weights possibly including some negative components fitting the given set of moments regardless of
the dimensionality of the problem or whether the given set of moments is realisable or not. Following this approach, both problems can be easily overcome without adapting the EQ method too much. Possible candidates for replacing the maximum-entropy distribution $x \log x$ are the L2 norm $x^2$ and the L1 norm $|x|$. As discussed earlier in section 4.4.2, independent from the application considered here, $x^2$ is always a candidate to replace $x \log x$ as they both aim to increase the uniformity of the reconstructed distribution as much as possible and $x^2$ satisfies at least some of Shannon’s requirements on a measure for uncertainty. Replacing the entropy by $|x|$ is also possible following the well-known concept of compressed sensing. Using this criterion, the number of nodes with a non-zero weight will be minimised. Despite always allowing for a feasible solution of the problem, replacing the maximum-entropy distribution by another distribution should only be done with care, since it introduces negative weights into the system. Introducing partly negative distribution functions may result in non-physical behaviour such as a negative density. In the case that the described approach is used, it is vital to use a stable algorithm and especially not a kinetic scheme for the calculation of the numerical flux because the later one is susceptible to non-physical weights. Nevertheless, this approach is a very promising approach to tackle both problems and will be shortly further examined in section 7. One interesting question concerning this approach is in which cases negative weights in the solution of this modified method occur and whether those cases correspond to the non-existence of a positive solution.

Regularised maximum-entropy closure A second idea which can be used to overcome both problems has been proposed in [4]. In [4], instead of finding the maximum-entropy distribution directly using (4.7), a regularised version of the maximum-entropy condition is considered. In this regularised version the exact fitting of the given moments is no longer enforced. In said approach, the maximum-entropy constrained optimisation problem (4.5) is replaced by the following unconstrained optimisation problem:

$$f^{(RME)} = \arg \max_{\tilde{f}} \left\{ \langle \tilde{f} \log \tilde{f} \rangle - \frac{1}{2\gamma} \| \langle \Phi \tilde{f} \rangle - u \|_2^2 \right\}$$

(5.16)

where $\gamma$ determines the "cost" of deviating from the given set of moments. Note that this unconstrained optimisation problem has always a solution independent from the realisability of $u$. Provided its existence, the local minimum of (5.16) is additionally always in the form of the model distribution of the usual maximum-entropy problem. The optimisation problem that needs to be performed to determine the parameters of the model distribution changes slightly from (4.7) to:

$$\alpha(u) = \arg \min_{\tilde{\alpha}} \left\{ \langle \exp[\tilde{\alpha} \Phi(c)] \rangle - \tilde{\alpha} u + \frac{\gamma}{2} \| \tilde{\alpha} \|_2^2 \right\}$$

(5.17)

This technique has several desirable properties, for instance, it reduces to the usual maximum-entropy closure as $\gamma \to 0$. Furthermore, most of the properties of the maximum-entropy closure are also properties of this regularised version. An additional advantage is that instead of considering the full-regularised version of the maximum-entropy closure, it is also possible to enforce the exact fitting of only the first $m$ moments, for example of the mass, momentum, and energy, and allowing for deviations only in the other components. In the partly regularised version the problem remains an constrained optimisation problem with some of its constraints relaxed:

$$f^{(RME)} = \arg \max_{\tilde{f}} \left\{ \langle \tilde{f} \log \tilde{f} \rangle - \frac{1}{2\gamma} \sum_{i=m+1}^{N} (\langle \Phi_i \tilde{f} \rangle - u_i)^2 \right\} \text{ s.t. } \langle \Phi_i \tilde{f} \rangle = u_i, \ i \in \{1, \ldots, m\}$$

(5.18)
Unfortunately, one of the drawbacks of the method is that sometimes even if a fitting set of weights exists, a set with a higher entropy that does not fit the given moments exactly is chosen. However, by choosing $\gamma$ sufficiently small thereby putting a considerable penalty on deviating from the solution it can be ensured that the overall accuracy is maintained [4]. In addition, the allowed deviation for the purpose of achieving a higher entropy can also be seen as advantageous if we put a greater focus on maximising the entropy or assume that there is some noise in the given moments.

Fortunately, due to the close connection between EQ and ME, it is directly possible to apply the work of [4] presented above to EQ. Therefore, only the integration of the maximum-entropy model distribution has to be replaced. The regularised version of EQ aiming to calculate the weights is analogous to (5.16):

$$f^{(RME)} = \arg \min_w \sum_{i=1}^{n} w_i \log w_i + \frac{1}{2\gamma} ||\langle \Phi f^{(EQ)} \rangle - \mathbf{u}||^2_2$$  \hspace{1cm} (5.19)

Its solution is still always in the form of the model distribution of the weights. As for the maximum-entropy approach, the unconstrained optimisation which needs to be performed to calculate the parameters of the model distribution is only slightly changed by adding $\frac{1}{2} ||\alpha||^2_2$ to (5.11).

The regularised version of the optimisation problem used to find the weights of the EQ-quadrature can be used to tackle both problems very effectively. Given a non-realisable set of moments as input, using this technique, a realisable reconstruction whose moments are close to the given set of non-realisable moments can be computed. The "closeness" of the reconstruction depends on the parameter $\gamma$, which balances the aim to maximise the entropy as much as possible with the aim of fitting the given moments as precisely as possible. Moreover, this method is also a very promising candidate for overcoming the problems of EQ in multi-dimensions in the case that no fitting set of non-negative weights exists. In this case, the regularised version of the maximum entropy condition (5.19) allows us to calculate a set of weights which is close to the original set of moments while maximising the entropy. This strategy is very promising because both the maximum entropy condition and the moment constraints are taken into account to overcome the problem that no positive set of weights exists. Such an approximative strategy is, for example, for the QMOM not possible because the whole method is constructed in a way that the weights are uniquely determined. In the case that the returned set of weights has negative entries, the only possibility in said closure is that a moment is discarded, whereas the proposed approach considers all of the moments given. A further advantage of this method is that, in the case that we accept the small error which might be introduced even for feasible problems, it is possible to directly consider this regularised version to find the weights of EQ. This has the considerable advantage that neither is it necessary to check for the realisability of the moments in advance nor to re-run the problem after the initial problem does not converge to a solution due to one of the two problems discussed in this section. All in all, this strategy can be seen as a promising candidate to face the problem of non-realisable sets of moments and a reasonable possibility to overcome the lack of realisability of the multidimensional EQ by choosing the weights that approximate the given moments as close as possible.

**Restrict the considered set of moments** A third possible approach is to reduce the considered set of moments and to calculate a smaller quadrature from this subset of moments. This quadrature can then be used to recalculate the higher order moments which could not be fitted and the higher order moments needed to close the spatial flux. For the
sake of dealing with non-realisability, the reduction of the set of moments can be made by calculating the Hankel determinants of the set of moments and by subsequently restricting the moment tensor to the tensor of maximal order for which the Hankel determinants are still all non-negative. However, this technique is computationally very expensive. Fortunately, concerning a set of one-dimensional moments, there already exists an adaptive version of the Wheeler algorithm, which has been proposed in [107] and is also used in the implementation of this thesis, which begins by computing the maximal number of nodes for which the quadrature remains well-defined as a first step. This quadrature corresponds to the quadrature calculated from the maximal set of realisable moments, since the Wheeler algorithm produces only well-defined nodes if and only if the given set of moments is realisable.

Concerning the treatment of multidimensional sets of moments where EQ does not produce a solution, it is possible to follow the same idea and to repeat EQ on the full moment tensor of order two lower than the previous one until a fitting quadrature is found.

**Perturbed set of moments**  One technique that can be only used to address the problem of a non-realisable set of moments as an input of the EQ-algorithm is to perturb the system in a way that the moments become realisable again. Several algorithms following this approach have already been presented [66, 104]. They all aim to perturb the system as less as possible varying in their reliability and speed (see [61] for a detailed explanation and comparison of the two approaches from [66] and [104]). This approach is applicable independent of the chosen closure but can become very computationally expensive and cannot prevent the system effectively from becoming non-realisable again. It can be therefore interpreted as plugging one hole after another, and it does not overcome the problem of non-realisable sets of moments in the long term.

6. Numerical scheme

Since it is impossible to keep track of the velocity distribution or its moments at each point in space and time, additional to a closure for the problem of moments, a spatial and temporal discretisation is needed. Therefore, both the spatial and temporal domain of interest are discretised into a finite number of blocks. The remaining part is to specify how the system of partial differential equation (3.1) can be rewritten to model the interaction and development of the velocity moments of these blocks in space and time. In the following, since it is a very intuitive and stable method, first, the finite volume method with a first-order time-stepping scheme step will be presented. After that, the discontinuous Galerkin method, which is fully capable of achieving high order spatial accuracy in various situations, will be discussed. To make full use of the improved spatial discretisation, also the higher order accurate Runge-Kutta time-stepping scheme will be explained.

6.1. Finite Volume Method

As the multidimensional finite volume method proceeds analogously, we will present the one-dimensional finite volume method in this section loosely following [55] see also [55] for further information and the extension to multiple dimensions). For the sake of simplicity, we will introduce the finite volume method as a discretisation of a general system of partial differential equations:

\[ \frac{\partial}{\partial t} \mathbf{u}(\mathbf{x}, t) + \frac{\partial}{\partial x} \mathcal{F}(\mathbf{u}(\mathbf{x}, t)) = 0. \]  

(6.1)
Note that the collisionless Boltzmann equation (3.1) is an exemplification of this equation. The one-dimensional finite volume method relies on a discretisation of the spatial domain \( \Omega_x = [x_L, x_R] \) into \( N_x \) cells. For the sake of readability, we will assume that the cells have uniform length:

\[ I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \text{ for } j = 1, \ldots, N_x \]

with cell size \( \Delta x = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}} \) and \( x_{\frac{1}{2}} = x_L \) and \( x_{N_x+\frac{1}{2}} = x_R \). In the finite volume method, the value of \( u \) in each of this cells \( j \) at timestep \( n \) is approximated by a constant \( u^n_j \) which is defined as the average cell value at timestep \( n \):

\[ u^n_j = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t_n) \, dx. \]

Consequently, the global approximation of \( u(x, t_n) \), which we denote as \( u^n(x) \), consists of piecewise constant reconstructions with discontinuities at the cell boundaries.

The ansatz for deriving the finite volume method is the observation that the value of \( u(x) \) is conserved over \( \Omega_x \) and that the only changes of \( u^n_j \) occur due to the spatial flux at the cell interfaces. The change of a conserved quantity can be intuitively calculated as the difference between the inflowing quantities and the outflowing quantities. This idea results in:

\[ \frac{d}{dt} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t) \, dx = \mathcal{F}(u(x_{j-\frac{1}{2}}, t)) - \mathcal{F}(u(x_{j+\frac{1}{2}}, t)). \]

Dividing this equation by \( \Delta x \) gives us an equation for the change of \( u^n_j \) in time. Using this, we can define an update step of \( u_j \) from \( u^n_j \) to \( u^{n+1}_j \) as:

\[ u^{n+1}_j = u^n_j + \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \mathcal{F}(u(x_{j-\frac{1}{2}}, t)) \, dt - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \mathcal{F}(u(x_{j+\frac{1}{2}}, t)) \, dt. \quad (6.2) \]

Unfortunately, we do not know the value of \( \mathcal{F}(u(x, t)) \) for \( t > t_n \) because it changes over time and depends on \( u(x, t) \) which we initially tried to advance in time using the value of the spatial flux. Since the exact result is not available, we will approximate \( \int_{t_n}^{t_{n+1}} \mathcal{F}(u(x, t)) \, dt \) by \( \Delta t \mathcal{F}(u(x, t_n)) \) which corresponds to an Euler time-stepping scheme of first order accuracy. An additional problem of (6.2), is that it is not possible to calculate \( \mathcal{F}(u(x_{j+\frac{1}{2}}, t)) \) exactly because the value of \( u(x, t) \) at this point is not known. Moreover, since this point lies on the boundary of two cells with different average values, in our piecewise constant reconstruction, \( u_h(x_{j+\frac{1}{2}}, t) \) is two-valued. Consequently, we have to rely on a numerical flux aiming to reconstruct the spatial flux at this point.

Let \( F^n_{j+\frac{1}{2}} \) denote the numerical flux at the interface \( x_{j+\frac{1}{2}} \) at time \( t_n \). Usually, \( F^n_{j+\frac{1}{2}} \) depends on the limits of \( u_h(x) \) from both sides and can be therefore written as a function of the limit from the left side denoted by \( u_h(x^-_{j+\frac{1}{2}}) \) and from the right side \( u_h(x^+_{j+\frac{1}{2}}) \):

\[ F^n_{j+\frac{1}{2}} = \mathcal{F}(u^n_h(x^-_{j+\frac{1}{2}}), u^n_h(x^+_{j+\frac{1}{2}})). \]

In the most simple first order version of the finite volume method a constant reconstruction of \( u(x) \) on each cell is used. Consequently, the limits correspond to the cell average: \( u_h(x^-_{j+\frac{1}{2}}) = u_{j-1} \), \( u_h(x^+_{j+\frac{1}{2}}) = u_j \) and \( u_h(x^+_{j+\frac{1}{2}}) = u_{j+1} \).

Using this notation, the general form of the finite volume method can be recovered:

\[ u^{n+1}_j = u^n_j - \Delta t \frac{\Delta x}{2} (F^n_{j+\frac{1}{2}} - F^n_{j-\frac{1}{2}}). \quad (6.3) \]

It remains to specify how the numerical flux \( F(u : h^n(x^-_{j+\frac{1}{2}}), u^n_h(x^+_{j+\frac{1}{2}})) \) is chosen (see section 6.1.1). Moreover, the reconstruction of \( u_h(x^\pm_{j+\frac{1}{2}}) \) in each cell introduces an additional
degree of freedom in the finite volume method and gives rise to a range of different finite volume methods aiming to achieve a higher order accuracy. For a concise review of higher order finite volume schemes, which were also used in the implementation of this thesis, see appendix C.1. Another important question which needs to be addressed is how the timestep size of the finite volume method should be chosen to ensure a stable simulation. This question and the formal order of convergence of the finite volume method, which is one of the most important properties of each discretisation scheme, will be discussed in C.3.

To use the finite volume method to discretise (3.1), it remains to specify how to handle non-conservative equations with a right hand side in the form $P(u)$. Usually, it is either possible to split the updating of the unknowns due to spatial transport from the updating due to a right hand side, as it is, for example, done in [33], or to include the effects of $P(u)$ into the update step [54]. Following the second approach, it is possible to extend (6.3) to:

$$u_j^{n+1} = u_j^n - \Delta t \frac{1}{\Delta x} (F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n) + \Delta t P_j^n,$$

with $P_j^n = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} P(u(x, t_n)) dx$.

### 6.1.1. Numerical flux functions

In the context of solving a system of non-linear partial differential equations, two of the most common choices for the numerical flux are the local and global Lax–Friedrichs flux [55]. In the local Lax-Friedrichs flux, the flux over an interface, where $a$ denotes the limit value of $u_h$ from the left side and $b$ the limit value from the right side, is defined as:

$$F^{LLF}(a, b) = \frac{1}{2} (F(a) + F(b)) - \frac{C^{LLF}}{2} (b - a)$$

where $C^{LLF} = \max_{\min(a, b) \leq s \leq \max(a, b)} |F'(s)|$, whereas the global Lax-Friedrichs flux is defined as:

$$F^{GLF}(a, b) = \frac{1}{2} (F(a) + F(b)) - \frac{C^{GLF}}{2} (b - a)$$

where $C^{GLF} = \max_{\min(a, b) \leq s \leq \max(a, b), \forall \tilde{u} = u(x^-_{j+\frac{1}{2}}), \tilde{b} = u(x^+_{j+\frac{1}{2}})} |F'(s)|$. Both Lax-Friedrichs methods satisfy the essential properties such as monotony of a numerical flux. In comparison, the global Lax-Friedrichs method is more general and stable, while the local one can be used to obtain a higher spatial accuracy [55]. Using one of these fluxes, the finite volume method (6.3) becomes:

$$u_j^{n+1} = u_j^n - \Delta t \frac{1}{\Delta x} \left[ \frac{1}{2} (F(u_h^n(x^-_{j+\frac{1}{2}})) + F(u_h^n(x^+_{j+\frac{1}{2}})) - C(u_h^n(x^+_{j+\frac{1}{2}}) - u_h^n(x^-_{j+\frac{1}{2}})) \right]$$

$$- \frac{1}{2} \left[ F(u_h^n(x^-_{j-\frac{1}{2}})) + F(u_h^n(x^+_{j-\frac{1}{2}})) - C(u_h^n(x^+_{j-\frac{1}{2}}) - u_h^n(x^-_{j-\frac{1}{2}})) \right]$$

Another possible numerical flux, which relies on a completely different philosophy and can be efficiently combined with EQ, is the kinetic-based flux which makes use of the reconstruction of the velocity distribution $f$ at the cell interfaces. This technique has been extensively studied by Fox [32], Fox [33], and Vikas et al. [101] in the context of proposing the QMOM. Since the QMOM and EQ share the quadrature-based reconstruction, this method is also well suited for EQ and will be presented in the context of discretising the collisionless Boltzmann equation (3.1). Before deriving this numerical flux, let us first
introduce some notation: Every moment that is necessary for the spatial transport can be split up into a contribution of particles with a negative and particles with a positive velocity in one spatial direction:

\[ u_{i+1,j,k}^\lambda = Q^-_{i+1,j,k} + Q^+_{i+1,j,k}, \]

where

\[ Q^-_{i+1,j,k} = \int_{\Omega_c} \min(c_1, 0) i^{i+1} c_2^{j} f(c) dc \]

and \( Q^+_{i+1,j,k} = \int_{\Omega_c} \max(0, c_1) i^{i+1} c_2^{j} f(c) dc \).

Using this splitting technique, also the calculation of moments using a quadrature-based method can be rewritten from:

\[ u_{i+1,j,k} = \sum_{l=1}^{n} \frac{\gamma^{i+1} \gamma^{j} \gamma^{k}}{\eta_{l}} w_{l} \]

to:

\[ u_{i+1,j,k} = Q^-_{i+1,j,k} + Q^+_{i+1,j,k} = \sum_{l=1}^{n} \min(\gamma_{l,1}, 0) i^{i+1} \gamma^{j} \gamma^{k} w_{l} + \sum_{l=1}^{n} \max(0, \gamma_{l,1}) i^{i+1} \gamma^{j} \gamma^{k} w_{l}. \]

Using this considerations, a kinetic representation of the numerical flux in form of a flux-splitting technique relies on the idea that the exact flux at position \( y \), \( F(u(y, t)) \), can be, in one dimension, split up in a similar way to:

\[ F(u(y, t))_{i} = \sum_{l=1}^{n} \min(\gamma_{l,1}, 0) \gamma^{i} \gamma^{j} w_{l} + \sum_{l=1}^{n} \max(0, \gamma_{l,1}) \gamma^{i} \gamma^{j} w_{l}. \]

Remembering that \( u_{h}(y, t) \) is two valued at the interface, we use this splitting technique to calculate the numerical flux using the particles with a positive velocity from the left side of the cell (to calculate the second term) and the particles with a negative velocity from the right side of the cell (to calculate the first term). Since the spatial flux over an interface can be seen as the combined effect of all inflowing (left side of the interface and positive velocity) and outflowing particles (right side of the interface and positive velocity), this splitting is somehow intuitive. Using this intuition, the kinetic-based numerical flux is defined as:

\[
\left( F^n \right)_i = Q^-_{i+1}(\gamma, w^r) + Q^+_{i+1}(\gamma, w^l) = \sum_{m=1}^{n} \min(\gamma_{m,1}, 0)(\gamma_{m}^{r}) w_{m} + \sum_{m=1}^{n} \max(0, \gamma_{m,1})(\gamma_{m}^{l}) w_{m},
\]

where \( (\gamma^r, w^r) \) denotes the quadrature obtained by EQ using \( u^r_{h}(x^r_{j+\frac{1}{2}}) \) and \( (\gamma^r, w^r) \) the quadrature obtained using \( u^r_{h}(x^r_{j+\frac{1}{2}}) \|_{33, 101} \). All in all, the finite volume method using this flux can be written as:

\[
u_{j}^{n+1} = u_{j}^{n} - \frac{\Delta t}{\Delta x} (Q^-(\gamma_{x_{j+\frac{1}{2}}, w^r_{x_{j+\frac{1}{2}}}) + Q^+(\gamma_{x_{j+\frac{1}{2}}, w^r_{x_{j+\frac{1}{2}}}) - Q^-(\gamma_{x_{j-\frac{1}{2}}, w^r_{x_{j-\frac{1}{2}}}) - Q^+(\gamma_{x_{j-\frac{1}{2}}, w^r_{x_{j-\frac{1}{2}}})).
\]

(6.4)

This kinetic interpretation of the numerical flux is very intuitive; however, its theoretical properties are quite unclear. One disadvantage of this method is that the numerical flux will only vary smoothly when \( u(x) \) varies smoothly if the abscissas calculated from \( u(x) \) vary smoothly in space [32]. If, for example, a small change in \( u(x) \) causes a change of the sign of several nodes, the flux will jump. Nevertheless, we will consider said kinetic-based flux because of its intuitive explanation and because it is possible to prove that the finite volume combined with this flux is realisability-preserving. In appendix C.2, the important and extraordinary realisability-preserving property of (6.4) will be shown and possible extensions of this realisability-preserving scheme to higher order finite volume methods will be discussed.
6.2. Discontinuous Galerkin method

The motivation behind developing the discontinuous Galerkin (DG) method as an additional method to discretize a system of partial differential equations was that all of the already proposed methods have some considerable disadvantages. For instance, despite allowing for higher-order reconstructions, the finite volume method is usually restricted to uniform grids when a global second order scheme is used [41]. Therefore, the finite volume method is flexible in the order of local accuracy but not flexible concerning the geometric grid when higher accuracy should be achieved. In contrast to this, the finite element method is hp-adaptive, which means that this method is capable of changes concerning the order of approximation and the structure of the grid at the same time. One disadvantage of the finite element method is, however, that the test functions used in this method are globally defined. As a result, the method requires a costly inversion of the global mass matrix at each timestep and is not well-suited to allow for different orders of accuracy in different elements. Furthermore, the type of test functions in FE methods is quite restricted [41].

Out of all these reasons and probably much, much more, the discontinuous Galerkin method has been introduced by Reed and Hill [74] in the context of solving the neutron transport equation. Subsequently, the theoretical properties and especially the formal accuracy of this method has been studied extensively. In a very influential series of papers, Cockburn and Shu [21, 22, 23, 25, 26] extended the method to the multidimensional case and developed the method further to be also able to describe non-linear and even non-smooth flows with high order accuracy (see [24] and [27] for a concise review).

Generally speaking, the discontinuous Galerkin method can be seen as a combination of the finite volume and the finite element method. The ideas of a local approximation using basis functions and using test functions to obtain a finite number of equations from the finite element method are combined with the concept of a numerical flux and slope limiters from the finite volume method. This combination results in a scheme which is, in contrast to FE methods, purely local, while being, in comparison to FV methods, able to achieve an arbitrary order of accuracy on unstructured grids.

Following the reasoning of [41], we will start by introducing the discontinuous Galerkin method in one dimension to discretize a system of partial differential equations in the form (3.2):

\[
\partial_t u(x,t) + \text{div} F(u(x,t)) = P(u(x,t)).
\] (6.5)

First note that, since (6.5) is formulated componentwise in \( u \), it is possible to treat each component of \( u \) independently. For the sake of readability, we will do so and will proceed with the scalar \( u \) instead of the vector \( u \). We will do the same with \( F \) and \( P \). Keep in mind that, to obtain the numerical scheme for \( u \), the described method has to be executed for all \( u_i \) independently.

In the discontinuous Galerkin method, as it was done in the finite volume method, the spatial domain \( \Omega \) is divided in a finite number of non-overlapping cells \( I \) with common boundaries. In contrast to the finite volume method, however, we do not assume that \( u(x,t) \) can be approximated by its average value in each cell. Instead, it is assumed that \( u(x,t) \) can be locally approximated by a polynomial \( u_k^i(x,t) \) of order \( N \) for \( x \in I_k \):

\[
u_k^i(x,t) = \sum_{n=1}^{N+1} \alpha_n^i(t) \varphi_n^k(x) = \sum_{n=1}^{N+1} u(x_n^i,t) \ell_n^k(x).
\] (6.6)

The first expression is known as the modal form where \( \varphi_k^j(x) \) denotes a set of polynomial basis functions which are locally defined on \( j \) and zero on all other elements. In this form,
the unknown variables $\alpha^k$ in each cell $k$ are $N$ coefficients which fully determine the local approximation. As we will show later in appendix D.2, this expression is equivalent to the second expression which is known as the nodal form. In this form, we assume that each cell is subdivided by $N$ local grid points $x^k_n \in \mathcal{I}_k$ and that the polynomial basis functions are defined such that:

$$
\ell^k_n(x^k_l) = \begin{cases} 
0 & \text{if } l \neq n \\
1 & \text{if } l = n.
\end{cases}
$$

Therefore, the second expression is an exemplification of the first one where, due to the constraints put on $\varphi$, the coefficients $\alpha$ are the values of $u(x, t)$ at the local grid points. Being computationally very efficient, we will use the nodal form in this thesis (see appendix D.2 for greater details).

Furthermore, we assume that the spatial flux function $F(u)$ can be locally approximated in a similar way as:

$$
F_h(u^k(x, t)) = \sum_{n=1}^{N+1} F(u(x^k_n, t)) \ell^k_n(x).
$$

(6.7)

with $x \in \mathcal{I}_k$. Besides (6.7), there also exist various other possibilities to approximate the flux function [41]. Unfortunately, since in (6.7) $F$ is only evaluated at a few grid points, some of the characteristics of $F$ in the cell may not be captured by (6.7). Thereby, it is possible that some aliasing error is introduced which can be addressed using the concept of filtering that will be introduced in appendix D.3 [41]. Usually, to overcome this issue, it would be advantageous to replace $F(u(x^k_n, t)))$ in (6.7) by $F_h(u(x^k_n, t))$ where $F_h$ is the projection of $F$ onto an $N$th order polynomial. This technique can be proven to be stable and energy conserving. Unfortunately, this projection is computationally expensive and only possible when $F$ is known explicitly. As a consequence, we will use (6.7) as the approximation of the flux, as $F$ is a non-linear operator and has to be approximated using a closure technique.

As in the finite volume method, using said local approximations, it is possible to obtain a global approximation $u_h(x, t)$ of $u$ as the union of local approximations, so that $u_h(x, t)|\mathcal{I}_k = u^k_h(x, t)$. Note that the global approximation for each of the entries of $u$ consists of piecewise polynomials of order $N$ on every cell $\mathcal{I}_k$ which are discontinuous and two-valued at the cell boundaries.

Having specified the structure of the local solution $u^k_h(x, t)$, now the problem in which sense this approximative solution has to satisfy (6.5) needs to be addressed. Using the above approximation it is clearly not possible to solve (6.5) directly and exactly. Therefore, we choose that the residual of the equation\textsuperscript{19} has to be orthogonal to a space of test functions $V_h$ with dimension $N+1$. $V_h$ is spanned by a set of $N+1$ linear independent test functions $\phi_1..., \phi_{N+1}$. Following the locality of the scheme, we define a space of local test functions $V^k_h$ spanned by $\phi_1|\mathcal{I}_k, ..., \phi_{N+1}|\mathcal{I}_k$ with $V_h$ being the union of all $V^k_h$. Requiring that the residual has to be orthogonal to all functions in $V_h$ is therefore equivalent to enforcing that the residual on each of the elements $\mathcal{I}_k$ has to be orthogonal to all of the basis functions of $V^k_h$. As a result, it is required that on each element $\mathcal{I}_k$ the following set of equations has to hold:

$$
\int_{\mathcal{I}_k} \phi^k_j(x)u^k_h(x, t) + \phi^k_j(x)\text{div}F(u^k_h(x, t)) - \phi^k_j(x)(P(u^k_h(x, t)))dx = 0, \forall j = 1, ..., N + 1
$$

(6.8)

\textsuperscript{19}The residual of (6.7) is defined as $R(x, t) = \partial_t u(x, t) + \text{div}F(u(x, t)) - P(u(x, t))$ and should ideally be zero. The residual can be interpreted as a measure of the error of the solution.
This set of equations puts \( N+1 \) constraints the local solution \( u_h^k \) in each cell \( I_k \). Recalling that the local solution \( u_h^k \) is defined by \( N+1 \) unknowns (either a general set of coefficients or a set of values of \( u(x,t) \) at local grid points), it is possible to calculate the time derivatives of the unknown using the local constraints (6.8) put on each of the local solutions.

Assuming that the test functions are smooth, we can apply the Gaussian theorem to split up the second term of (6.8) and to rearrange (6.8) to obtain the weak form of the DG method\(^20\):

\[
\int_{I_k} \phi_j^k(x)u_h^k(x,t) - \frac{\partial \phi_j^k(x)}{\partial x} F(u_h^k(x,t)) dx + \int_{\partial I_k} \mathbf{n} \phi_j^k(x) F^*(u_h(x,t)) - \phi_j^k(x) P(u_h^k(x,t)) dx = 0
\]

(6.9)

where \( \mathbf{n} \) denotes the outward pointing normal vector on the considered boundary and \( F^* : \mathbb{R}^n \rightarrow \mathbb{R} \) denotes the numerical flux aiming to calculate the flux over a boundary. As in the finite volume method, the usage of a numerical flux is necessary, since \( u_h \) is two-valued at the boundary. In our implementation of DG, the numerical fluxes presented in section 6.1.1 are used. However, producing the least amount of artificial viscosity, Cockburn and Shu \(^{24}\) suggested that to solve non-linear problems usually the Godunov flux should be used. Unfortunately, it is not possible to use some popular fluxes such as the Goudnov flux, since \( F \) is a highly non-linear term an it is no closed form expression available. Fortunately, Cockburn and Shu \(^{24}\) observed that the local Lax-Friedrichs flux usually produces similar results as the Goudnov flux thereby justifying our choice of this fluxes.

Before proceeding with the rearrangement of (6.8), a few remarks concerning the set of test functions need to be made. In principle, it is possible to rearrange (6.9) in a way that arbitrary sets of test functions, even delta functions, can be used.\(^{21}\) Here, however, following the Galerkin approach, it is assumed that the set of test and basis functions are identical both spanning a space of polynomials up to order \( N \). Consequently, we set \( \phi_j^k(x) = \ell_j^k(x) \).

Using this assumptions, it is possible to simplify (6.9). In one dimension, it is even possible to evaluate the volume integral explicitly, which will be replaced by the left \( x_{k-\frac{1}{2}} \) and right \( x_{k+\frac{1}{2}} \) boundary points with the corresponding normals \(-1\) and \(1\), respectively. As a result, we obtain the following discretisation of (6.5):

\[
\mathcal{M}^k \partial_t u(x^k,t) = S^k F(u(x^k,t)) - F^*(u_h(x_{k-\frac{1}{2}},t),u_h(x_{k+\frac{1}{2}},t)) \ell_j^k(x_{k-\frac{1}{2}}) \\
+ F^*(u_h(x_{k+\frac{1}{2}},t),u_h(x_{k-\frac{1}{2}},t)) \ell_j^k(x_{k+\frac{1}{2}}) + \mathcal{M}^k P(u(x^k,t))
\]

(6.10)

where

\[
u(x^k,t)_j = u(x^k_j,t), \quad \mathcal{M}_{ij} = \int_{I_k} \ell_i^k(x) \ell_j^k(x) dx \quad \text{and} \quad S_{ij}^k = \int_{I_k} \ell_i^k(x) \frac{\partial \ell_j^k(x)}{\partial x} dx.
\]

By multiplying both sides with \( (\mathcal{M}^k)^{-1} \), the time derivatives of the unknowns can be obtained.

All in all, to discretise (6.5) using the discontinuous Galerkin method, at each timestep, it is necessary to calculate the time derivative of the unknown \( u(x^k,t) \) at each cell for each moment using (6.10). This involves calculating the fluxes at the local grid points out of test functions, it is necessary to use the strong form of the DG method.
of the given values of \( u \) at this points and the numerical fluxes at the interfaces. Using this information, it is possible to calculate the right-hand side of (6.10) and through subsequent multiplication with \((\mathcal{M}^k)^{-1}\) to obtain the time derivatives of the values of \( u(x,t) \) at the local grid points.

Fortunately, the presented discontinuous Galerkin method is of a highly local nature. The different elements are only coupled by the numerical flux between two adjacent cells. Since - in contrast to FE methods - \( \mathcal{M} \) and \( S \) are also local, the discontinuous Galerkin method is highly parallelisable and able to handle different mesh adaptation strategies. Furthermore, the presented DG method is of arbitrary order and satisfy the hp-adaptivity property.

For a more detailed derivation of the discontinuous Galerkin method and a derivation in two dimensions see appendix D. Additionally, in appendix D.3, a short introduction to the main problem of the described method to handle discontinuities, and possible techniques to overcome this problem is given.

6.2.1. Timestepping schemes

In the presented derivation of the discontinuous Galerkin method, we treated time as a continuous quantity thereby obtaining a semidiscrete scheme. However, in addition to a spatial discretisation, also a temporal discretisation dividing the time of interest \([0,T]\) into a finite number of timesteps: \( t_1, \ldots, t_n, t_{n+1}, \ldots \) with \( \Delta t = t_{n+1} - t_n \) is needed. In the finite volume method, we already introduced a first order timestep method which is known as the Euler method. However, to be able to achieve a high order accuracy of the solution of non-linear problems not only higher order spatial discretisation schemes but also higher order time-stepping schemes are needed. Furthermore, the stability regions of higher-order time-stepping schemes are usually much larger than the one of the Euler method thereby allowing for larger timesteps. Without the use of higher order time-stepping schemes, the DG method sometimes produces inaccurate descriptions of the solution even under quite restrictive conditions on the timestep size [41].

Independent from the method chosen for the spatial discretisation, at the end of each step of the numerical scheme, an approximation of the temporal derivative of the unknowns at the current timestep is obtained, which can be written in the form \( \frac{\partial u^k_{i,j}(t_n)}{\partial t} = R(u^k_{i,j}(t_n)) \). For example, in the presented DG method \( u^k_{i,j} = u(x^k_{i,j}, t) \). In the finite volume method, we just use the first order Euler scheme:

\[
U^k_{i,j}(t_{n+1}) = U^k_{i,j}(t_n) + \Delta t R(U^k_{i,j}(t_n)).
\]

As mentioned above, due to its restricted range of stability and low order accuracy, the Euler method is usually not suited to be used in combination with the high order accurate discontinuous Galerkin method. The reason for this is that the Euler method assumes that the time derivative of \( U^k_{i,j} \) in \([t_n, t_{n+1}]\) remains constant. Otherwise, an interpolation of the solution at, for example, \( t_{n+\frac{1}{2}} \) is needed and two time-derivatives of the unknown need to be calculated.

The higher order Runge-Kutta scheme, which has been introduced in [87], is generally used in combination with the DG method\(^{22}\). This method utilises the idea of computing values of \( U^k \) at different steps and writes the updated value as a linear combination of intermediate values and approximations of the updates at the different intermediate steps.

\(^{22}\)Shu and Cockbourn even gave the combination of both methods a name: Runge-Kutta discontinuous Galerkin method (RKDG) and introduced them in combination.
The general rth order Runge-Kutta scheme can be expressed as \[25\]:

\[
(u_{i,j}^k)^{(m)} = \sum_{l=0}^{m-1} [\alpha_{ml}(u_{i,j}^{k(l)}) + \beta_{ml} \Delta t R((u_{i,j}^{k(l)}))], \forall m = 1, \ldots, r
\]

with \((u_{i,j}^{k(0)}) = u_{i,j}(t_n)\) and \((u_{i,j}^{k(r)}) = u_{i,j}(t_{n+1})\). In the work of Cockburn and Shu [20, 21, 24], usually second and third order Runge-Kutta schemes with the following coefficients have been considered:

\[
\begin{array}{ccc}
 r=2 \text{ (second order)} & \alpha_{10} = 1 & \beta_{10} = 1 \\
 & \alpha_{20} = \alpha_{21} = 0.5 & \beta_{20} = 0, \beta_{21} = 0.5 \\
 r=3 \text{ (third order)} & \alpha_{10} = 1 & \beta_{10} = 1 \\
 & \alpha_{20} = \frac{3}{4}, \alpha_{21} = \frac{1}{4} & \beta_{20} = 0, \beta_{21} = \frac{1}{4} \\
 & \alpha_{30} = \frac{1}{3}, \alpha_{31} = 0, \alpha_{32} = \frac{2}{3} & \beta_{30} = 0, \beta_{31} = 0, \beta_{32} = \frac{2}{3}
\end{array}
\]

where \(t_l\) denotes the time at which the calculation of the \(l\)th intermediate value takes place. This value can be used as an additional parameter for \(R\) to ensure that time-dependent boundary conditions are correctly enforced. Note that the Runge-Kutta method does not have any impact on our spatial discretisation. The only difference between using this method or the Euler scheme is that for each full timestep, (6.10) has to be solved \(r\) times with different intermediate values of \(u_{i,j}^k\). In our experiments, usually considering high order spatial discretisation up to fifth order, we used the third order Runge-Kutta method as said method provides a suitable large region of stability. We also tested the considered spatial discretisation schemes in combination with temporal discretisation methods of different orders. Different time-stepping schemes usually lead to similar or slightly worse results. Therefore and since the third order method offers improved linear stability properties, we decided to always use the third order Runge-Kutta method for the sake of comparability between different approaches. A general discussion of a suitable timestep size and further theoretical properties of the discontinuous Galerkin method, such as the formal order of convergence can be found in appendix D.4.

7. Numerical results

Analysing the performance of EQ as a closure of the Boltzmann equation is a very challenging task. The main difficulty of this task is that only the performance of the whole solver can be measured directly. This solver, however, usually also involves several additional steps which simplify the problem, e.g. a temporal and spatial discretisation scheme. It is, therefore, sometimes unclear which simplifications account for the measured error of the solution. Nevertheless, in the following chapter, we will try to analyse the capabilities of EQ to simulate equilibrium and non-equilibrium flows by solving the Boltzmann equation.

To test EQ, a solver of the Boltzmann equation following the reasoning of this thesis has been implemented. In the solver and also in the course of the thesis, several steps have been made to simplify the Boltzmann equation and to make its solution computationally feasible. To analyse the performance of EQ, we will shortly review the important steps which have been made. The main simplification steps taken in this thesis are depicted in figure 2. Unfortunately, each of these steps introduces some error in the computed solution. To identify possibilities for an in-depth analysis, in the following, the way in which and the reason why said steps introduce an error in the solution will be reviewed shortly.
In the first step, some error is introduced because by replacing the exact velocity distribution with a finite number of its moments some knowledge about the system is lost. Furthermore, as mentioned earlier, the underlying distribution cannot be uniquely reconstructed from a finite number of its moments. Therefore, transporting the moments instead of the velocity distribution in time will result in some form of error. As a consequence of the first step, it is necessary to use a closure to close the spatial transport term in the Boltzmann equation. As every closure makes some assumptions about the distribution function which needs to be reconstructed such as its general form, no closure can always reconstruct the actual underlying distribution function. As a consequence, some error is introduced in the solution. Since the aim of this chapter is to examine the capabilities of EQ, we will use this closure in conducting the experiments. It is, however, as mentioned in chapter 5, possible to interpret EQ as a sparse version of the maximum-entropy closure. Following this interpretation and aiming to build upon the work conducted on the maximum-entropy closure, in the analysis of the simplification steps taken, we assume that as a first step the maximum-entropy closure is used to close the Boltzmann equation formulated in terms of the moments of the underlying velocity distribution. Unfortunately, also this mostly very accurate closure will introduce some error because the underlying distribution function is not always in the form of a maximum-entropy distribution.

In the next step, the costly full-scale maximum-entropy closure is replaced by the proposed sparse EQ. This will introduce an error in the maximum-entropy closure because instead of the maximum-entropy distribution only a sparse maximum-entropy quadrature is reconstructed.

In the final step, since it is impossible to keep track of the moments of the velocity distribution at each point in space and time, different space and time discretisation schemes have been used. As mentioned in appendix D and appendix C, this discretisation schemes will introduce some error with varying magnitude, too.

As a consequence, testing EQ as a closure of the Boltzmann equation is very challenging because comparing the calculated solution with the analytical solution of the Boltzmann equation, which is, in fact, only very rarely available, one often cannot be sure which of the simplifications accounts for the measured error. This problem is intensified by the fact that sometimes errors balance each other out. Therefore, in the following, we will test as many parts of the process in isolation as possible. Even when considering the performance of the whole solver, several attempts to identify the step which accounts for the most significant part of the error will be made.

Turning to a description of possible attempts to test the different steps in isolation, the first step cannot be tested in this way because the unclosed system of moments is
underspecified and without a closure it is not possible to advance the moments in time. In previous work, however, it has been shown that it is possible to use the method of moments in combination with different closures to provide very accurate approximations of the solution of the Boltzmann equation [94]. Fortunately, the subsequently used maximum-entropy closure has already been extensively tested and examined by various authors. They all concluded that the maximum-entropy closure is a very promising candidate for describing a variety of flows in different regimes which usually yields very accurate results [56, 64, 80, 81, 82]. It is especially remarkable that it has been shown that the maximum-entropy closure is also capable of modelling strong non-equilibrium flows.

The third simplification step, however, which involves EQ as a sparse version of the maximum-entropy closure, has never been taken before due to the novelty of the proposed closure. Therefore, the first subsection of this chapter will be devoted to testing the capabilities of EQ to approximate the maximum-entropy solution.

Concerning the spatial and temporal discretisation, which is done in step 4, as described in appendix D and appendix C, the accuracy of the finite-volume scheme and the discontinuous Galerkin method can even be theoretically proven. To validate our implementation of the used discretisation methods, we also conducted some experiments to explore the accuracy of this methods in practice. The results of this experiment can be found in appendix E. The main conclusion of this section is that our implementation is able to reach the theoretically proven optimal rate of convergence for test cases involving smooth solutions. Finally, to test the capabilities of the whole solver, which has been described in this thesis, we will also conduct various test cases to explore its capabilities to describe equilibrium and non-equilibrium flows in different regimes.

In the following chapter, for the sake of simplicity and to save computation time, we will limit our experiments mainly to one dimension and will only sometimes consider the two dimensional case. An extension to three dimensions is, however, directly and intuitively possible.

7.1. Entropic Quadrature for predicting higher order moments

Solving the Boltzmann equation, the main challenge is to close the spatial transport term by predicting some higher order moments out of a given set of lower order moments. To examine the ability of EQ to solve the Boltzmann equation, we will now study the performance of EQ as a solution of the problem of moments. Recalling that EQ can be seen as a sparse version of the full-scale maximum-entropy closure, as depicted in figure 2, we will compare the higher order moments predicted by the method of entropic quadrature with those predicted by the maximum entropy approach.

In the following, the performance of EQ in two different types of test cases will be studied. First, the ability of entropic quadrature to approximate the maximum-entropy solution will be examined. Since it is known that given a set of moments the maximum-entropy closure will always return a distribution in the form of $f^{(ME)}$ fitting the given set of moments, said comparison reduces to examining the capability of EQ to capture distributions in the form of the maximum-entropy distribution $f^{(ME)}$. After that, functions in the form of the sum of two maximum-entropy distributions, for example the sum of two Gaussian distributions, which arise quite commonly in practice, will be considered.

7.1.1. Maximum-entropy distribution in one dimension

To study the ability of EQ to approximate the solution of the maximum-entropy closure, we have constructed some maximum-entropy distributions and tested the ability of EQ
to predict their higher order moments out of a given set of lower order ones. Therefore, a set of 1.000 maximum-entropy distributions in the form:

\[ h(x) = e^{\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4}, \text{ with } \theta_4 \in [-1, -0.1] \text{ and } \theta_3, \theta_2, \theta_1, \theta_0 \in [-1, 1] \]

has been randomly generated. Subsequently, we calculated the first \( M \) moments of this distribution function and used the EQ-algorithm to construct a distribution \( f^{(EQ)} \) fitting the \( M \) moments of \( h(x) \). To evaluate \( f^{(EQ)} \), we calculated the relative error of its prediction of the next higher order moment, which is, in one dimension, the only moment needed to close the Boltzmann equation: \[ \left| \frac{\langle x^M f^{(EQ)}(x) \rangle - \langle x^M h(x) \rangle}{\langle x^M h(x) \rangle} \right| . \] In table 1, the most important results of the conducted experiment are depicted. In the first seven columns, the performance of EQ using algorithm 2 as usual to calculate the abscissas are displayed. To give a detailed overview of the results several statistical measures of the one thousand measured relative errors have been included with Q1 denoting the first quantile of the measured errors and Q3 the third one. Furthermore, we included the average number of iterations of the multidimensional Newton method required to find the maximum of the Lagrange dual functions which is necessary to find the maximum-entropy weights. Additionally, we included two reference solutions in the table. In the ninth column, the average error of the next higher order moment computed using the QMOM is displayed. Finally, since this technique can be a very promising approach for tackling problems with non-realisability, we also replaced the maximum-entropy criterion with \( x^2 \), as it was mentioned in chapter 5.4. The results of this slightly changed form of EQ are presented in the tenth column.

<table>
<thead>
<tr>
<th>M</th>
<th>Mean</th>
<th>Median</th>
<th>Q1</th>
<th>Q3</th>
<th>Min</th>
<th>Max</th>
<th>Nlte</th>
<th>QMOM</th>
<th>EQX^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.095</td>
<td>0.065</td>
<td>0.025</td>
<td>0.15</td>
<td>6.7 × 10^{-1}</td>
<td>0.71</td>
<td>5.34</td>
<td>0.32</td>
<td>0.071</td>
</tr>
<tr>
<td>6</td>
<td>0.004</td>
<td>0.0032</td>
<td>0.0013</td>
<td>0.006</td>
<td>1.2 × 10^{-5}</td>
<td>0.03</td>
<td>6.01</td>
<td>0.14</td>
<td>0.0037</td>
</tr>
<tr>
<td>8</td>
<td>0.0029</td>
<td>0.0021</td>
<td>0.00036</td>
<td>0.0045</td>
<td>3 × 10^{-8}</td>
<td>0.017</td>
<td>6.07</td>
<td>0.059</td>
<td>0.0028</td>
</tr>
<tr>
<td>10</td>
<td>0.002</td>
<td>0.0016</td>
<td>0.00038</td>
<td>0.0033</td>
<td>2.1 × 10^{-12}</td>
<td>0.0077</td>
<td>6.08</td>
<td>0.024</td>
<td>0.0021</td>
</tr>
<tr>
<td>12</td>
<td>0.00088</td>
<td>0.00063</td>
<td>0.00014</td>
<td>0.0015</td>
<td>3.1 × 10^{-12}</td>
<td>0.0034</td>
<td>6.09</td>
<td>0.01</td>
<td>0.00092</td>
</tr>
</tbody>
</table>

Table 1: Relative error of the predicted higher order moment of a set of maximum-entropy distributions

Concerning the analysis of the obtained results, it is, first of all, remarkable to observe that the error of the predicted moment decreases constantly when the number of moments given is increased. Despite being intuitively plausible, the strength of this decrease is quite surprising. Especially the improvement achieved by increasing the number of given moments from four to six is very drastic. However, this result can be explained by the fact that the number of parameters of the model distribution of the EQ-weights is equal to the number of moments given. With only four moments given said distribution cannot capture the more complex distribution \( h(x) \). With five moments, however, this is possible leading to this substantial decrease of the relative error. Nevertheless, this observation is very interesting because it strengthens the relationship between the model distribution of the EQ-weights and the maximum-entropy distribution. Furthermore, it also shows how important a thoughtful choice of the number of given moments really is and that this choice should depend on the complexity of the distribution function which presumably needs to be approximated.

Concerning the computational cost of EQ, the number of matrix-inversions necessary to find the maximum-entropy weights increases slightly when the number of given moments increases. Furthermore, the size of the matrix which needs to be inverted increases, too.
Nevertheless, the whole conducted experiment takes only around 50ms on a standard laptop.

Comparing the results of EQ with the results from the QMOM, the predictions using the QMOM are significantly worse than the ones of EQ. This is, however, not surprising as the underlying distribution function is in the form of a maximum-entropy distribution and the degrees of freedom introduced by EQ into the QMOM are always used to increase the fit with said distribution. Nevertheless, it is worth noticing that EQ also produces a better reconstruction of the next higher order moment when the same number of abscissas is used. Comparing the average error of EQ with six moments and the QMOM with 12 moments, the optimisation criterion used by EQ seems to be advantageous over the knowledge of additional moments in the QMOM. This is a strong hint that using EQ it could be possible to reduce the number of moments considered compared to the QMOM while still being able to model more complex non-equilibrium flows.

The most surprising insights are offered by the comparison between EQ and the modified version of EQ with the optimisation criterion $x^2$ (EQ$X^2$). As mentioned earlier, using EQ$X^2$ has the significant advantage of always producing a fitting set of weights on the cost that negative weights may be produced. In the conducted experiment, surprisingly, EQ$X^2$ often produces better results than EQ. This result is not intuitively explainable because, in theory, EQ fills the degree of freedom in the quadrature optimal in the sense that the reconstructed quadrature is as close as possible to the maximum-entropy distribution, whereas EQ$X^2$ optimises the quadrature to $x^2$. One possible explanation is that since the quadrature points are not uniform and their number is very small, the possible fit with the maximum-entropy distribution is somewhat limited. Furthermore, $x^2$ often approximates $x \log x$ locally quite well with the additional advantage of an increased freedom by allowing for negative weights. All in all, these results indicate that EQ$X^2$ is an up-and-coming candidate to replace EQ when realisability issues are faced.

However, the picture changes when the degree of freedom and therefore the impact of the optimisation criterion is increased. In figure 3, the development of the average relative error of different versions of EQ for $M = 8$ is depicted when the number of additional inserted nodes increases. The usual Wheeler-based EQ, which uses algorithm 2 for the calculation of the abscissas, which keeps the nodes from the Wheeler algorithm, (purple) is compared with EQ in combination with algorithm 3 which calculates a uniform quadrature in a larger domain (green) and a combined version of both algorithm where a uniform quadrature in the domain of the nodes of algorithm 2 is calculated (blue). Furthermore, we included EQ$X^2$ using nodes calculated with algorithm 3. To interpret the calculated results adequately, it is first important to note that a logarithmic y-axis is used. Concerning EQ$X^2$, as pointed out at the beginning of this paragraph, the additional degree of freedom introduced by the insertion of additional nodes is disadvantageous for approximating the
maximum-entropy distribution because due to the changed optimisation criterion, this degree of freedom is instead used to approximate a function in the form $\sum x^2$. Consequently, it is possible to conclude that both the maximum-entropy and quadratic optimisation criterion produce quite similar results when the available degree of freedom is limited while differ significantly when it is increased. Furthermore, the behaviour of $EQX^2$ concerning the production of undesirable negative weights changes drastically, too. In the default version of $EQ$ where the number of nodes is only doubled and the degree of freedom is limited $EQX^2$ is not only in excellent agreement with $EQ$ but also only rarely produces negative weights (in around 5% of the cases). However, when the degree of freedom is increased, negative weights occur in nearly all calculated quadratures. To ensure that the number of calculated negative weights is limited, $EQX^2$ should only be used with a minimal multiplicative factor. All in all, $EQX^2$ seems to be a great candidate to tackle problems with non-realisable sets of moments and realisability problem of $EQ$. However, to face said problems this strategy should not be combined with an increased number of nodes, which is an alternative to tackle said problems.

Turning to the comparison of $EQ$ combined with different algorithms to find the nodes, it is, first of all, essential to note that both versions of $EQ$ which use a uniform quadrature were not always able to reconstruct the given moments correctly when the number of nodes is only doubled in comparison to the nodes calculated from the Wheeler algorithm. This is due to the fact that the Wheeler-nodes are no longer included in these quadratures and consequently the existence of a non-negative set of weights is no longer guaranteed. However, surprisingly, for all considered test cases when the multiplicative factor of the nodes is increased to 4 or higher, all given moments were fitted exactly. Further comparison of the results of $EQ$ using a uniform quadrature and $EQ$ using a Wheeler-based quadrature leads to the insight that while it is beneficial to use the Wheeler-based quadrature for a minimal multiplicative factor because here the existence of a non-negative set of weights can be guaranteed, for a higher degree of freedom it is usually advantageous to use a uniform-distributed set of nodes obtained by using algorithm 3. The reasons, why the uniform quadrature is advantageous over the Wheeler-based quadrature for a higher multiplicative factor, can be identified by considering the results of the uniform quadrature calculated in the bounds of the Wheeler-based quadrature. For higher multiplicative factors, this quadrature performs significantly better than the Wheeler-based quadrature indicating that a uniform quadrature can be beneficial as discussed in chapter 5.2.3, whereas performing drastically worse than the uniform quadrature on a larger domain. The importance of an increasing domain of the quadrature is also supported by the observation that the error of the uniform quadrature on an increasing domain decreases substantially when the multiplicative factor is increased, while the errors of the other two quadratures witness no significant changes. It is further interesting to note that, in practice, the error of $EQ$ using algorithm 3 for the calculation of the nodes converges to machine precision when the multiplicative factor is increased.

Referring back to the discussion of possible practical problems of the infeasible Newton method in chapter 5.2.3, on the considered test cases here, we can support the assumption. Using the default $EQ$-algorithm with the infeasible newton method instead of the Lagrange dual function to calculate the $EQ$-weight, on the one thousand 12-moment test cases considered here, the infeasible newton method fails to converge on 23 test cases. In most of these cases the set of moments lie on the boundary of realisability e.g. the determinant of the Hankel matrix is very low (around $1e^{-12}$). Since optimising using the Lagrang dual function is much more stable, consequently, said method should be chosen in situations where moments which lie on the boundary of realisability are involved.
To conclude, the conducted experiments indicate that EQ can be, in fact, seen as a sparse way of the maximum-entropy closure because this closure is well-suited to predict the next higher order moment of a maximum-entropy distribution with a very high accuracy. Furthermore, the accuracy of the method can be improved by increasing the number of moments given and the additional nodes inserted as well as using uniformly distributed nodes instead of Wheeler-based ones.

### 7.1.2. Sums of two maximum-entropy distributions

In the following, as they commonly arise in practice, sums of maximum-entropy distributions will be shortly considered. Examining the ability of EQ to predict the higher order moments of those distributions is very interesting because the underlying distribution function is no longer in the maximum-entropy form. Consequently, the maximum-entropy closure does not achieve a perfect fit of the missing higher-order moments and it is rather unclear how the error of the maximum-entropy closure and the QMOM relates to each other in said case. Similar to the proceeding in section 7.1.1, we generated 500 functions in the form $g(x) = h'(x) + h''(x)$ with $h(x)$ being a randomly generated maximum-entropy distribution. The main results of our analysis are visualised in figure 4 which is constructed analogous to figure 3.

However, the first important difference between the two figures is that the y-scale is no longer logarithmic. Consequently, the differences between the error of the three quadratures are no longer as significant as in the previous subsection. The stagnation of the error of the uniform quadrature indicates that the usage of the maximum-entropy closure would approximately result in the same error.

All in all, despite seeming to be slightly advantageous in the considered test case the maximum-entropy solution seems to be well approximated by EQ in cases where the underlying distribution function is no longer a maximum-entropy distribution pointing to a very promising approximation of the maximum-entropy solution by EQ in practical experiments.

### 7.1.3. Maximum entropy distribution in two dimensions

We also extended our tests from section 7.1.1 to two dimensions. Here, three hundred two dimensional maximum-entropy distributions in the form:

$$h(x) = e^{\theta_0 x + \theta_1 y + \theta_2 x^2 + \theta_3 xy + \theta_4 y^2 + \theta_5 x^3 + \theta_6 x^2 y + \theta_7 xy^2 + \theta_8 x y^3 + \theta_9 x^4 + \theta_{10} x^3 y + \theta_{11} x^2 y^2 + \theta_{12} x y^3 + \theta_{13} y^4},$$

with $\theta_9, \theta_{10}, \theta_{11}, \theta_{12}, \theta_{13} \in [-1, -0.1]$ and $\theta_8, \theta_7, \theta_6, \theta_5, \theta_4, \theta_3, \theta_2, \theta_1, \theta_0 \in [-1, 1]$ have been randomly generated. In two dimensions, we keep all moments up to order $M - 1$ as the input of the problem of moments. Since we aim to close the spatial transport term
of the Boltzmann equation, the closure needs to predict all moments of order $M$. The error of the reconstruction is measured as the average relative error of these predictions. In contrast to the one-dimensional test case, in two dimensions our first concern is to analyse the capabilities of EQ to find a set of non-negative weights. For $M = 6$, in none of the examined cases do the Wheeler-based EQ algorithm not fit the given moments. For problems of moments including seventh order moments, however, problems of EQ to find a non-negative set of weights for the Wheeler-based nodes algorithm arise. In the conducted test cases, said problems could be effectively solved by inserting additional nodes as it was suggested in chapter 5.4. Interestingly, we also tried to rotate the given system of moments to increase the probability that there exists a non-negative set of weights for the tensorial system of nodes. Despite being intuitively very reasonable, on average, this technique does not lead to any benefits concerning precision and the existence of a non-negative set of weights.

Turning to the bigger picture, since the main conclusions of the previous section such as that the measured error decreases when the number of given moments is increased, also hold true for the two-dimensional case, we will only shortly discuss a small part of the results which are plotted in figure 5. Here, as usual, the results of EQ combined with different algorithms to find the nodes are depicted. All moments up to fifth order, which are 25 in two dimensions, are given. Concerning the analysis of the computed results, it is, first of all, interesting to note that the measured errors are all very small and for the Wheeler-based EQ even in the same order of magnitude as in one dimension. One significant difference, however, is that the error of the Wheeler-based EQ decreases substantially when the number of additional inserted nodes is increased. Simultaneously, the error of the uniform quadrature remains more or less constant. The first observation can be explained by the fact that, in two dimensions, the considered distributions are presumably much more complex than in one dimension. Consequently, only by considering an increased number of nodes it is possible to capture all their characteristics. On the other hand, the second observation can be explained in two ways: Either is the multiplicative factor chosen too low to effectively approximate the maximum-entropy solution or the numerics involved in the algorithm witness some problems in the case that the number of nodes is very high.

Nevertheless, the conducted experiments indicate that EQ is also well-suited to approximate maximum-entropy distribution and thereby the solution of the maximum-entropy closure in two dimensions. However, the application of EQ to multiple dimensions is more challenging than in the one-dimensional case because of the faced realisability problems and the increased complexity of the involved numerical algorithms.
7.2. Kinetic equations

To test the capabilities of EQ to close kinetic equations, we will analyse the performance of the described solver in different test cases involving equilibrium and non-equilibrium flows. For the sake of simplicity and saving computation time, we will mainly focus on experiments in one dimension before turning to one two-dimensional test case. Despite ultimately aiming to accurately describe flows in the transition regime, out of various reasons, we will often consider the collisionless Boltzmann equation, e.g. \( Kn \rightarrow \infty \), instead. The initial motivation for considering this equation is that, only in the collisionless case, an analytical solution of the Boltzmann equation is available. Without the collision integral, the Boltzmann equation is analytically solvable, and the solution is always in the following form:

\[
f(x, c, t) = f_0(x - tc, c)
\]  

with \( f_0 \) being the initial condition of the problem. The availability of an analytical solution is very useful because it allows for comparing the computed solution with the real one and measuring the error precisely.

Moreover, the collisionless Boltzmann equation can also be considered as the hardest possible test case for EQ. First of all, the closely-related QMOM struggles with the accurate description of flows at high or infinite Knudsen number [33]. In said case, due to the discrete nature of this closure, usually delta shocks and vacuum states are produced [33, 35]. Furthermore, in the collisionless case in multi-dimensions, the QMOM encounters problems with negative weights more frequently [33]. Fox [33] even concluded that “because of the strong dependence of solutions to the collisionless kinetic equation on initial and boundary conditions, it is unlikely that any moment method can provide an accurate solution for all cases.” As a result, since the accuracy of the QMOM decreases with increasing Knudsen number, we can also expect that the accuracy of EQ will increase when considering flows in the transition regime instead of in the collisionless case. Furthermore, the collisionless case is also one of the most difficult ones for the maximum-entropy closure because the collision term which draws the local velocity distributions to a local thermodynamic equilibrium is no longer present. Additionally, the main focus of the present thesis is on the spatial transport term of the Boltzmann equation and not on different models for the collision operator of the Boltzmann equation. As a result, it is somehow natural to only focus on the effects of the spatial transport term leaving aside inter-particle collisions.

7.2.1. Smooth perturbation

For the sake of comparability with work conducted on the maximum-entropy closure, we will first examine a test case introduced in Schaefer et al. [81]. In the following, we consider a gas which is in a local equilibrium with a smooth perturbation of the mass density. The corresponding initial conditions are:

\[
f_0(x, c) = f_M(x, c; \rho_0(x), v_0(x), \theta_0(x)) \quad \text{with} \forall x \in \Omega_x = [-78, 78]
\]

\[
\rho_0(x) = 1 + 4.7N(x; 0, 122.7), v_0(x) = 0, \theta_0(x) = 1,
\]

with \( N(x; \mu, \sigma^2) \) being the normal distribution in one variable. The chosen spatial domain of interest is \( \Omega_x = [-78, 78] \) and initially the considered final time is \( T = 5 \). Since the analytical solution is available, we use analytic boundary conditions. In figure 6a, the initial and final conditions of this problem are plotted. Additionally, the computed solution is visualised. The solution was computed using the basic EQ-algorithm on the
first four moments combined with the discontinuous Galerkin method of first order on a uniform grid consisting of 100 cells. To encode these configurations concisely, we will use the abbreviations depicted in the nearby table in this chapter.

The computed solution using EQ and the analytical solution, which are depicted in figure 6a, are visually in a perfect agreement. Despite the low number of moments used and the quite sparse discretisation, the error of the computed solution is very small indicating that EQ is well suited to model very smooth flow problems. Due to the visual agreement of the solutions, in the following, instead of analysing the solution visually, we will calculate the relative error of the density as:

\[
e(\rho) = \frac{||\rho - \rho^{(ref)}||_1}{||\rho^{(ref)}||_1},
\]

where \(\rho^{(ref)}\) denotes the density of the reference solution which is, in this case, the analytical solution of the problem.

In the following, we aim to analyse how \(e(\rho)\) depends on the parameters of the solver. We start by examining how the error changes when the number of transported moments is increased. Since the method of moments offers a hierarchy of approximations for the Boltzmann equation, it would be very desirable that the error decreases when the number of moments is increased. In fact, this behaviour can be observed in all conducted experiments. The error rates for different numbers of transported moments are presented in table 2.

In this table, the relative error of the solution at time \(T = 10\) and \(T = 20\) for the default EQ as well as some reference solutions are depicted. We will start by considering only the first two columns before turning to a general comparison of the four methods considered. For \(T = 10\), a substantial drop in the error of EQ can be observed in the case that the number of moments is increased from 4 to 6 and from 6 to 8. After that,
Convergence rate

Table 2: $e(\rho)$ of the solution calculated by different closures for the smooth perturbation test case for different numbers of moments with DG, $N_x = 100$, $D=1$, LLF

<table>
<thead>
<tr>
<th>M</th>
<th>EQ (T=10)</th>
<th>EQ (T=20)</th>
<th>QMOM (T=10)</th>
<th>QMOM (T=20)</th>
<th>EQX (T=10)</th>
<th>EQX (T=20)</th>
<th>EQ-ME (T=10)</th>
<th>EQ-ME (T=20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$8 \times 10^{-4}$</td>
<td>$6.0 \times 10^{-1}$</td>
<td>$3.9 \times 10^{-1}$</td>
<td>2</td>
<td>$1.1 \times 10^{-1}$</td>
<td>$6.8 \times 10^{-1}$</td>
<td>$3.2 \times 10^{-2}$</td>
<td>$7 \times 10^{-1}$</td>
</tr>
<tr>
<td>6</td>
<td>$2 \times 10^{-2}$</td>
<td>$1.1 \times 10^{-1}$</td>
<td>$2.7$</td>
<td>$1.4 \times 10^{-2}$</td>
<td>$1.4 \times 10^{-1}$</td>
<td>$5.1 \times 10^{-3}$</td>
<td>$1.8 \times 10^{-1}$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$4.7 \times 10^{-2}$</td>
<td>$8 \times 10^{-2}$</td>
<td>$3.3 \times 10^{-2}$</td>
<td>$8.1 \times 10^{-1}$</td>
<td>$4.5 \times 10^{-3}$</td>
<td>$7 \times 10^{-1}$</td>
<td>$2.8 \times 10^{-3}$</td>
<td>$5.4 \times 10^{-2}$</td>
</tr>
<tr>
<td>10</td>
<td>$2.3 \times 10^{-1}$</td>
<td>$3.1 \times 10^{-2}$</td>
<td>$9.8 \times 10^{-4}$</td>
<td>$5.3 \times 10^{-1}$</td>
<td>$2.3 \times 10^{-3}$</td>
<td>$2.9 \times 10^{-1}$</td>
<td>$2.2 \times 10^{-3}$</td>
<td>$1.7 \times 10^{-2}$</td>
</tr>
<tr>
<td>12</td>
<td>$1.8 \times 10^{-1}$</td>
<td>$1.4 \times 10^{-2}$</td>
<td>$3.0 \times 10^{-4}$</td>
<td>$3.3 \times 10^{-1}$</td>
<td>$1.8 \times 10^{-3}$</td>
<td>$1.2 \times 10^{-1}$</td>
<td>$1.9 \times 10^{-3}$</td>
<td>$4.9 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

Table 3: $e(\rho)$ and order of convergence of the DG method for various degrees with LLF, $T = 5$

<table>
<thead>
<tr>
<th>D</th>
<th>M</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>Convergence rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.9</td>
<td>$1.2 \times 10^{-2}$</td>
<td>$3.5 \times 10^{-3}$</td>
<td>$9.3 \times 10^{-4}$</td>
<td>1.8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.3</td>
<td>$1.9 \times 10^{-3}$</td>
<td>$2.7 \times 10^{-4}$</td>
<td>$3.3 \times 10^{-5}$</td>
<td>2.9</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.8</td>
<td>$1.5 \times 10^{-4}$</td>
<td>$5.9 \times 10^{-6}$</td>
<td>$5.7 \times 10^{-7}$</td>
<td>4.1</td>
<td></td>
</tr>
</tbody>
</table>

the change is no longer so drastic. Especially increasing the number of moments from 10 to 12 changes the error only slightly. There are different possible reasons for this observation. One hypothesis is that the usage of more than ten moments for EQ is no longer beneficial. However, the findings from the previous section and the measured error for the test case with $T = 20$ contradicts this assumption. For $T = 20$, the error of the solution computed using EQ decreases substantially even when increasing the number of moments from 10 to 12. It is, furthermore, important to note that for this quite long simulation the usage of only four or six moments is no longer sufficient to obtain a very accurate result (see also figure 6b). As a result, the number of moments considered should always be chosen depending on the desired degree of accuracy and the problem. However, the general trend that a higher number of moments leads to an increased precision can be found when analysing the results of this test case.

It remains to analyse why the error for $T = 10$ decreases only slightly when increasing the number of moments from 10 to 12. Another hypothesis to explain this behaviour is that the observed error is no longer an error introduced by the closure (e.g. in simplification (2) or (3)) but instead introduced by the used discretisation scheme. This hypothesis is supported by the error rates shown in figure 3 and 3. In these tables, the error of the computed solution for $M = 12$ and $T = 5$ combined with the discontinuous Galerkin method and the finite volume method for varying $D$ and $N_x$ as well as the order of convergence are presented. The observed order of convergence is optimal for the discontinuous Galerkin method and the finite volume method. This is a remarkable finding because it suggests that the error in the solution is mainly due to the discretisation scheme and that, in the case of 12 transported moments, EQ solves the Boltzmann equation nearly analytically in the considered test case. The fact that the velocity distributions which arise in the analytical solution of the problem are not in the form of a maximum-entropy distribution even strengthens the strength of this observation indicating that EQ is also well-suited to approximate general smooth functions.

As EQ can be interpreted as an extension to the QMOM, for the sake of comparability, we also conducted the described experiments using said closures. The measured error rates are included in 2. The most important insight of the comparison of both methods is
Table 4: $e(\rho)$ and order of convergence of the FV method for various degrees with KF, $T = 5$

<table>
<thead>
<tr>
<th>D</th>
<th>M</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>Convergence rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$2 \times 10^{-1}$</td>
<td>$1.1 \times 10^{-4}$</td>
<td>$5.8 \times 10^{-2}$</td>
<td>$3 \times 10^{-2}$</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>$6 \times 10^{-2}$</td>
<td>$1.9 \times 10^{-2}$</td>
<td>$5.2 \times 10^{-3}$</td>
<td>$1.2 \times 10^{-3}$</td>
<td>2.9</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>$1 \times 10^{-2}$</td>
<td>$1.6 \times 10^{-5}$</td>
<td>$2.2 \times 10^{-4}$</td>
<td>$2.7 \times 10^{-5}$</td>
<td>4.1</td>
</tr>
</tbody>
</table>

that, in all cases, EQ yields better results. One can argue that this comes at the cost of performing a costly multidimensional Newton method. However, the computing time only approximately doubles when EQ instead of the QMOM is used. The reason for this only very small increase is that depending on the chosen stopping criterion usually only 4 or 5 iterations of the Newton method are required. In this test case, achieving significantly better results while only doubling computation time, the usage of EQ instead of the QMOM is definitely beneficial. For $T = 20$, the QMOM even loses its validity when only 4 or 6 moments are considered. A comparison between the computed solution for this problem with $M = 6$ of both closures is shown in figure 6b. Furthermore, we can use the obtained results to address our question raised in chapter 5 whether EQ or the QMOM performs better when the same number of abscissas are constructed. An equal number of abscissas, however, requires the number of given moments for the QMOM to be twice as high as the number of moments given for EQ. Usually, one would expect that, in the described case, the QMOM always produces better results because the knowledge of an increased number of moments of a distribution offers important additional information. In practice, however, both approaches perform comparably in the case $T = 10$, whereas EQ outperforms the QMOM when the duration of the simulation is increased to $T = 20$. This observation is a strong indication that the degree of freedom introduced by EQ into the QMOM is filled in a very beneficial way and that the usage of the maximum-entropy condition is even advantageous over the knowledge of additional moments of the underlying distribution function.

Furthermore, on the other hand, it is interesting to compare the solution of EQ with the maximum-entropy solution. For the sake of saving computation time, instead of using a full-scale maximum-entropy closure, we just used EQ with a uniform quadrature and a multiplicative factor of six for the nodes. This approach is usually a quite good approximation of the maximum-entropy closure because, in the previous section, it has been shown that this version of EQ is capable of approximating the maximum-entropy distribution very accurately. Comparing the results of both version of EQ, the errors for $T = 10$ are in good agreement, whereas the denser version of EQ performs slightly better for $T = 20$. Nevertheless, these results indicate that the error of EQ and the ME are presumably in the same order of magnitude. However, concerning computation time, using EQ is advantageous because the computation time approximately doubles depending on the number of moments considered when the multiplicative factor 2 of the default EQ is increased to 6 for computing a more maximum-entropy like solution.

Figure 6b also includes the measured error for $E Q X^2$. Since the measured errors of EQ and $E Q X^2$ are nearly the same, our intuition that $E Q X^2$ is a well-suited approach to tackle realisability issues is confirmed. An additional remarkable observation is that $E Q X^2$ does not produce any negative weights during the whole simulation. However, if additional freedom by the introduction of additional nodes is inserted, the simulation immediately produces non-physical results.
Last but not least, an additional interesting question which we tackled using the presented test case is how the different numerical fluxes change the computed solution. Therefore, all experiments shown so far were also conducted using the global Lax-Friedrichs flux and the kinetic-based flux. In all cases, changing the flux changes the error only slightly (at maximum the error changes around 5%).

All in all, the performance of EQ in the considered test case is extraordinary. The measured errors are even in the case that the simulation lasts for a quite long time very low. In some cases, it seems that the main parts of the error are introduced by the discretisation scheme. Furthermore, in comparison to the QMOM, EQ has proven to be advantageous, while also performing only slightly worse than a dense approximation of EQ to the ME solution. To conclude, EQ has proven to be well-suited to fast and accurately describe equilibrium-flows.

7.2.2. Two-beam problem

Having studied the capabilities of EQ to describe equilibrium flows, we will now conduct different test cases involving strong non-equilibrium flows. In such test cases, it is usually, due to strong discontinuities in the solution, very difficult to distinguish between the error introduced by EQ and the error introduced by the numerical discretisation. Consequently, it is quite difficult to isolate the impact of only one simplification step or the influence of different parameters. Therefore, we will focus on the ability of the whole solver to compute strong non-equilibrium flows and consider only different numbers of given moments and different discretisation schemes. We start by examining the behaviour of EQ in a two-beam problem as it was done in [79] using the following initial conditions:

\[
f_0(x, c) = \begin{cases} 
  f_M(c; 1, v_0, 1), & x \leq 0 \\
  f_M(c; 1, -v_0, 1), & x > 0 
\end{cases}
\]

Using (7.1), it is possible to show that:

\[
\rho(x, t) = \frac{1}{2}(1 + \text{erf}\left(\frac{v_0 + \frac{x}{2}}{\sqrt{2}}\right)\text{erfc}\left(\frac{-v_0 + \frac{x}{2}}{\sqrt{2}}\right)).
\]

In the following, we will consider a standard domain \(\Omega_x = [-1, 1]\) with \(T = 0.2\). For the sake of readability, the visualisation of the computed solution has been outsourced to appendix A. We start our analysis by choosing \(v_0 = 0.5\) because for this test case both the discontinuous Galerkin method as well as the finite volume method are able to produce stable results.

In figure 8 and 9 the results of the solver using the discontinuous Galerkin method and the finite volume method are depicted. Comparing both solutions, one could gain the surprising impression that the finite volume method seems to produce significant better results than the discontinuous Galerkin method. However, these improved results are only due to the well-known smearing effects which occur when a first order finite volume scheme in the presence of a discontinuous solution is used. This error introduced by the numerical scheme balances out with the error introduced by the closure and leads to the quite accurate result. Nevertheless, such balancing-effects are not desirable because the balancing is very problem dependent and disturbs the analysis of the closure. The assumption that strong smearing effects take place in the solution of the finite-volume method is also supported by considering the solution of higher order finite volume schemes which are visualised in figure 10. Here, the higher order finite volume schemes prevent the smearing effects thereby reducing the accuracy of the solution and making it more
similar to the one computed using the discontinuous Galerkin method. It is especially worth noting that the first order finite volume method using a local Lax-Friedrichs flux produces nearly perfect results due to the very strong smearing effects thereby highlighting the importance of a closer examination of obtained results.

Turning now to the analysis of EQ, besides problems of the chosen numerical discretisation, EQ produces always a stable solution. This insight is supported by the stable solution of EQ of the physically harder two-beam problem with \( v_0 = 2 \), which is depicted in figure 11.

Furthermore, the calculated solutions capture the basic form of the analytical solution. However, the frequently observable pattern of different packages of particles travelling at the same speed, which results in a piecewise constant but discontinuous solution, can be observed. Therefore, the solution calculated using EQ can be interpreted as a staircase function. The staircases are especially clearly visible in the solution calculated using the discontinuous Galerkin method. To show the form of the computed solution in the case that a very accurate numerical scheme is used we also included a solution obtained using a second-order discontinuous Galerkin method limited with a Biswas-limiter. Concerning the accuracy of the calculated solution, despite the discontinuities in the computed solution, it is still possible to obtain a reasonably accurate approximation of the solution when the number of moments given is increased. In all depicted results, including additional moments results in an increased number of stairs in the computed staircase function and thereby in an increased accuracy of the approximation.

All in all, the conducted experiments indicate that EQ is also capable of simulating strong non-equilibrium flows with the desirable property that the error decreases in the case that the number of moments given is increased. Nevertheless, the spatial discretisation scheme for such problems involving strong discontinuities should be chosen with great care keeping in mind that they may significantly change the calculated solution. As the QMOM breaks down in the considered test case, the results obtained using EQ indicate an increased capability of this closure to describe non-equilibrium flows. An additional standard test case to further examine the capabilities of EQ to model non-equilibrium flows is conducted in appendix F.

7.2.3. Stationary shock

Additional to the two non-stationary test cases considered above, we will now also consider a special version of the Riemann problem: a stationary shock profile. This test case is very popular for examining different closures of the method of moments because, since the shock is steady, the computed solution can be observed very comfortably. [64, 95]. In the stationary shock test case, two equilibrium states are connected via a stationary shock. The equilibrium state before the shock is called the upstream region and is defined by \( \rho_0, v_0, \theta_0 \), whereas the equilibrium state after the shock is called downstream region and is defined by \( \rho_1, v_1, \theta_1 \). To set up a stationary shock problem, it is important to choose the six parameters in a way that the resulting system is in a steady state, e.g. the temporal derivatives of the macroscopic quantities need to be zero. Since the collision operator conserves mass, momentum and energy, the corresponding spatial derivatives have to be always zero. Therefore, it is necessary that the terms, from which the spatial derivatives are taken, are constant over the domain of interest and especially at the point between the two regions. For example, to ensure that the density is steady, the mass flux before and after the shock needs to be the same, e.g. \( \rho_0 v_0 = \rho_1 v_1 \). In a similar manner, to ensure that all involved terms are constant, it is possible to set up a system of three
equalities which gives us a framework to determine the six parameters. This system of equalities determining which equilibrium conditions can be connected by a steady shock are the well-known Rankine–Hugoniot conditions of the Euler equations. Fortunately, the Rankine–Hugoniot conditions provide a parametrised description of these states. Apart from some scaling, the most important parameter which one can choose is the velocity of the upstream region determining the speed of incoming particles which can be observed at the shock. It is also possible to reformulate the parametrised description in terms of the Mach number of the resulting shock problem which acts as a parameter for the strength of the shock by relating said number to the velocity in the upstream region.

In one-dimension, the Rankine–Hugoniot conditions result in the following parametrised set of parameters for the equilibrium states in the downstream and the upstream region:

\[
\begin{align*}
\rho_0 &= 1, \quad v_0 = \sqrt{\frac{5}{3}} M, \quad \theta_0 = 1 \\
\rho_1 &= \frac{10M^2}{9 + 5M^2}, \quad v_1 = \sqrt{\frac{5M^2 + 9}{30M}}, \quad \theta_1 = \frac{(5M^2 - 3)(5M^2 + 9)}{60M^2}
\end{align*}
\]

Since the initial conditions are discontinuous, the solution produces some very small sub shocks before reaching a steady state. To simplify the start of the simulation and prevent possible problems of the discretisation scheme with discontinuities, the initial conditions of each macroscopic quantity \( \eta \) are smoothed using: \( \eta(x) = \eta_0 + (\eta_1 - \eta_0) * \frac{\tanh(\frac{x}{\epsilon}) + 1}{2} \). In the presented experiments, we choose to use \( \epsilon = 0.1 \). Analogous to [64], we computed the steady shock for three different Mach numbers (\( M = 2, M = 4 \), and \( M = 8 \)). For the computation, we set \( Kn = 0.05 \). In this test case, however, the Knudsen number determines only the width of the steady shock and therefore up to a rescaling of the axis, the results are for all Knudsen numbers the same.

In the conducted simulations, reflecting boundary conditions are used to enable the small sub shocks to flow out of the domain and the whole simulation to converge to the steady shock. Furthermore, we used a simple first-order finite volume method, as this test case involves discontinuities and difficult physical conditions. Consequently, we cannot hope for higher than first-order accuracy at the shock, and this is already achieved by the finite volume method. Furthermore, the first-order finite volume method is usually more stable in the presence of discontinuities, strong local extrema and sub-shocks and does not require the adjustment of complex limiters. We used a uniform grid consisting of 200 elements on the domain \([-1, 1]\) for \( Ma = 2 \) and \( Ma = 4 \) and 400 elements for \([-2, 2]\) for \( Ma = 8 \) because the width of the stationary shock increases with increasing Mach number. For the numerical flux, we used the local Lax-Friedrichs flux.

In appendix A in 12, 13 and 14, visualisations of the calculated shocks for the three different Mach numbers considered are presented. In the conducted experiments the solution converges to the stationary shock around \( T = 1 \) and remains unchanged until the end of the simulation at \( T = 1.5 \).

Turning to the analysis of the computed solution, first of all, the fact that EQ is able to produce a steady solution for such strong stationary shocks considered in this section is a clear indicator that this closure is very stable and robust. Furthermore, the obtained results are also quite accurate. For \( Ma = 2 \), the solutions for \( M = 6 \) and \( M = 10 \) are very similar indicating that they have converged to the actual solution of the problem. Moreover, they are visually in a very good agreement with the figures in [64]. Even EQ based on four moments produces quite similar results already showing the fundamental properties of the solution.
For $Ma = 4$, the computed solutions deviate a little bit more. For four moments, the solver produces a small non-physical discontinuity at the beginning of the shock. Apart from this discontinuity, however, the produced result is in a good agreement with the computed solution for $M = 6$. The solution for ten moments considered is again slightly different. Nevertheless, they are all visually in good agreement [64] and the small changes indicate a convergence to the actual solution of the problem and a close fit of this solution with the computed solutions. For $Ma = 8$ with $M = 10$ the simulation encounters some problems because the 10th moment of the corresponding equilibrium distribution is very large (e.g. it is initially up to $1 \times 10^{22}$) thereby causing some problems for the necessary numerics. Furthermore, using a Lax-Friedrichs flux, the moments encounter some realisability problems. As a result, we use the kinetic-based flux in combination with the finite volume method because the resulting scheme is realisability-preserving. The calculated solutions are shown in 14 for $M = 4$ and $M = 6$. The solution computed from four moments exhibits again a small discontinuity before the shock, whereas for six moments this discontinuity is removed, while the rest of the solution is in a quite good agreement.

All in all, the ability of EQ to compute stationary shocks up to $Ma = 8$ which involves strong non-equilibrium flows points toward the surpassing capabilities of this method to handle strong non-equilibrium fluxes. It is, furthermore, interesting to note that the QMOM is not capable of simulating such flows and breaks down in the computation of the solution of the presented stationary shock test case.

7.2.4. Smooth perturbation in two dimensions

In the following, we will shortly extend our analysis of the capabilities of EQ to two dimensions. Analogous to the one dimensional test case, we consider again a gas which is locally in a thermodynamic equilibrium with a perturbed density. The chosen initial conditions are:

$$f_0(x, \mathbf{c}) = f_M(x, \mathbf{c}; \rho_0(x, y), \mathbf{v}_0(x, y), \theta_0(x, y))$$

$$\rho_0(x) = 1 + \mathcal{N}(x; 0.5, 0.02)\mathcal{N}(y; 0.5, 0.02), \mathbf{v}_0(x) = 0, \theta_0(x) = 1$$

on $\Omega = [0,1] \times [0,1]$ discretised using a Cartesian grid with $T = 0.05$. The main findings of the one-dimensional smooth perturbation test case also hold true in the considered two-dimensional test case. For example, it is possible to observe a substantial decrease in the error when the number of moments is increased. To ultimately prove that the error introduced by EQ is negligible, as in one dimension, we also aimed to observe the order of convergence of the discretisation scheme in the error. Therefore, the errors of the computed density in infinity norm for different cell width are plotted in figure 7 using logarithmic scales.

```
Figure 7: Error of the density in the infinity norm calculated using EQ and DG with various degrees in two dimensions with $M = 21$
```
to easily observe the underlying order of convergence. For the first order discontinuous Galerkin scheme, the results are extraordinary, since the theoretical optimal second-order convergence is even outdone decreasing to a minimal error of around $1 \times 10^{-5}$ per cell. This observed order of convergence and the very low error indicate that the error introduced in the solution of the Boltzmann equation is presumably mainly due to the numerical discretisation. However, aiming to achieve a third order convergence using a second order discontinuous Galerkin method, we encountered some problems. There exist different possible explanations for this problem. For example, it is possible that the Lax-Friedrichs flux loses its similarity to an upwind flux when the number of nodes on each boundary increases. Moreover, also in one-dimensions, we observed for some test cases that the first order discontinuous Galerkin method outperforms the second order one (table 7). Additionally, it is also possible that the closure-error accounts for the observed error and that the solution computed with the second order discontinuous Galerkin method is optimal given the number of moments considered.

All in all, despite the problem with the second order DG method, it can be concluded that EQ is also well-suited to model equilibrium flows in multiple dimensions. Since the measured error is very small and during the simulation no problems of EQ with sets of moments which cannot be fitted arise, it can be concluded that EQ calculates also in two dimensions very fast and reliable quadrature-based approximations of smooth distribution functions. Consequently, with a very high probability, the abilities of EQ to capture strong non-equilibrium flows generalise to two dimensions, too. However, the implementation of a solver for said cases is much more difficult compared to one dimension because the necessary numerical discretisation techniques require advanced ideas such as multidimensional limiters.

8. Conclusion

In this thesis, a new closure for the Boltzmann equation, which we named entropic quadrature, has been developed. This closure aims at combining the strong theoretical justification of the maximum-entropy closure with the efficiency of the quadrature method of moments. As argued in section 5.3.1, since EQ shares important properties with both closures, EQ can indeed be interpreted as a combination of the two. Especially the interpretation of EQ as a sparse version of the maximum-entropy closure and as an extended version of the QMOM has witnessed a great attention in the course of this thesis. Consequently, concerning its theoretical properties, EQ fits well into the set of already proposed closures. In chapter 7, several experiments examining the capabilities of this closure to model equilibrium and non-equilibrium flows have been conducted. Since EQ has proven to be able to simulate equilibrium flows very accurately and capture non-equilibrium flows, the method of entropic quadrature appears to be a very promising closure for the Boltzmann equation in different regimes. Nevertheless, additional experiments and comparisons to other closures are necessary to further justify this conclusion. All in all, by introducing a small additional degree of freedom in the QMOM and filling it with the maximum-entropy condition, EQ was able to extend the accuracy and validity of the QMOM significantly.

One of the most desirable properties of the described solver is that its accuracy can be increased in multiple ways thereby offering a very configurable approximation of the Boltzmann equation. First of all, EQ fits well into the hierarchy of approximation offered by the method of moments. The significant improvement of accuracy that can be achieved by considering a higher number of moments is one of the main findings in chapter
Moreover, an additional important parameter controlling the sparsity of the computed quadrature and thereby its computational cost is the number of additional inserted nodes. By increasing said number, the difference between EQ and the maximum-entropy closure can be decreased while increasing the accuracy of the closure. Moreover, by combining EQ with the Runge-Kutta discontinuous Galerkin method, which is capable of achieving any order of desired accuracy, the described solver is also able to attain arbitrary accuracy concerning the spatial and temporal discretisation. Hence, by controlling these parameters, the accuracy of the proposed closure can be adaptively increased to any desired degree at the expense of an increased computational cost.

The opportunities concerning future work on this closure are manifold. First of all, experiments further examining the performance and capabilities of this closure in two dimensions and testing its capabilities in three dimensions need to be conducted. The extension to three dimensions is directly possible and using EQ instead of ME is especially advantageous because, in three dimensions, the number of problems of moments which need to be solved increases drastically. To make the usage of EQ in two and three dimensions more affordable by decreasing its computation time and increasing its stability, it would be beneficial to examine the numerical algorithms involved in the calculation of the maximum-entropy weights more closely and optimise them where it is possible. Furthermore parallelising the presented solver and using GPUs to speed up computations is also a very promising approach because the utilised discontinuous Galerkin method is of a highly local nature and solving different problems of moments using EQ can be easily done in parallel. For a possible guideline of how such an optimisation of the solver can proceed see [81] where this task was addressed for the maximum-entropy closure. Besides the application of EQ to the three-dimensional version of (3.1), due to the flexibility and sparsity of the quadrature-based reconstruction, EQ should also be well-suited to solve the Boltzmann equation including, for example, fluid drag terms and complex models of the collision operator.

Moreover, to understand EQ and its capabilities in more detail, further basic research on EQ needs to be conducted. First of all, the ideas, how to deal with non-realisable sets of moments and multidimensional sets of moments, where EQ is initially not able to find a fitting set of non-negative weights, which were outlined in section 5.4, need a systematic evaluation. Building upon the ideas from section 5.4, we can imagine that the problems of EQ in multi-dimensions can be overcome and that EQ can make a valuable contribution to the problem of non-realisable sets of moments. The significant advantage of EQ to tackle the problem of non-realisable moments, which has witnessed an increased attention in the field of computational fluid dynamics [4], is the sparseness and flexibility of the quadrature-based reconstruction.

Another important future task is to further examine the freedom in EQ concerning the considered set of nodes. In section 7.1.1 we have already discussed the advantages and disadvantages of the two algorithms for calculating the EQ-nodes which were used in this thesis to some extent. Nevertheless, a further in-depth comparison could lead to useful insights into the properties of EQ. The development of new algorithms to determine the EQ-nodes is also promising.

Having shown that EQ is a suitable closure for the Boltzmann equation, it could also be beneficial to apply this closure to problems of moments arising in different fields. Generally speaking, since they have similar properties, it is possible to replace the maximum-entropy closure with the EQ in all its applications. Fortunately, the maximum-entropy closure has already been applied to problems of moments arising in different disciplines thereby offering a range of different possible applications of EQ. Closely related to the application
considered here, the maximum entropy closure has also been successfully applied to the
collection balance equation [5, 53, 72, 98] which is interestingly also one of the most
important applications of the QMOM [60, 60, 62]. Both closures of said equation have
been already compared in [70]. The success of the maximum-entropy closure and the
QMOM in this field points towards a possible very promising application of EQ to solve
said equation.

Additionally, the maximum-entropy closure has been applied in the field of econometrics
[105], hydrology and water resources [88], and ensemble forecasting [84]. Further possible
applications can be found in [58, 69]. All these applications of the maximum-entropy
closure are also possible applications of the proposed method of entropic quadrature and
deserve further examination.
A. Visualisation of the numerical results

A.1. Two-beam problem

Figure 8: Solutions of the two-beam problem with \( v_0 = 0.5 \) calculated using the discontinuous Galerkin method
Figure 9: Solutions of the two-beam problem with $v_0 = 0.5$ calculated using the finite volume method
Figure 10: Comparison of different finite volume schemes for solving the two-beam problem with \( v_0 = 0.5 \)

Figure 11: Solutions of the two-beam problem with \( v_0 = 2 \) calculated using the finite volume method
A.2. Stationary shock

Figure 12: Stationary shock profile for Ma=2; FV, D=1, LLF, \( N_x = 200 \)

Figure 13: Stationary shock profile for Ma=4; FV, D=1, LLF, \( N_x = 200 \)
Figure 14: Stationary shock profile for $Ma=8$; FV, D=1, KF, $N_x = 400$
B. Infeasible Newton method

B.1. Minimising a quadratic function under linear constraints

As mentioned in section 5.2.1, as part of the infeasible Newton method, we will have to minimise a quadratic function under linear constraints. This problem arises in the following form:

\[
\begin{align*}
\text{arg min}_{x} & \quad \frac{1}{2}x^TPx + p^Tx + c \\
\text{subject to} & \quad Ax = b
\end{align*}
\]

(B.1)

where \(c \in \mathbb{R}, x \in \mathbb{R}^n, p \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}, P \in \mathbb{R}^{n \times n}\) with \(m < n\), \(\text{rank}(A) = m\) and \(P\) being a diagonal matrix with positive entries. In this case, (B.1) is convex and the remaining part is to find the local and, consequently, global minimum of (B.1). We will solve (B.1) using the method of Lagrange multipliers. Therefore, as a first step, we define the Lagrange function of (B.1) as:

\[
L(x, \nu) = \frac{1}{2}x^TPx + p^Tx + c + (Ax - b)^T\nu
\]

with \(\nu \in \mathbb{R}^m\). After that, theory tells us that the minimum of the constrained optimisation problem (B.1) is also a critical point of the Lagrange function [13]. To calculate the critical point of \(L(x, \nu)\), we set the partial derivatives of the function equal to zero thereby forming a system of linear equations:

\[
\begin{align*}
\delta x: & \quad Px + A^T\nu = 0 \\
\delta \nu: & \quad Ax = b
\end{align*}
\]

This system is well known as the Karush-Kuhn-Tucker system and can be put in matrix notation:

\[
\begin{pmatrix}
P & A^T \\
A & 0
\end{pmatrix}
\begin{pmatrix}
x \\
\nu
\end{pmatrix}
= \begin{pmatrix}
-p \\
b
\end{pmatrix}
\]

(B.2)

where the first matrix is called the KKT matrix. To solve this system efficiently, we transform the system by multiplying both sides with \(D = \text{diag}(\sqrt{P_{11}}, ..., \sqrt{P_{nn}}, 1, ..., 1) \in \mathbb{R}^{n+m\times n+m}\) and inserting \(D^{-1}D\) between the KKT matrix and the vector of unknowns. The resulting system can be written as:

\[
D\begin{pmatrix}
P & A^T \\
A & 0
\end{pmatrix}D^{-1}D\begin{pmatrix}
x \\
\nu
\end{pmatrix} = D\begin{pmatrix}
-p \\
b
\end{pmatrix}
\]

which can be rewritten to:

\[
\begin{pmatrix}
I & \tilde{A}^T \\
\tilde{A} & 0
\end{pmatrix}
\begin{pmatrix}
\tilde{x} \\
\nu
\end{pmatrix}
= \begin{pmatrix}
-p \\
b
\end{pmatrix}
\]

(B.3)

with \(\tilde{p} = \text{diag}(\sqrt{P_{11}^{-1}}, ..., \sqrt{P_{nn}^{-1}}) \ast p, \tilde{x} = \text{diag}(\sqrt{P_{11}^{-1}}, ..., \sqrt{P_{nn}^{-1}}) \ast x\) and \(\tilde{A}^T = \text{diag}(\sqrt{P_{11}^{-1}}, ..., \sqrt{P_{nn}^{-1}}) \ast A^T\). From this system, the following equalities can be extracted:

\[
\tilde{x} + \tilde{A}^T\nu = -\tilde{p}, \tilde{A}\tilde{x} = b
\]

Substituting the first equality into the second one leads to:

\[
-\tilde{A}\tilde{p} - \tilde{A}\tilde{A}^T\nu = b \iff \tilde{A}\tilde{A}^T\nu = -\tilde{A}\tilde{p} - b \iff \nu = (\tilde{A}\tilde{A}^T)^{-1}(-\tilde{A}\tilde{p} - b).
\]

Consequently, solving (B.3) requires the inversion of \(\tilde{A}\tilde{A}^T\). Fortunately, \(\tilde{A}\tilde{A}^T\) is symmetric positive definite because \(A\) has full rank and can be inverted by the stable and fast Cholesky decomposition. As a final step, the calculated solution \(\nu\) can be plugged into the first equality to obtain the solution \(x\). All in all, one step of this method requires only the inversion of a positive definite \(m \times m\) matrix.
B.2. Infeasible Newton method

Using the method for solving quadratic functions under linear constraints presented above, we will now provide a detailed description of how to solve (5.6).

(5.6) can be solved by using the method presented in the previous section because it is in the form of (B.1) and the derivatives of the corresponding Lagrange function can be written in the form of (B.2):

\[
\begin{pmatrix}
\frac{1}{w_1} & 0 & \ldots & 0 \\
0 & \frac{1}{w_2} & \ldots & 0 \\
\vdots & 0 & \ddots & \vdots \\
0 & \ldots & 0 & \frac{1}{w_n}
\end{pmatrix} \begin{pmatrix} x_0 \\
x_1 \\
x_2 \\
\vdots \\
x_n \\
\nu_0 \\
\nu_1 \\
\vdots \\
\nu_m
\end{pmatrix} + A^T \begin{pmatrix}
-1 - \log w_1 \\
-1 - \log w_2 \\
-1 - \log w_3 \\
\vdots \\
-1 - \log w_n \\
\Delta b_0 \\
\Delta b_1 \\
\vdots \\
\Delta b_m
\end{pmatrix}
\]

where the constraint \(Ax = \Delta b\) is used with \(\Delta b = b - Aw^{(i)}\).

All in all, the infeasible Newton method has many desirable properties. For example, no feasible solution is required as a starting point while making effective use of a good initial guess. Furthermore, assuming that a feasible solution exists, this method converges and usually can even achieve locally quadratic convergence [13]. Unfortunately, the restricted domain of the entropy function poses a serious challenge to the infeasible Newton method.

Since the domain is restricted to the positive real numbers, we have to ensure that after each update step the weights remain positive. Therefore, in the update step, a damping factor \(\eta\) is added which should be chosen in a way that \(w^{(i+1)} = w^{(i)} + \eta x\) remains non-negative.

Since the method starts at an infeasible point, in contrast to the feasible Newton method, it cannot be proven that each step \(x\) goes in a descent direction of \(f\). Instead, the progress of the method can be measured by using the following two residuals:

\[r_{\text{dual}}(w, \lambda) = \nabla f(x) + A^T \lambda, \quad r_{\text{primal}}(w, \lambda) = Aw - b,\]

where \(\lambda\) denotes the Lagrange multipliers in the Lagrange function of the original problem (5.5) Note that \((w^*, \lambda^*)\) is the solution of (5.5) if the gradient of the Lagrange function (5.10) is zero at \((w^*, \lambda^*)\) which is equal to:

\[r(w^*, \lambda^*) = (r_{\text{dual}}(w^*, \lambda^*), r_{\text{primal}}(w^*, \lambda^*)) = (0, 0).\]

Using these residuals, it is possible to show that the method progresses in each step by showing a decrease of the norm of the two residuals in the direction of a Newton step:

\[
\frac{d}{d\eta} \left| r(w + \eta x, \lambda + \eta \Delta \lambda) \right|_2^2 \bigg|_{\eta = 0} = -2r(w, \lambda)^T r(w, \lambda).
\]

This can be rewritten to:

\[
\frac{d}{d\eta} \left| r(w + \eta x, \lambda + \eta \Delta \lambda) \right|_2 \bigg|_{\eta = 0} = -\|r(w, \lambda)\|_2 \leq 0.
\]

Consequently, it is possible to use \(\|r(w, \lambda)\|_2\) as a measure of the progress of this algorithm.
The last step remaining is to define the method which calculates $\eta$ and to define the stopping criterion. Concerning the calculation of $\eta$, to make the method more efficient and make use of the necessary damping factor, we use a backtracking line search to determine $\eta$. A backtracking line search aims to find a damping factor to ensure that the resulting step width has reasonable length and that the updated solution remains in the domain of the function.

**Algorithm 6:** Backtracking line search

**Data:** Current point $w \in \text{dom}f$, update $x$ in descending direction, $\alpha \in (0,0.5)$ and $\beta \in (0,1)$

1. $\eta := 1$
2. while $w + \eta x \notin \text{dom}f$ or $f(w + \eta x) > f(w) + \alpha \eta \nabla f(w)^T x$ do
3. \[ \eta := \beta \eta \]
4. end

To judge the quality of a step and the corresponding step size, this method compares the value of the updated point with the predicted value of the point using a linear approximation of the function at the current point. In the case that both values differ too much (e.g. using the calculated gradient at the current point, the value of the updated point would be predicted to be much lower than it actually is), the step width is decreased until the condition is no longer fulfilled. In this method, the parameter $\alpha$ defines which difference between the actual and the predicted value is considered significant, while $\beta$ defines the fineness of the search.

For both the stopping criterion and the backtracking line search, we will use $r(x,\lambda)$ because this quantity defines the progress of the method. All in all, with $\Delta \lambda := \nu - \lambda$ the infeasible Newton can be summarised as:

**Algorithm 7:** Infeasible Newton method

**Data:** Start point $w^{(0)} \in \text{dom}f_0$, $\lambda^{(0)}$, tolerance $\epsilon > 0$, $\alpha \in (0,0.5)$ and $\beta \in (0,1)$

1. while not ($A w = b$ and $\|r(w^{(i)},\lambda^{(i)})\|_2 \leq \epsilon$) do
2. \[ \eta := 1; \quad \text{▷ Backtracking line search to calculate } \eta \]
3. \[ \text{while } w^{(i)} + \eta x < 0 \text{ or } \|r(w^{(i)} + \eta x,\lambda^{(i)} + \eta \Delta \lambda)\|_2 > (1 - \alpha \eta)\|r(w,\lambda)\|_2 \text{ do} \]
4. \[ \eta := \beta \eta \]
5. end
6. Update: $w^{(i+1)} := w^{(i)} + \eta x$ and $\lambda^{(i+1)} := \lambda^{(i)} + \eta \Delta \lambda$
7. end

Note that, despite starting at an infeasible point, all points in the sequence $w^{(0)},...,w^{(n)}$ will be feasible once a full step ($\eta = 1$) is taken and that then the infeasible Newton method reduces to the feasible Newton method with a direct convergence of $f(w^{(i)})$. Furthermore, assuming that the problem is feasible (there exists a positive $w$ with $A w = b$) it can be proven that algorithm 7 always converges to the solution of (5.5) [13]. This can be easily done by recalling that $\|r\|_2$ decreases at each step.
C. Further on Finite volume methods

C.1. Reconstruction techniques: Higher order finite volume methods

In the following, different techniques to reconstruct $u_{j+\frac{1}{2}}$ from the given cell averages will be presented. It is important to consider those reconstruction techniques because they increase the formal order of convergence of the considered finite volume method.

The most simple reconstruction, which has been already mentioned above, assumes that $u_{h,i}(x)$ is constant on each cell. Assuming that this property holds, the first order finite volume method can be obtained by choosing $u_{j+\frac{1}{2}} = u_j$. Despite its low order of convergence, this technique has various desirable properties. Firstly, simplifying computations, usually, some of the terms in the full description of the finite volume method cancel out. Secondly, this method is realisability-preserving as it will be discussed in the next section. Thirdly, the first order finite volume method is very stable but suffers from a loss of accuracy, for example, for simulations involving strong discontinuities [55].

To increase the formal order of convergence, we cannot hope for higher than first order accuracy. In smooth regions, however, we want to keep the slopes unchanged aiming to achieve second order accuracy. Consequently, above, we need for detecting discontinuities in the solution and, subsequently, setting the corresponding slopes to zero. Fortunately, this problem is well-studied and a range of different measures, called slope limiters, have been developed. The most famous and usually used one is [55]:

$$\minmod limiter: \sigma_i^n = \minmod\left(\frac{u_i^n - u_{i-1}^n}{\Delta x}, \frac{u_{i+1}^n - u_i^n}{\Delta x}\right),$$

Unfortunately, these slopes are only stable when the solution is sufficiently smooth [55]. To obtain this insight, consider an advection equation with a piecewise constant but discontinuous initial data with $u_{j-1} = 1$, $u_j = 1$, $u_{j+1} = 0$ and $\Delta x = 1$ [55]. Using an downwind slope results in $\sigma_j^0 = -1$ and consequently, in a non-physical value at the left interface of $I_j$ as the reconstructed value on the left interface is: $u_{j-\frac{1}{2}} = 1 - 1 \times (-0.5) = 1.5$. This overshoot in the reconstruction will lead to values higher than one for $u_j^1$, whereas the value of $u_{j-1}$ will become less than one in the following timesteps. In fact, in the outlined scenario, $u_j^1$ will be always larger than one unless $\sigma_j^0 = 0$. The oscillation inserted by this discontinuity will grow over time becoming worse and worse [55].

To be able to mimic also non-smooth situations, it is, therefore, necessary to limit the slope $\sigma_i$ by setting it to zero when discontinuities occur. At discontinuities, as explained above, we cannot hope for higher than first order accuracy. In smooth regions, however, we want to keep the slopes unchanged aiming to achieve second order accuracy. Consequently, there is a need for detecting discontinuities in the solution and, subsequently, setting the corresponding slopes to zero. Fortunately, this problem is well-studied and a range of different measures, called slope limiters, have been developed. The most famous and usually used one is [55]:

$$\minmod limiter: \sigma_i^n = \minmod\left(\frac{u_i^n - u_{i-1}^n}{\Delta x}, \frac{u_{i+1}^n - u_i^n}{\Delta x}\right),$$

83
where the minmod function is defined as:

$$\text{minmod}(a, b) = \begin{cases} 
\text{sign}(a) \min(|a|, |b|) & \text{if } \text{sign}(a) = \text{sign}(b) \\
0 & \text{otherwise}
\end{cases}$$

In this limiter, instead of always using the upwind or downwind flux, both fluxes are compared. In the case that both slopes have different signs, the limiter concludes that a local maximum or minimum, for example in the form of a discontinuity, is present and consequently sets the slope to zero. In the other case, returning the minimum of both slopes the limiter is quite conservative. Usually, using a minmod limiter results in a quite stable solution with second order accuracy in smooth regions [55]. However, the chosen slope can sometimes be reasonably increased leaving room for sharper reconstruction with a higher accuracy for non-smooth solutions [55]. One possible limiter, which calculates sharper slopes for non-smooth regions, while keeping second order accuracy for smooth regions, is:

superbee limiter [76] : \( \sigma_i^n = \maxmod(\sigma_i^{(1)}, \sigma_i^{(2)}) \),

where the maxmod function is defined analogous to the minmod function with the only difference that in the case that both slopes have identical signs the maximum slope is returned and:

\[
\sigma_i^{(1)} = \minmod\left( \frac{u_{i+1}^n - u_i^n}{\Delta x}, 2\frac{u_i^n - u_{i-1}^n}{\Delta x} \right) \\
\sigma_i^{(2)} = \minmod\left( 2\frac{u_{i+1}^n - u_i^n}{\Delta x}, \frac{u_i^n - u_{i-1}^n}{\Delta x} \right)
\]

In this limiter, each of the one sided fluxes are compared to twice the other flux. This limiter usually returns the larger of the two slopes in smooth regions. In non-smooth regions, very sharp slopes are computed preventing a smearing of the discontinuity. Sometimes, this limiter even connects discontinuities over two cells. However, the superbee method comes at the disadvantage of reconstructing occasionally too steepen slopes.

A limiter which aims to overcome the disadvantages of both limiters is the monotonized central-difference limiter (MC limiter) [99] :

MC limiter : \( \minmod\left( \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}, 2\frac{u_i^n - u_{i-1}^n}{\Delta x}, 2\frac{u_{i+1}^n - u_i^n}{\Delta x} \right) \)

This limiter usually returns the centred slope in smooth regions, which usually provides a very accurate description. At the same time, at discontinuities, to prevent smearing effects larger slopes compared to the minmod limiter are returned. LeVeque [55] suggest to use this limiter as a default because this is a compromise between the other two.

Aiming to explore the limits of the finite volume method in smooth situations, a non-limited third order reconstruction will also be used in our simulations. In this method, the values in cell \( j \) are reconstructed using a three point stencil assuming that on cell \( j \) \( u_h(x) \) can be approximated by a quadratic function:

\[
u_h(x) = a_j + b_j(x - x_j) + \frac{c_j}{2}(x - x_j)^2.
\]

To derive the coefficients, we enforce that the average of \( u_h(x) \) in \( I_j \) is preserved, while the derivatives of \( u_h(x) \) at the interfaces should be approximated with second order accuracy [19]. This leads to:

\[
u_{j-\frac{1}{2}}^n = \frac{1}{3}(u_j^n - u_{j-1}^n) + \frac{1}{6}(u_{j+1}^n - u_j^n),
\]
\[ u^n_{j+\frac{1}{2}} = u^n_j - \left[ \frac{1}{6}(u^n_j - u^n_{j-1}) + \frac{1}{3}(u^n_{j+1} - u^n_j) \right]. \]

One important remark that has been made by Vikas et al. [101] in the context of proposing the QMOM is that, using a kinetic-based flux, there is some ambiguity concerning the variables which are reconstructed at the cell interfaces. Following the presented approach, the moments at the cell interfaces are reconstructed. Subsequently, these moments are used to obtain a quadrature-based reconstruction of the velocity function at the interface. Aiming to reduce the computational cost and make the algorithm more stable, it is also possible to directly reconstruct the quadrature at the cell interfaces using the calculated quadratures within a small stencil. The only difference to the scheme above is that, instead of \( u_j, w_j \) and \( \gamma_j \) are reconstructed. In this method, both reconstructions proceed independently. The corresponding formulas can be obtained by replacing \( u_j \) in the component-wise reconstruction described earlier directly by \( w_j \) and \( \gamma_j \) (see [101] for further details).

### C.2. Realisability

As already mentioned in section 5.4, one big concern of most closure methods for the method of moments is that they fail when the moments become non-realisable. Furthermore, it was also mentioned that a set of moments can become non-realisable due to very small changes. These changes can, for example, occur due to some errors introduced by the numerical scheme. Therefore, to prove that a scheme is realisability-preserving is a very desirable and strong property. Unfortunately, this is only possible for some finite volume methods and impossible for higher order methods such as the discontinuous Galerkin method.

Following the reasoning of [28, 101], it will be argued that the first order finite volume method relying on a kinetic-based flux combined with a Euler time-stepping scheme guarantees the non-negativity of the first moment and even preserves realisability. Furthermore, it will be shortly explained that it is possible to derive a quasi-second order method with the same properties.

We start by proofing that the first order finite volume method using a kinetic-based flux can guarantee the non-negativity of the first moment. Therefore, we assume that \( u^n_{i,1} \) is non-negative. Our aim is to shown that given this assumption \( u^{n+1}_{i,1} \) is also non-negative. To do so, let use first recall that in a first order scheme \((\gamma^r_{x_{i-\frac{1}{2}}}, w^r_{x_{i-\frac{1}{2}}}) = (\gamma^r_{x_{i+\frac{1}{2}}}, w^r_{x_{i+\frac{1}{2}}}) = (\gamma_i, w_i)\) holds. Consequently, the scheme can be written as:

\[ u^{n+1}_i = u^n_i + \frac{\Delta t}{\Delta x} \left( -Q^- (\gamma_{i+1}, w_{i+1}) - Q^+ (\gamma_i, w_i) + Q^-(\gamma_i, w_i) + Q^+(\gamma_{i-1}, w_{i-1}) \right) \]

Since it is obvious that \( Q^- \) is always non-positive, while \( Q^+ \) is always non-negative, it follows that the first and last part in the brackets contributes only positive to \( u^{n+1}_{i,1} \). The only part that remains is to show that:

\[ u^n_{i,1} + \frac{\Delta t}{\Delta x} \left( - Q^+ (\gamma_i, w_i) + Q^- (\gamma_i, w_i) \right) = \sum_{m=1}^{n} \left( 1 - \frac{\Delta t}{\Delta x} |\gamma_{i,m}| \right) \gamma_{i,m} w_{i,m} \geq 0. \quad (C.1) \]

It follows that the first moment remains always non-negative in the case that:

\[ (1 - \frac{\Delta t}{\Delta x} |\gamma_{i,m}|) \geq 0, \forall m \iff \Delta t \leq \frac{\Delta x}{\max_m |\gamma_{i,m}|}. \]
By setting
\[ \Delta t = \frac{\Delta x}{\max_{j,m}|\gamma_{j,m}|} \quad \text{(C.2)} \]
we can ensure that the first moment remains non-negative on the whole domain. This property usually cannot be proven for higher order finite volume schemes, since for these schemes the quadrature within one cell is usually not constant: \((\gamma_{i, -\frac{1}{2}}, w_{i, -\frac{1}{2}}) \neq (\gamma_i, w_i)\). Consequently, it is not possible to summarise the terms as it was done in (C.1). Vikas et al. [101] proposed a group of new quasi-higher order schemes where this is still possible. This schemes rely on the observation that the grouping in (C.1) is only possible because the abscissas are fixed over the whole cell. To achieve this also for higher order schemes, only the weights are reconstructed using one of the techniques introduced in the previous section, whereas the abscissas remain unchanged thereby ensuring that: \(\gamma_{i, -\frac{1}{2}} = \gamma_{i + \frac{1}{2}} = \gamma_i\) and allowing for the grouping of the terms. Vikas et al. [101] has proven that the resulting scheme preserves the positivity of the first moment in the case that a condition on the timestep size similar to (C.2) is fulfilled.

Building upon our reasoning that the non-negativity of the first moment is always preserved, it is also possible to prove that the introduced scheme preserves the realisability of the moments given. To proof this, using the definition of \(Q^\pm\), we rewrite \(u_j^{n+1}\) to:

\[
(u_j^{n+1}) = \int_R c^i f_j^{(EQ)}(c) dc + \frac{\Delta t}{\Delta x} \left( - Q_{i+1}^- f_{j+1}^{(EQ)}(c) - Q_{i+1}^+ f_{j-1}^{(EQ)}(c) \right) + \frac{\Delta t}{\Delta x} \left( - Q_{i+1}^+ f_{j}^{(EQ)}(c) + Q_{i+1}^- f_{j-1}^{(EQ)}(c) \right) = \int_R c^i h(c)
\]

with
\[
h(c) = f_j^{(EQ)}(c) + \frac{\Delta t}{\Delta x} \left( - \min(c, 0) f_{j+1}^{(EQ)}(c) - \max(0, c) f_{j}^{(EQ)}(c) \right) + \frac{\Delta t}{\Delta x} \left( - \min(c, 0) f_{j}^{(EQ)}(c) + \max(0, c) f_{j-1}^{(EQ)}(c) \right) = (1 - \frac{\Delta t}{\Delta x}[c]) f_j^{(EQ)}(c) - \min(c, 0) f_{j+1}^{(EQ)}(c) + \max(0, c) f_{j-1}^{(EQ)}(c).
\]

where, we used the same grouping-technique as in the simpler proof earlier for the second step.

With the condition (C.2) on the timestep holding, \(h(v)\) is obviously non-negative on \(\Omega_x\). Consequently, \(h(c)\) is a valid phase density function and \(u_j^{n+1}\) a set of realisable moments. It is also possible to prove that the quasi-higher order schemes preserve realisability using the fact that they guarantee the non-negativity of the first moment [101].

**C.3. Further remarks on theoretical properties**

Before making practical use of the finite volume method, the question of how to choose the timestep size \(\Delta t\) needs to be discussed because for some \(\Delta t\) the simulation will become unstable. Unfortunately, the stability region of each time-stepping scheme depends on the spatial discretisation scheme, the chosen geometry, the problem and the computed solution. As a result, usually, only a guideline on a suitable timestep can be given. The results obtained from the previous section are a first hint on how to choose a suitable timestep. It is interesting to note that using our approximation of the characteristic
speed $\lambda$ for quadrature-based methods, (C.2) corresponds to the commonly used CFL condition, which is necessary to guarantee the stability and convergence of the supplied scheme. The CFL condition states - a little bit cryptic - that to ensure convergence the true domain of dependence of the problem has to be entailed in the numerical domain [55]. This means that the distance that any information travels during one timestep should be below the size of one grid cell. Following [55], consider the linear advection equation: $u_t + \lambda u_x = 0$ which is discretised using a finite volume method. With $\Delta t > \frac{\Delta x}{\lambda}$ the resulting scheme would be unstable because in one timestep the information from one cell should reach two cells in the direction of $\lambda$. In practice, however, in the first order finite volume method, the updated cell average depends only on the neighbour elements. As a result, information cannot be transported across more than one cell in each timestep. All in all, the usual CFL condition which needs to be satisfied by ever finite volume scheme is:

$$\Delta t < \frac{\Delta x}{\lambda} \quad \text{(C.3)}$$

where as usual $\lambda$ denotes the largest eigenvalue of $|F'|$. Note that $\lambda$ usually depends on the current solution and is consequently not fixed during the simulation. As a result, the timestep size is usually adaptively chosen.

As mentioned earlier, (C.3) corresponds to (C.2) thereby justifying our choice of $\lambda$ for quadrature-based methods. Furthermore, it is interesting to note that the first order finite volume method is realisability-preserving under a timestep condition which should be fulfilled regardless of the realisability-preserving property of the method to ensure the general stability of the scheme.

Since the CFL condition is only a necessary condition, in practice, a slightly changed guideline including a problem and scheme dependent CFL-like constant $C$ lying in $O(1)$ is used [55]:

$$\Delta t = C \frac{\Delta x}{\lambda} \quad \text{(C.4)}$$

Another formal property of the finite volume method which is important to verify our implementation of the finite volume method and to compare this method to other methods is the formal order of convergence. To measure the $L_p$ accuracy of the calculated solution after timestep $t_n$, we usually compare the calculated solution on each element with the true solution using the $p$ norm:

$$\|u(x,t_n) - u^n_h\|_{L_p} = \sum_{j=1}^{N_x} \Delta x_j \left| \frac{1}{\Delta x_j} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} (u(x,t_n) - u^n_j) dx \right|_p, \quad \text{(C.5)}$$

where we compare our computed approximative solution which aims to approximate the cell averages with the actual cell average. A method is called $p$-convergent if a constant $C$ depending only on $u$, $N$ and $t$ exists so that: $\|u - u_h\|_{\Omega_x} \leq C h^p$ where $h$ denotes the maximal size of one cell. Since the constant $C$ is usually not known, we do not obtain any explicit bounds on the error. Instead, we know how the error behaves when the grid or the order of the approximation is changed. For example, using this bound, we know that when a 2-convergent method is used, and the cell size is divided by two, the error is quartered.

To obtain upper bounds on the convergence of a numerical scheme, usually, two steps are taken: First, the error which is introduced by the numerical scheme in one timestep (local error) is studied. Those considerations fall into the study of the consistency of the method. After that, the stability of the scheme is considered. Stability means that the
local errors do not grow undesirably, for example exponentially, in time [55]. Using this technique, it is possible to show that a first-order finite volume method utilising a Lax-Friedrichs flux is first order accurate for sufficiently smooth solutions and flux functions [55]. Obtaining convergence rates for general non-smooth solutions is very challenging, since it can be proven that even for linear systems a 0.5 convergence of the L2-norm is sharp [9]. The presented second and third order methods lead to schemes with the same order of accuracy [55, 100]. Again this property only holds for sufficiently smooth functions. For such higher order methods, local extrema usually pose additional problems because at these points the accuracy usually degenerates to first order [55].

D. Further on the discontinuous Galerkin method

D.1. Detailed derivation of the Discontinuous Galerkin method in two dimensions

In the following, a detailed derivation involving some quite extensive intermediate steps of the discontinuous Galerkin method for the two dimensional version of (6.5) is presented following [41]. Since all definitions are completely analogous in two dimensions, we start from (6.8):

\[
\int_{I_k} \phi^k_j(x,y) u^n_h(x,y,t) + \phi^k_j(x,y) \text{div} \mathbf{F}(u^n_h(x,y,t)) - \phi^k_j(x,y)(P(u^n_h(x,y,t))) dxdy = 0, \forall j = 1, ..., N + 1
\]

where \( F: \mathbb{R}^m \rightarrow \mathbb{R}^2 \). In the following, we will drop the time dependence of \( u \) for the sake of readability. Using a corollary obtained from Gauss divergence theorem:

\[
\int_V (\mathbf{F} \cdot (\nabla g) + g \text{div} \mathbf{F}) dV = \oint_{\partial V} (g \mathbf{F} : n) d\partial V
\]

it is possible to rewrite the second term of (D.1) as:

\[
\int_{I_k} \phi^k_j(x,y) \text{div} \mathbf{F}(u^n_h(x,y)) dxdy = -\int_{I_k} \nabla \phi^k_j(x,y) \cdot \mathbf{F}(u^n_h(x,y)) dxdy + \int_{\partial I_k} \phi^k_j(x,y) \mathbf{F}^*(u_h(x,y)) \cdot \hat{n} dxdy
\]

where we again can rewrite the first part of the expression as:

\[
\int_{I_k} \nabla \phi^k_j(x,y) \cdot \mathbf{F}(u^n_h(x,y)) dxdy = \int_{I_k} \left[ \partial_x \phi^k_j(x,y) \partial_y \phi^k_j(x,y) \right] \left[ \begin{array}{c} F^{(x)}(u^n_h(x,y)) \\ F^{(y)}(u^n_h(x,y)) \end{array} \right] dxdy
\]

where \( \mathbf{F}_1 = F^{(x)} \) and \( \mathbf{F}_2 = F^{(y)} \). Using the nodal definition of \( u_h \) and \( \mathbf{F}_h \) this can be further rewritten to:

\[
\sum_{n=1}^{N+1} F^{(x)}(u^n_h(x_n,y_n)) \int_{I_k} \partial_x \phi^k_j(x,y) \partial_y \phi^k_j(x,y) dxdy + \sum_{n=1}^{N+1} F^{(y)}(u^n_h(x_n,y_n)) \int_{I_k} \partial_y \phi^k_j(x,y) \partial_y \phi^k_j(x,y) dxdy
\]

The second term of (D.1) can also be rewritten to:

\[
\int_{\partial I_k} \phi^k_j(x,y) \mathbf{F}^*(u_h(x,y)) \cdot \hat{n} dxdy = \sum_{e \in \mathcal{K}(I_k)} \int_e \phi^k_j(x,y)(\mathbf{F}^{*(x)}(u_h(x,y))n^e_x + \mathbf{F}^{*(y)}(u_h(x,y))n^e_y) dxdy,
\]

where \( \mathcal{K}(I_k) \) denotes the set of edges of the element \( I_k \). In the case that a Lax-Friedrichs flux is used this can be rewritten to:

\[
\sum_{n=1}^{N+1} \left( \int_{I_k} \phi^k_j(x,y) \mathbf{F}^{*(x)}(u^n_h(x_n,y_n)) dxdy + \int_{I_k} \phi^k_j(x,y) \mathbf{F}^{*(y)}(u^n_h(x_n,y_n)) dxdy \right)
\]
\[ \sum_n (\sum_{n=1}^{N+1} \left( \frac{1}{2} \left( F^{(x)}(u_k^n(x^n, y^n)) + F^{(y)}(u_k^n(x^n, y^n)) + n_k^n - \frac{C}{2} u_k^n(x^n, y^n) I_n \right) + \frac{C}{2} u_k^n(x^n, y^n) I_n \right) dx dy \]

where \( I_k \) denotes the element which is connected to \( I_k \) over \( \kappa \). Using the above simplifications, we can rewrite (D.1) to:

\[ M^k \partial_t u_h(x^k, y^k) = S^{k,(x)}(x^k, y^k) + S^{k,(y)}(x^k, y^k) + M^k P(x^k, y^k) \]

\[ - \sum_{\kappa} N^{\alpha,k}(1) \left( F^{(x)}(x^k, y^k) * n_k^x + F^{(y)}(x^k, y^k) * n_k^y + C u_h(x^k, y^k) \right) \]

\[ - \sum_{\kappa} N^{\alpha,k}(1) \left( F^{(x)}(x^k, y^k) * n_k^x + F^{(y)}(x^k, y^k) * n_k^y - C u_h(x^k, y^k) \right) \]

(D.3)

with

\[ u_h(x^k, y^k)_{ij} = u_h(x^k_j, y^k_j), \quad F^{(x)}(x^k, y^k)_{ij} = F^{(x)}(u_h(x^k_j, y^k_j)), \quad M_{ij} = \int_{I_k} \ell_k^x(x,y) \ell_k^y(x,y) dx dy, \]

\[ S_{ij}^{k,(x)} = \int_{I_k} \ell_k^x(x,y) \frac{\partial \phi_k(x,y)}{\partial x} dx dy, \quad N_{ij}^{\alpha,k} = \int_{I_k} \ell_k^x(x,y) \ell_j^y(x,y) dx dy, \quad N_{ij}^{\alpha,k} = \int_{I_k} \ell_k^x(x,y) \ell_j^y(x,y) dx dy. \]

For further details on how to extend the method to three dimensions and further theoretical and practical considerations see [41]. In our implementation of the DG method, we have also loosely followed [41].

**D.2. Relationship between the nodal and the modal form**

We will now expand a little bit on the form of the local representation of the solution. The local approximation is either in the modal or in the nodal form:

\[ u_h(x^n, t) = \sum_{n=1}^{N+1} \alpha_n(t) \varphi_n(x) = \sum_{n=1}^{N+1} u(x^n, t) \ell_n^x(x). \]  

(D.4)

Concerning the modal approach, it is, first of all, an interesting question how the coefficients \( \alpha_n \) from a given function \( u(x) \) can be obtained. One possible approach is to do so by projection leading to the following system: \( M \alpha = \hat{u} \) where \( M_{ij} = \langle \varphi_i(x) \varphi_j(x) \rangle \) and \( \hat{u} = \langle u(x) \varphi_i(x) \rangle \). Since \( M \) is also used as a central part in the calculation of the time derivatives, it is important to choose the basis functions \( \varphi \) in a way that \( M \) is reasonably conditioned to enable an accurate calculation of \( \alpha \) and the time derivatives. To achieve this, usually the normalised Legendre polynomials are chosen as basis functions because then \( M \) reduces to the identity matrix [41]. However, \( \hat{u} \) still needs to be calculated using a numerical quadrature.

To relate the two representation the following observation is necessary:

\[ V \alpha = u, \]  

(D.5)

with \( V_{ij} = \hat{P}_{j-1}(\xi_i) \) and \( u_i = u(\xi_i) \), where \( \hat{P}_j \) denotes the \( j \)th order Legendre polynomial and \( \xi \) a set of \( N + 1 \) distinct grid points. Using (D.5), it is possible to convert the modal representation into the nodal one and vice versa. Both representations are identical because the value of an \( N \)th order polynomial at \( N + 1 \) points determines the polynomial uniquely. In practice, however, both representations lead to slightly different solutions because, for example, the initial conditions can be different because the parameters \( \alpha \) calculated by projection may not result in an exact interpolation at the chosen grid points.
The only part remaining is to choose an optimal set of grid points for the nodal form. One possible criterion is to choose them in a way that $V$ is far away from singularity because, in practice, some conversion of the two representations is needed. Another criterion is that the Lebesque constant of the resulting Lagrange polynomials should be minimised because it is possible to relate this measure to the distance of $u_h$ to an optimal $N$th order polynomial representation of an arbitrary function $u$ [41]. Hesthaven and Warburton [41] indicated that the usage of Legendre-Gauss-Lobatto quadrature points is beneficial because they are optimal concerning both criteria and lead to a reliable interpolation of any polynomial function.

In our implementation, we choose to use a nodal approach. Mostly because with a solely modal approach it is very hard to approximate the flux function because to calculate its coefficients an integration of the flux function is needed which is, in the context of solving the Boltzmann equation, obviously not possible because the function is unknown. Furthermore, the nodal approach is often computational easier to handle, for instance, the initial conditions are much easier to calculate or to visualise.

D.3. Limiter and Filtering

One of the most serious problems of the presented discontinuous Galerkin method is its behaviour near discontinuities. The following chapter gives a brief overview of how the problem of discontinuities in the solution can be addressed. Usually, discontinuities cause significant problems to the convergence of the solution because point-wise convergence at the discontinuity is lost and spurious oscillations are introduced by the shock. The underlying phenomenon known as Gibbs phenomenon is, however, well-studied and understood [41]. Fortunately, it can be shown that, in general, such oscillations do not destroy all information necessary to achieve high order accurate solution in smooth regions of the solution [41]. However, to recover this accuracy in smooth regions and to make the simulation stable for reasonable timesteps, some “postprocessing” of the solution in each step is needed. In the following, two techniques aiming to overcome said problems will be presented.

One less common technique aiming to reduce the aliasing error, which is closely related to the oscillations introduced in the solution by discontinuities, is filtering. Filter try to tackle this issue by introducing a controlled amount of dissipation in the solution while maintaining some of the most important properties of the solution such as the mean. The filter is applied to each term of the modal local approximation in each cell and can be interpreted as a reweighing of the coefficients:

$$\tilde{u}_h^j(x, t) = \sum_{n=1}^{N+1} \sigma(n) \alpha_n(t) \varphi_n(x).$$

One filter, which is often very useful to reduce the aliasing error in the solution, is the exponential filter [41]:

$$\sigma(\eta) = \exp(-\alpha \eta^s),$$

where $\alpha$ and $s$ are parameters determining the “strength” of the filter. Generally speaking, $s$ is the order of the filter and impacts the result significantly. The order of the filter and its strength are inversly proportional. Consequently, low order filtering can lead to overly dissipated solution, while when choosing $s$ sufficiently high, filter can stabilise the solution and ensure high order accuracy away from the discontinuity and even in regions where the discontinuity has passed through [41]. In the implementation of the thesis, more complex
filter are used [41]:

$$\sigma(\eta) = \begin{cases} 1 & 0 \leq \eta \leq \eta_c = \frac{N_c}{N} \\ \exp(-\alpha(\frac{\eta - \eta_c}{1 - \eta_c})^s) & \eta_c \leq \eta \leq \eta \end{cases}$$

which allows for leaving some of the nodes unchanged through introducing a threshold variable $N_c$. Despite enabling for higher order accuracy away from the discontinuity, filters fall short when it comes to tackling oscillations in the solution around the shock. For filters of order higher than two, these oscillations are usually still present, whereas for lower order filters the oscillations are gone at the expanse of smearing out the discontinuity. Unfortunately, such oscillations are a serious problem for some applications because they can lead, for example, to non-physical solutions [41].

To tackle the problem of handling such oscillations at the discontinuity, we make again use of the concept of limiting the solutions as it was done for the finite volume methods. To derive the already introduced slope limiters as a possibility to avoid oscillations, it is necessary to introduce the total variation of a numerical solution. The total variation norm is defined as:

$$|u_h|_{TV} = \sum_{k=1}^{N_x-1} |u_{k+1}^h - u_k^h|.$$ 

For an advection equation, for example, the true solution has a constant $|u|_{TV}$ value over time. For numerical solutions, however, this is not necessarily the case and the development of $|u_h|_{TV}$ over time can be seen as an indicator whether or not oscillations have been introduced. Since oscillations usually increase $|u_h|_{TV}$, one reasonable requirement for a numerical scheme is that $|u_h|_{TV}$ is non-increasing over time. A method that can guarantee this property is called total variation dismissing (TVD). Cockburn [20] proved that the DG method under a few conditions is total variation dismissing in the mean (TVDM) which means that $|\bar{u}|_{TV}$ is non-increasing over time. This property is slightly weaker than TVD but essentially sufficient to avoid any form of oscillation. The assumptions under which TVDM is proven are assumptions on the numerical solution:

$$\text{sign}(\bar{u}_{k+1} - \bar{u}_k) = \text{sign}(\bar{u}_{k}^h(x_{k+\frac{1}{2}}) - \bar{u}_k^h(x_{k-\frac{1}{2}})),$$

$$\text{sign}(\bar{u}_{k+1}^h - \bar{u}_k^h) = \text{sign}(u_{k+1}^h(x_{k+\frac{1}{2}}) - u_k^h(x_{k-\frac{1}{2}})),$$

$$\text{sign}(\bar{u}_{k+1}^h - \bar{u}_k^h) = \text{sign}(p(u_{k+1}^h) - p(u_k^h))$$

with $p(u_k^h) = \bar{u}_k^h - \frac{\Delta t}{h} f^+(u_k^h(x_{k+\frac{1}{2}})) + \frac{\Delta t}{h} f^-(u_k^h(x_{k-\frac{1}{2}}))$ and $f^+(a) = \frac{1}{2}(f(a) + Ca)$ and $f^-(b) = \frac{1}{2}(f(b) - Cb)$ being the splitted Lax-Friedrichs flux. These conditions are in general not enforced by the DG method. To enforce them, slope limiters can be used. In contrast to the finite volume method, in the context of DG methods, the slope limiters are not applied during the calculation of the update; instead, they are applied to the solution at the end of each timestep. A limiter should, general speaking, satisfy the following conditions [20]:

**List of requirements 1.**

1. **Enforce (D.6).**
2. **Do not violate conservation.**
3. **Do not decrease the formal accuracy of the method away from the discontinuity.**
It is worth noticing that, at the discontinuity, we cannot expect higher than first order accuracy.

As mentioned above, a first option to satisfy (D.6) is the use of slope limiters as they have been introduced in (C.1). However, using this method comes at the disadvantage that only the first two properties of the list of requirements 1 are fulfilled because it is assumed that the approximated solution can be represented by a linear function in each cell:

\[ u_h^k(x) = \pi_h^k + (x - x_0^k)(u_h^k)_x, \]

where \((u_h^k)_x\) denotes the slope of \(u_h(x)\) in cell \(k\). Postponing this issue for a while, we proceed with this very intuitive form of limiting. As in FV methods, the slope of \(u_h^k\) is replaced by a limited slope. A first candidate for a limited solution fulfilling (D.6) is [41]:

\[ \Pi^1 u_h^k(x) = \pi_h^k + (x - x_0^k) \text{minmod}((u_h^k)_x, \frac{\pi_h^{k+1} - \pi_h^k}{0.5h}, \frac{\pi_h^k - \pi_h^{k-1}}{0.5h}) \]  
(D.7)

where \(h = \max_i \Delta x_i\) and the minmod function is used to indicate and limit a possible oscillation. In the case that the slope in three adjacent elements have different sign a possible oscillation is detected and the minmod function will return zero thereby setting back the solution to a piecewise constant one as it is the case in the finite volume method. An alternative to (D.7) is the MUSCL limiter [55]:

\[ \Pi^1 u_h^k(x) = \pi_h^k + (x - x_0^k) \text{minmod}((u_h^k)_x, \frac{\pi_h^{k+1} - \pi_h^k}{h}, \frac{\pi_h^k - \pi_h^{k-1}}{h}) \]  
(D.8)

which is slightly more dissipative.

As mentioned earlier, both limiters do not fulfil the third property of the list of requirements 1 out of two reasons. First of all, at the end of each step, each cell is reconstructed as a linear function thereby destroying any advantages of using higher order DG methods. Secondly, not only does the accuracy degenerates to the one of a first order DG scheme in smooth regions but also does the limiter detect local extrema as possible oscillations and thereby degenerate the accuracy at said point to first order. Unfortunately, the last issue is a result of a general conflict between the first and third condition of the list of requirements 1 because at local extrema in smooth regions the TVDM assumptions are usually not fulfilled and the accuracy is therefore correctly reduced to first order by the limiter.

One possibility to tackle the former problem was proposed by Cockburn [20] as a general slope limiter \(\Pi^N\) which tries to detect the cells in which oscillations may have appeared in beforehand and limit only those cells, whereas the higher-order approximations in all other cells are left unchanged. \(\Pi^N(u_h^k)\) proceeds as follows:

1. Compute \(v_r^k\) and \(v_l^k\) as the limited values of the solution at the interface
2. If \(v_r^k = u_h^k(x_{k-\frac{1}{2}})\) and \(v_l^k = u_h^k(x_{k+\frac{1}{2}})\) do not change \(u_h^k\), otherwise set \(u_h^k = \Pi^1 \tilde{u}_h^k\)

where \(\tilde{u}_h\) is the approximation of \(u_h\) up to first order and:

\[ v_r^k = \pi_h^k + \text{minmod}(u_h^k(x_{k+\frac{1}{2}}) - \pi_h^k, \pi_h^k - u_h^{k-1}, \pi_h^{k+1} - \pi_h^k), \]

\[ v_l^k = \pi_h^k - \text{minmod}(\pi_h^k - u_h^k(x_{k-\frac{1}{2}}), \pi_h^k - u_h^{k-1}, \pi_h^{k+1} - \pi_h^k). \]

This generalised slope limiter leads to improved accuracy in smooth regions thereby fulfilling at least partly the third property from list 1 [20, 41].
To tackle the problem of lost accuracy at local smooth extrema, it is possible to provide a bound on the second derivative at the local extrema and to use this bound to detect only oscillations and to exclude local extrema. This strategy, however, has the significant drawback that one has to know this problem dependent constant in advance [41]. An alternative and problem independent strategy has been proposed in Biswas et al. [12]. Here, instead of either leaving the solution unchanged or restricting it to a linear or constant approximation, an adaptive limiting technique is applied to the modal form of approximation. This strategy starts with limiting the highest order coefficient and in the case that the limiting changes the coefficient, proceeds with the coefficient one order lower using the following formulae:

\[
\tilde{\alpha}^{j}_{n+1} = \frac{1}{2n + 1} \minmod((2n + 1)\alpha^{j}_{n+1}, \alpha^{j}_{n} - \alpha^{j-1}_{n}, \alpha^{j+1}_{n} - \alpha^{j}_{n})
\]  

(D.9)

The method proceeds in this way until the zeroth order coefficient is limited. It has been indicated that this method applies limiting only where needed and is capable of maintaining high order accuracy in smooth regions [12]. Burbeau et al. [18] proposed an extension of this method which additionally controls the amount of viscosity introduced to prevent the smearing of local extrema even more effective.

D.4. Time step size and order of convergence

Having already briefly discussed the timestep size and order of convergence of finite volume methods, we will now do the same for the RKDG method presented in section 6.2. For non-limited higher order DG methods, the usual CFL condition is not sufficient to guarantee stability [109]. The reason for this is that not only the eigenvalue spectrum of the differential equation under consideration (e.g. \( \lambda \)) has to lie in the stability region of the time-stepping scheme but also the spectral radius of the discretised differential operator of the chosen numerical scheme. For low order finite volume methods, this usually does not make any difference. For higher order unlimited DG methods using an upwind flux, however, it can be shown that this spectrum grows asymptotically in \( O(N^2) \) [41, 102]. Fortunately, it can be additionally proved that this growth rate has only an effective influence on the spectral radius when \( N \) becomes large and that for \( N < 10 \) the growth can be described by \( O(N) \).

Let \( \Delta r_i \) denote the distance between two local grid points projected onto the unit cell. Since \( \Delta r_i \) is inversely proportional to \( N \), one general guideline based on observations on the spectral radius of the DG method for choosing the timestep is [41]:

\[
\Delta t \leq \frac{C}{\lambda} \min_{k,i} \frac{h_k}{2} (\Delta r_i).
\]

Another possible guideline which is more analogous to the CFL condition for finite volume methods is [109]:

\[
\Delta t = C \frac{(\Delta x)^{\frac{4}{3}}}{\lambda}.
\]  

(D.10)

Concerning the usual CFL condition (C.4), Cockburn and Shu [24] even included a table for which values of \( C \) Runge-Kutta methods of different order combined with DG methods of different order are stable. Nevertheless, when using the discontinuous Galerkin method a smaller timestep size compared to the finite volume method needs to be used. This is completely reasonable because the degrees of freedom per cell are \( k + 1 \) in comparison to
1 in FV methods. In addition, it is also possible to prove that limited versions of the DG methods are also stable under usual CFL conditions [24].

Turning to the order of convergence of the DG method, there exist tones of different bounds proven for the convergence behaviour of the discontinuous Galerkin method. They vary in the number of dimensions considered, form of the solution and problem, type of scheme, and type of geometry considered. To begin, we first define a measure for the accuracy of a solution of the DG method analogous to (C.5) as:

\[
\| u(t^n) - u^n_h \|_{I,p} = \sum_{j=1}^{N_x} \int_{I_j} \| u(x,t^n) - \sum_{n=1}^{N+1} u(x^n_j,t) \rho_j(x) \|_{p} dx,
\]

where we now compare the numerical solution with the precise solution instead of just the average because the discontinuous Galerkin method claims to approximate not only the cell average but also the precise solution.

Turning to error bounds and order of convergence, generally speaking, most of the error bounds in the literature only apply to sufficiently smooth solutions [109]. As expected, the error bounds for non-smooth cases are very poor and usually proven under quite rigorous constraints. Considering in the following only error bounds for smooth solutions of non-linear problems in one dimension, the most general and common bound is the following (for a detailed proof see [109]):

\[
\| u(t^n) - u^n_h \|_{I,2} \leq \tilde{C}(h^{N+\sigma} + (\Delta t)^2) = Ch^{N+\sigma}
\]

which holds in the case that the timestep is sufficiently small ((D.10) has to hold) and the grid sufficiently fine (h is sufficiently small), where \( \tilde{C} \) is a constant depending only on N and \( u \), while C additionally depends on \( \Delta t \). \( \sigma \) is a flux dependent constant. For general monotonic fluxes \( \sigma = \frac{1}{2} \) holds, while for any upwind flux, such as the Roe and Godonov flux, \( \sigma = 1 \) holds. An upwind flux is a numerical flux satisfying:

\[
F^*(a,b) = \begin{cases} 
F(a) & \text{if } F'(c) \geq 0, \quad \forall c \in [\min(a,b), \max(a,b)] \\
F(b) & \text{if } F'(c) < 0, \quad \forall c \in [\min(a,b), \max(a,b)] 
\end{cases}
\]

Consequently, the optimal rate of convergence of the RKDG method without any post-processing of the solution is: \( O(h^{N+1}) \). Unfortunately, in the context of solving the Boltzmann equation it is, in practice, not possible to use upwind fluxes because of the non-linearity of the closures involved in the spatial transport term. Instead, as mentioned earlier, we usually use the local Lax-Friedrichs flux, which only guarantees an order of convergence of \( O(h^{N+\frac{1}{2}}) \). Fortunately, in most problems the local Lax-Friedrichs flux produces results which are quite similar as the results from the upwind Godonov flux [24]. As a result, we can hope to observe optimal convergence for most of the problems considered.

Considering multi-dimensions, the optimal order of convergence that can be achieved depends on the grid chosen. For regular rectangular grids, as well as some form of structured triangular grids, it can be proven that the order of convergence is similar to (D.11) [75, 109]. For general unstructured grids, however, it is known that the order of convergence usually does not exceed \( O(h^{N+\frac{1}{2}}) \) [71].

E. Validation of numerical scheme

To validate our implementation of the Runge-Kutta discontinuous Galerkin method in one dimension, we conducted several test cases measuring the error of the calculated
solution. Since the error which the discontinuous Galerkin method produces is problem dependent, we are mainly interested in the order of convergence of the solution. As discussed in appendix D, the order of convergence of the discontinuous Galerkin method can be proven to be $O(h^{N+1})$ when an upwind flux is used. Since we use the local Lax-Friedrichs flux, which is not an upwind flux, the observed order of convergence should, therefore, lie in between of $O(h^{N+1})$ and $O(h^{N+\frac{1}{2}})$. In the following, the error of the discontinuous Galerkin method of different degrees on different meshes in the context of solving a linear and non-linear problem will be presented.

### E.1. Advection equation

In the following, the results of our implementation of the discontinuous Galerkin method applied to the one dimensional advection equation will presented. We solve the one dimensional advection equation:

$$\partial_t f(x, t) + c \partial_x f(x, t) = 0.$$  

using the initial condition $f_0(x) = \text{abs}(\sin(\pi * x))$ on the domain $\Omega = [-1, 1]$ with periodic boundary conditions for $c = 2$ and $T = 2$. In table 5, the errors of the solution of the discontinuous Galerkin method using different degrees and mesh sizes are displayed. Additionally, the average measured order of convergence as well as the order of convergence of each of the mesh-refinement steps are included in the table. For our analysis, it is important to note that the Lax-Friedrichs flux fulfills the requirements of an upwind flux in the context of solving an advection equation. The measured convergence rates are all close to optimal and sometimes even above the theoretically provable convergence rate. Considering the advection equation with different initial conditions and parameters leads to usually identical results. However, it is not always possible to observe the optimal convergence rates. For example, in the case that initial error is already very small, the convergence rates are usually not optimal. Furthermore, especially at higher order discontinuous Galerkin schemes the error can sometimes no longer be decreased because machine precision is reached.

### E.2. Euler equations

Since the Boltzmann equation is a highly non-linear equation, we also tested the discontinuous Galerkin method to solve non-linear problems. In such problems, the local Lax-Friedrichs is no longer an upwind flux, and it is rather unclear which convergence rate one has to expect. We choose to solve the well-studied Euler equations. Unfortunately, due to the non-linearity of the problem usually no analytical solution of the Euler equations is available. Therefore, we applied the technique of manufactured solutions to the problem. In this technique, a right side is added to each of the equations which is

<table>
<thead>
<tr>
<th>D</th>
<th>M</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>Overall CR</th>
<th>1. CR</th>
<th>2. CR</th>
<th>3. CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.4 × 10^{-1}</td>
<td>4.6 × 10^{-2}</td>
<td>8.2 × 10^{-3}</td>
<td>1.7 × 10^{-3}</td>
<td>2.4</td>
<td>2.4</td>
<td>2.5</td>
<td>2.24</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.5 × 10^{-2}</td>
<td>1.8 × 10^{-3}</td>
<td>2.3 × 10^{-4}</td>
<td>2.9 × 10^{-5}</td>
<td>3</td>
<td>3.1</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.2 × 10^{-3}</td>
<td>8.3 × 10^{-4}</td>
<td>3.2 × 10^{-5}</td>
<td>3.1 × 10^{-6}</td>
<td>3.9</td>
<td>3.8</td>
<td>4.0</td>
<td>4.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9.6 × 10^{-4}</td>
<td>3 × 10^{-5}</td>
<td>1 × 10^{-6}</td>
<td>4.9 × 10^{-8}</td>
<td>4.7</td>
<td>5</td>
<td>4.9</td>
<td>4.3</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Error of solution of the discontinuous Galerkin for the advection equation
constructed in a way that a former chosen solution is the analytical solution of the equations [59]. We used both a steady and transient manufactured solution which are both in the form of:

$$\rho(x) = \rho_0 + \rho_x \sin(a_{\rho_x} \frac{x}{L}) + \rho_t \sin(a_{\rho_t} \frac{t}{L})$$

$$v(x) = v_0 + v_x \sin(a_{v_x} \frac{x}{L}) + v_t \cos(a_{v_t} \frac{t}{L})$$

$$p(x) = p_0 + p_x \cos(a_{p_x} \frac{x}{L}) + p_t \cos(a_{p_t} \frac{t}{L})$$

with \(\rho_t = p_t = v_t = 0\) for the steady case. For the steady solution we used \(\Omega_x = [-0.5, 2.5]\), while we used \(\Omega_x = [0, 9.5]\) for the simulation of the transient solution. For both cases we choose \(T = 1\). The corresponding errors are presented in table 7 and 6 as well as the measured order of convergence. The measured order of convergence is usually in very close agreement with the theoretical provable optimal order of convergence. However, sometimes very odd pattern occur in the data. For example, in the transient case, the first order discontinuous Galerkin method is more accurate than the second order one.

All in all, using the described experiment, we were able to validate our implementation of the DG and show its great capability to solve linear and non-linear equations.

**F. Riemann problem**

For the sake of comparability, since it is a standard test case in computational fluid dynamics, we will also shortly present the results of the solver on a standard Riemann test case [6]:

$$f_0(x, c) = \begin{cases} 
  f_M(c; 5, 0, 1), & x \leq 0 \\
  f_M(c; 1, 0, 1), & x > 0
\end{cases}$$

which corresponds to Mach number 1.369. We consider the problem on a standard domain \(\Omega_x = [-1, 1]\) with \(N_x = 300\) and \(T = 0.2\). In figure 15 the results of EQ combined

---

**Table 6: Error of solution of the discontinuous Galerkin for a steady manufactured solution of the Euler equations**

<table>
<thead>
<tr>
<th>D</th>
<th>M</th>
<th>40</th>
<th>80</th>
<th>160</th>
<th>320</th>
<th>Overall CR</th>
<th>1. CR</th>
<th>2. CR</th>
<th>3. CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.3 × 10^{-2}</td>
<td>7.2 × 10^{-3}</td>
<td>1.2 × 10^{-3}</td>
<td>2.1 × 10^{-4}</td>
<td>2.8</td>
<td>3.1</td>
<td>2.5</td>
<td>2.6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.9 × 10^{-1}</td>
<td>4.5 × 10^{-2}</td>
<td>4.8 × 10^{-3}</td>
<td>8.4 × 10^{-4}</td>
<td>2.6</td>
<td>2.1</td>
<td>3.2</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9.2 × 10^{-4}</td>
<td>1.2 × 10^{-4}</td>
<td>7.5 × 10^{-6}</td>
<td>5.8 × 10^{-7}</td>
<td>3.6</td>
<td>2.9</td>
<td>4</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.9 × 10^{-4}</td>
<td>1.6 × 10^{-6}</td>
<td>7.2 × 10^{-6}</td>
<td>2.5 × 10^{-8}</td>
<td>4.5</td>
<td>4.2</td>
<td>1.1</td>
<td>8.3</td>
<td></td>
</tr>
</tbody>
</table>

**Table 7: Error of solution of the discontinuous Galerkin for a transient manufactured solution of the Euler equations**

<table>
<thead>
<tr>
<th>D</th>
<th>M</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>Overall CR</th>
<th>1. CR</th>
<th>2. CR</th>
<th>3. CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.7 × 10^{-1}</td>
<td>2.4 × 10^{-2}</td>
<td>3.8 × 10^{-3}</td>
<td>5.9 × 10^{-4}</td>
<td>2.7</td>
<td>2.8</td>
<td>2.7</td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.5 × 10^{-1}</td>
<td>2.9 × 10^{-2}</td>
<td>5.8 × 10^{-3}</td>
<td>1.1 × 10^{-3}</td>
<td>2.3</td>
<td>2.4</td>
<td>2.3</td>
<td>2.3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.9 × 10^{-2}</td>
<td>1.1 × 10^{-4}</td>
<td>7.4 × 10^{-6}</td>
<td>5.9 × 10^{-6}</td>
<td>3.9</td>
<td>4.2</td>
<td>3.8</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.9 × 10^{-3}</td>
<td>9.6 × 10^{-6}</td>
<td>4.2 × 10^{-6}</td>
<td>1.9 × 10^{-7}</td>
<td>4.4</td>
<td>4.3</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.3 × 10^{-4}</td>
<td>1.6 × 10^{-6}</td>
<td>3.5 × 10^{-8}</td>
<td>5.7 × 10^{-10}</td>
<td>6</td>
<td>6.3</td>
<td>5.6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7.9 × 10^{-6}</td>
<td>9.1 × 10^{-8}</td>
<td>1.1 × 10^{-9}</td>
<td>1.8 × 10^{-11}</td>
<td>6.3</td>
<td>6.4</td>
<td>6.3</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
with the discontinuous Galerkin method and the finite volume method are visualised. The main properties of the solution are similar to the solution of the two-beam problem. Again strong smearing effects take place in the finite volume method, while the discontinuous Galerkin method produces a staircase like function including sharp discontinuities. Furthermore, by increasing the number of moments considered, it is possible to increase the number of steps in the solution thereby increasing its fit with the smooth analytical solution. We also included the results of the solver for the Riemann problems in different regimes in 16. The results show the expected pattern that the calculated solution is smoothed in the case that the Knudsen number decreases.
References


