

**Application of the Finite Pointset Method to moving
boundary problems for the BGK model of rarefied gas
dynamics**

**vom Fachbereich Mathematik
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(doctor rerum naturalium, Dr. rer. nat.)
genehmigte
Dissertation**

D 386

Maria Kobert

Datum der Disputation: 22.07.2015

1. Gutachter: Prof. Dr. Axel Klar
2. Gutachter: Prof. Dr. Giovanni Russo

Acknowledgements

First of all I want to thank Prof. Dr. Axel Klar, Dr. Jörg Kuhnert and Dr. Sudarshan Tiwari for supervising and advising me during the last few years. Especially, I thank Axel Klar for his understanding and support after me having a child.

I also want to express my gratitude to Prof. Dr. Giovanni Russo for reviewing previous versions of this work and for giving me a lot of constructive and helpful remarks.

Furthermore, I thank the Department Transport Processes (TV) from the Fraunhofer Institute for Industrial Mathematics (ITWM) for the financial support during my PhD studies.

Of course I am deeply indebted to my family and my friends for being around and encouraging me all the time. In particular, I thank my husband, Kassian Kobert, for his patience and relief during challenging times.

Finally, I want to thank the grandparents of our son Paul for the extra-time they gave me by taking care of him whenever they could.

To my husband, Kassian Kobert,
who always trusts in me.

Contents

1	Introduction	9
1.1	Motivation and related work	9
1.2	Outline of the work	13
2	The Boltzmann equation and the kinetic BGK model	15
2.1	Boltzmann equation	15
2.2	BGK model	19
2.3	Boundary conditions	23
3	Finite Pointset Method (FPM)	27
3.1	Initializing the point cloud	28
3.2	Neighbor searching	28
3.3	Finding holes and removing points	29
3.4	Approximation of function values and differential operators	30
3.5	Numerical tests	36
3.5.1	Dirichlet boundary value problem	37
3.5.2	Neumann boundary value problem	37
3.5.3	Mixed boundary value problem	38
4	Regular grid methods for solving the BGK model equation	41
4.1	Direct discretization methods	43
4.1.1	Implicit-Explicit scheme	43
4.1.2	Semi-Lagrangian scheme	47
4.2	Moment methods	51
4.2.1	13 Moment Equations	51
4.2.2	Sectional Quadrature Method of Moments	53

4.3	Comparison of the methods	56
5	Semi-Lagrangian scheme in the FPM framework	63
5.1	Numerical scheme in one dimensional spacial and velocity space	63
5.1.1	Reconstruction of the distribution function	65
5.1.2	Evaluation of the moments	67
5.1.3	Numerical example: Shock tube problem	68
5.2	The scheme in two dimensional spatial and velocity space	69
5.2.1	Reconstruction of the distribution function	70
5.2.2	Evaluation of the moments	73
5.2.3	Numerical example: Driven Cavity flow	73
6	Application to moving boundary problems	79
6.1	One dimensional piston problem	79
6.1.1	Boundary conditions	80
6.1.2	Results	87
6.2	Moving piston in a two dimensional square	90
6.2.1	Specular reflection boundary conditions	91
6.2.2	Diffusive reflection boundary conditions	95
6.2.3	Results	98
6.3	Rotating slab	102
6.3.1	Specular reflection boundary conditions	103
6.3.2	Results	104
7	Conclusions and outlook	107
	Bibliography	110

Chapter 1

Introduction

1.1 Motivation and related work

Modeling rarefied gas flows plays an important role in many fields of applications. It is used in vacuum engineering to design vacuum pumps for vacuum tubes, electric lamps, medical applications used for radiotherapy, milking machines and many more [61]. Furthermore simulation of rarefied flows is common in aerospace design to describe the reentry of space shuttles into the atmosphere [8], shock waves around jets or flows of gas fuel within a jet engine. More recently it is also used in nanotechnology, for example to build Micro-Electro-Mechanical-Systems (MEMS) [13], which are present in accelerometers, micro pumps, micro engines and so on.

The research in the field of rarefied gas dynamics began in the latter half of the nineteenth century and has continued to the present. Because of growing interest in space and flight at extreme altitude attention was drawn to a flow regime that had not been seriously considered before. Another scientific field was the gas dynamic isotope separation technique and the development of flow cooling to aid in the spectroscopic study of complicated molecules which had arisen during the 1970s. For the 1980s and 1990s the unique properties of cluster, created by gas-dynamic sources provided possibilities to study surface dispositions of cluster complexes, cluster structures and cluster chemistry [39].

As we can see rarefied gas dynamics with the rest of fluid dynamics applies to a variety of technological devices in various scientific areas.

The analysis of rarefied flows can be divided into three flow regimes, depending on the

value of the so called *Knudsen number* Kn , which is the ratio of the mean free path and a characteristic length. The mean free path of molecules is the average distance between collisions with other molecules in the gas. The characteristic length may be chosen, for example as the mean pipe diameter in a vacuum pump.

When the mean free path is much smaller than the pipe diameter, i.e. $Kn < 0.01$, the gas flow can be seen as continuous fluid. That means the number of particles, of which the gas consists is large enough to neglect the motion of each individual particle. In this case one can employ Euler or Navier-Stokes equations to describe the flow. To solve these equations Computational Fluid Dynamics (CFD) approaches [68], which consist of discretization methods such as the finite element, finite volume or the finite differences methods can be applied.

In the case where the mean free path is about the same as or even greater than the characteristic length ($Kn \geq 1$) one speaks of molecular flows. Here, the density of the molecules in a gas is low, so that the probability of collisions between particles is smaller than the probability of particle-wall collisions. The motion of the fluid appears random. Thus, statistical methods, such as the Direct Simulation Monte Carlo (DSMC) method [47] are used for simulation. These are based on a kinetic description of the gas through the Boltzmann equation, which was established by Ludwig Boltzmann in 1872 [2, 3, 11].

The transition region ($0.01 < Kn < 1$) between the continuum flow regime and the molecular flow regime is referred to as the Knudsen or transition flow regime. In this region Navier-Stokes equations are not adequate to model gas flows. Because of the rapid increase of particle collisions the DSMC method tends to fail, as well.

Therefore an important issue in the field of simulation of rarefied gas flows is the development of methods, which are able to cover all ranges of the Knudsen number. One idea is to apply common CFD methods to kinetic Boltzmann model equations, see for example [1, 10, 40, 44, 48, 50]. Other known methods are based on the macroscopic moments of the Boltzmann equation, which provide partial differential equations suitable for application of CFD methods. One of these methods is the 13 Moment Method [56, 57, 58, 59, 65]. Another method is the Quadrature Method of Moments [19, 23]. In this work we also evaluate the application of the Sectional Quadrature Method of Moments (SQMOM) to kinetic equations, which until now is mainly known from the application to Population Balance Equations (PBE) [5].

The main goal of the following work is to develop a method, which is suitable to simulate problems of rarefied gases inside geometries with moving boundaries for any Knudsen number. This intention is motivated by the complexity of the simulation of gas flows inside vacuum pumps, since here one often deals with all three flow regimes of a diluted gas and with moving boundaries. As an example we consider the functional principle of a rotary vane pump [60], which is illustrated in Figure 1.1¹. The pumping

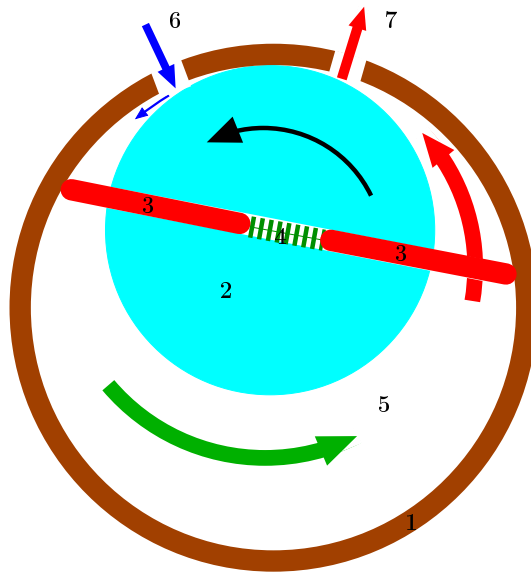


Fig. 1.1: *Schematic representation of a rotary vane pump.*

system consists of a housing (1), a rotor (2), vanes (3) that move radially under a spring (4) force and an inlet (6) and outlet (7). The working chamber (5) is located inside the housing. Rotor and vanes divide it into three separate spaces having variable volumes. As the rotor turns, gas flows into the enlarging chamber until it is sealed off by the second vane. Then the gas is compressed until the outlet valve opens against atmospheric pressure. During this process all three flow regimes are present, from molecular flow as the gas is spreading into the working chamber to a continuous fluid in the end as the gas is compressed.

¹Picture created with Xfig [64]

In the last few years moving boundary problems have become a popular topic in the field of kinetic theory of gases and their applications. There are already many attempts to solve such problems in literature. One of them is, of course the DSMC method [47, 7], which is not an optimal method in the case of transition region flows (see above). In connection to moving boundaries one has to take the ensemble average over many independent runs in order to reduce the fluctuation inherent to the method, which is very time consuming.

There also exist some works, which use deterministic methods to solve model Boltzmann equations with the help of known techniques in CFD. The most common ones are moving mesh methods [17], which are associated with high computational costs because of the complexity near the moving parts.

Another known approach is the immersed boundary method [18]. In this approach the equation is solved on a Cartesian grid, where even the solid part of the domain is meshed, but in the solid cells the equation is not solved. These cells are instead filled with boundary conditions. Thus, the main difficulty is to choose these conditions properly.

Recently, Russo and Filbet [50] proposed a Semi-Lagrangian scheme for one dimensional problems, in which the solution is computed on a fixed Cartesian Grid, as well. Boundary conditions are treated by using ghost cells and the source term of the kinetic equation is treated implicitly. However, near the boundary one needs to use more complex interpolation techniques, which again increases the computational costs.

To overcome the last problem we intend to use is the Finite Pointset Method (FPM) [62, 63, 52], which is a mesh free numerical method developed at the Fraunhofer Institute for Industrial Mathematics (ITWM) Kaiserslautern and is mainly used to solve fluid dynamical problems. In this approach the medium is represented by a finite set of points, which are not required to be regularly arranged. Each of the points is endowed with the local properties as the density, velocity, pressure and temperature. The sampling points can move with the medium or they may be fixed in space while the medium flows through them. The advantage of this method over the classical CFD methods is more evident in the case of problems with complicated geometries and/or moving boundaries. Here, the software complexity that would be required to handle these features with topological data structures, commonly used in CFD, is avoided.

In this work this method is applied to the Bhatnagar-Gross-Krook (BGK) model [6, 45], in which the complex collision term of the Boltzmann equation is replaced by a simple relaxation expression. It is assumed that the net effect of collisions makes the velocity distribution function relax toward a local equilibrium distribution over a characteristic time. This is done in order to avoid difficulties, which arise by solving the nonlinear multidimensional integral collision term in the Boltzmann equation.

1.2 Outline of the work

In Chapter 2 we begin with an insight into the kinetic theory of gases by shortly explaining the derivation of the Boltzmann equation in Section 2.1. In Section 2.2 we introduce the kinetic BGK model (2.19), which plays a super ordinate role throughout this work. A short introduction into boundary conditions assigned to kinetic models is given in Section 2.3.

Chapter 3 serves to give an insight into the Finite Pointset Method, which is the main numerical tool used in this work. In this context it is worth to mention the Weighted Least Squares Method, see Section 3.4. The numerical tests, presented in Section 3.5 serve to show the accuracy of FPM.

Furthermore, in Chapter 4 some numerical methods for solving the BGK model equation in the regular grid setting are presented. The methods are compared to each other and the different merits are pointed out in Section 4.3. Special attention should be pointed to the Semi-Lagrangian method, see Section 4.1.2. This method represents the basis for the numerical method derived in this work.

In Chapters 5 and 6 we are concerned with the implementation of our numerical method using FPM and the idea of the Semi-Lagrangian method for one and two dimensional spatial and velocity space. Some numerical results are presented in Sections 5.1.3 and 5.2.3. The results are compared to the results given through DSMC- and CFD methods, respectively.

In Sections 6.1 - 6.3 our method is applied to moving boundary problems in one and two dimensional spatial and velocity space. The focus lies in the implementation of

boundary conditions, see Sections 6.1.1, 6.2.1, 6.2.2, 6.3.1. Furthermore, some merits of FPM over regular grid methods are pointed out.

Finally, in the last chapter some conclusions and a few ideas for future work in the field of this thesis are given.

Chapter 2

The Boltzmann equation and the kinetic BGK model

The following chapter serves to give an insight into kinetic theory of gases. The main tool to describe the behavior of such gases is the Boltzmann equation. The BGK model, which represents the main component in this work, is a simplification of this equation. Furthermore, common models for kinetic boundary conditions are presented in the end of this chapter.

2.1 Boltzmann equation

The Boltzmann equation is an integrodifferential equation describing the behavior of a dilute gas and was established by Ludwig Boltzmann in 1872 [2, 3, 11]. It forms the basis for the kinetic theory of gases. In the following section we describe the derivation of this equation. The following explanations are based on the book by Cercigniniani from 1976 [11].

Let us consider a macroscopic volume of gas, for example a 1cm^3 cube, which is a system of a very large number, for instance 10^{20} molecules moving in a rather irregular way. We may assume, that the molecules are particles obeying the laws of classical mechanics and the laws of interaction between the molecules are perfectly known. From this it follows that the evolution of the system is computable, if suitable initial data are given. If our molecules, for example, are mass points, the equations of motion

are:

$$\begin{aligned}\dot{x}_i &= v_i, \\ \dot{v}_i &= X_i, \quad i = 1, \dots, N,\end{aligned}\tag{2.1}$$

where N is the number of particles and x_i and v_i the corresponding positions and velocities.

It holds, that:

$$X_i = \frac{F}{m},\tag{2.2}$$

where $X = (X_1, X_2, \dots, X_N) \in \mathbb{R}^N$, F is the external force and m is the mass of the particles, which is supposed to be the same for each particle.

Furthermore, let $x_i = (x_i^{(1)}, x_i^{(2)}, x_i^{(3)}) \in \mathbb{R}^{3N}$, $v_i = (v_i^{(1)}, v_i^{(2)}, v_i^{(3)}) \in \mathbb{R}^{3N}$.

Remark 2.1. Note, that x_i and v_i are functions of t .

To compute the time evolution of the system, one has to solve $6N$ first-order differential equations with $6N$ initial conditions:

$$x_i(0) = x_{i0}, \quad \dot{x}_i(0) = v_i(0) = v_{i0}.\tag{2.3}$$

As one can see, the computational effort required to solve this number of equations, would be enormous. Thus, we need to find a more efficient way to describe the problem.

We proceed using the following steps.

First, we rewrite system (2.1) into one equation:

$$\dot{z} = \frac{\partial z}{\partial t} = Z\tag{2.4}$$

with $z = (x, v)$ and $Z = (\dot{x}, \dot{v}) = (v, X)$.

In principle, we have to consider the evolution of the state of particles in a phase space, which is a $6N$ dimensional space, where the coordinates are the $3N$ components of the N position vectors x_i and the $3N$ components of the N velocities v_i . It follows that the state of the system at time t is a point, whose coordinates are the $6N$ values of the position vectors and velocities of the N particles.

Let $P_N(z, t)$ be the probability density of the system of N particles at state z at any

time t . That means, $P_N(z)dz$ gives the probability that the point z is present in the interval $z + dz$. The probability density fulfills:

$$\int_{\mathbb{R}^{6N}} P_N(z, t) dz = 1. \quad (2.5)$$

In other words, we replace the representative point by a continuous distribution with density proportional to the probability density. It means that the system of mass points is replaced by a sort of fluid with density proportional to P and with velocity $\dot{z} = Z$. Conservation of mass yields:

$$\frac{\partial P_N}{\partial t} + \text{div}(P_N Z) = 0. \quad (2.6)$$

Furthermore, it holds, that:

$$\text{div}(P_N Z) = Z \nabla P_N + P_N \text{div} Z. \quad (2.7)$$

Usually, we have $\text{div} Z = 0$. In fact, since x_i and v_i are independent variables and the force per unit mass is also velocity independent, it follows, that:

$$\text{div} Z = \sum_{i=1}^N \left(\frac{\partial}{\partial x_i} \cdot v_i + \frac{\partial}{\partial v_i} \cdot X_i \right) = \sum_{i=1}^N N \frac{\partial}{\partial v_i} \cdot X_i = 0. \quad (2.8)$$

We get the following equation, also called *Liouville equation*:

$$\frac{\partial P_N}{\partial t} + Z \cdot \frac{\partial P_N}{\partial z} = 0, \quad (2.9)$$

which can be rewritten into:

$$\frac{\partial P_N}{\partial t} + \sum v_i \cdot \frac{\partial P_N}{\partial x_i} + \sum X_i \cdot \frac{\partial P_N}{\partial v_i} = 0, \quad (2.10)$$

where $\frac{\partial P_N}{\partial x_i}$ are the gradients in the 3-dimensional space of the positions of the i^{th} particle and $\frac{\partial P_N}{\partial v_i}$ the gradients of the velocities. The probability for the point representing the system to be found in a region R of the phase space at time t is:

$$\text{Prob}(z \in R) = \int_R P_N(z, t) dz. \quad (2.11)$$

When the external force X_i , $i = 1, \dots, N$ is absent, the Liouville equation becomes:

$$\frac{\partial P_N}{\partial t} + \sum v_i \cdot \frac{\partial P_N}{\partial x_i} = 0. \quad (2.12)$$

The probability function P_N describes the state of all N particles at a certain time t . Now we consider only one particle, for example $i = 1$ in the state (x_1, v_1) at time t . That means, we search for the probability for one particle to be in the element $dx dv$ at time t .

In our case we deal with a gas of hard spheres. That is, as we were looking at a system of many "billiard balls". It is a system of spheres with diameter σ , which can not overlap in space, so the particles can not interact, if the distance between them is larger than σ , but they do interact, if their distance is exactly equal to σ . Let $P_N^{(1)}$ be the distribution function for the state of one particle. $P_N^{(1)}$ is related to P_N by:

$$P_N^{(1)}(x_1, v_1, t) = \int P_N \prod_{i=2}^N dx_i dv_i. \quad (2.13)$$

If the particle would move without interaction with the other particles, it would satisfy the Liouville equation, but if it is interacting with the other particles, we have:

$$P_N = 0, \quad |x_i - x_j| < a, \quad i \neq j. \quad (2.14)$$

In this case the Liouville equation (2.12) doesn't hold, if we integrate it over the space $\mathbb{R}^{6(N-1)}$. It holds, that:

$$\frac{\partial P_N^{(1)}}{\partial t} + Z \cdot \frac{\partial P_N^{(1)}}{\partial z} = S, \quad (2.15)$$

where S has a complicated integral form.

Remark 2.2. The description of S is skipped here, since it is not relevant for this work. However, it can be found in the previously mentioned book [11].

For a very large number of particles ($N \rightarrow \infty$ and $\sigma \rightarrow 0$) it is appropriate to introduce another one particle expected mass density function:

$$f(t, x, v) = N \cdot m P_N^{(1)}, \quad (2.16)$$

where $f(t, x, v) dx dv$ is the expected mass of particles in the volume between (x, v) and $(x + dx, v + dv)$. If we now rewrite the Liouville equation in terms of f , we get:

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} = Q(f, f), \quad (2.17)$$

where Q is called *collision integral*. This equation is called **Boltzmann equation**. In our case, for the gas of hard spheres, we can write the collision integral in the following form:

$$Q(f, f) = \frac{\sigma^2}{m} \int (f' f'_* - f f_*) |V \cdot n| dn dv_*, \quad (2.18)$$

where:

$$\begin{aligned} V &= v - v_*, \\ v' &= v - n(n \cdot (v - v_*)), \\ v'_* &= v_* + n(n \cdot (v - v_*)). \end{aligned}$$

Here, v is the current velocity, v_* the integration variable and v' , v'_* are the velocities corresponding to v and v_* after collision, $|n| = 1$ and $n \cdot (v - v_*) > 0$. Furthermore, $f' = f(v')$, $f_* = f(v_*)$ and $f'_* = f(v'_*)$, respectively.

2.2 BGK model

Since the collision operator of the Boltzmann equation (2.18) has a rather complicated integral form, there are many alternative, simpler expressions proposed for the right hand side of the Boltzmann equation (2.17). They are known as *collision models*.

The idea behind these models is that a large amount of the details of two-body interaction, which is contained in the collision term in general does not influence the values of many experimentally measured quantities. That is, in most situations the collision operator $Q(f, f)$ can be replaced by a somehow blurred image which only retains the qualitative and average properties of the true collision operator.

One of the most popular collision models is the so called **Bhatnagar-Gross-Krook (BGK) model** [6], which reads:

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} = \frac{1}{\tau} (M[f] - f), \quad (2.19)$$

where $x \in \mathbb{R}^{d_x}$ and $v \in \mathbb{R}^{d_v}$. Here, $d_x = 1, 2, 3$ denotes the dimension of the physical- and $d_v = 1, 2, 3$ of the velocity space, respectively.

Furthermore, M is the so called *Maxwellian*:

$$M[f] = \frac{\rho}{(2\pi RT)^{\frac{d_v}{2}}} \exp\left(-\frac{\|v - u\|^2}{2RT}\right), \quad (2.20)$$

where ρ , u and T denote the macroscopic fields, namely the density, mean velocity and the temperature. Moreover, R is the gas constant.

This model describes a gas which is tending to a local Maxwellian. This tendency is also reflected in the fact that, for the spatially homogeneous gas, we have exact solutions.

Remark 2.3. The Maxwellian can be calculated by setting the collision operator of the Boltzmann equation $Q(f, f) = 0$. More exactly, the states of the thermodynamic equilibrium, that is, states, in which $\frac{\partial f}{\partial t} = 0$ and $\frac{\partial f}{\partial x} = 0$ are characterized by the Maxwellian distribution.

Furthermore, τ denotes the relaxation time and may be a function of the density ρ and the temperature T . Several models are discussed in [30].

Remark 2.4. For some computations in this work we use the following definition for τ [16]. As already mentioned in the motivation, the Knudsen number is the ratio of the mean free path λ to the representative physical length scale L :

$$Kn = \frac{\lambda}{L}, \quad (2.21)$$

where λ is computed as:

$$\lambda = \frac{\kappa_B}{\sqrt{2.0\pi\rho R}d^2}, \quad (2.22)$$

where κ_B is the Boltzmann constant and is set to $1.3806e^{-23}JK^{-1}$ and d is the particle hard shell diameter.

Then, from equations (2.21) and (2.22) it follows, that τ is related to Kn through:

$$\tau = \frac{KnL}{\sqrt{\left(\frac{8RT}{\pi}\right)}}. \quad (2.23)$$

However, most of the computations in the following work are performed by assuming that τ is constant. If we write the BGK equation in non dimensional form, then τ even becomes the Knudsen number.

The main advantage of the BGK-model is that for any given problem one can deduce integral equations for the macroscopic quantities by evaluating the so called *moments* of the distribution function f :

$$(\rho, \rho u, E) = \langle f \phi(v) \rangle = \int_{\mathbb{R}^{d_v}} \phi(v) f(t, x, v) dv, \quad (2.24)$$

where $\phi(v) = (1, v, 0.5|v|^2)^T$ are the so called *collision invariants*. The quantity E is the total energy density and is related to the temperature by the internal energy e :

$$e(x, t) = \frac{d_v}{2} RT(x, t), \quad \rho e = E - \frac{1}{2} \rho u^2. \quad (2.25)$$

The collision invariants satisfy the following equation:

$$\int_{\mathbb{R}} \phi(v) Q(f, f) dv = 0. \quad (2.26)$$

By the above definition of moments, we can get the conservation laws for the mass density, momentum density and energy density:

$$\begin{aligned} \phi(v) = 1 &\Rightarrow \frac{\partial \langle f \rangle}{\partial t} + \nabla_x \cdot \langle f v \rangle = 0, \\ \phi(v) = v &\Rightarrow \frac{\partial \langle f v \rangle}{\partial t} + \nabla_x \cdot \langle v \times v f \rangle = 0, \\ \phi(v) = \frac{1}{2} |v|^2 &\Rightarrow \frac{\partial \langle \frac{1}{2} |v|^2 f \rangle}{\partial t} + \nabla_x \cdot \langle \frac{1}{2} |v|^2 f v \rangle = 0. \end{aligned} \quad (2.27)$$

This equations are in general not closed, since the number of unknowns is larger than the number of equations.

In the fluid dynamic limit, as $\tau \rightarrow 0$, the distribution function f converges to a local Maxwellian and the system (2.27) becomes a closed system. The conserved quantities satisfy the classical Euler equations of gas dynamics for a mono atomic gas:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \sum_{j=1}^{d_x} \frac{\partial \rho u_j}{\partial x_j} &= 0, \\ \frac{\partial \rho u_i}{\partial t} + \sum_{j=1}^{d_x} \frac{\partial}{\partial x_j} (\rho u_i u_j + p \delta_{ij}) &= 0, \\ \frac{\partial E}{\partial t} + \sum_{j=1}^{d_x} \frac{\partial}{\partial x_j} (u_j (E + p)) &= 0. \end{aligned} \quad (2.28)$$

System (2.28) consists of $2 + d_x$ equations in $3 + d_x$ unknowns. The pressure p is related to e by the following equation:

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho|u|^2\right), \quad (2.29)$$

where the poly-tropic constant $\gamma = \frac{d_v+2}{d_v}$ represents the ratio between specific heat at constant pressure and at constant volume.

The BGK model contains most of the basic features of the Boltzmann collision integral. However, it also has its shortcomings. Some of them can be avoided by suitable modifications. These modifications, however, come at the expense of the simplicity of the model.

For example, τ can vary with the molecular velocity, so it can depend on v instead being locally constant. This modification is justified by the circumstance that a computation of τ for physical models of the molecules, shows that it varies with the molecular velocity. In this case most of the properties of the BGK model are retained, but the macroscopic quantities, which appear in the Maxwellian are not the density, velocity and temperature of the gas, but some fictitious local parameters [37].

Another model is the so called (ellipsoidal statistical) *ES-BGK model* [12]. In this model, the local Maxwellian is substituted by a three dimensional Gaussian. This is done to adjust the model to give the same Navier-Stokes equations as the full Boltzmann equation.

A different model is offered by a Fokker-Planck collision term [29]. This model has been first introduced in connection with the kinetic theory of liquids. In some cases it has the same kind of non-linearity as the true Boltzmann equation.

Furthermore, one has to notice, that the idea of kinetic models can be naturally extended to mixtures and polyatomic gases, see [11].

Remark 2.5. In this work we just consider the BGK model. An extension of the presented numerical method should be straightforward to other kinetic models.

The BGK model has been extensively theoretically investigated [43] and many numerical computations have been carried out in order to validate its properties [15, 71]. The advantage of this model for the discretization via a deterministic method lies in the possibility of linear costs. In addition, the BGK-model is known to be very accurate in

near-equilibrium regions since it has the same fluid limit as the Boltzmann equation. In some cases it also works far from the equilibrium, see [25]. This model is especially used in transitional regions, where the Navier-Stokes equations fail and the Boltzmann equation is very expensive.

2.3 Boundary conditions

For many applications the gas flow, which is described by the Boltzmann equation takes place in a region bounded by one or several solid bodies. In these cases boundary conditions have to be prescribed to characterize the behavior of the gas near the walls [9].

In this work we refer to the boundary conditions Maxwell proposed in a paper of 1879 [35].

The first condition he proposed corresponds to a simple gas-solid interaction where we assume that the wall is smooth and perfectly elastic, so that the particles of the gas are specularly reflected, see left part of Figure 2.1. The condition writes:

$$f(t, x, v) = f(t, x, v - 2n(n \cdot v)), \quad x \in \Gamma, \quad v \cdot n > 0, \quad (2.30)$$

where Γ denotes the boundary and n is the unit vector to the surface at x .

Maxwell noticed that equation (2.30) means that the gas can exert any stress on the surface only in the direction of the normal. This assumption is extremely unrealistic and can only be used in particular cases, since in many situations the gas can exert stresses in oblique directions. This is why he also introduced another type of boundary condition corresponding to a more complex gas-solid interaction. Physically he supposed that the wall consists of fixed elastic spheres and that the molecules going from the gas to the wall must collide once or more with the spheres. In this case, the particle is reflected into the gas with a velocity taken with the probability whose density corresponds to the equilibrium state of the gas. Thus, the boundary condition writes:

$$f(t, x, v) = \frac{\rho_\Gamma}{2\pi(RT_\Gamma)} \int_{v' \cdot n < 0} (|v' \cdot n| f(t, x, v') dv') \left(\exp\left(-\frac{\|v\|^2}{2RT_\Gamma}\right) \right), \quad v \cdot n > 0, \quad (2.31)$$

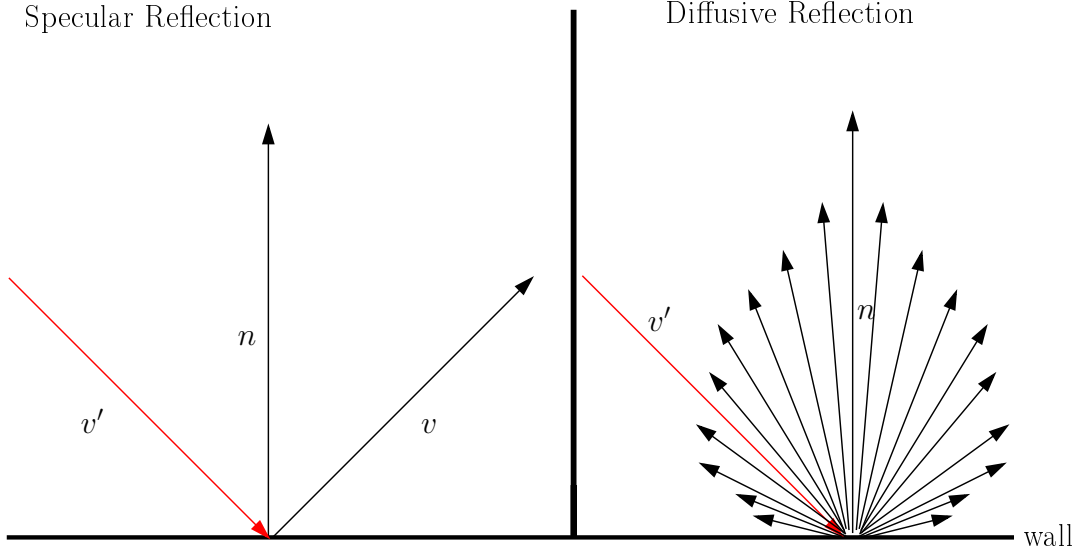


Fig. 2.1: *Specular and diffusive reflection.*

where T_Γ is the temperature and ρ_Γ the density at the wall. This model is known as *diffusive reflection condition*, see also right part of Figure 2.1.

Finally, there is also a more complicated intermediate situation which is devoted to be more physically realistic. Maxwell postulated that there is a fraction of the gas which accommodates to the temperature of the solid and another one which is reflected by the solid. The mixed boundary condition reads:

$$f(t, x, v) = (1 - \alpha)f(t, x, v - 2n(n \cdot v)) + \alpha \frac{\rho_\Gamma}{2\pi(RT_\Gamma)} \int_{v' \cdot n < 0} (|v' \cdot n| f(t, x, v')) \exp\left(-\frac{\|v\|^2}{2RT_\Gamma}\right), v \cdot n > 0, \quad (2.32)$$

where $\alpha \in [0, 1]$ is called the *accommodation coefficient*. It represents the tendency of the gas to accommodate to the wall. It means that the fraction $(1 - \alpha)$ of molecules satisfies specular boundary conditions whereas a fraction of α satisfies diffuse boundary conditions. If $\alpha = 0$ we have specular boundary conditions and if $\alpha = 1$ we recover diffusive boundary conditions.

Remark 2.6. Note, that the above formulas are only valid, if the wall is at rest. If

we deal with a moving boundary, we have to replace v by $v - u_\Gamma$, where u_Γ denotes the velocity of the wall.

Chapter 3

Finite Pointset Method (FPM)

As already mentioned, we use the Finite Pointset Method (FPM) [62, 63, 52] for our numerical computations.

FPM is a grid free method for solving fluid dynamical problems. The basis for the computations is a point cloud, which represents the flow field. The points of this cloud are carriers of all relevant physical information, like density, velocity, pressure and so on. It has to cover the whole flow domain. The boundaries are also represented by a finite number of points, on which the boundary conditions are prescribed. Since it is a grid free method and the distribution of particles can be quite arbitrary, the method is suitable for flows with complicated geometries, for example with moving boundaries or multiphase flows.

Another known grid free method is the method of Smoothed Particle Hydrodynamics (SPH) [26], which was originally invented to solve astrophysical problems without boundaries. The main difficulty of this method is the implementation of boundary conditions.

A further approach for solving fluid dynamic equations in the grid free framework is the Moving Least Squares or the Weighted Least Squares (WLSQ) method [62, 52]. In this approach boundary conditions can be implemented naturally by just placing the particles on the boundary and prescribing boundary conditions on them.

The main difference between FPM and the classical finite differences is, that the points don't have to be regularly arranged. If the point cloud would reduce to a regular point grid, then FPM would become a typical finite differences method.

A step by step explanation of how FPM works in a two dimensional physical space

is given in the following sections. First the FPM particles or points are irregularly distributed inside the computational domain. As a next step, we establish lists of neighboring particles for each particle, which are needed for numerical approximation. After that a hole filling and removing algorithm is applied to guarantee a good distribution of points. As a last step we show how to approximate function values and differential operators using the WLSQ method.

3.1 Initializing the point cloud

First we introduce the function $h(x, y)$, which describes the average distance between particles. Throughout this thesis, h is chosen to be a constant.

The computational domain has a boundary with piecewise linear shell functions, for example line segments in $2D$. In the first step we set particles in the center of each shell element. These particles are then the sources for other new particles, which have minimum distance $\beta h(x, y)$ and maximum distance $h(x, y)$, where $0 < \beta \leq 1$. Then we use these particles as sources for new particles as long as the boundary is filled.

To establish interior particles we use the boundary particles as sources. After we have constructed one layer inside the domain, the available particles again are sources for other new interior particles. We successively insert new particles until the center is reached and the minimum and maximum conditions are satisfied.

To maintain a qualitative particle distribution, it means to prevent clustering or scattering of the particles we have to apply a hole filling and removing algorithm, which is described in the following sections.

3.2 Neighbor searching

Now, that the particles are numbered from $j = 1$ to N_{xy} and we have direct access to position $(x_j, y_j)^T$, we search for neighbors of each particle inside a ball with radius h . In our simulations we have assumed constant h all over the domain.

To obtain the neighborhoods we construct a voxel data structure. The voxels form a regular grid of squares in $2D$ and have side length h .

Then we establish three types of lists. The first loops over all points and computes its voxel. The second list can be obtained from the first by sorting with respect to the voxel indices. For a point $(x_j, y_j)^T$ all points inside the ball $B((x_j, y_j)^T, h)$ need to be found. The second list gives direct access to the voxel, where this point is contained in. Now all points in the voxel and its neighboring voxels are tested to be in the ball $B((x_j, y_j)^T, h)$. The indices of the points in these voxels are given by the first list. Finally, the neighborhood information is saved in the third list [20].

3.3 Finding holes and removing points

The next step is to find holes in the point cloud, that is to find points, where the distance between the points is larger than βh . If it is the case, we have to add new points to remove the hole.

Also, we have to find points which are too close to each other, that is closer than $\beta h(x, y)$ and remove those. If we find two such points, both are removed and replaced by a new particle. The new particle is then inserted in the center of mass of the two particles, which were removed and the data on this particle is interpolated [20].

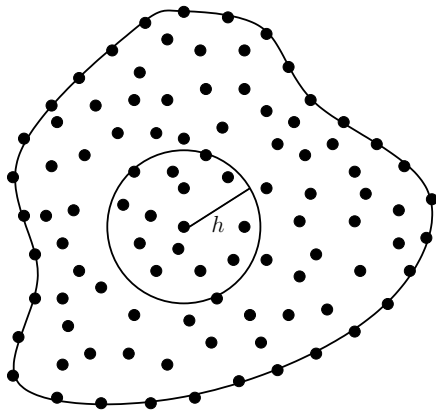


Fig. 3.1: *Distribution of computational points inside a two dimensional domain in FPM.*

3.4 Approximation of function values and differential operators

In the next step we have to evaluate the values we are interested in, on every particle position. We approximate the function values and the derivatives of the functions using the Weighted Least Squares method.

The FPM particle positions are themselves points of the geometrical grid. Thus, the approximation of derivatives of a function at each grid point means their approximation on each particle position.

Let u be a function in $2D$ and u_j its values at z_j for $j = 1, 2, \dots, N_{xy}$, where $z_j = (x_j, y_j)^T$. We want to construct the function values and the derivatives of $u(z)$ at every particle position z from its neighboring cloud of points. Let z be an arbitrary particle position and $P(z, h) = \{z_k : k = 1, 2, \dots, N_{xy}, \|z_k - z\| \leq h\}$ be the set of m neighbor points of z in a ball with radius h . Without loss of generality, we assume, that $P(z, h) = \{z_1, \dots, z_m\}$. The central particle z is also contained in $P(z, h)$. For consistency reasons we require that there should be at least 6 particles including the central particle.

We use a weight function $w = w(r)$ with small compact support, where $r = \frac{\|z - z_k\|}{h}$ and h determines the size of the support. We need this function to make sure, that the values on the particles which are closer to z have a higher impact on the interpolation than those which are far from it.

Our aim is to try to find a function which satisfies the following condition:

$$w(r) \rightarrow \begin{cases} 0, & \text{if } r \rightarrow 1 \\ 1, & \text{if } r \rightarrow 0 \end{cases} \quad (3.1)$$

For example, we can use a Gaussian function of the type:

$$w(r) = \begin{cases} \exp(-\theta r^2), & \text{if } r < 1 \\ 0, & \text{otherwise} \end{cases}, \quad (3.2)$$

where θ is a positive constant.

However, we also require w to be n -times continuously differentiable for $r \in [0, 1]$ and continuous in \mathbb{R}^+ .

For example, we can use:

$$w(r) = \begin{cases} \exp(-\eta \cdot r^2) - \exp(-\eta), & \text{if } r \leq 1 \\ 0, & \text{otherwise} \end{cases}, \quad (3.3)$$

where η is a positive constant.

Another choice of w is given by:

$$w(r) = \begin{cases} (1 - r^2)r, & \text{if } r \leq 1 \\ 0, & \text{otherwise} \end{cases}. \quad (3.4)$$

We can group the weight values in the so called *weight matrix*, which has a diagonal form:

$$W = \begin{pmatrix} w_1 & 0 & 0 & \cdot & \cdot & 0 \\ 0 & w_2 & 0 & \cdot & \cdot & 0 \\ 0 & 0 & w_3 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & w_m \end{pmatrix}.$$

As in the classical discretization method, we have to construct shape functions ϕ_k^* , $k = 1, \dots, m$ (here $*$ is a placeholder for the numerous differential operators) such that:

$$\begin{aligned} u &\approx \sum_{k=1}^m \phi_k^0 u_k, \\ u_x &\approx \sum_{k=1}^m \phi_k^x u_k, \\ u_y &\approx \sum_{k=1}^m \phi_k^y u_k, \\ \Delta u &\approx \sum_{k=1}^m \phi_k^\Delta u_k. \end{aligned} \quad (3.5)$$

We show how the method works on the example of the Laplacien operator. We want to approximate:

$$\Delta u \approx \sum_{k=1}^m \phi_k^\Delta u_k. \quad (3.6)$$

We use the Taylor's series expansion and the least squares approximation to construct ϕ_k^Δ . We consider the Taylor expansion of $u(z_k)$ around z :

$$u(z_k) = u(z) + dx_k u_x(z) + dy_k u_y(z) + 1/2 dy_k^2 u_{xx}(z) + dx_k dy_k u_{xy}(z) + 1/2 dy_k^2 u_{yy}(z) + e_k \quad (3.7)$$

for $k = 1, \dots, m$, where $dx_k = x_k - x$, $dy_k = y_k - y$ and e_k the error in the Taylor series expansion at the point z_k .

The next step is to multiply equation (3.7) by ϕ_k^Δ and sum over 1 to m :

$$\begin{aligned} \sum_{k=1}^m \phi_k^\Delta u_k &= \left(\sum_{k=1}^m \phi_k^\Delta \right) u + \left(\sum_{k=1}^m \phi_k^\Delta dx_k \right) \cdot u_x + \left(\sum_{k=1}^m \phi_k^\Delta dy_k \right) \cdot u_y + 1/2 \left(\sum_{k=1}^m \phi_k^\Delta \cdot dy_k^2 \right) \cdot u_{xx} \\ &\quad + \left(\sum_{k=1}^m \phi_k^\Delta dx_k \cdot dy_k \right) \cdot u_{xy} + 1/2 \left(\sum_{k=1}^m \phi_k^\Delta \cdot dy_k^2 \right) \cdot u_{yy} + e_k. \end{aligned} \quad (3.8)$$

From the expression (3.8) we obtain the approximation of the Laplacien operator

$$\Delta u \approx \sum_{k=1}^m \phi_k^\Delta u_k + e_k, \quad (3.9)$$

if the following conditions are satisfied:

$$\begin{aligned} \sum_{k=1}^m \phi_k^\Delta &= 0, \\ \sum_{k=1}^m \phi_k^\Delta dx_k &= 0, \\ \sum_{k=1}^m \phi_k^\Delta dy_k &= 0, \\ \sum_{k=1}^m \phi_k^\Delta dx_k^2 &= 2, \\ \sum_{k=1}^m \phi_k^\Delta dx_k dy_k &= 0, \\ \sum_{k=1}^m \phi_k^\Delta dy_k^2 &= 2. \end{aligned} \quad (3.10)$$

These conditions simply mean, that if ϕ_k^Δ is applied to constant or linear functions it delivers 0 and if applied to quadratic functions it delivers 2.

The 6 constraints from (3.10) are denoted by K_1, \dots, K_6 .

To obtain a stable linear system we add one more condition that the coefficient on the central particle must be negative, similar to the classical finite differences discretization. Therefore, we consider the following condition:

$$\phi^\Delta(z_k = z) = -\frac{\kappa}{h^2} = K_7, \quad (3.11)$$

where κ is a non-negative small number. Hence, we have 7 constraints for the approximation of the Laplacien operator.

Now, we minimize the functional:

$$J = \sum_{k=1}^m \frac{(\phi_k^\Delta)^2}{w_k} \quad (3.12)$$

with respect to K_1, \dots, K_7 . From the Euler-Lagrange theorem it follows, that:

$$\frac{\partial J}{\partial \phi_k^\Delta} + \sum_{j=1}^7 \lambda_j \frac{\partial K_j}{\partial \phi_k^\Delta} = 0, \quad (3.13)$$

where $\lambda_1, \dots, \lambda_7$ are the Lagrange multipliers. More explicitly we express

$$\frac{2\phi_k^\Delta}{w_k} + \lambda_1 + dx_k \lambda_2 + dy_k \lambda_3 + dx_k^2 \lambda_4 + dx_k dy_k \lambda_5 + dy_k^2 \lambda_6 + \lambda_7 \delta_{kz} = 0 \quad (3.14)$$

for $k = 1, \dots, m$, where

$$\delta_{kz} = \begin{cases} 1, & \text{if } z_k = z \\ 0, & \text{else.} \end{cases} \quad (3.15)$$

That means we have a system of equations, which can be written as:

$$Ma = b, \quad (3.16)$$

where:

$$M = \begin{pmatrix} 1 & dx_1 & dy_1 & dx_1^2 & dx_1 dy_1 & dy_1^2 & \delta_{1z} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & dx_m & dy_m & dx_m^2 & dx_m dy_m & dy_m^2 & \delta_{mz} \end{pmatrix}$$

and $a = (\lambda_1, \dots, \lambda_7)^T$, $b = -2(\frac{\phi_1^\Delta}{w_1}, \dots, \frac{\phi_m^\Delta}{w_m})^T$. In general we have $m > 7$ and therefore the linear system (3.16) is overdetermined and we obtain the solution from the WLSQ method:

$$\min_a (Ma - b)^T W (Ma - b). \quad (3.17)$$

The solution is explicitly given by:

$$a = (M^T W M)^{-1} (M^T W) b, \quad (3.18)$$

if $M^T W M$ is non singular, where $W = \text{diag}[w_1, \dots, w_m]$.

The vector $(M^T W) b$ can be written explicitly with the help of the constraints K_l , $l = 1, \dots, 7$:

$$(M^T W) b = (-2)(0, 0, 0, 2, 0, 2, -\frac{\kappa}{h^2})^T. \quad (3.19)$$

We can compute the Lagrange multipliers λ_i , $i = 1, \dots, 7$ and the shape functions ϕ_k^Δ , $k = 1, \dots, m$ can be expressed by:

$$\phi_k^\Delta = -\frac{w_k}{2} (\lambda_1 + dx_k \lambda_2 + dy_k \lambda_3 + dx_k^2 \lambda_4 + dx_k dy_k \lambda_5 + dy_k^2 \lambda_6 + \delta_{kx} \lambda_7). \quad (3.20)$$

Similarly we also can compute other derivatives and the function values itself, where the matrix M remains the same, only the constraints will be different, which results in different vectors $(M^T W) b$. For example to compute u_x :

$$(M^T W) b = -2(0, 1, 0, 0, 0, 0, \frac{-\kappa}{h^2}), \quad (3.21)$$

or to compute u_{xx} :

$$(M^T W) b = -2(0, 0, 0, 2, 0, 0, \frac{-\kappa}{h^2}). \quad (3.22)$$

Once we compute the Lagrange multipliers as above, we obtain the approximation Δu at particle z .

This procedure can also be applied for solving partial differential equations. For example, suppose we want to solve the Poisson equation of the form:

$$\Delta u = f, \quad (3.23)$$

with the boundary conditions:

$$\begin{aligned} u &= f \text{ on } \Gamma_D, \\ \frac{\partial u}{\partial n} &= h \text{ on } \Gamma_N, \end{aligned}$$

where n is the outer normal at the boundary and Γ_D and Γ_N denote the Dirichlet and Neumann boundary, respectively.

If the particle z for example belongs to the Neumann boundary, then the constraints mentioned above, are given by:

$$\begin{aligned} \sum_{k=1}^m \phi_k^\Delta &= 0, \\ \sum_{k=1}^m \phi_k^\Delta dz_k &= n. \end{aligned} \tag{3.24}$$

Now, if we consider z_j , an arbitrary particle and z_{jk} its neighbors of number $m(j)$, then we have to solve the following sparse linear system of equations:

$$\sum_{k=1}^{m(j)} \phi_{jk}^\Delta u_j = f_j, \tag{3.25}$$

where the vector consisting of the unknowns u_j , $j = 1, \dots, N_{xy}$ represents the solution of equation (3.23).

In this system the neighbor particle index gives the entry of the matrix components. Furthermore, it can be solved by any iterative solvers, such as for example the stabilized conjugate gradient method.

We iterate the procedure, until the error is smaller than a given threshold ϵ (see also [62]). Formally, the iteration stops, if:

$$\frac{\sum_{j=1}^{N_{xy}} |u_j^{(\tau+1)} - u_j^\tau|}{\sum_{j=1}^{N_{xy}} |u_j^{(\tau+1)}|} < \epsilon, \tag{3.26}$$

where τ denotes the number of iterations. The solution is then defined by $u_j = u_j^{\tau+1}$ for $j = 1, \dots, N_{xy}$ as τ tends to infinity. Here, ϵ is a very small positive constant and can differ according to the problem and the value of h .

3.5 Numerical tests

The numerical experiments in this section are performed in the two dimensional space. We compute a Poisson equation in a unit square, where the analytical solutions are also available. In the distribution of points we replace the boundaries with regular points of spacing 0.04 and the points inside the domain are distributed as was described in section 3.1, see also Fig. 3.2². The size of the support h is set to $h = 0.1$. As initial guess for $p^{(0)}$ we choose $p^{(0)} = 0$ at all points.

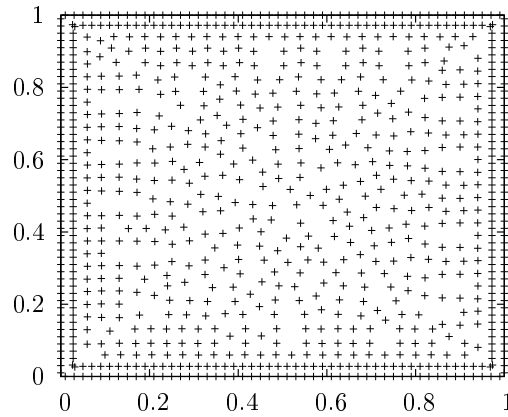


Fig. 3.2: *Distribution of computational points inside the domain $[0, 1]^2$.*

Remark 3.1. For details on the application of the Weighted Least Squares method to a Poisson equation, see for example [24, 62].

²Figure created with Gnuplot [69]

3.5.1 Dirichlet boundary value problem

We consider the Dirichlet boundary value problem (BVP):

$$\begin{aligned}\Delta p &= 2((y^2 - 1) + (x^2 - 1)) \text{ on } (0, 1) \times (0, 1), \\ p &= 0 \text{ on } x = 1 \text{ and } y = 1, \\ p &= -(y^2 - 1) \text{ on } x = 0, \\ p &= -(x^2 - 1) \text{ on } y = 0,\end{aligned}$$

where the analytical solution is given by:

$$p(x, y) = (x^2 - 1)(y^2 - 1).$$

In the following figure (Fig. (3.3)) we can see the comparison between the numerical and the exact solution of the problem:

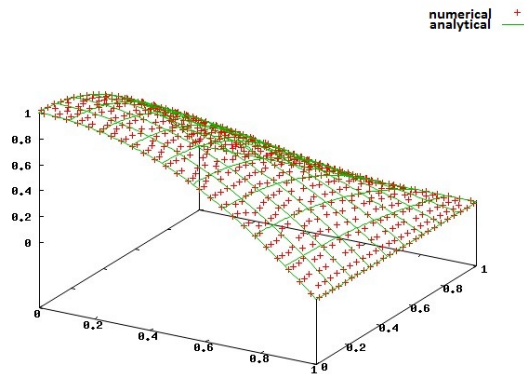


Fig. 3.3: Comparison between the analytical and numerical solutions for the Dirichlet BVP.

3.5.2 Neumann boundary value problem

Next, we consider the Neumann boundary value problem:

$$\begin{aligned}\Delta p &= -\cos(\pi x) \text{ on } (0, 1) \times (0, 1), \\ \frac{\partial p}{\partial n} &= 0 \text{ on } x = 0 \text{ and } y = 0, \\ \frac{\partial p}{\partial n} &= 0 \text{ on } x = 1 \text{ and } y = 1,\end{aligned}$$

with the analytical solution of the form:

$$p(x, y) = \frac{1}{\pi} \cos \pi x.$$

Again here we can see (Fig. 3.4) that the analytical and the numerical solution do conform to each other:

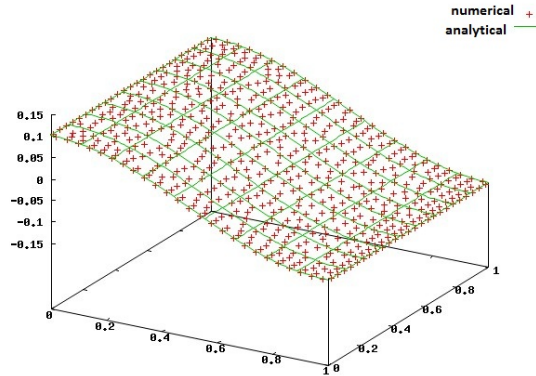


Fig. 3.4: Comparison between the analytical and numerical solutions for the Neumann BVP.

3.5.3 Mixed boundary value problem

Finally, we consider a mixed boundary value problem:

$$\begin{aligned} \Delta p &= -2(2y^2 - 3y^3 + 1) + 6(1 - x^2)(2y - 1) \text{ on } (0, 1) \times (0, 1), \\ \frac{\partial p}{\partial n} &= 0 \text{ on } y = 0 \text{ and } y = 1, \\ p &= 2y^3 - 3y^2 + 1 \text{ on } x = 0, \\ p &= 0 \text{ on } x = 1, \end{aligned}$$

where the analytical solution is defined as:

$$p(x, y) = (1 - x^2)(2y^3 - 3y^2 + 1).$$

Here, we also can see (Fig. 3.5) the comparison between the two solutions:

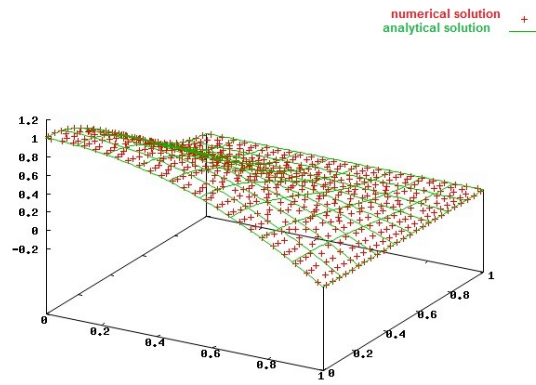
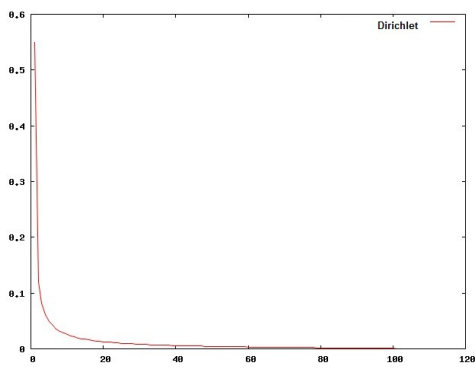
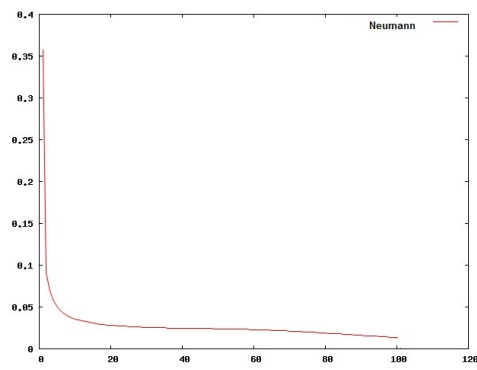


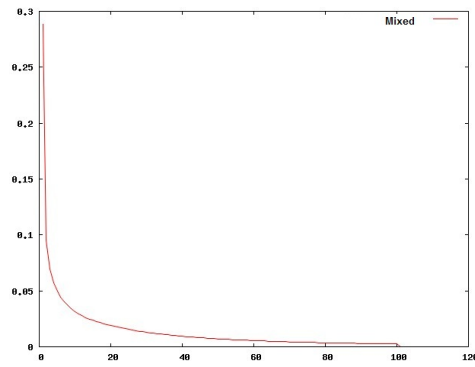
Fig. 3.5: Comparison between the analytical and numerical solutions for the mixed BVP.



(a) Dirichlet BVP



(b) Neumann BVP



(c) Mixed BVP

Fig. 3.6: Error vs. iterations

In Figure (3.6) the error, as defined on the left hand side of equation (3.26), for all three boundary value problems are plotted for the first 100 iterations. In all cases the errors converge to some small positive constants. Furthermore, we can see that the scheme is stable, i.e. that there are no oscillations.

The following table shows the maximum error between the numerical and the exact solutions for the Dirichlet boundary value problem for different values of h . One can see, that the error decreases with smaller values of h .

h	L^2 error
0.1	0.00408
0.05	0.00119
0.025	0.000307
0.0125	$5.25 \cdot 10^{-5}$

Chapter 4

Regular grid methods for solving the BGK model equation

In the following chapter we present some deterministic numerical methods, which are used to solve the BGK model equation on a regular grid. Furthermore, we compare these approaches with respect to their efficiency and the suitability for using the ideas in the FPM framework.

As already mentioned in the motivation of this work, there exist many methods to solve kinetic model equations, which can be grouped into two categories, namely statistical and deterministic ones.

The most popular statistical method is the Direct-Simulation Monte Carlo (DSMC) method [7]. This procedure uses simulation molecules which represent a large number of real molecules in a probabilistic simulation. The main assumption of the DSMC method is, that the molecular movement and the collision phases can be decoupled over time periods that are smaller than the mean collision time.

Unlike statistical techniques, deterministic ones don't work with probabilistic assumptions. They are based on the discretization of the equation in time-, physical- and velocity space, see for example [1, 10, 40, 44, 48, 50]. The main tools to do this are common CFD techniques [68], which are mainly used to solve fluid dynamic equations. In this chapter we describe some deterministic approaches, which already exist for fixed Cartesian grids. We show the problems, advantages and disadvantages of those and their suitability for implementation in a mesh-free framework.

First we introduce two methods, where the BGK model equation is directly discretized in space, time and velocity by CFD discretization techniques. After that, two ap-

proaches are presented, where partial differential equations are derived using the moments of the BGK equation. These equations are also solved by simple discretization procedures.

Furthermore, the methods will be compared with respect to their efficiency and the suitability for adapting them in a mesh-free framework.

Remark 4.1. The following methods are presented for a one dimensional spatial and velocity space.

Before we start to explain and compare the methods, we recall the one dimensional form of the BGK model equation, which reads:

$$\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial x} = \frac{1}{\tau}(M[f] - f), \quad (4.1)$$

where:

- τ is the relaxation time and is supposed to be a constant in the following chapter,
- $M[f]$ is the Maxwellian distribution with $M = \frac{\rho}{\sqrt{2\pi RT}} \exp(-\frac{|v-u|^2}{2RT})$

and:

- ρ is the density,
- R the gas constant,
- T the temperature and
- u the mean velocity.

The macroscopic quantities ρ , u and T can be calculated using the moments of f in the following way:

$$\begin{aligned} \rho &= \int_{\mathbb{R}} f(t, x, v) dv, \\ \rho u &= \int_{\mathbb{R}} v f(t, x, v) dv \\ E &= \int_{\mathbb{R}} \frac{1}{2} v^2 f(t, x, v) dv, \end{aligned} \quad (4.2)$$

where $E(x, t)$ is the total energy density and is related to the temperature by the internal energy e :

$$e(x, t) = \frac{1}{2}RT(x, t),$$

$$\rho e = E - \frac{1}{2}\rho u^2.$$

In section 2.2 we have already seen that if we multiply the BGK equation by the collision invariants and integrate it over v , for $\tau \rightarrow 0$ we get the Euler equations of gas dynamics:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0, \\ \frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x}(\rho u u + p) &= 0, \\ \frac{\partial E}{\partial t} + \frac{\partial}{\partial x}(\rho u E + u p) &= 0. \end{aligned} \tag{4.3}$$

4.1 Direct discretization methods

In the context of direct discretization methods the BGK equation is first discretized in velocity space. Then the resulting equations are solved using common CFD methods.

4.1.1 Implicit-Explicit scheme

The Implicit-Explicit scheme (*IMEX-Euler*) is a high-order scheme in space and in time [44]. It is based on a high order Implicit-Explicit discretization of the time derivative with the Runge-Kutta IMEX schemes [4]. The time discretization is carried out before the approximation of the space derivatives and the discretization of velocity space. In particular, the convective term is treated explicitly, while the source term is integrated implicitly. This approach allows treating even problems with infinite stiffness, which so far required the solution of large linear systems.

In the implicit step, first the macroscopic moments are evaluated and then the distribution function is updated, so conservation is naturally enforced.

Indeed, when a grid in velocity space is introduced, exact conservation no longer holds

at the discrete level. However, the error in the conserved quantities seems to be negligible.

Time discretization

For the sake of simplicity we consider a one dimensional problem in space and introduce a uniform grid in time with spacing Δt . Let $f^n(x, v) = f(x, v, n\Delta t)$. The updated distribution function is then given by:

$$f^{n+1}(x, v) = f^n(x, v) - \Delta t \sum_{i=1}^{\mu} \tilde{\omega}_i v \partial_x f^{(i)} + \Delta t \sum_{i=1}^{\mu} \frac{\omega_i}{\tau} (M^{(i)}(x, v) - f^{(i)}(x, v)), \quad (4.4)$$

where $M^{(i)}(x, v), i = 1, \dots, \mu$ are the Maxwellian functions obtained from the moments of the intermediate stages $f^{(i)}(x, v)$, which are defined as:

$$f^{(1)} = f^n + \Delta t \frac{a_{11}}{\tau} (M^{(1)} - f^{(1)}), \quad (4.5)$$

$$f^{(i)} = f^n - \Delta t \sum_{l=1}^{i-1} \tilde{a}_{il} v \partial_x f^{(l)} + \Delta t \sum_{l=1}^i \frac{a_{il}}{\tau} (M^{(l)} - f^{(l)}). \quad (4.6)$$

The coefficients $a_{il}, \tilde{a}_{il}, \omega_i, \tilde{\omega}_i$ define the IMEX Runge-Kutta scheme [44].

Let $\phi(v) = (1, v, \frac{1}{2}||v||^2)^T$ be the vector of collision invariants of f . To compute the moments of f for the first stage values we have:

$$\int_{\mathbb{R}} f^{(1)} \phi dv = \int_{\mathbb{R}} f^n \phi dv + \Delta t \frac{a_{11}}{\tau} \int_{\mathbb{R}} ((M^{(1)} - f^{(1)}) \phi) dv. \quad (4.7)$$

Since

$$\int_{\mathbb{R}} ((M^{(1)} - f^{(1)}) \phi) dv = 0, \quad (4.8)$$

we can immediately find the macroscopic variables $\rho^{(1)}, u^{(1)}, T^{(1)}$ corresponding to $f^{(1)}$ and with these we can compute the Maxwellian $M^{(1)}$:

$$M^{(1)} = \frac{\rho^{(1)}}{(2\pi RT^{(1)})^{\frac{1}{2}}} \exp\left(-\frac{||v - u^{(1)}||^2}{2RT^{(1)}}\right) \quad (4.9)$$

Once the Maxwellian is known we can find $f^{(1)}$ by solving equation (4.6). Here, we only consider the IMEX-scheme of first order, that is, $\mu = 1$. The resulting scheme reads:

$$f^{(1)} = \frac{\tau f^n + \Delta t a_{11} M^{(1)}}{\tau + \Delta t a_{11}}, \quad (4.10)$$

$$f^{n+1} = f^n - \Delta t \tilde{\omega}_1 v \partial_x f^{(1)} + \Delta t \frac{\omega_1}{\tau} (M^{(1)} - f^{(1)}). \quad (4.11)$$

One has to remark that the IMEX Runge-Kutta discretization conserves density, momentum and energy.

Space discretization

A uniform grid in space with grid points x_j is introduced. Let $f_j^n(v) = f(x_j, v, n\Delta t)$. The grid spacing is denoted by $\Delta x = x_{j+1} - x_j$. It is important that the space discretization provides an accurate approximation of $\partial_x f$. Furthermore, the discretization should be non-oscillatory and conservative.

Therefore a piecewise polynomial function \hat{f} such that $f_j = \frac{1}{\Delta x} \int_{I_j} \hat{f} dx$, where $I_j = (x_j - \frac{\Delta x}{2}, x_j + \frac{\Delta x}{2})$, is introduced. For the sake of simplicity $x_j - \frac{\Delta x}{2} = x_{j-\frac{1}{2}}$ and $x_j + \frac{\Delta x}{2} = x_{j+\frac{1}{2}}$.

The simplest approach is to use an upwinding scheme:

$$\partial_x f_j^n = \frac{1}{\Delta x} (\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}) = \begin{cases} \frac{1}{\Delta x} (f_j^n - f_{j-1}^n), & \text{if } v_i \geq 0 \\ \frac{1}{\Delta x} (f_{j+1}^n - f_j^n), & \text{else.} \end{cases} \quad (4.12)$$

If we substitute the upwinding term into equation (4.11), we have:

$$f^{n+1} = f^n - \tilde{\omega}_1 v \frac{\Delta t}{\Delta x} (\hat{f}_{j+\frac{1}{2}}^{(1)} - \hat{f}_{j-\frac{1}{2}}^{(1)}) + \Delta t \frac{\omega_1}{\tau} (M^{(1)} - f^{(1)}). \quad (4.13)$$

This scheme is conservative by construction, see [44].

Velocity discretization

The discretization of velocity induces some problems. The main difficulty is, that the moments of f are calculated through a quadrature formula. Then it is no longer true,

that f and M have the same moments.

The second problem is due to the difficulty to select a suitable grid. A fine velocity grid yields a better accuracy. However, a finer grid also leads to higher computational costs.

We choose a velocity grid and let v_k , $k \in K$ be the set of grid points. In this work we use a regular grid, where the bounds are dependent on the distribution function f . More details on that are given in Chapter 5.

Given any function $g : \mathbb{R}^N \rightarrow \mathbb{R}$, let:

$$\langle g \rangle_K = Q(g, v_k) \approx \int_{\mathbb{R}^N} g(v) dv. \quad (4.14)$$

That is, $\langle g \rangle_K$ denote the approximation of $\langle g \rangle$ using a suitable quadrature rule $Q(\cdot, v_k)$ built on the nodes v_k .

The macroscopic variables now depend on the quadrature rule and on the grid used in velocity space. Let:

$$\begin{aligned} \rho_K &= \langle f \rangle_K, \\ (\rho u)_K &= \langle f v \rangle_K, \\ E_K &= \langle \frac{1}{2} f v^2 \rangle_K. \end{aligned}$$

Now we can construct an approximate Maxwellian with the formula:

$$M_K(f)(x, v, t) = \frac{\rho_K(x, t)}{2\pi RT_K(x, t)^{\frac{N}{2}}} \exp\left(-\frac{|v - u_K(x, t)|^2}{2RT_K(x, t)}\right). \quad (4.15)$$

The problem is that f and $M_K(f)$ do not have to have the same discrete moments. In general it follows:

$$\langle (f - M_K(f))\phi \rangle_K \neq 0. \quad (4.16)$$

However in [36] it is proven that it is possible to find a discrete Maxwellian \tilde{M} , such that:

$$\begin{aligned} \langle (f - \tilde{M}(f))\phi \rangle_K &= 0, \\ \tilde{M} &= \exp(\alpha(x, t) \cdot \phi(v)), \end{aligned} \quad (4.17)$$

where $\alpha(x, t)$ is an unknown vector which depends on the macroscopic quantities. Now the scheme reads:

$$f_{jk}^{n+1} = f_{jk}^n - \Delta t \partial_x f_{jk}^{(1)} + \Delta t \frac{\omega_1}{\tau} (\tilde{M}(f_j^{(1)}) - f_{jk}^{(1)}), \quad (4.18)$$

$$f_{jk}^{(1)} = f_{jk}^n + \Delta t \frac{a_{11}}{\tau} (\tilde{M}(f_j^{(1)}) - f_{jk}^{(1)}), \quad (4.19)$$

for $f_{jk}^n = f(x_j, v_k, t^n)$. To get the values $f_{jk}^{(1)}$ we have to compute $\tilde{M}(f_j^{(1)})$.

However we use a simpler way by looking at the following equation:

$$\langle f_j^{(1)} \phi \rangle_K = \langle f_j^n \phi \rangle_K + \Delta t \frac{a_{11}}{\tau} \langle (\tilde{M}(f_j^{(1)}) - f_j^{(1)}) \phi \rangle_K. \quad (4.20)$$

The last term in this equation drops out since f and \tilde{M} have the same moments. It follows:

$$\langle f_j^{(1)} \phi \rangle_K = \langle f_j^n \phi \rangle_K. \quad (4.21)$$

Thus, the scheme looks like this:

$$f_{jk}^{(1)} = f_{jk}^n + \Delta t \frac{a_{11}}{\tau} (M_{jk}^{(1)} - f_{jk}^{(1)}), \quad (4.22)$$

where $M_{jk}^{(1)}$ is the Maxwellian computed from the discrete moments. This solution is much faster, since we don't have to calculate \tilde{M} though it is used. In the numerical tests [44], it is shown that the errors in the conserved variables are indeed very small.

The main merit of this method is, that using the technique described above schemes of any order of accuracy in both, space and time can be constructed. Furthermore, since the source term is treated implicitly even problems with great stiffness can be integrated with relatively large time steps.

4.1.2 Semi-Lagrangian scheme

The Semi-Lagrangian method [50, 51, 53] also works with a direct discretization of the BGK equation. The advantage over the previous scheme is the Lagrangian nature, which makes it appropriate for problems with moving boundaries and alleviates the CFL restriction on the time step. Furthermore, the implicit treatment of the source term leads to an Asymptotic Preserving behavior.

Again we describe the method in one dimension in space and in time.

We solve the following initial value problem:

$$\begin{aligned}\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} &= \frac{1}{\tau}(M - f), \\ f(t, x, v) &= f_0(x, v),\end{aligned}\tag{4.23}$$

where $v \in \mathbb{R}$, $x \in \mathbb{R}$ and $t > 0$.

We want to integrate the equation up to a fixed time $t = t_f$. Furthermore, we assume a constant time step $\Delta t = \frac{t_f}{N_t}$ and a uniform grid in physical and velocity space with mesh spacing Δx and Δv . We denote the grid points by $t_n = n\Delta t$, $x_i = i\Delta x$, $i = 1, \dots, N_x$, $v_j = j\Delta v$ and $j = -N_v, \dots, N_v$. We assume that the distribution function is negligible for $|v| > v_{max} = N_v\Delta v$. Let $f_j(t, x)$ be a numerical approximation $f(t, x, v_j)$. The evolution equation for $f_j(t, x)$ along the characteristics between time step n and time step $n + 1$ is obtained by writing (4.23) in characteristic form for $f_j(x, t)$:

$$\begin{aligned}\frac{df_j}{dt} &= \frac{1}{\tau}(M_j - f_j), \\ \frac{dx}{dt} &= v_j, \\ x(t_n) &= \tilde{x}, \quad f_j(t_n) = f_j^n(\tilde{x}), \quad t \in [t_n, t_{n+1}].\end{aligned}\tag{4.24}$$

Here, M_j denotes the local Maxwellian with the same moments as $f(t, x, \cdot)$ evaluated at velocity v_j .

Now, a first order scheme is obtained by discretizing (4.24) in time. Since we are interested in a scheme which is stable for small relaxation time, we discretize equation (4.24) by an implicit Euler scheme:

$$f_{ij}^{n+1} = \tilde{f}_{ij}^n + \frac{\Delta t}{\tau}(M_{ij}^{n+1} - f_{ij}^{n+1}),\tag{4.25}$$

$$x_i = \tilde{x}_{ij} + v_j\Delta t, \quad i = 0, \dots, N_x, \quad j = -N_v, \dots, N_v.\tag{4.26}$$

The values \tilde{f}_{ij}^n are reconstructed at positions \tilde{x}_{ij} by a suitable reconstruction. One can employ linear interpolation by using the values at the grid points, where \tilde{x}_{ij} lies between. For complicated problems also high order reconstruction is possible, for example a piecewise cubic polynomial which is obtained by Hermite interpolation in each interval $[x_i, x_{i+1}]$. The first derivatives of the function at x_i , $\frac{\partial f}{\partial x}$, which are

needed for Hermite interpolation, can be computed by second order central differences. Depending on the problem other reconstructions than the Hermite polynomial are possible. For example one can use WENO reconstructions, see [53].

Remark 4.2. One has to remark, that if $\Delta x = \Delta v \Delta t$, the foot of the characteristic would be a grid point, so that no interpolation is required at the point \tilde{x}_{ij} . In such a case the method would become a Lattice-Boltzmann method (LBM) [38], which is usually used to model fluid dynamic equations.

Next, we have to evaluate the values M_i^{n+1} . By definition, the Maxwellian M_i^{n+1} and the function f_i^{n+1} have the same moments ρ_i^{n+1} , $(\rho u)_i^{n+1}$, E_i^{n+1} . Hence, to calculate the Maxwellian one needs to calculate the moments of f_i^{n+1} . Consider equation (4.26), multiply both sides by the collision invariants ϕ and integrate over v :

$$\int_v \phi_i f_i^{n+1} dv = \int_v \phi_i \tilde{f}_i^n dv + \frac{\Delta t}{\tau} \left(\int_v \phi_i M_i^{n+1} dv - \int_v \phi_i f_i^{n+1} dv \right). \quad (4.27)$$

It is obvious that:

$$\left(\int_v \phi_i M_i^{n+1} dv - \int_v \phi_i f_i^{n+1} dv \right) = 0. \quad (4.28)$$

Hence, it follows that:

$$\int_v \phi_i f_i^{n+1} dv = \int_v \phi_i \tilde{f}_i^n dv. \quad (4.29)$$

This means, one can compute the Maxwellian by using the moments of \tilde{f} . If we assume, that the distribution function is smooth and the energy in velocity outside the domain can be neglected, these moments can be approximated by summation over v_j :

$$\begin{aligned} \rho_i^{n+1} &= \sum_j \gamma_j \tilde{f}_{ij}^n, \\ (\rho u)_i^{n+1} &= \sum_j \gamma_j \tilde{f}_{ij}^n v_j, \\ E_i^{n+1} &= \frac{1}{2} \sum_j \gamma_j \tilde{f}_{ij}^n |v_j|^2, \end{aligned} \quad (4.30)$$

where γ_j is a constant which depends on the quadrature rule. In this work we use a uniform grid in velocity space. Therefore, we usually employ the trapezoidal rule,

where $\gamma_j = \frac{\Delta v}{2}$.

We have to remark that the moments which are computed by summation are not exactly equal to the moments computed by integrating the continuous Gaussian. This may lead to a small inconsistency of the method. To overcome this problem Mieussens introduced a discrete Maxwellian, see previous section and [36]. However, to compute this Maxwellian one has to solve a nonlinear system. Furthermore, the difference between the moments of the discrete Maxwellian and the standard Gaussian is very small due to spectral accuracy of the quadrature formulas, see also [44].

Once the moments are computed the Maxwellian is evaluated by plugging in the values for ρ , u and T . It follows:

$$f_{ij}^{n+1} = \frac{\tau \tilde{f}_{ij}^n + \Delta t M_{ij}^{n+1}}{\tau + \Delta t}. \quad (4.31)$$

One has to remark, that we can also use the higher order IMEX schemes for the time-discretization, see [53], where another conservative version of the method is presented. Note, that for $\tau \rightarrow 0$ the distribution function f is projected onto the Maxwellian. Furthermore, in this limit the scheme becomes a relaxation scheme for the Euler equations. We can say that the scheme is Asymptotic Preserving. This term was, amongst others, introduced by Axel Klar [31], in the context of kinetic equations close to the low Mach number limit.

For numerical methods of differential equations this property is the capability of a method applied to a stiff system to become a consistent discretization of the limit equation and is strongly related to L-stability property of the method. In the context of hyperbolic equations with relaxation, high order Implicit-Explicit schemes that are AP have been derived, see [36].

In 2010 Russo, Santagati and Yun demonstrated, that the Semi-Lagrangian scheme converges in a weighted L^1 norm to the unique smooth solution by deriving an explicit error estimate. For details, see [51].

The main advantage of the Semi-Lagrangian scheme is, that because of its Lagrangian nature it can be easily adapted to problems with moving boundaries and it allows the use of large CFL numbers. Furthermore, the domain can be discretized by a simple

Cartesian Grid and the implicit treatment of the source term allows capturing the fluid dynamic limit.

4.2 Moment methods

The difference of moment methods to direct discretization methods is, that here we don't discretize the BGK equation directly, but we work with the macroscopic quantities. These quantities result by integrating the BGK equation in velocity space. This integration leads to a transformation of the BGK equation into some partial differential equations which only depend on the physical space variables. These equations can be simply solved by known CFD methods.

4.2.1 13 Moment Equations

The 13 Moment Equations approach [56, 57] works with macroscopic quantities like ρ , u and T . In this method first a set of meaningful macroscopic equations is derived, which have the common structure of partial differential equations and do not depend on the velocity space anymore. This set of equations can simply be solved by any known discretization method.

One has to mention, that there are other attempts to derive macroscopic equations out of the Boltzmann- or the BGK model. The Navier-Stokes-Fourier (NSF) equations, which are derived of the Chapman-Enskog expansion [11, 49] fail to describe processes in rarefied gases for Knudsen numbers above 0.05. The Burnett or super-Burnett equations [54], which are a second or third order Chapman-Enskog expansions are known to be unstable in transient processes.

Another well known method for obtaining suitable equations is Grad's method of moments [27], which provides stability at any level. However, Grad's equations fail to describe smooth shock structures for low Mach numbers and the equations are not related a priori to the Knudsen number as smallness parameter.

In fact, the 13 Moment Equations are a regularized version of Grad's equations. These equations are unconditionally stable. Furthermore, the equations were derived for

the Boltzmann equation itself up to third order accuracy. In this work we skip the complicated derivation but we write down the 13 Moment Equations for the BGK model.

Moments

We define the general irreducible moments of the density function f as:

$$m_{u_1 \dots u_n}^a = \int C^{2a} C_{\langle i_1} C_{i_2} \dots C_{i_n \rangle} f dv, \quad (4.32)$$

where $C = v - u$ denotes the peculiar velocity, where v is the velocity of the particle and u is the mean velocity of the medium. Indices in angular brackets denote the symmetric and trace-free part of a tensor.

Some of the moments have a particular interpretation:

$$\begin{aligned} m^0 &= \rho, \\ m_i^0 &= 0, \\ m^1 &= 2\rho e = 3\rho\theta = 3p, \\ m_{ij}^0 &= \sigma_{ij}, \\ m_i^1 &= 2q_i, \end{aligned}$$

where ρ is the density, e is the internal energy, p is the pressure, σ_{ij} the irreducible part of the pressure tensor and q_i the heat flux.

13 Moment Equations in second order of the Knudsen number

First we look at the equations for the conservation laws:

$$\frac{D\rho}{Dt} + \rho \frac{\partial v_k}{\partial x_k} = 0, \quad (4.33)$$

$$\frac{3}{2}\rho \frac{D\theta}{Dt} + \rho\theta \frac{\partial v_k}{\partial x_k} + \frac{\partial q_k}{\partial x_k} + \sigma_{kl} \frac{\partial v_k}{\partial x_l} = 0, \quad (4.34)$$

$$\rho \frac{Dv_i}{Dt} + \rho \frac{\partial \theta}{\partial x_i} + \theta \frac{\partial \rho}{\partial x_i} + \frac{\partial \sigma_{ik}}{\partial x_k} = 0. \quad (4.35)$$

Here, we can see that this system is not a closed set of equations since we need equations for σ_{ij} and q_i :

$$\begin{aligned} \frac{D\sigma_{ij}}{Dt} + \frac{4}{5} \frac{\partial q_{<i}}{\partial x_{j>}} + 2\sigma_{k<i} \frac{\partial v_{j>}}{\partial x_k} + \sigma_{ij} \frac{\partial v_k}{\partial x_k} + 2\rho\theta \frac{\partial v_{<i}}{\partial x_{j>}} &= -\frac{\rho\theta}{\mu} \sigma_{ij}, \\ \frac{Dq_i}{Dt} + \frac{5}{2} \sigma_{ik} \frac{\partial \theta}{\partial x_k} - \sigma_{ik} \theta \frac{\partial \ln \rho}{\partial x_k} + \theta \frac{\partial \sigma_{ik}}{\partial x_k} + \frac{7}{5} q_i \frac{\partial v_k}{\partial x_k} + \frac{7}{5} q_i \frac{\partial v_i}{\partial x_k} \\ &+ \frac{2}{5} q_i \frac{\partial v_k}{\partial x_k} + \frac{5}{2} \rho\theta \frac{\partial \theta}{\partial x_i} = -\frac{5}{2} \frac{\rho\theta}{\kappa} q_i, \end{aligned}$$

where μ is the viscosity and is given by:

$$\frac{1}{2\tau} = \frac{\rho\theta}{2\mu} \quad (4.36)$$

and κ the heat conductivity:

$$\frac{1}{10\tau} = \frac{1}{2} \frac{1}{\kappa} \rho\theta. \quad (4.37)$$

The main advantage of this method is, that depending on the physical space dimension of the BGK model we have at most to solve 13 partial differential equations, which only depend on x and on t . Thus, this method is completely independent on the dimension in velocity space. Furthermore, one has to note, that in this method we directly compute the macroscopic quantities from the above equations.

In this work we use a simple finite differences scheme to solve the equations.

4.2.2 Sectional Quadrature Method of Moments

Originally the Sectional Quadrature Method of Moments (SQMOM) was developed to solve the population balance equation (PBE) [5, 33]. Our aim is, to show how to use this method for solving the BGK equation.

The SQMOM is a method which combines the advantages of the quadrature method of moments and the finite differences methods. These methods have already been applied to kinetic equations [23]. One has to remark, that the PBE is also an integro-differential equation with a different source term than that of the BGK model equation.

Again we consider the initial value problem (4.23). We divide the integral $[v_{min}, v_{max}]$ into $2N_v + 1$ sections with size $\Delta v = \frac{v_{max} - v_{min}}{2N_v + 1}$ and define sectional moments by:

$$m_k^{(i)} = \int_{v_{i-\frac{1}{2}}}^{v_{i+\frac{1}{2}}} v^k f(t, x, v) dv, \quad (4.38)$$

where $v_{i\pm\frac{1}{2}} = v_{min} + (i \pm \frac{1}{2})\Delta v$, $i = -N_v, \dots, N_v$. Then $m_k = \sum_{i=-N_v}^{N_v} m_k^{(i)}$.

We consider one quadrature per section:

$$\begin{aligned} f(t, x, v) &= \omega \delta(v - u), \\ \Rightarrow m_k^{(i)} &= \omega^{(i)} (v_i)^k, k = 0, 1, 2, i = -N_v, \dots, N_v \end{aligned} \quad (4.39)$$

Then, we transform equation (4.23) into sectional moments:

$$\frac{\partial m_k^{(i)}}{\partial t} + \frac{\partial m_{k+1}^{(i)}}{\partial x} = \frac{1}{\tau} (M_k^{(i)} - m_k^{(i)}), \quad (4.40)$$

where: $M_k^{(i)} = \int_{v_{i-\frac{1}{2}}}^{v_{i+\frac{1}{2}}} v^k f_M dv$.

Consider the physical space $[a, b]$ and N_x grids with size $\Delta x = \frac{b-a}{N_x}$. Note that from (4.39) it follows:

$$\begin{aligned} m_{k+1} &= \omega \cdot u^{k+1} = u \cdot \omega \cdot u^k = u \cdot m_k, \\ \Rightarrow m_0 &= \omega = \rho, \\ \Rightarrow m_1 &= \rho \cdot u = m_0 \cdot u, \\ \Rightarrow u &= \frac{m_1}{m_0}. \end{aligned}$$

Thus, the moment equations can be written as:

$$\frac{\partial m_k^{(i)}}{\partial t} + \frac{\partial}{\partial x} (u^{(i)} m_k^{(i)}) = \frac{1}{\tau} (M_k^{(i)} - m_k^{(i)}). \quad (4.41)$$

Note, that the one point quadrature gives the moment closure relation.

Now we have to discretize equations (4.41) in time and in space. Since we need a scheme which is also conservative for small relaxation time, again we have to use a

Semi-Implicit method. This means that we discretize the source term implicitly and the convection term with an explicit upwinding scheme.

The scheme is the following:

$$(m_k^{(i)})_j^{n+1} = \frac{\tau(m_k^{(i)})_j^n - \tau \frac{\Delta t}{\Delta x} (F_k^{(i)})_j^n u_k^{(i)n} + \Delta t (M_k^{(i)})_j^{n+1}}{\tau + \Delta t}, \quad (4.42)$$

where:

$$(F_k^{(i)})_j^n = \begin{cases} (m_k^{(i)})_j^n - (m_k^{(i)})_{j-1}^n, & \text{if } u^{(i)} \geq 0 \\ (m_k^{(i)})_{j+1}^n - (m_k^{(i)})_j^n, & \text{else} \end{cases}.$$

One has to note, that in $1D$ velocity space we need 3 moments per section to calculate the macroscopic quantities ρ , u and T , in $2D$ one needs 5 moments and in $3D$ there are 7 moments needed. Thus, in $3D$ velocity space we have $7(2N_v + 1)$ PDE's to solve, whereas in the direct discretization methods this number reduces to $2N_v + 1$.

4.3 Comparison of the methods

In the following section we compare the numerical accuracy and efficiency of the methods described above. Therefore, we apply the methods to Sod's Shock tube problem in one dimension in space and velocity.

We solve the BGK equation for $x \in [0, 1]$, $v \in [-10, 10]$ and $t \in [0, 0.2]$ with the following initial conditions:

$$\begin{aligned} T_0(x) &= 1.0, \\ \rho_0(x) &= \begin{cases} 1.0, & \text{if } x \leq 0.5 \\ 0.125, & \text{else,} \end{cases} \\ v_0(x) &= 0.0, \\ f_0(x, v) &= \frac{\rho_0}{(2\pi RT_0)^{\frac{1}{2}}} \exp\left(-\frac{|v - u_0|^2}{2RT_0}\right) \end{aligned}$$

In Figure 4.1 one can see, that for $\tau \rightarrow 0$ the solutions of the Implicit-Explicit and the Semi-Lagrangian scheme, as well as those of the SQMOM and the 13 Moment Equations are converging to the exact solution of the Euler equations. That is, as one expects in the fluid dynamic limit.

Next, Figure 4.2 represents, that for increasing values of τ the solutions of all methods become more smooth. This behavior is expected as the collision frequency is becoming smaller.

Furthermore, in Figure 4.3 we show, that the solution of SQMOM is less exact than that of the two direct discretization methods, for the same numerical effort. Therefore, we plot the result of the Implicit-Explicit and the Semi-Lagrangian scheme for $\Delta v = 0.5$ and that of the SQMOM for $\Delta v = 1.5$. We do this for fair comparison, since for SQMOM we have three times more equations to solve as for the other methods for the same Δv .

The next phenomena, we investigate is, that for increasing values of τ the solution develops small discontinuities for the direct discretization schemes, see Figure 4.4. This is due to the fact, that for large values of τ , there is a lack of relaxation, which makes the particles travel undistributed along the characteristics. With increasing τ , we can decrease Δv to overcome this problem, see Figure 4.5.

As expected, all three methods, namely the SQMOM and the both, Implicit-Explicit and Semi-Lagrangian schemes become more exact, the smaller Δv is chosen, see Figure 4.6.

Furthermore, as one can see in Figure 4.7, the result of the 13 Moment Equations seems to be better than that of the other methods, for bigger values of τ . However, this approach also seems to develop discontinuities in the regions of molecular flows. We know, that in the direct discretization schemes one can overcome this problem by increasing the number of grids in velocity space, see Figure 4.5. In the case of 13 Moment Equations this seems not to be as easy.

In conclusion we can say, that the Implicit-Explicit and the Semi-Lagrangian schemes offer similar results. This is not surprising, since both methods are related to each other through similar discretization methods. The major difference is that the second scheme works with a Lagrangian approach, whereas the Implicit-Explicit scheme uses a time-splitting approach. However, the discretization in spatial and velocity space is similar.

The SQMOM seems to be the most time consuming method, since in every velocity node we have to solve at least three equations, depending on the dimension of the problem. The 13 moment approach is the fastest one, since here we have only 13 equations to solve in two dimensional physical and velocity space. However, the implementation of boundary conditions is not straight forward, see [65], as for the direct discretization methods, where we can use kinetic boundary conditions, see Chapter 2. From the four methods introduced, we choose the Semi-Lagrangian approach for implementation in the FPM framework. As we have seen it performs well with respect to accuracy and computational time. Furthermore, it is straight forward in its implementation. The final reason is, that because of the Semi-Lagrangian behavior this method is simply applicable to problems with moving boundaries, which is the main topic of this work. In the next chapters we will see that the implementation of boundary conditions is even easier in FPM, since we use an irregular cloud of points.

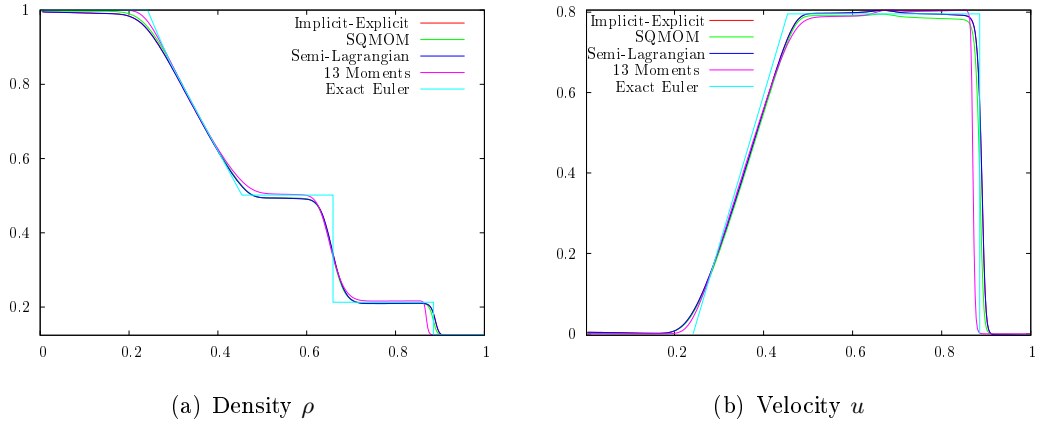


Fig. 4.1: All Methods for $\tau = 10^{-5}$, $v \in [-10, 10]^3$. Convergence towards the exact Euler solution can be observed.

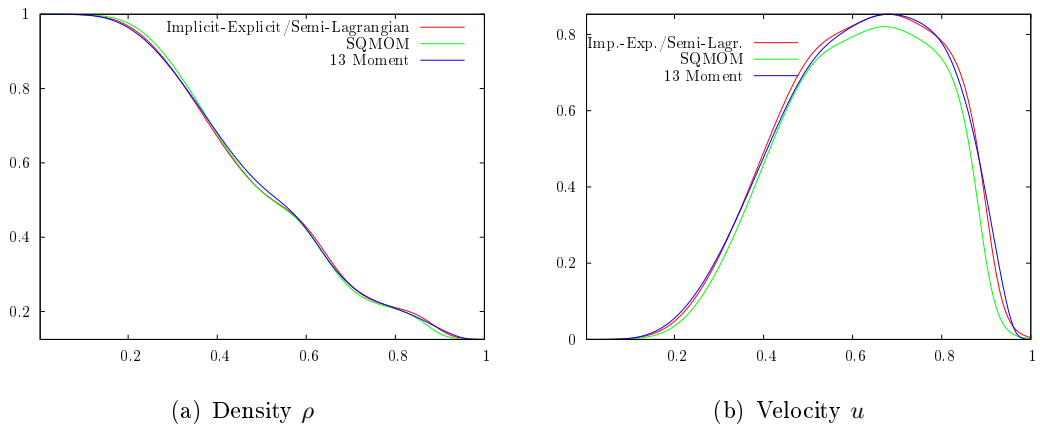


Fig. 4.2: Plots for $\tau = 10^{-2}$. All Methods are calculated with $\Delta v = 0.5$. Compared to Figure 4.1, higher values of τ lead to a smoothing of the solution.

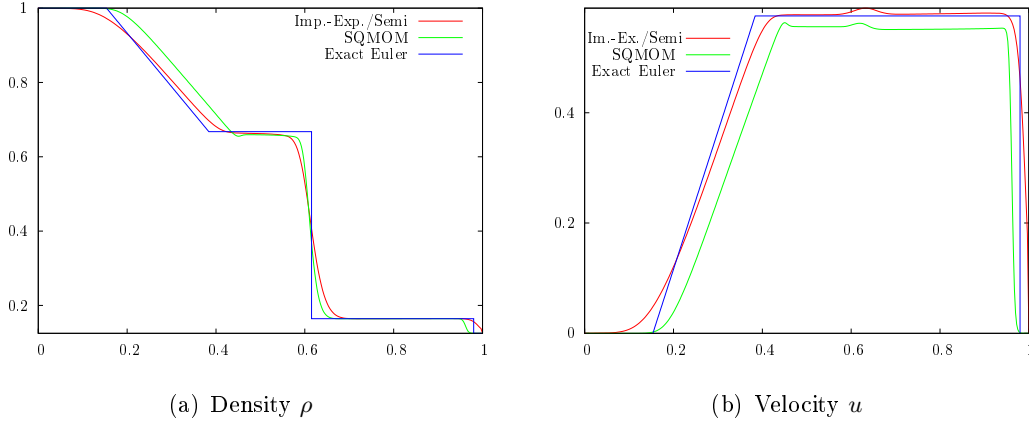


Fig. 4.3: Plots for $\tau = 10^{-4}$, $v \in [-10, 10]$. The SQMOM is calculated for $\Delta v = 1.5$, all other methods for $\Delta v = 0.5$. One can see, that the SQMOM performs worse than the other methods for the same numerical effort.

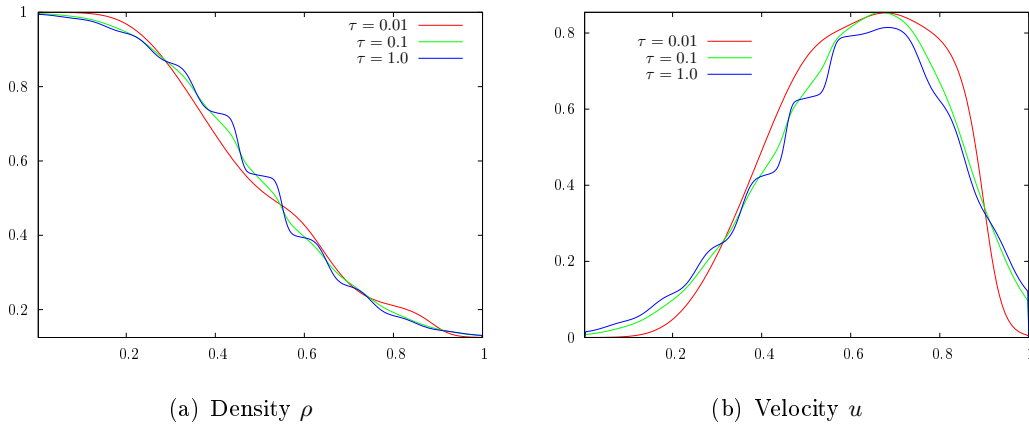


Fig. 4.4: Solution of the Semi-Lagrangian scheme for $\Delta v = 0.5$ (the SQMOM and Implicit-Explicit methods behave similarly). Discontinuities are developed for larger values of τ .

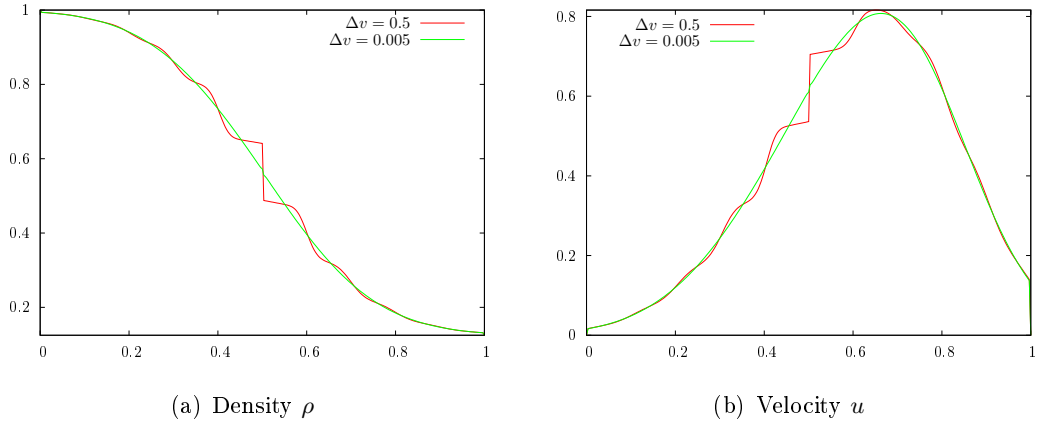


Fig. 4.5: The result of Implicit-Explicit/Semi-Lagrangian methods for $\tau = 10^0$. Discontinuities due to large τ can be avoided by choosing smaller values for Δv .

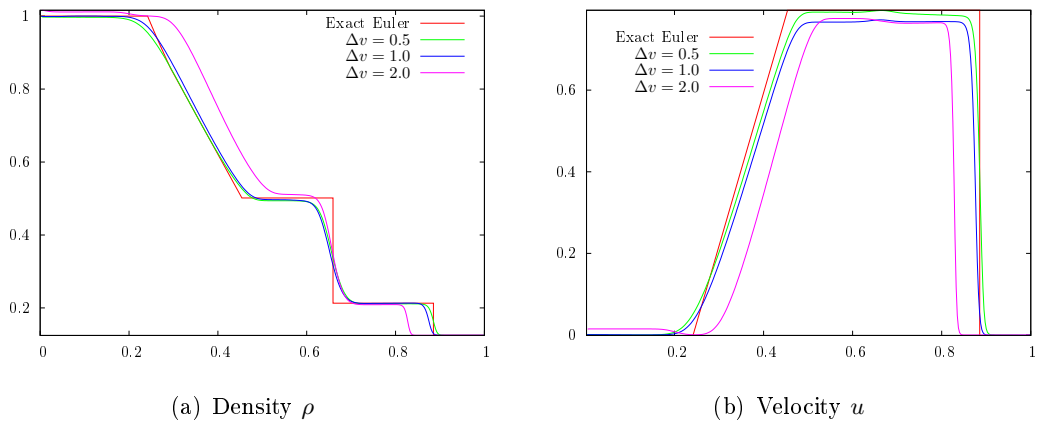
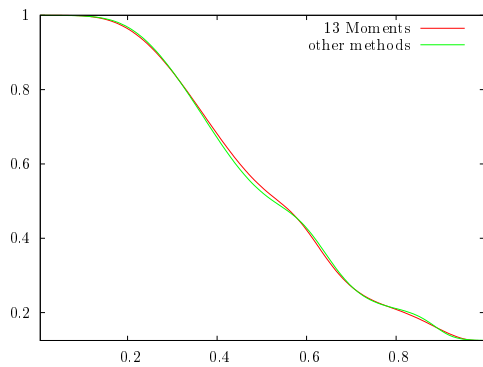
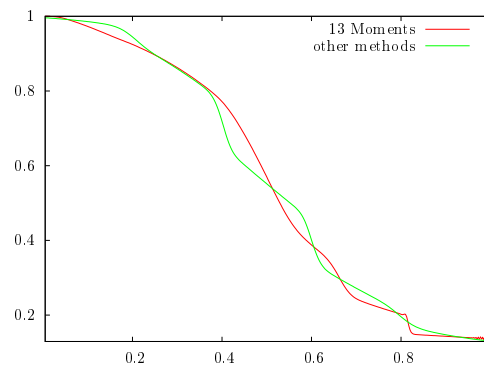


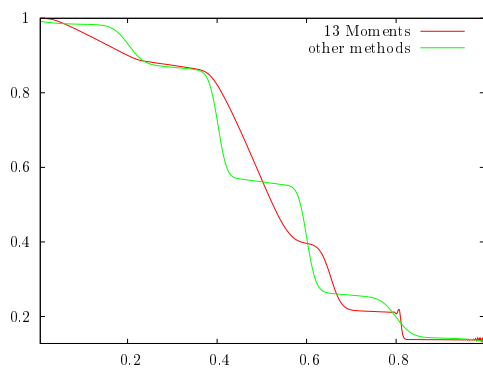
Fig. 4.6: Solution for $v \in [-10, 10]^3$, $\tau = 10^{-5}$, $\Delta v = 0.5$, $\Delta v = 1.0$, $\Delta v = 2.0$ for the Semi-Lagrangian Scheme. The solution becomes more exact, the smaller Δv is chosen.



(a) Density ρ for $\tau = 10^{-3}$ and $\Delta v = 0.001$



(b) Density ρ for $\tau = 10^{-1}$ and $\Delta v = 0.01$



(c) Density ρ for $\tau = 10^0$ and $\Delta v = 0.1$

Fig. 4.7: *The result of 13 Moment Equations and the other methods for different τ . The solution of 13 Moment Equations develops discontinuities for large values of τ , which can not be avoided by decreasing Δv .*

Chapter 5

Semi-Lagrangian scheme in the FPM framework

As already mentioned in the previous chapter, we choose the Semi-Lagrangian idea (see Section 4.1.2) to use it in the FPM framework. In the following chapter we present a modified Semi-Lagrangian scheme, which is suitable for being used in a mesh-free context. We also show some numerical examples in one dimensional and two dimensional spatial and velocity space.

Remark 5.1. In this work, in contrast to other FPM applications (see, e.g. [63]), we restrict ourselves to the case, where the FPM particles do not move with time. That means our points are irregularly distributed, but they do not change their positions during computational time. We will leave the moving mesh framework to further investigations.

5.1 Numerical scheme in one dimensional spacial and velocity space

We recall the description of the Semi-Lagrangian scheme from Section 4.1.2 and modify it for use in an irregular point setting.

Suppose we want to solve the following initial value problem:

$$\begin{aligned}\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} &= \frac{1}{\tau}(M - f), \\ f(t = 0, x, v) &= f_0(x, v),\end{aligned}\tag{5.1}$$

where:

$$M = \frac{\rho}{\sqrt{2\pi RT}} \exp\left(-\frac{|v - u|^2}{2RT}\right),\tag{5.2}$$

Furthermore, $t > 0$, $v \in [v_{min}, v_{max}] \subset \mathbb{R}$ and $x \in [0, x_{max}] \subset \mathbb{R}$.

Remark 5.2. The interval $[v_{min}, v_{max}]$ depends on the initial value f_0 of the distribution function. It is chosen in such a way, that outside this interval the values of f_0 are negligible, see also Figure 5.1.

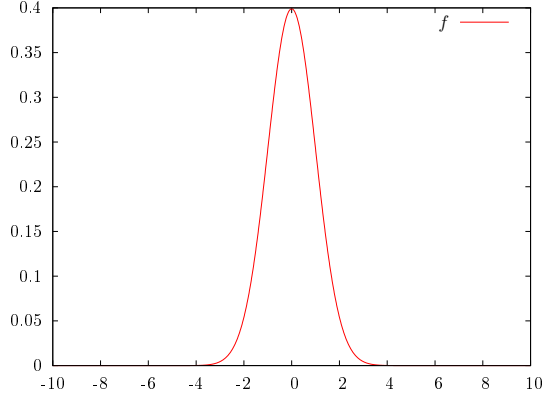


Fig. 5.1: Distribution function f_0 in velocity space. For $v < -10$ and $v > 10$ the values are negligible.

Let $f_l(t, x)$ be the discretization of $f(t, x, v)$ in velocity space. The discretization is regular, where $l = -N_v, \dots, N_v$ and $N_v = \frac{v_{max} - v_{min}}{2\Delta v}$.

We can get the evolution equation for $f_l(t, x)$ along the characteristic between time step n and time step $n + 1$ by writing equation (5.1) in characteristic form, see also [50]:

$$\begin{aligned}
 \partial_t f_l &= \frac{1}{\tau} (M_l - f_l), \\
 \frac{\partial x}{\partial t} &= v_l, \\
 x(t_n) &= \tilde{x}, \\
 f_l(t_n) &= f_l^n(\tilde{x}) = \tilde{f}_l.
 \end{aligned} \tag{5.3}$$

For the discretization of (5.3) we use the implicit Euler scheme:

$$\begin{aligned} f_{jl}^{n+1} &= \tilde{f}_{jl}^n + \frac{\Delta t}{\tau} (M_{jl}^{n+1} - f_{jl}^{n+1}), \\ \tilde{x}_{jl} &= x_j - v_l \Delta t, \quad j = 1, \dots, N_x. \end{aligned} \quad (5.4)$$

where N_x is the number of our arbitrarily arranged particles. An illustration of the scheme is shown in Figure 5.2.

From equation (5.4) it is seen that we need the values \tilde{f}_{jl}^n and M_{jl}^{n+1} to calculate f_{jl}^{n+1} .

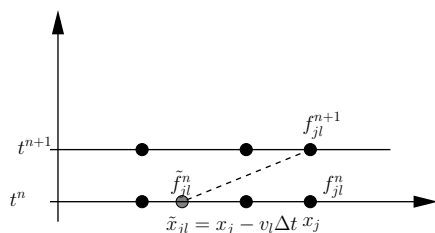


Fig. 5.2: Space-time grid for the computation of f_{jl}^{n+1} .

5.1.1 Reconstruction of the distribution function

To reconstruct \tilde{f}_{jl} we use second order interpolation in connection with the Weighted Least Squares method, see Section 3.4.

For ease of notation, let us skip the dependence of f on v and t , that is $f(t, x, v) = f(x)$. Suppose we want to interpolate $f(\tilde{x})$. For interpolation we use the values $f(x_i) = f_i$ with $x_i \in P(\tilde{x}, h) = \{x_i : \|\tilde{x} - x_i\| \leq h\}$ for $i = 1, \dots, m$, where \tilde{x} is the particle with the minimum distance to \tilde{x} (see also Figure 5.3):

$$\tilde{x} = \min_j \|\tilde{x} - x_j\|, \quad j = 1, \dots, N_x \quad (5.5)$$

In the next step, as was described in Section 3.4, we use the second order Taylor series expansion of $f(x_i)$ around \tilde{x} :

$$f(x_i) = f(\tilde{x}) + d\tilde{x}_i f_x(\tilde{x}) + \frac{1}{2} d\tilde{x}_i^2 f_{xx}(\tilde{x}) + e_i, \quad (5.6)$$

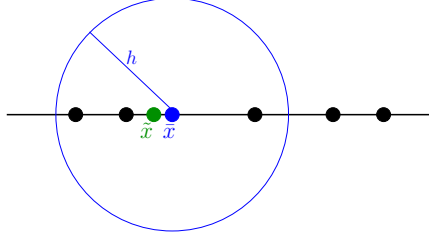


Fig. 5.3: For interpolation of \tilde{x} we use the neighboring points of \tilde{x} .

where $d\tilde{x}_i = x_i - \tilde{x}$.

Then we minimize the error e_i by using the Weighted Least Squares method. It means we have the following system of normal equations to solve:

$$(M^T W M)a = (M^T W)b, \quad (5.7)$$

where the weighting matrix W consists of values from Gaussian weighting functions (see equation (3.2)) and:

$$M = \begin{pmatrix} 1 & (x_1 - \tilde{x}) & (x_1 - \tilde{x})^2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 1 & (x_m - \tilde{x}) & (x_m - \tilde{x})^2 \end{pmatrix}.$$

The solution to our problem reads:

$$a = (M^T W M)^{-1} (M^T W)b, \quad (5.8)$$

where $a = (f(\tilde{x}), f_x(\tilde{x}), f_{xx}(\tilde{x}))^T$ and $b = (f(\tilde{x}_1), \dots, f(\tilde{x}_m))^T$. We are only interested in the first value of vector a . Thus, $f(\tilde{x}) = ((M^T W M)^{-1})_1 (M^T W)b$, where the index 1 denotes the first row of the matrix $(M^T W M)^{-1}$.

Remark 5.3. Of course, higher order reconstruction of \tilde{f} is possible at this point. One simply has to extend the method described above to higher order, which also means choosing an appropriate value for h to have enough neighboring particles for an accurate interpolation.

Remark 5.4. Although not occurring in this work, in the context of interpolation, the possibility of oscillations near shocks is present. One natural idea in these situations would be to use non-oscillatory reconstructions near discontinuities, for example types of Essentially Non-Oscillatory (ENO) or Weighted Essentially Non-Oscillatory (WENO) schemes [44, 55]. The idea of these schemes is to use a convex combination of low-order polynomials to construct high-order reconstructions on a given set of mesh points. Smoothness indicators and nonlinear weights are used to decrease the order of the polynomials near non-smooth regions. For details on the application of WENO type interpolation in connection with the WLSQ method in meshless frameworks, see for example [70, 72].

5.1.2 Evaluation of the moments

As in Section 4.1.2, the value M^{n+1} is calculated from the moments of \tilde{f}^n , using suitable quadrature formulas:

$$(\rho, \rho u, E) = \int_{v_{min}}^{v_{max}} v^\mu \tilde{f}^n dv = \sum_{l=-N_v}^{N_v-1} \gamma_l v_l^\mu \tilde{f}_l, \quad \mu = 0, 1, 2 \quad (5.9)$$

where γ_l are dependent on the quadrature rule we choose. In the following applications we use the trapezoidal rule, which reads:

$$\int_{v_{min}}^{v_{max}} v^\mu \tilde{f}^n = \frac{\Delta v}{2} \sum_{l=-N_v}^{N_v-1} (v_{l+1} + v_l)^\mu (\tilde{f}_{l+1} + \tilde{f}_l), \quad \mu = 0, 1, 2. \quad (5.10)$$

Remark 5.5. As already mentioned in Chapter 4, the calculation of the moments by discrete summation may lead to small inconsistency of the method. However, if the solution is regular this error is negligible, see [44].

After we have the values for \tilde{f}^n and M^{n+1} we can calculate the values for f^{n+1} using equation 5.4 in a slightly rewritten form:

$$f_{lj}^{n+1} = \frac{\tau \tilde{f}_{lj}^n + \Delta t M_{lj}^{n+1}}{\tau + \Delta t}. \quad (5.11)$$

5.1.3 Numerical example: Shock tube problem

As one dimensional numerical example we consider the Shock tube problem. That is, we have a $1D$ tube which is divided into two equal regions. Each region is filled with the same gas, but with different density or pressure, respectively. Thus, initially a discontinuity is created between the two regions. With time, the density or pressure of the gas disperses and a shock wave develops. In our case we look at $x \in [0, 1]$ and $v \in [v_{min}, v_{max}] = [-10, 10]$ with the following initial conditions:

$$\begin{aligned} \rho_0(x) &= \begin{cases} 1.0, & \text{if } x \leq 0.5, \\ 0.125, & \text{else,} \end{cases} \\ u_0(x) &= 0.0, \quad T_0(x) = 1.0, \\ f_0(x, v) &= \frac{\rho_0}{(2\pi RT_0)^{\frac{1}{2}}} \exp\left(-\frac{|v - u_0|^2}{2RT_0}\right), \end{aligned} \tag{5.12}$$

where $R = 1$. We choose $v_{max} = -v_{min} = 10$ since outside this region the values of the distribution function are negligible (see Figure 5.1). Furthermore, we choose $\Delta t = 10^{-3}$, $\Delta v = 0.25$ for the near continuum case and $\Delta v = 0.1$ for the molecular case. The interaction radius h is chosen to be 0.05 in both cases.

Results

Figures 5.4 and 5.5 show the velocity and density profiles at time $t = 0.2$. The results are compared to the exact Euler solution for $\tau = 10^{-4}$ and to the DSMC solution for $\tau = 10^{-1}$. It is seen that the results gained through FPM are close to the reference solutions in both cases. Furthermore, in Figure 5.5 we can see that the FPM result avoids the fluctuations inherent to the DSMC method.

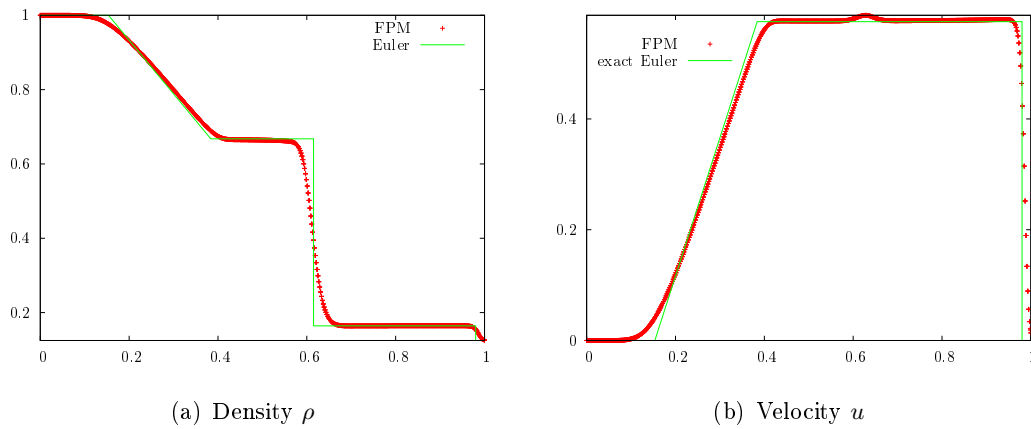


Fig. 5.4: Density and velocity plot for $\tau = 10^{-4}$, compared to exact Euler solution.

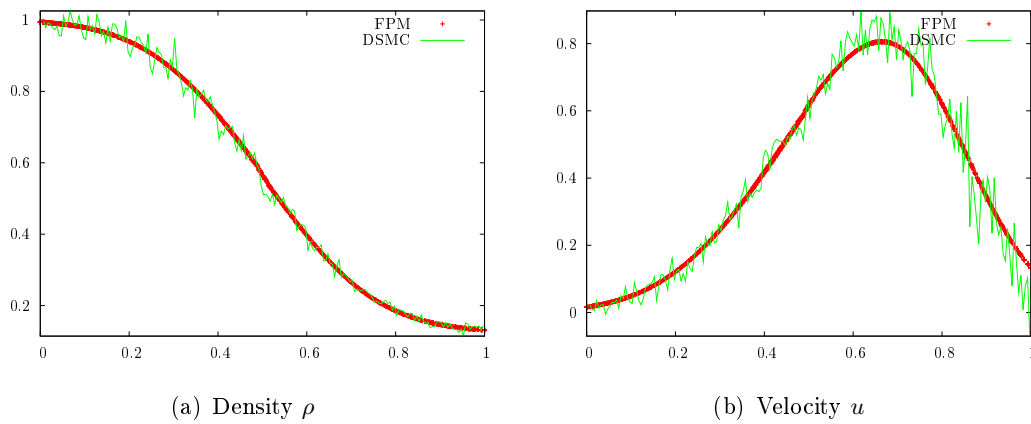


Fig. 5.5: Density and velocity plot for $\tau = 10^{-1}$, compared to DSMC result.

5.2 The scheme in two dimensional spatial and velocity space

The next step is to extend the scheme described in the previous section to a two dimensional space in position and velocity.

Let $z = (x, y) \in [0, x_{max}] \times [0, y_{max}] \subset \mathbb{R}^2$ be the irregularly arranged positions and $v = (v_x, v_y) \in [v_{x_{min}}, v_{x_{max}}] \times [v_{y_{min}}, v_{y_{max}}] \subset \mathbb{R}^2$ the velocities. Then the initial value

problem reads:

$$\begin{aligned}\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} + v_y \frac{\partial f}{\partial y} &= \frac{1}{\tau}(M - f), \\ f(0, z, v) &= f_0(z, v).\end{aligned}\tag{5.13}$$

where:

$$M = \frac{\rho}{(2\pi RT)} \exp\left(-\frac{\|v-u\|^2}{2RT}\right),\tag{5.14}$$

where $u = (u_x, u_y)^T$ denotes the mean velocity of the gas.

Let $f_{il}(t, z)$ be a numerical approximation of $f(t, z, v_{x_i}, v_{y_l})$ in velocity space. The discretization is regular with $i = -N_{v_x}, \dots, N_{v_x}$ and $l = -N_{v_y}, \dots, N_{v_y}$, where $N_{v_x} = \frac{v_{x_{max}} - v_{x_{min}}}{2\Delta v_x}$ and $N_{v_y} = \frac{v_{y_{max}} - v_{y_{min}}}{2\Delta v_y}$. Then we can get the evolution equation for $f_{il}(t, z)$ along the characteristic between time step n and time step $n+1$ by writing equation (5.13) in characteristic form:

$$\begin{aligned}\partial_t f_{il} &= \frac{1}{\tau}(M_{il} - f_{il}), \\ \frac{\partial x}{\partial t} &= v_x, \\ \frac{\partial y}{\partial t} &= v_y, \\ z(t_n) &= \tilde{z}, \\ f_{il}(t_n) &= f_{il}^n(\tilde{z}) = \tilde{f}_{il}.\end{aligned}\tag{5.15}$$

For the discretization of (5.15) we use the implicit Euler scheme. Let $z_j = (x_j, y_j)$:

$$\begin{aligned}f_{jil}^{n+1} &= \tilde{f}_{jil}^n + \frac{\Delta t}{\tau}(M_{jil}^{n+1} - f_{jil}^{n+1}), \\ \tilde{x}_{ji} &= x_j - v_{x_i} \Delta t, \\ \tilde{y}_{jl} &= y_j - v_{y_l} \Delta t,\end{aligned}\tag{5.16}$$

where $j = 1, \dots, N_{xy}$ denotes the discretization in spatial space.

5.2.1 Reconstruction of the distribution function

The values \tilde{f}_{jil} are interpolated as follows:

For the sake of simplicity in the following we skip the indices j, i, l . Suppose we want

to interpolate the value of f at \tilde{z} . As in the one dimensional case, for that we use the values on the neighboring particles of \tilde{z} , that is, the values $f(\tilde{z}_k)$, where $\tilde{z}_k \in P(\tilde{z}, h)$ and \tilde{z} is the point, which has the smallest distance to \tilde{z} :

$$\tilde{z} = \min_j \|\tilde{z} - z_j\|, \quad j = 1, \dots, N_{xy}. \quad (5.17)$$

Finding the nearest neighbor to \tilde{z}

To find the next point to \tilde{z} we use a similar procedure as for establishing the lists of neighboring particles for z_j , see Section 3.2. That means we establish a voxel data structure. The voxels form a regular grid of squares in $2D$ and have side length $l = \frac{1}{\sqrt{2}} \frac{h}{\xi}$, where ξ is a positive constant, which depends on the users choice.

Let us denote the squares by B_{kl} , where $k = 1, \dots, N_{B_x}$, $l = 1, \dots, N_{B_y}$, $N_{B_x} = \lfloor \frac{L_x}{l} + 0.5 \rfloor$, $N_{B_y} = \lfloor \frac{L_y}{l} + 0.5 \rfloor$ where L_x and L_y are the side lengths of the geometry. Now we loop over all points and establish a matrix $G = (g_{kl})_{kl}$, where:

$$g_{kl} = \begin{cases} j, & \text{for some } z_j \in B_{kl} \\ -1, & \text{if } \nexists z_j \in B_{kl} \end{cases} \quad \forall k = 1, \dots, N_{B_x} \quad l = 1, \dots, N_{B_y}. \quad (5.18)$$

Since an entry of $g_{kl} = -1$ is not useful to us, we recursively fill up G . That is, if a box $B_{\hat{k}\hat{l}}$ is empty, we iteratively loop over the closest neighboring boxes. If at least one of them is associated with a particle $z_{\hat{j}}$ through G , we pick this particle to define the entry of G . That is $g_{\hat{k}\hat{l}} = \hat{j}$.

For our test, with $\xi = 3$, a particle was always found within one of the eight neighboring boxes, if none was present in the box itself.

After we have established matrix G , we easily can get access to the particle z , which is located in the same box B_{kl} as \tilde{z} or close to it if no such particle exists.

Next we have two cases. Either the particle z is already near enough to \tilde{z} , so we can take the neighboring particles of z for interpolation or we have to look for $\tilde{z} \in P(z, h)$ which is the nearest particle to \tilde{z} . We do this in the following way:

$$\tilde{z} = \begin{cases} z, & \text{if } \|\tilde{z} - z\| < \beta h \\ \min_k \|\tilde{z} - z_k\| \quad \forall z_k \in P(z_j, h), & \text{else} \end{cases}, \quad (5.19)$$

where βh is the smallest distance between two points.

Next we interpolate the value \tilde{f} by using the values on the neighboring particles of \tilde{z} , which we denote by \tilde{z}_k , $k = 1, \dots, m$. For the illustration, see also Figure 5.6. For

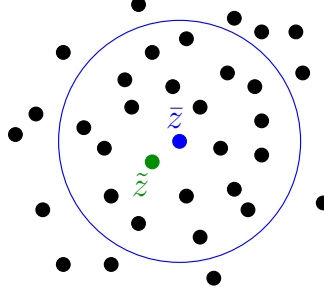


Fig. 5.6: For interpolation of \tilde{z} we use the points inside the blue circle, i.e. the neighboring particles of \tilde{z} .

the interpolation we use the Weighted Least Squares method, which was described in Section 3.4. We want to use a second order interpolation, so the matrix M consists of test functions up to second order:

$$M = \begin{pmatrix} 1 & (\bar{x}_1 - \tilde{x}) & (\bar{y}_1 - \tilde{y}) & (\bar{x}_1 - \tilde{x})^2 & (\bar{x}_1 - \tilde{x})(\bar{y}_1 - \tilde{y}) & (\bar{y}_1 - \tilde{y})^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & (\bar{x}_m - \tilde{x}) & (\bar{y}_m - \tilde{y}) & (\bar{x}_m - \tilde{x})^2 & (\bar{x}_m - \tilde{x})(\bar{y}_m - \tilde{y}) & (\bar{y}_m - \tilde{y})^2 \end{pmatrix}.$$

As weighting matrix W , we use a matrix, which consists of values from Gaussian weighting functions.

We have the following solution to our problem:

$$a = (M^T W M)^{-1} (M^T W) b, \quad (5.20)$$

where $a = (f(\tilde{z}), f_x(\tilde{z}), f_y(\tilde{z}), f_{xx}(\tilde{z}), f_{yy}(\tilde{z}))^T$ and $b = (f(\tilde{z}_1), \dots, f(\tilde{z}_m))^T$. We are only interested in the first value of vector a , since we do not need the derivatives of f at \tilde{z} . Thus, $f(\tilde{z}) = ((M^T W M)^{-1})_1 (M^T W) b$, where the index 1 denotes the first row of the matrix.

5.2.2 Evaluation of the moments

As in the one dimensional case, the Maxwellian M^{n+1} is calculated from the moments of \tilde{f}^n by summation:

$$(\rho, \rho u, E) = \int_{v_{xmin}}^{v_{xmax}} \int_{v_{ymin}}^{v_{ymax}} v_x^\mu v_y^\nu \tilde{f}^n dv = \sum_{i=-N_{vx}}^{N_{vx}-1} \sum_{l=-N_{vy}}^{N_{vy}-1} \gamma_{il} v_{x_i}^\mu v_{y_l}^\nu \tilde{f}_{il}, \quad \mu, \nu = 0, 1, 2 \quad (5.21)$$

where γ_{il} depends on the quadrature rule.

For the following examples we choose the two dimensional trapezoidal rule:

$$\int_{v_{xmin}}^{v_{xmax}} \int_{v_{ymin}}^{v_{ymax}} v_x^\mu v_y^\nu \tilde{f}^n dv = \frac{\Delta v_x \Delta v_y}{4} \sum_{i=-N_{vx}}^{N_{vx}-1} \sum_{l=-N_{vy}}^{N_{vy}-1} (v_{x_i} + v_{x_{i+1}})^\mu (v_{y_l} + v_{y_{l+1}})^\nu (\tilde{f}_{il} + \tilde{f}_{i+1l} + \tilde{f}_{il+1} + \tilde{f}_{i+1l+1}). \quad (5.22)$$

Note, that T can be calculated using equations 2.25.

5.2.3 Numerical example: Driven Cavity flow

As two dimensional example we pick the Driven Cavity flow.

Let $z = (x, y)^T \in [0, 1]^2$ and $v = (v_x, v_y)^T \in [-2000, 2000]^2$. We consider a flow inside a two dimensional square, where the upper boundary is moving with a velocity u_Γ , see Figure 5.7. Caused by this movement a vortex develops with time.

The initial and boundary conditions are:

$$\begin{aligned}
 u(0, z) &= u_0 = 0.0, \\
 \rho(0, x) &= \rho_0, \\
 T(0, z) &= T_0 = 300.0, \\
 f(0, z, v) &= f_0 = \frac{\rho_0}{(2.0\pi RT_0)} \exp\left(\frac{-||v - u_0||^2}{2.0RT_0}\right), \\
 u_x(0, y) &= u_x(1, y) = u_x(x, 0) = u_y(0, y) = u_y(1, y) = u_y(x, 0) = u_y(x, 1) = 0 \\
 u_x(x, 1) &= u_\Gamma = 10.
 \end{aligned}$$

The value ρ_0 depends on how we choose the Knudsen number Kn and the relaxation

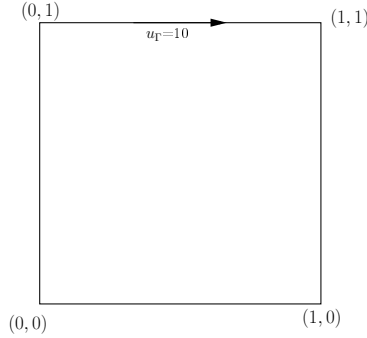


Fig. 5.7: Geometrical set up for the Driven Cavity flow

time τ , respectively. The dependency of ρ to τ and Kn was already stated in Section 2.2, see equations (2.22) and (2.23).

Boundary conditions

As boundary conditions we use **diffusive reflection** conditions at all four boundaries. For explanation, see also Section 2.3 and Figure 2.1.

In the following we restrict ourselves to the description of the implementation of boundary conditions on the upper boundary, that is, $\Gamma = [x, 1]$, $n = (0, -1)^T$. The conditions for the other three boundaries are straightforward.

By $f_{\Gamma il}$, where $i = -N_{v_x}, \dots, N_{v_x}$ and $l = -N_{v_y}, \dots, N_{v_y}$, we denote the density function values, which belong to the upper boundary. These values are computed as follows: For $v_{il} \cdot n < 0$, that is, for $v_{y_l} > 0$ they are calculated from the evolution equation. For velocities $v_{y_l} \leq 0$ we use a Maxwellian with average velocity equal to the velocity at the wall, and with a density such that the net mass flux across the wall is zero.

For $v_{y_l} > 0$ we have:

$$f_{\Gamma il}^{n+1} = \tilde{f}_{\Gamma il}^n + \frac{\Delta t}{\tau} (M_{\Gamma il}^{n+1} - f_{\Gamma il}^{n+1}), \text{ for } v_{y_l} > 0, \quad (5.23)$$

where $\tilde{f}_{\Gamma il} = f(\tilde{z}_{\Gamma il})$ and $\tilde{z}_{\Gamma il} = z_{\Gamma} - v_{il} \Delta t$.

For $v_{y_l} \leq 0$ we use the following Maxwellian:

$$f_{\Gamma il}^{n+1} = M_{\Gamma il}^{n+1} = \frac{\rho_{\Gamma}}{2.0\pi RT_{\Gamma}} \exp\left(-\frac{\|v_{il} - u_{\Gamma}\|^2}{2RT_{\Gamma}}\right) \text{ for } v_{y_l} \leq 0, \quad (5.24)$$

where ρ_{Γ} , T_{Γ} and u_{Γ} denote the density, temperature and the mean velocity at the wall.

The value ρ_{Γ} is computed by imposing that:

$$\sum_{v_{x_i}} \sum_{v_{y_l} \leq 0} M_{\Gamma il} (v_{il} - u_{\Gamma}) \cdot n + \sum_{v_{x_i}} \sum_{v_{y_l} > 0} f_{\Gamma il} (v_{il} - u_{\Gamma}) \cdot n = 0. \quad (5.25)$$

For computing $\tilde{f}_{\Gamma il}^n$ at the points $\tilde{z}_{\Gamma il}$, which belong to characteristics leaving the domain we simply use the values at the points inside the domain, see Figure 5.8. The Maxwellian for these velocities is computed in the following way:

If we multiply by the collision invariants, we obtain:

$$\sum_i \sum_l \phi_{il} M_{\Gamma il}^{n+1} = \sum_i \sum_l \phi_{il} f_{\Gamma il}^{n+1}. \quad (5.26)$$

Observing that:

$$M_{\Gamma il}^{n+1} = f_{\Gamma il}^{n+1}, \quad \forall v_{y_l} \leq 0 \quad (5.27)$$

equality (5.26) is also valid if we only sum over $v_{y_l} > 0$:

$$\sum_{v_{x_i}} \sum_{v_{y_l} > 0} \phi_{il} M_{\Gamma il}^{n+1} = \sum_{v_{x_i}} \sum_{v_{y_l} > 0} \phi_{il} f_{\Gamma il}^{n+1} = \sum_{v_{x_i}} \sum_{v_{y_l} > 0} \phi_{il} \tilde{f}_{\Gamma il}^n. \quad (5.28)$$

Once we have the three parameters to compute the fraction of Maxwellian for $v_{y_l} > 0$, we can compute the values of the distribution function as usual from equation (5.23). After we have obtained the values f_Γ for all v_{il} , we can interpolate the values at \tilde{z} as follows:

If \tilde{z} is located inside the domain, near the upper boundary, we go on with the interpolation as it is described in the previous section using the values on f_{kil} at the points $z_k \in P(\tilde{z}, h)$. Note that z_Γ can also belong to the cloud of neighboring particles of \tilde{z} and so the precomputed values f_Γ can be used for interpolation. If a characteristic is entering the domain, i.e. $\tilde{y} > 1$, we can set $\tilde{f}_{jil} = f_{\Gamma il}$. We can do this, by using the argument that the characteristic belonging to $\tilde{y} > 1$ intersects the upper boundary at time t_m , $t^n < t_m < t^{n+1}$. For details, see Figure 5.8.

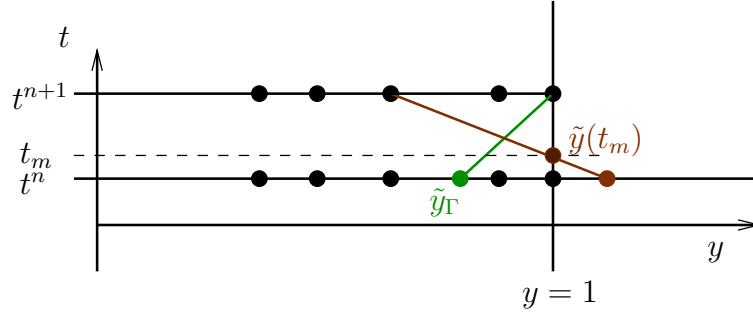


Fig. 5.8: Diffusive boundary conditions at the upper boundary. The values of the function for values belonging to the brown characteristic are set to $f_\Gamma = f(x, 1)$.

Results

The results are computed for $\tau = 10^{-1}$ and $\tau = 10^{-4}$, $N_{v_x} = N_{v_y} = 100$, $h = 0.05$ and $\Delta t = 10^{-4}$. The overall number of particles depends on the value of h and on the order of interpolation in the WLSQ. In this case we use second order interpolation, which means we need at least 6 particles in each ball with radius h . Therefore, the lower bound for the overall number of particles is set to be $N_{xy} \approx \frac{6}{\pi h^2}$.

The solution becomes stationary at $t = 0.05$. Figure 5.9 shows the u_x and u_y velocity

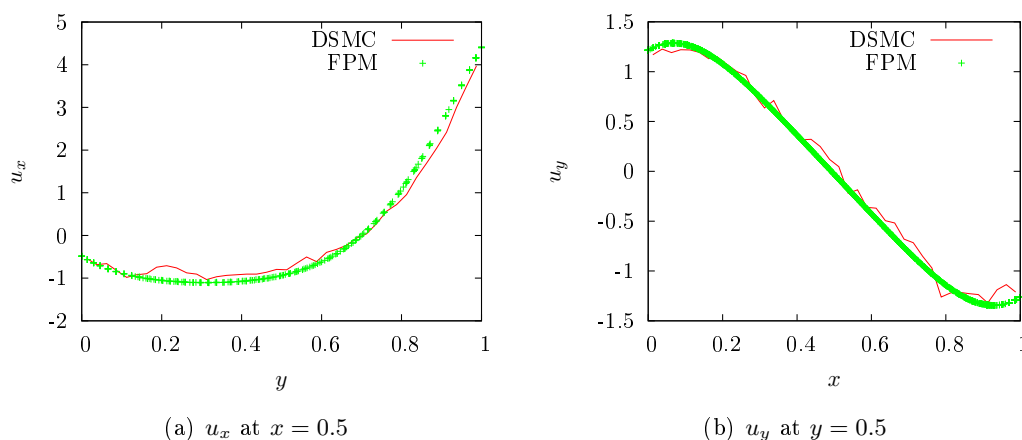


Fig. 5.9: Velocity profiles for $\tau = 10^{-1}$ at $t = 0.05$

profiles along the y and x axis, respectively. The FPM results for $\tau = 10^{-1}$ are compared to the results gained through DSMC method with the following parameters. The number of cells in x as well as in y direction is set to 40, the number of particles per cell is also set to be 40. The time step is chosen to be $\Delta t = \frac{0.025}{2RT_0}$. The simulation is stopped at $t = 0.05$. As can be seen the results are in good agreement. However, as expected FPM yields a smooth solution while DSMC fluctuates.

In Figure 5.10³ one can see the velocity vector fields for high Knudsen number (right) and low Knudsen number (left). As expected, after some time the movement of the upper wall forces the creation of a vortex inside the computational domain. One can see, that for higher Knudsen number the vortex is shifted more to the center of the

³Illustration created with ParaView [28].

square.

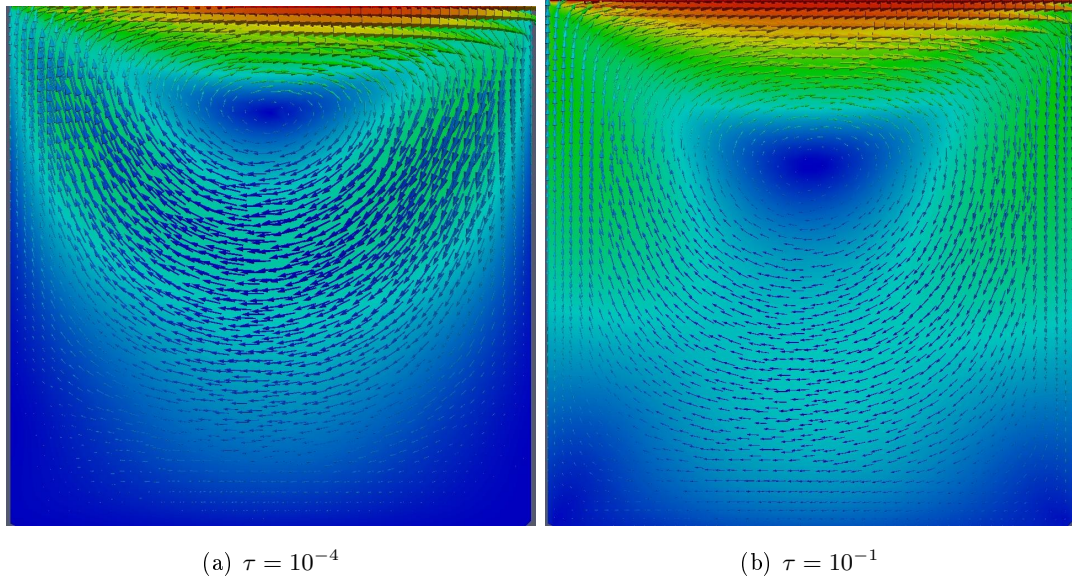


Fig. 5.10: Velocity vector fields for Driven Cavity flow at $t = 0.05$.

Chapter 6

Application to moving boundary problems

In the following sections we focus our attention on moving boundary problems. We show how to handle moving boundaries in one and two dimensional spatial and velocity space, using the approach discussed in Sections 5.1 and 5.2. Furthermore, the advantages of grid free methods such as FPM over the methods, which use a regular Cartesian Grid are pointed out.

6.1 One dimensional piston problem

Motivated by the application to, for example MEMS (see motivation), we consider a simple one dimensional problem in a domain with a moving boundary.

As computational domain we choose $x \in [0, x_p]$ for the spatial space and $v \in [-16, 16]$ for the velocity space. Here, x_p denotes the position of an infinitely high and thin piston, which is initially located at $x = x_p(0)$ and is moving to the left with velocity u_p , see Figure 6.1. The position x_p and the velocity u_p are functions of time, that is: $x_p : t \in \mathbb{R} \rightarrow x_p(t) \in [0, x_p(0)]$. As the piston moves, the computational domain changes, whereas the positions of our FPM points remain the same. That means, by and by the number of particles located inside the domain decreases. The initial

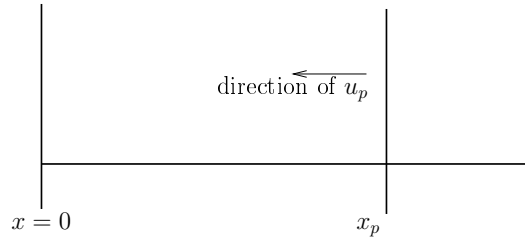


Fig. 6.1: Geometrical set up for the one dimensional moving piston problem. The piston is initially located on the right side of the geometry and is moving with velocity u_p to the left. The wall at $x = 0$ remains fixed.

conditions are:

$$\begin{aligned} u(0, x) &= u_0 = 0.0, \\ \rho(0, x) &= \rho_0 = 1.0, \\ T(0, x) &= T_0 = 3.0, \\ f(0, x, v) &= f_0 = \frac{\rho_0}{\sqrt{2\pi RT_0}} \exp\left(\frac{-|v - u_0|^2}{2RT_0}\right), \end{aligned}$$

where u is the mean velocity of the gas, T the temperature and ρ its density. R is the gas constant and is set to be 1.0 in this example.

6.1.1 Boundary conditions

As already mentioned in Section 2.3, there are three different types of boundary conditions for kinetic equations. The specular and diffusive boundary conditions and a mix of the two of them. In this work we only consider the specular and diffusive case and leave the mixed case for future investigations.

Specular reflection

First we describe specular reflection boundary conditions, recall Section 2.3.

At the left boundary we require:

$$f(t, 0, v) = f(t, 0, -v) \text{ for } v > 0. \quad (6.1)$$

This condition is equivalent to replacing the wall by a gas with specular symmetric distribution for $x < 0$. With this in mind we can easily convert the boundary condition into an initial condition for the so called **ghost points**, see also [50]. We do this using the following procedure:

First we compute the positions for the ghost points by mirroring the points located near the boundary at $x = 0$. Since we have an irregular grid, this new particles can be defined as grid points themselves. After that we prescribe boundary conditions at the ghost points by using the values at the particles lying inside the domain and the specular reflection condition. In the end we can interpolate the values at the positions \tilde{x} located near the boundary by using the values at interior points and at ghost points. For computing the positions of the ghost points that are needed for an accurate interpolation near the boundary, we use the following steps:

First, we introduce the position \tilde{x}^* :

$$\tilde{x}^* = \begin{cases} -\tilde{x}, & \text{if } \tilde{x} < 0 \\ \tilde{x}, & \text{if } 0 < \tilde{x} < \epsilon(h) \end{cases}, \quad (6.2)$$

where $\epsilon(h)$ is a function, which helps to indicate the \tilde{x} which lie inside the domain, but don't have enough particles on the left for an accurate interpolation. For example, $\epsilon(h) = \frac{h}{2}$. As we will see later on, \tilde{x}^* is only needed to find the appropriate points to mirror.

As a next step we search for the particle \bar{x} , which has the smallest distance to \tilde{x}^* and consider the particles $x_k \in P(\bar{x}, h)$, $k = 1, \dots, m$. Then we introduce ghost points by mirroring x_k in the following way, see also Figure 6.2:

$$x_k^* = -x_k, \text{ if } ||x_k^* - \tilde{x}|| \leq h. \quad (6.3)$$

Now we prescribe boundary conditions at the ghost points:

$$f(t, x_k^*, v_l) = \begin{cases} f(t, x_k, -v_l), & \text{for } v_l > 0 \\ f(t, x_k, v_l), & \text{else} \end{cases}. \quad (6.4)$$

In the end we can easily interpolate the values \tilde{f} using the values at x_k and x_k^* , which are located in a ball with radius h around \tilde{x} . At the piston this is less trivial. Here,

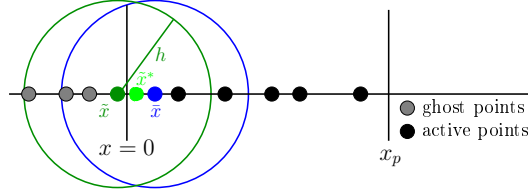


Fig. 6.2: Active and ghost points for the piston problem at the left wall. For the interpolation of f at \tilde{x} the values at the points inside the blue circle are taken.

the boundary condition is given by:

$$f(t, x_p, v) = f(t, x_p, 2.0u_p - v), \quad \forall v \cdot n > u_p. \quad (6.5)$$

Remark 6.1. Incidentally, the condition for the stationary wall follows from setting $u_p = 0$.

We convert the condition into a boundary value problem for the ghost points using the following argument. We approximate the motion of the piston by a piecewise linear function in time, that means, we assume that the velocity of the piston remains unchanged in the interval $[t_n, t_{n+1}]$. Then, the procedure is similar to that at the left wall. The main difference is, that we have to introduce new ghost points in each time step, since the geometry near the piston changes with time.

We interpolate the value $f(\tilde{x})$ for \tilde{x} located near the piston by using the values at ghost points x_k^* :

$$x_k^* = 2.0x_p - x_k, \quad \text{if } |x_k^* - \tilde{x}| \leq h, \quad x_k \in P(\tilde{x}, h), \quad \tilde{x} = \min_x \|x - \tilde{x}^*\| \quad (6.6)$$

where:

$$\tilde{x}^* = \begin{cases} 2.0x_p - \tilde{x}, & \text{if } \tilde{x} > x_p \\ \tilde{x}, & \text{if } x_p - \epsilon(h) < \tilde{x} < x_p \end{cases} \quad (6.7)$$

and $\epsilon(h)$ is defined as in equation (6.2).

The values of the distribution function at the ghost points are the following:

$$f(t, x_k^*, v_l) = \begin{cases} f(t, x_k, 2.0u_p - v_l), & \text{for } v_l < u_p \\ f(t, x_k, v_l) & \text{else} \end{cases}. \quad (6.8)$$

Finally, we can interpolate \tilde{f} using the values at the following set of points:

$$\tilde{P} = \{x_k^* : \|x_k^* - \tilde{x}\| \leq h\} \cup \{x_k \in P(\bar{x}, h) : \|x_k - \tilde{x}\| \leq h\}, \quad (6.9)$$

where x_k^* are the ghost- and x_k the inner points, respectively.

To interpolate \tilde{f} in velocity space at the new velocity $v^* = 2.0u_p - v$ we use linear interpolation with the two neighbors of v^* in the regular velocity grid.

Remark 6.2. The velocity grid does not mandatorily have to be regular. An idea could be to choose a velocity grid, which is more dense in the area where the values of the distribution function are high and to place only some points outside this region. As long as the grid is symmetric around $v = 0$ it would not affect the interpolation procedure in velocity space.

Observation 6.3. The values of the distribution function f at the ghost points require no interpolation in x direction.

An interpolation in physical space is not necessary, since in FPM we have an irregular cloud of points and are able to define ghost points, which are a reflection of inner points, as grid points themselves. In a regular grid the ghost points are not mandatorily reflected on the piston. Thus, an interpolation in physical space is required before prescribing boundary conditions at those points. For more details on that, see Figure 6.3. This illustrates **one important** advantage of FPM over regular grid methods. That is, we only have to interpolate in v direction whereas in a regular grid, for example a finite differences scheme we first would have to use more complicated interpolation in x direction which would lead to a loss of accuracy near the piston and also to larger computational times.

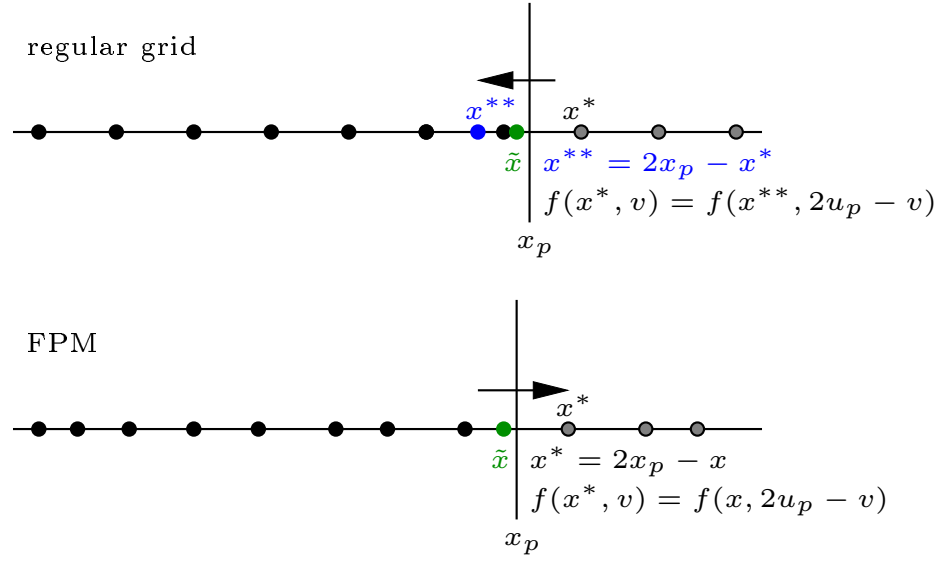


Fig. 6.3: *Specular reflection at the piston. In the regular point setting the ghost points are not reflected on the piston. Thus, one has first to interpolate $f(x^{**})$ to get the value at x^* . In FPM no interpolation in x direction is needed.*

Diffusive reflection

The implementation of diffusive boundary conditions at the left wall works similarly to the 2D case for stationary walls, see Section 5.2.3.

Formally the diffusive boundary conditions at the piston reads:

$$f(t, x, v) = \frac{\rho_p}{2\pi(RT_p)} \int_{v' \cdot n < 0} |v' \cdot n| f(t, x, v') dv' \exp\left(-\frac{\|v - v_p\|^2}{2RT_p}\right), v \cdot n > 0, \quad (6.10)$$

where T_p , ρ_p and v_p are the values for temperature, density and velocity at the piston and n is the normal at the piston pointing inside the domain.

To implement this condition we proceed in the following way:

Let $\tilde{x}_{pl} = x_p - v_l \Delta t$, $l = -N_v, \dots, N_v$ and $f_{pl} = f(t, x_p, v_l)$ be the values which belong to the piston. There are two cases we have to distinguish. Either $v_l \leq u_p$ or $v_l > u_p$. In the first case we set the distribution function f_p^{n+1} to be the local Maxwellian with average velocity equal to the piston velocity and with a density such that the mass flux across the piston is zero. In the second case we compute it from the evolution

equation.

We proceed in several steps:

1. For $v_l > u_p$ we reconstruct the values \tilde{f}_{pl}^n using the values which are located inside the domain, that is, the values at the points on the left side of the piston and the values f_{pl}^n from the previous time step.
2. Now we can compute the Maxwellian:

$$f_{pl}^{n+1} = M_{pl}^{n+1} = \frac{\rho_p}{\sqrt{2.0\pi RT_p}} \exp\left(-\frac{(v_l - u_p)^2}{2RT_p}\right), \quad v_l \leq u_p \quad (6.11)$$

using the values \tilde{f}_{pl}^n and the following arguments for computing ρ_p .

We impose:

$$\sum_{v_l < u_p} M_{pl}^{n+1}(v_l - u_p) + \sum_{v_l > u_p} \tilde{f}_{pl}^n(v_l - u_p) = 0. \quad (6.12)$$

We can use \tilde{f}_{pl}^n in equation (6.12), since we have:

$$f_{pl}^{n+1} = M_{pl}^{n+1}, \quad \forall v_l \leq u_p \quad (6.13)$$

and therefore we obtain:

$$\sum_{v_l > u_p} \phi_l M_{pl}^{n+1} = \sum_{v_l > u_p} \phi_l f_{pl}^{n+1} = \sum_{v_l > u_p} \phi_l \tilde{f}_{pl}^n, \quad (6.14)$$

where ϕ_l are the collision invariants.

3. Now we have all the values we need to compute f_{pl}^{n+1} through the evolution equation:

$$f_{pl}^{n+1} = \tilde{f}_{pl}^n + \frac{\Delta t}{\tau} (M_{pl}^{n+1} - f_{pl}^{n+1}), \quad \forall v_l > u_p. \quad (6.15)$$

4. In the last step we can approximate the values at \tilde{x} located inside the domain near the piston by using the values at interior points and the values f_{pl}^n and f_{pl}^{n+1} . For \tilde{x} which belong to characteristics starting outside the domain we first search for the intersection point $(t_m, \tilde{x}(t_m))^T$ between the characteristic and the line which connects x_p^n and x_p^{n+1} in time space (see Figure 6.4). Then we can easily interpolate the value at $\tilde{x}(t_m)$ using the values at neighboring points and the values at x_p^n and x_p^{n+1} .

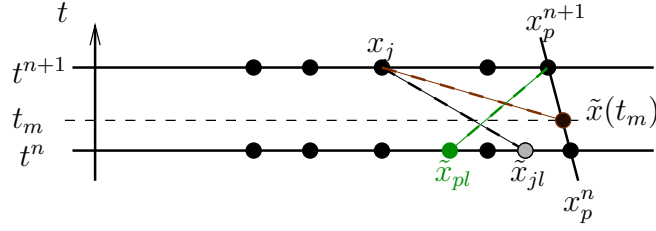


Fig. 6.4: *Diffusive Boundary conditions at the piston. For \tilde{x} belonging to characteristics entering the domain (dashed brown line) we use values at x_p^n and x_p^{n+1} for interpolation.*

Of course, this kind of interpolation is somehow one-sided, since there are several points on the left side of $\tilde{x}(t_m)$ and just one point on the right.

To change that one could again introduce ghost points and use the inverse Lax-Wendroff method, recently proposed by Filbet and Yang [22] to extrapolate the values of the distribution function at those ghost points.

Let x_k^* , $k = 1, \dots, m$ be the ghost points, which are irregularly arranged outside the computational domain and have a maximum distance of βh , $0 < \beta \leq 1$, to the piston, see also Figure 6.5. We have two cases:

1. For $v_l \leq u_p$ we can use the values which are located inside the domain and the values f_{pl}^n and f_{pl}^{n+1} to extrapolate the values at x_k^* .
2. For $v_l > u_p$ it is not that easy, since in this case we can not use the values at particles located inside the domain for extrapolation. Thus, we use the inverse Lax-Wendroff method [22]. At the piston a first order series expansion yields:

$$f(x_k^*, v_l) = f_{pl} + (x_k^* - x_p) \frac{\partial f}{\partial x} + \epsilon(\Delta x^2), \quad \forall v_l > u_p. \quad (6.16)$$

Hence, a second order approximation at ghost points is:

$$f(x_k^*, v_l) = f_{pl} + (x_k^* - x_p) \frac{\partial f}{\partial x}. \quad (6.17)$$

Since we already computed f_{pl} we just need an approximation of the first derivative, which we can get by reformulating equation (5.1):

$$\frac{\partial f_p}{\partial x} = \frac{1}{v} \left(-\frac{\partial f_p}{\partial t} + \frac{1}{\tau} (M - f_p) \right). \quad (6.18)$$

Now, instead of the approximation of the space derivative $\partial_x f$ we need an approximation of the time derivative $\partial_t f$. A first order one is:

$$\frac{\partial f}{\partial t} \approx \frac{(f_{pl}^{n+1} - f_{pl}^n)}{\Delta t}. \quad (6.19)$$

Higher order interpolation can be obtained by additionally using values from previous time steps.

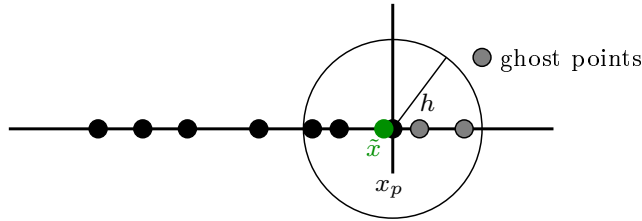


Fig. 6.5: Ghost points for diffusive boundary conditions. Differently to the case of specular reflection, the ghost points do not have to be reflected on the piston.

Remark 6.4. Note, that if $v \rightarrow 0$ or $\tau \rightarrow 0$ at the boundary, the Lax-Wendroff procedure may lead to oscillation near the piston. Therefore, a limitation procedure based on WENO interpolation can be applied. For more details on that, see [22].

A similar problem turns up for the extrapolation at the ghost points, when a shock goes out of the boundary. In this case a WENO type or other kind of non-oscillatory interpolation has to be considered, as well.

6.1.2 Results

In Figure 6.6 we can see results for the moving piston problem with specular reflection boundary conditions at the piston and at the wall. The piston is initially located at $x_p = 18.0$ and moves with velocity $u_p = -0.25 \sin(t)$. It can be seen that a shock wave is created by the motion of the piston, which propagates inside the gas. The density and velocity profiles are illustrated at time $t = 1, 3, 4$. One can see that with smaller τ the contours of the shock wave become sharper. This behavior is expected, since the smaller τ , the more diluted the gas and the result is more “exact”. If one compares

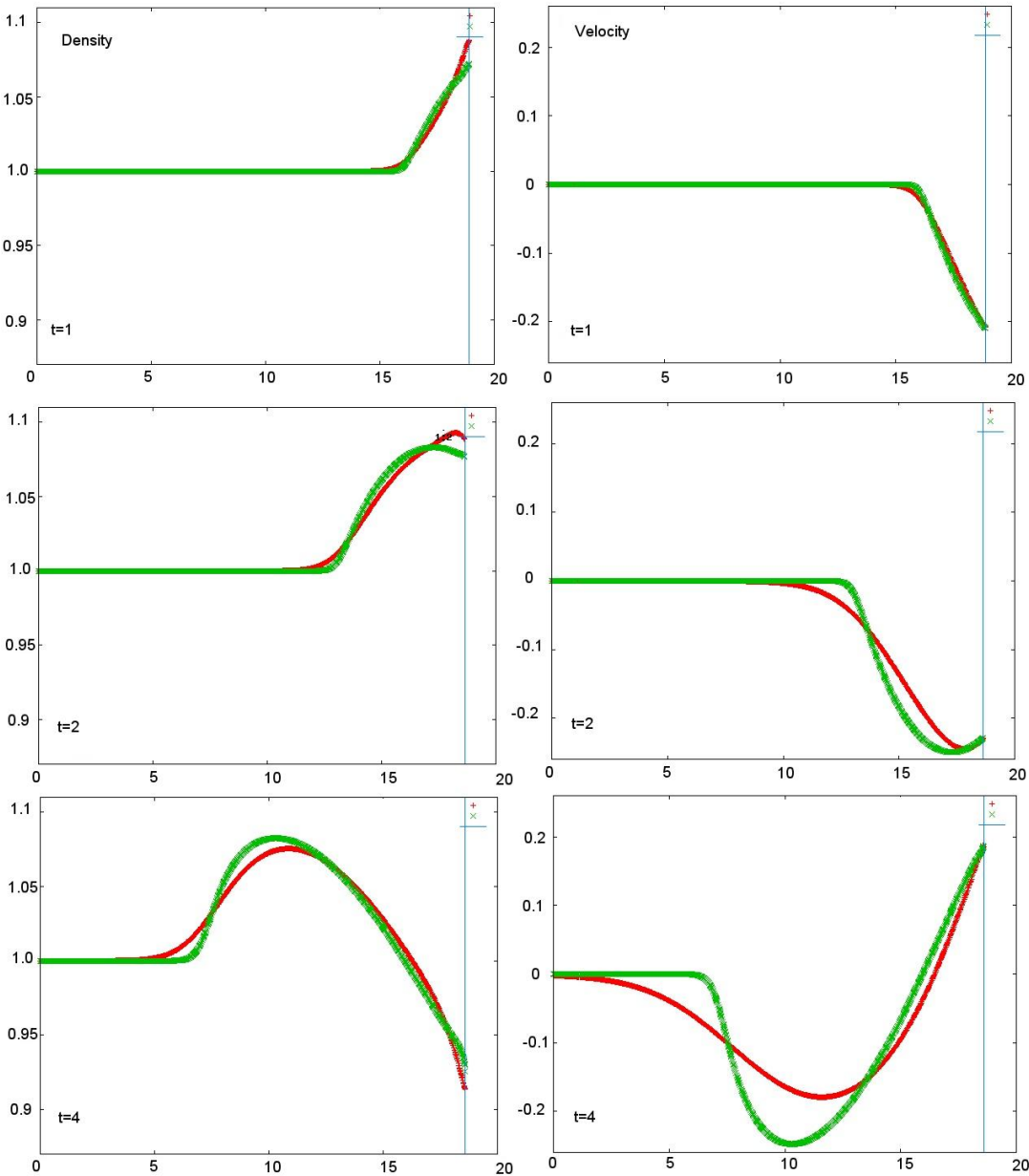
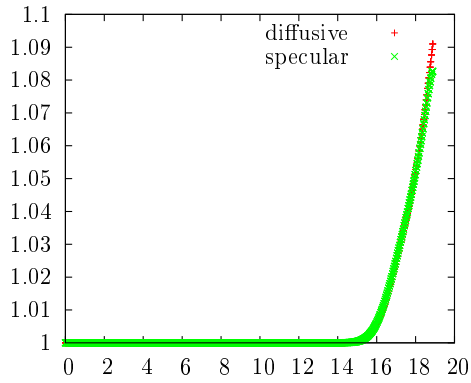


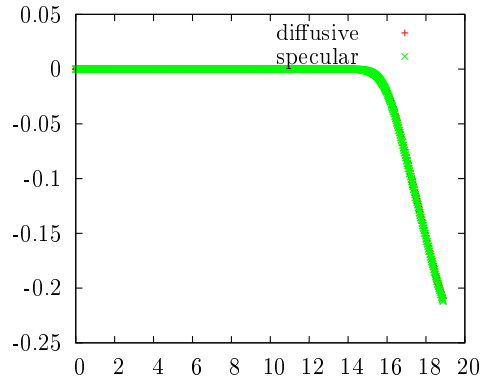
Fig. 6.6: Density and velocity plots for the one dimensional piston problem with specular reflection boundary conditions, red: $\tau = 10^{-1}$, green: $\tau = 10^{-4}$

our results to the results of Russo and Filbet in [50], one can see that the FPM results are in good agreement to the results gained through a finite differences scheme.

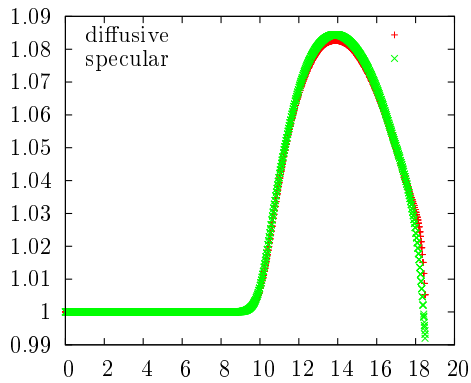
In Figure 6.7 we can see that the results obtained from diffusive reflection boundary conditions are comparable to the results concerning specular reflection conditions.



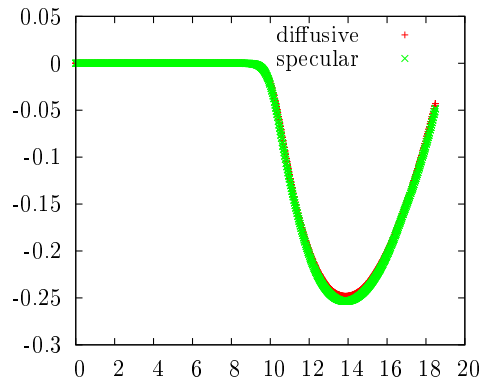
(a) Density ρ for $t = 1$ and $\tau = 10^{-1}$



(b) Velocity u for $t = 1$ and $\tau = 10^{-1}$



(c) Density ρ for $t = 3$ and $\tau = 10^{-4}$



(d) Velocity u for $t = 3$ and $\tau = 10^{-4}$

Fig. 6.7: Comparison between results belonging to specular and diffusive boundary conditions.

6.2 Moving piston in a two dimensional square

The example presented in the following section serves as an extension of the one dimensional moving piston problem to two dimensions in spacial and velocity space. As computational domain we choose $z = (x, y)^T \in [0, 1]^2$ and $v = (v_x, v_y)^T \in [-10, 10]^2$. The piston is again infinitely high and thin and has a length of 0.4. It starts near the left bottom corner of the square, see Figure 6.8 and moves with velocity $u_p = (u_{x_p}, u_{y_p})^T = (0.25 \sin(\frac{1}{2}\pi t), 0.25 \sin(\frac{1}{2}\pi t))^T$. The initial conditions are:

$$\begin{aligned} u(0.z) &= u_0 = 0.0, \\ \rho(0.z) &= \rho_0 = 1.0, \\ T(0.z) &= T_0 = 1.0, \\ f(0, z, v) &= f_0 = \frac{\rho_0}{(2.0\pi RT_0)} \exp\left(\frac{-\|v-u_0\|^2}{2.0RT_0}\right), \end{aligned}$$

where $u = (u_x, u_y)^T$ denotes the mean velocity of the gas and the gas constant R is set to be one.

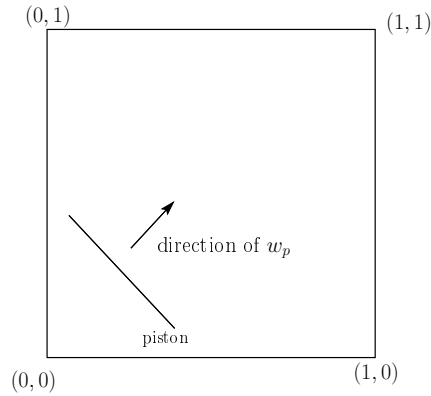


Fig. 6.8: Moving piston in 2D.

6.2.1 Specular reflection boundary conditions

As in the one dimensional piston problem we first explain specular reflection boundary conditions at the walls and at the piston.

At the walls the boundary conditions are similar to those at the left boundary in the example of the one dimensional piston problem, see Section 6.1.1, Figure 6.2.

Let Γ denote the set of points which belong to the right wall, i.e. $\Gamma = \{z : x = 1, y \in [0, 1]\}$. Then the set of ghost points at this wall can be computed as follows.

First we need the positions \tilde{z}^* :

$$\tilde{z}^* = \begin{cases} (2.0 - \tilde{x}, \tilde{y})^T, & \text{if } \tilde{x} > 1.0 \\ (\tilde{x}, \tilde{y})^T, & \text{if } 1.0 - \epsilon(h) < \tilde{x} \leq 1.0 \end{cases}. \quad (6.20)$$

Then we pick the \bar{z} , which is the nearest neighbor to \tilde{z}^* and look at the points $z_k \in P(\bar{z}, h)$, $k = 1, \dots, m$. We can easily get the ghost points z_k^* :

$$z_k^* = (2.0 - x_k, y_k)^T, \quad \forall z_k \in P(\bar{z}, h) \text{ with } \|z_k^* - \tilde{z}\| \leq h. \quad (6.21)$$

We skip the explanation for the other three walls, since there the procedure is completely analogous.

The values at the ghost points are:

$$f(t, z_k^*, v_{il}) = \begin{cases} f(t, z_k, v_{il} - 2n(n \cdot v_{il})), & \text{for } v_{il} \cdot n > 0 \\ f(t, z_k, v_{il}) & \text{else} \end{cases} \quad (6.22)$$

for $v_{il} = (v_{x_i}, v_{y_l})$, $i = -N_{v_x}, \dots, N_{v_x}$ and $l = -N_{v_y}, \dots, N_{v_y}$. Furthermore, $n = (n_1, n_2)^T = (-1, 0)^T$ is the normal at the right wall pointing inside the domain.

At the piston we also have to introduce ghost points. The difference to the one dimensional case is, that we have to reflect our points on a line in $2D$.

Formally the boundary condition reads:

$$f(t, z, v) = f(t, z, (v - u_p) - 2n(n \cdot (v - u_p))), \quad z \in \Gamma_p, \quad (v - u_p) \cdot n > 0, \quad (6.23)$$

where Γ_p denotes the set of points belonging to the piston, $n = (n_1, n_2)^T$ is the normal at the piston directed into the computational domain and $u_p = (u_{x_p}, u_{y_p})^T$ its velocity.

Let us define the set Γ_p of points which denote the position of the piston as:

$$\Gamma_p = \{z_{p_m} = (x_{p_m}, y_{p_m})^T : m = 1, \dots, M_p\} \quad (6.24)$$

Additionally, we can describe the piston as a linear function g such that $g(x_{p_m}) = y_{p_m} = a_p \cdot x_{p_m} + b_p$.

Remark 6.5. One has to remark that z_{p_m} do not belong to the set of FPM particles. In fact, we introduce these points for mirroring purposes to get the set of ghost points, as we will see later on.

For the definition of boundary particles we have to know on which side of the slab the particles are located. As computational domain we define the area in which the currently considered point z is located. In our case the piston is initially moving with positive velocity. Suppose, the computational domain is the area where the piston is moving to. Then the point $z_j = (x_j, y_j)$ is located inside the computational domain, if:

$$[g_p(x_j) - y_j] \cdot [g_p(n_1 + x_p) - (n_2 + y_p)] > 0, \quad (6.25)$$

where $(x_j, y_j)^T$ denotes the position of an arbitrary FPM particle, $(x_p, y_p)^T$ is an arbitrary point located at the piston and n_1, n_2 denote the components of the unit normal in the direction of the piston velocity, see Figure 6.9.

Let us define the set D as the set of points which lie on the diagonal band on which the piston moves, see Figure 6.9. Let D^C be the complement of D .

Then, we denote the set of points which belong to the computational domain by:

$$\Omega_I(n) = \{z_j : [g_p(x_j) - y_j] \cdot [g_p(n_1 + x_p) - (n_2 + y_p)] > 0\} \cup D^C \quad (6.26)$$

and the set of the points which are outside the domain by:

$$\Omega_O(n) = \Omega_I(n)^C. \quad (6.27)$$

For illustration on $\Omega_I(n)$ and $\Omega_O(n)$, see Figure 6.10.

Then the specular reflection boundary conditions are implemented in the following way (see also Figure 6.12). As in the previous cases, we first introduce the points \tilde{z}^* :

$$\tilde{z}^* = \begin{cases} 2.0 \frac{(\tilde{z} - z_p) \cdot l}{l \cdot l} \cdot l - \tilde{z} + 2.0 z_p, & \text{if } \tilde{z} \in \Omega_O \\ \tilde{z} & \text{if } \tilde{z} \in \Omega_I \end{cases}, \quad (6.28)$$

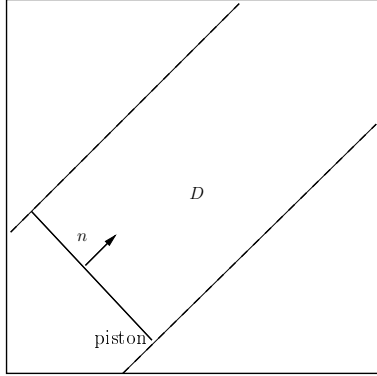


Fig. 6.9: D is the set of points which lie on the diagonal band on which the piston moves.

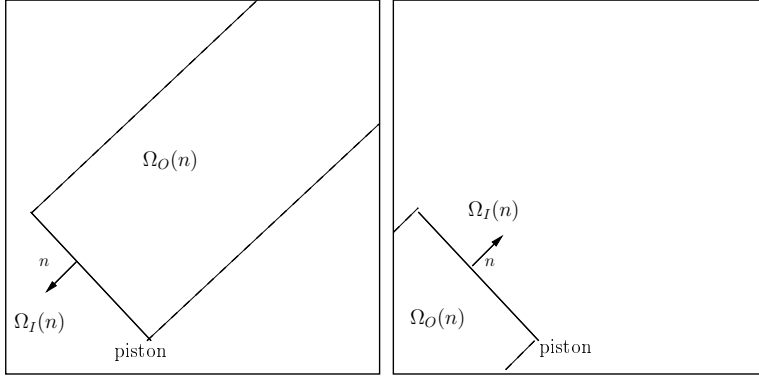


Fig. 6.10: Illustration of the set of points which are inside or outside the computational domain.

where z_p is an arbitrary point which belongs to the piston and l is a vector which lies on the piston. This means, that \tilde{z} is reflected on the slab, if the characteristic to which it belongs is starting outside the domain, see also Figure 6.11 for explanation. Suppose $z_k \in P(\tilde{z}, h)$, $k = 1, \dots, m$ are the neighboring particles of $\tilde{z} = \min_z \|z - \tilde{z}^*\|$. Then we introduce ghost points in the following way:

$$z_k^* = 2.0 \frac{(z_k - z_p) \cdot l}{l \cdot l} \cdot l - z_k + 2.0z_p, \quad \forall z_k \in P(\tilde{z}, h) \text{ with } \|z_k^* - \tilde{z}\| \leq h. \quad (6.29)$$

Thus, the reflected points which have a distance more than h to \tilde{z} are not mirrored on

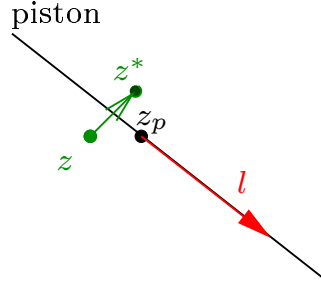


Fig. 6.11: Reflection of points on a slab, z is the point located outside the computational domain, z^* is the position which is reached by the reflection of z , z_p is an arbitrary point which belongs to the piston, l is a vector that is parallel to the piston.

the piston.

Then, the specular reflection boundary conditions at the ghost points read:

$$f(t, z_k^*, v_{il}) = \begin{cases} f(t, z_k, (v_{il} - u_p) - 2n(n \cdot (v_{il} - u_p))), & \text{for } (v_{il} - u_p) \cdot n > 0 \\ f(t, z_k, v_{il}) & \text{else} \end{cases} \quad (6.30)$$

After we have placed the ghost points and introduced the values at those, we can interpolate \tilde{f} using the values at the following set of points (see also Figure 6.12):

$$\tilde{P} = \{z_k^* : \|z_k^* - \tilde{z}\| \leq h\} \cup \{z_k \in P(\tilde{z}, h) : \|z_k - \tilde{z}\| \leq h\}, \quad (6.31)$$

where z_k^* denote the ghost points and z_k the inner points.

Observation 6.6. As for the one dimensional case, no interpolation of the distribution function f is required for the ghost points in z direction.

As one can see this makes the interpolation in the irregular point setting easier and faster than on a regular grid.

Suppose we had a regular grid. Then the ghost points would lie on this regular grid, see also the right part of Figure 6.12. That means that the ghost points would not mandatorily be reflected on the piston. To interpolate the value at \tilde{z} , we would have to reflect some ghost points back inside the domain and use the values of inner points

to interpolate the values at these reflected points. After this we would be able to interpolate \tilde{f} .

Of course, \tilde{f} has also to be interpolated with respect to the new values $\tilde{v} = (v - u_p) - 2n(n \cdot (v - u_p))$ in velocity space. Therefore, we simply use bi-linear interpolation, since the velocity grid is regular for this example.

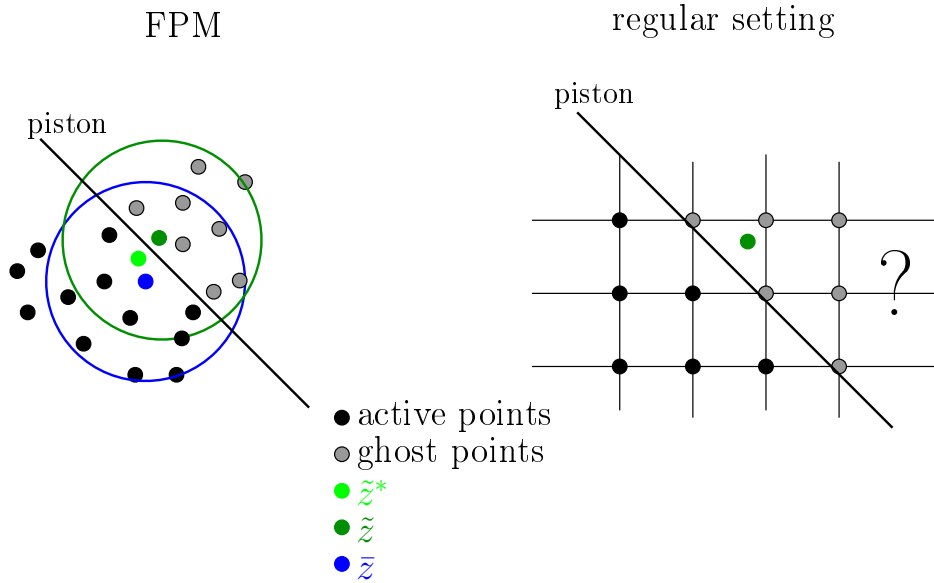


Fig. 6.12: *Specular reflection at the piston. In the FPM setting the points inside the green circle are used for interpolation. In the regular setting it is more complicated, since the ghost points are not reflected on the piston. Thus, the values at the ghost points in FPM do not have to be interpolated in spatial space.*

6.2.2 Diffusive reflection boundary conditions

For the explanation of the implementation of diffusive boundary conditions at the walls, see Section 5.2.3.

At the piston the procedure is similar to that in the one dimensional case, see Section 6.1.1.

Let $z_{p_m} \in \Gamma_p$, $m = 1, \dots, M_p$ be the points which represent the position of the piston and $\tilde{z}_{p_{mil}} = z_{p_m} - v_{il}\Delta t$. The procedure to compute $f_{p_{mil}}^{n+1}$ is the following:

1. For $(v_{il} - v_p) \cdot n \leq 0$ we reconstruct the values $\tilde{f}_{p_{mil}}^n$ using the values which are located inside the domain and the values $f_{p_{mil}}^n$. For the interpolation we use the Weighted Least Squares method, as usual.
2. After we have the values $\tilde{f}_{p_{mil}}^n$ we can compute the Maxwellian:

$$f_{p_{mil}}^{n+1} = M_{p_{mil}}^{n+1} = \frac{\rho_{p_m}}{2.0\pi RT_{p_m}} \exp\left(-\frac{\|v_{il} - u_p\|^2}{2RT_{p_m}}\right), \quad \forall (v_{il} - u_p) \cdot n > 0 \quad (6.32)$$

using the values $\tilde{f}_{p_{mil}}^n$ for computing ρ_p with the following arguments.

To compute ρ_p we impose:

$$\sum_{(v_{il}-u_p)\cdot n > 0} M_{p_{mil}}^{n+1} (v_{il} - u_p) \cdot n + \sum_{(v_{il}-u_p)\cdot n < 0} \tilde{f}_{p_{mil}}^n (v_{il} - u_p) \cdot n = 0 \quad (6.33)$$

We can use $\tilde{f}_{p_{mil}}^n$ in equation (6.33), since we have:

$$f_{p_{mil}}^{n+1} = M_{p_{mil}}^{n+1}, \quad \forall (v_{il} - u_p) \cdot n > 0 \quad (6.34)$$

and therefore we obtain:

$$\sum_{(v_{il}-u_p)\cdot n < 0} \phi_{il} M_{p_{mil}}^{n+1} = \sum_{(v_{il}-u_p)\cdot n < 0} \phi_{il} f_{p_{mil}}^{n+1} = \sum_{(v_{il}-u_p)\cdot n < 0} \phi_{il} \tilde{f}_{p_{mil}}^n, \quad (6.35)$$

where ϕ_{il} are the collision invariants.

3. Now we have all the values we need to compute $f_{p_{mil}}^{n+1}$ through the evolution equation:

$$f_{p_{mil}}^{n+1} = \tilde{f}_{p_{mil}}^n + \frac{\Delta t}{\tau} (M_{p_{mil}}^{n+1} - f_{p_{mil}}^{n+1}), \quad \forall (v_{il} - u_p) \cdot n \leq 0 \quad (6.36)$$

4. As a final step we can compute the values at $\tilde{z} \in \Omega_I$, using WLSQ interpolation as usual with the values at neighbors located in Ω_I and the values $f_{p_m}^n$ and $f_{p_m}^{n+1}$. For $\tilde{z} \in \Omega_O$ we first have to find the intersection points between the corresponding characteristic and the plain which connects the points $z_{p_m}^n$ and $z_{p_m}^{n+1}$ in time space. For that we introduce the points \hat{z} and $\hat{\tilde{z}}$, which represent the projections of the

points z and \tilde{z} on the piston, see Figure 6.13. For detecting the intersection point (red point in Figure 6.13) we reduce the problem to one dimension by looking at the distances between z and \hat{z} and \tilde{z} and $\hat{\tilde{z}}$. Then we search for the intersection point between the lines illustrated on the right part of Figure 6.13. For interpolating the value at this point we can use the values at neighboring particles located inside the domain plus the values $f_{p_m}^n$ and $f_{p_m}^{n+1}$.

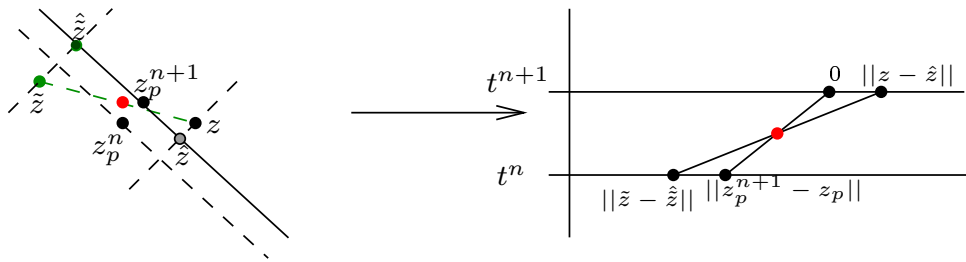


Fig. 6.13: For characteristics entering the domain we use the intersection point between the two lines (left part of the figure) and interpolate the value on this point as usual.

Remark 6.7. Of course, in the two dimensional setting we can also introduce ghost points and use a similar procedure as in the one dimensional case. We just have to extend the inverse Lax-Wendroff procedure, see 6.1.1 to two dimensions.

Remark 6.8. There is another problem we have to do remarks on. Suppose we have two pistons. The piston at time step t^n and that at t^{n+1} . It can happen that there are points which lie between the two slabs. These points belong to one region at time t^n and to another region at time t^{n+1} . Thus, they have to be extrapolated before we continue with the computations. For this extrapolation we use points which lie not on the same side as the point in question, see Figure 6.14. We use extrapolation of first order with linear test functions.

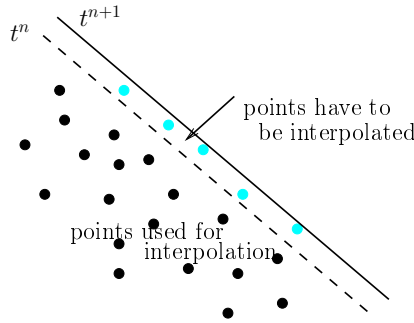


Fig. 6.14: Area of points which is overtaken by the piston in one time step

6.2.3 Results

In Figures 6.15 and 6.16 we can see the results for the $2D$ piston problem for $\tau = 10^{-1}$ and $\tau = 10^{-4}$, $h = 0.05$, $\Delta t = 20^{-3}$ and $\Delta v = 0.25$. The results are computed using specular reflection boundary conditions at the slab and at the walls.

One can see that while the velocity of the piston increases, for example in Figure 6.15 (a) and (b) for $t = 1$, the density above the slab increases whereas below the slab it decreases. The velocity vectors, see Figure 6.16 (a) and (b) point into the direction of the piston motion. At both ends of the piston a vortex develops.

For $t = 2$, the velocity of the slab is nearly zero, so the density tends to 1 near the piston, see Figure 6.15 (c) and (d). Understandably the mean velocity of the gas is very small in this region, see Figure 6.16 (c) and (d). This effect happens, since the gas has had enough time to disperse.

As the velocity of the piston reverses, for $t = 3$, the density profiles are reversed to that of $t = 1$, see Figure 6.15 (e) and (f). Now the density below the slab is high and that above it is low. The velocity vectors also point into the reverse direction to those for $t = 1$, see Figure 6.16 (e) and (f). Furthermore, if we compare the velocity vector fields for $\tau = 10^{-1}$ (left side of Figure 6.16) to those for $\tau = 10^{-4}$ (right side of Figure 6.16), we can see that the profiles for the smaller values of τ are more precise.

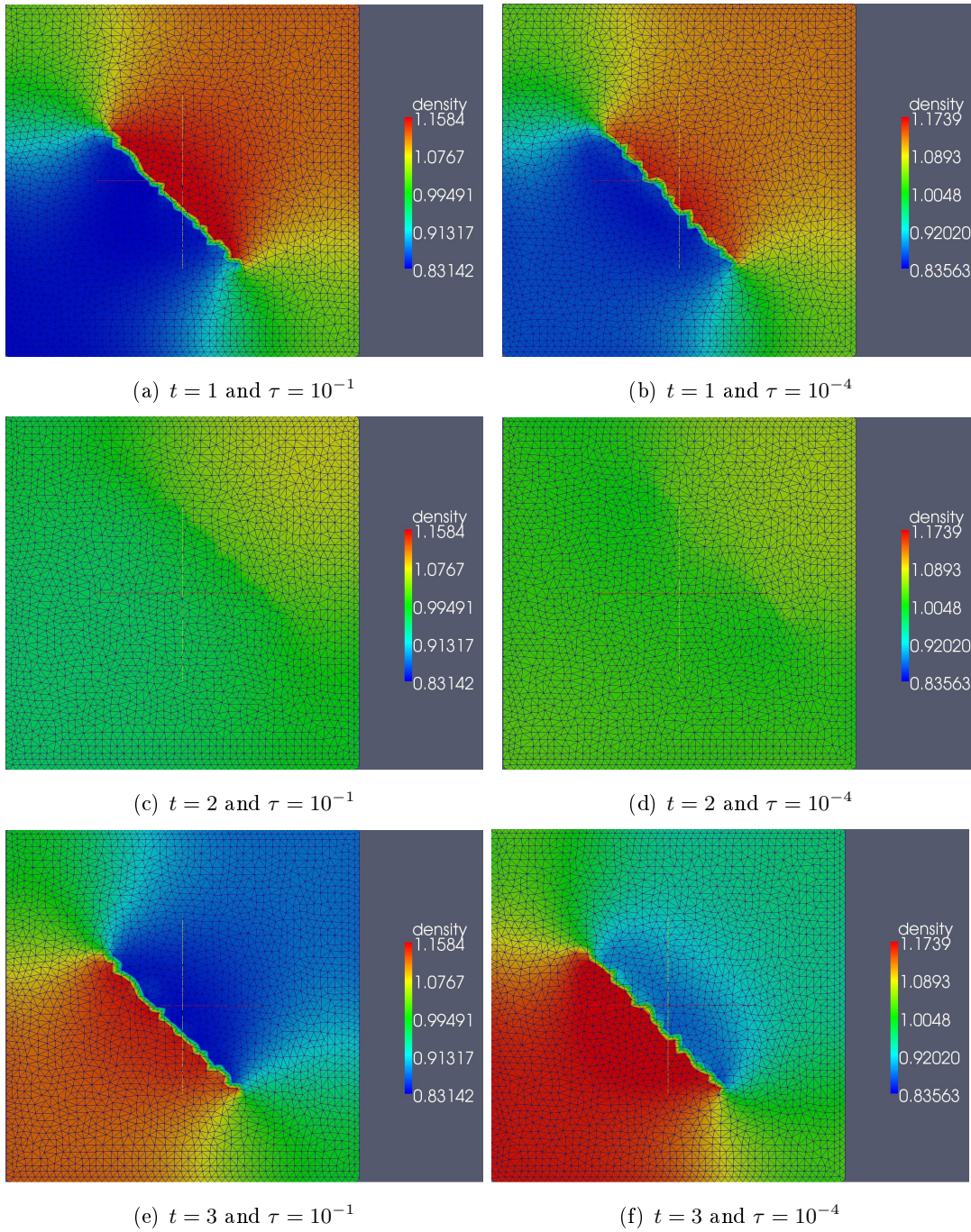


Fig. 6.15: *Density fields for the 2D piston problem.*

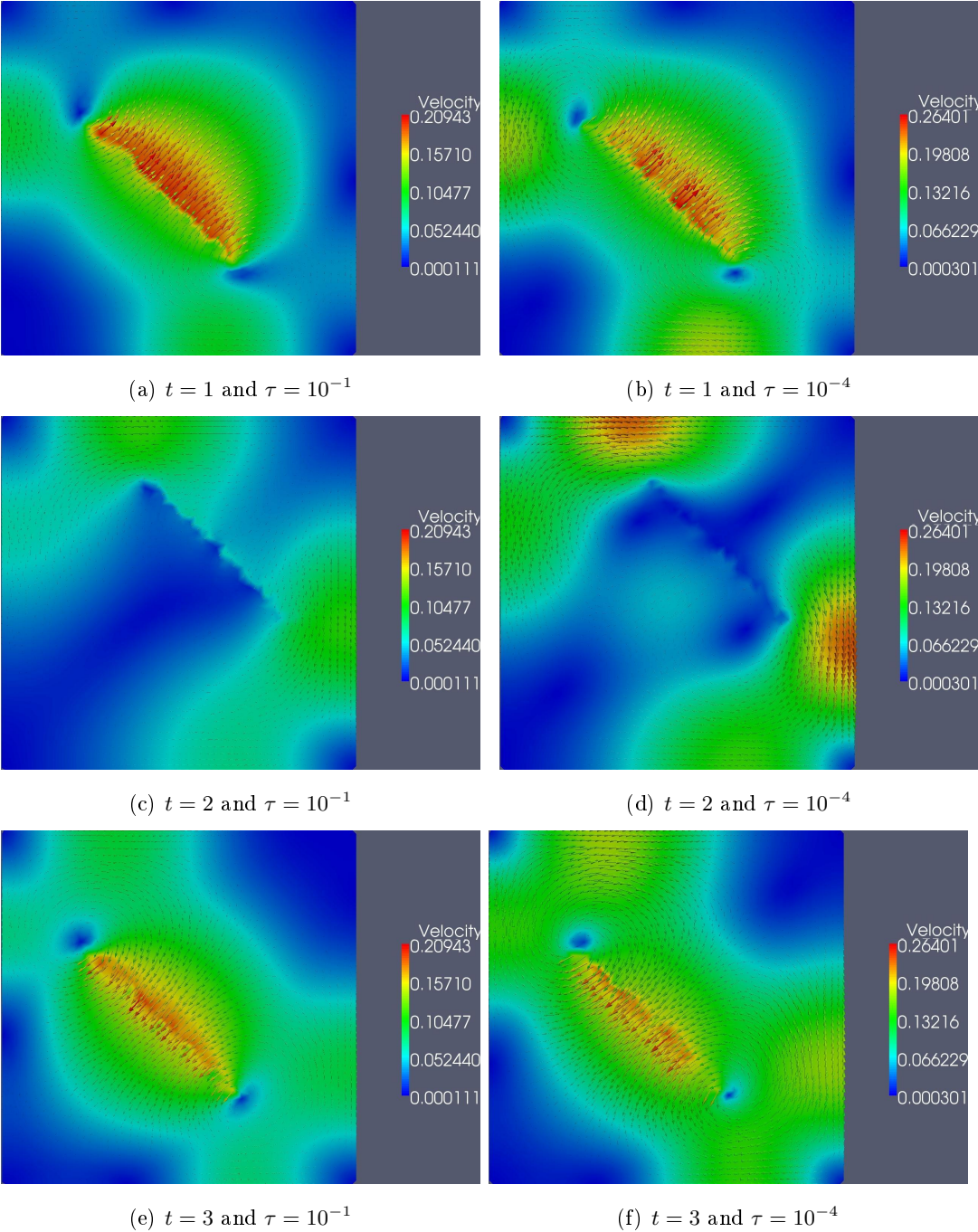


Fig. 6.16: Velocity vector fields for the 2D piston problem.

Remark 6.9. The regular triangulations appearing in the figures are just for plotting purposes. Furthermore, the shape of the piston appears irregular, since usually there are no computational points which lie exactly on the piston. Thus, the figure only show values at FPM points, which are naturally irregularly arranged.

In Figure 6.17 we can see results for diffusive reflection boundary conditions at the piston. It is seen, that the results are comparable to those for specular reflection conditions, see Figures 6.15 and 6.16.

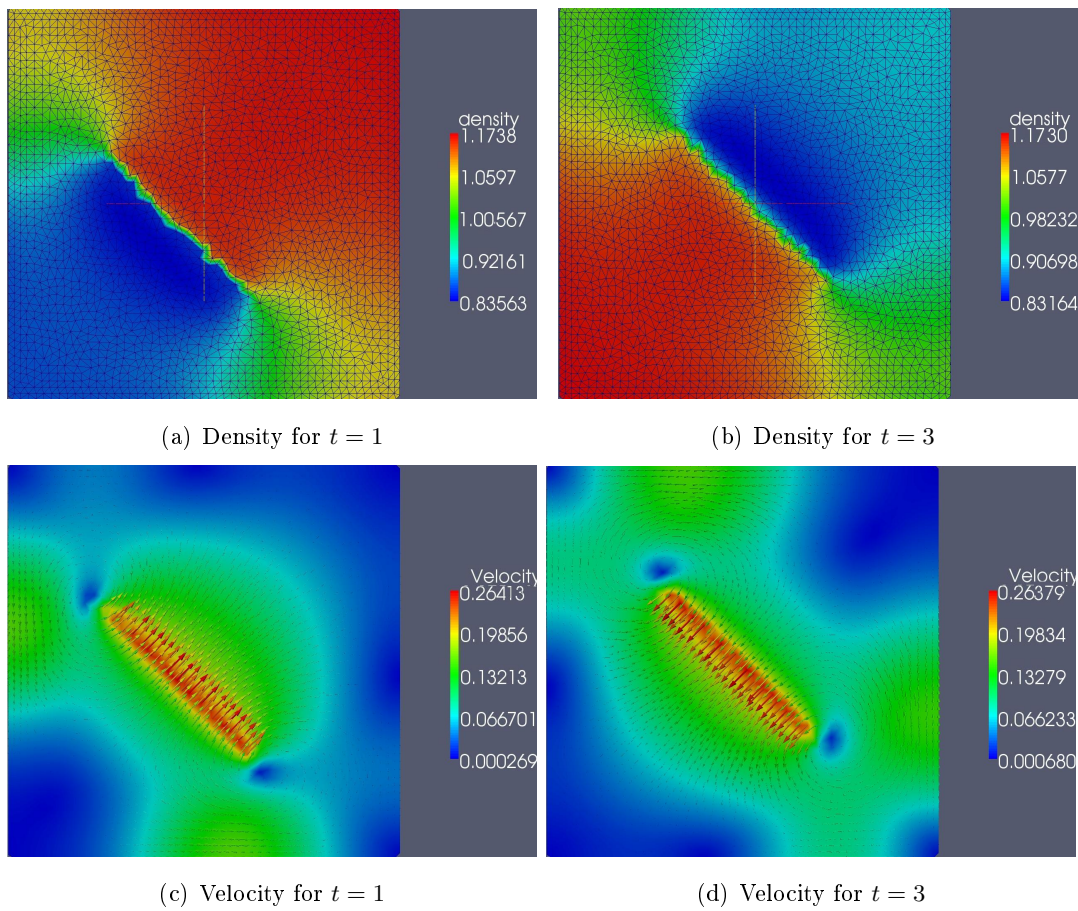


Fig. 6.17: Density fields and velocity vector fields with diffusive reflection at the piston for the 2D piston problem.

6.3 Rotating slab

Motivated by the function of a rotary vane pump (see motivation), we now have a look at a rotating slab in $2D$.

For the spatial space we choose the computational domain to be the same as for Driven Cavity flow and the moving piston in $2D$. The slab is infinitely high and thin. It rotates around the point $z_M = (x_M, y_M)^T = (0.5, 0.5)^T$, has a length of 0.4 and an angular rate $\omega = \frac{2.0\pi}{T}$, where $T = 4.0$.

The initial conditions are:

$$\begin{aligned} u(0.z) &= u_0 = 0.0, \\ \rho(0.z) &= \rho_0 = 1.0, \\ T(0.z) &= T_0 = 1.0, \\ f(0, z, v) &= f_0 = \frac{\rho_0}{(2.0\pi RT_0)} \exp\left(\frac{-\|v-u_0\|^2}{2.0RT_0}\right). \end{aligned}$$

The gas constant is set to be $R = 1$ and $v \in [-10, 10]^2$.

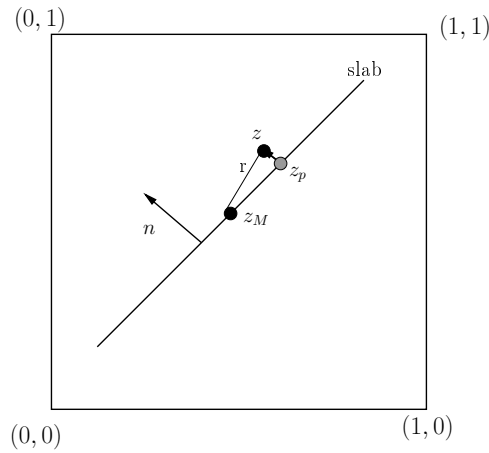


Fig. 6.18: Set up for the rotating slab problem.

6.3.1 Specular reflection boundary conditions

At the walls the boundary conditions are similar to those for the non-rotating problem, see equations 6.20, 6.21 and 6.22.

The main difference for the boundary conditions at the slab is that it is rotating and so its velocity is different at each point z_{pm} .

We define the velocity u_p , which belongs to the slab, at an arbitrary FPM particle z through:

$$u_p(z) = \omega \cdot n \cdot r(z), \quad (6.37)$$

where n is the normal at the slab which is pointing into the computational domain, r is the radius at z and ω the angular rate, see also Figure 6.18.

Then the specular boundary conditions at the piston are defined as follows.

Again we make use of the sets Ω_O and Ω_I , which have a slightly different configuration as in the previous case. Since the slab is rotating the set D consists of the following points, see also Figure 6.19:

$$D = \{z_j : \|z_M - z_j\| \leq r_M\}, \quad (6.38)$$

where r_M denotes the radius of the circle the slab is passing during the computational time. Using this definition of D we can define the sets Ω_O and Ω_I in the same way as in equations (6.26) and (6.27). Note, that for that we have to make the assumption that the computational domain is the area, where the vector n is pointing to. Let $\tilde{z} \in \Omega_I$ or $\tilde{z} \in \Omega_O$. If \tilde{z} is the nearest neighbor to \tilde{z}^* (for definition of \tilde{z}^* , see equation 6.28), then the ghost points have the following positions:

$$z_k^* = 2.0 \frac{(z_k - z_p) \cdot l}{l \cdot l} \cdot l - z_k + 2.0z_p, \quad \forall z_k \in P(\tilde{z}, h), \|z_k^* - \tilde{z}\| < h, \quad (6.39)$$

where z_p and l have already been defined in equation 6.29. Furthermore, the values at the ghost points are:

$$f(t, z_k^*, v_{il}) = \begin{cases} f(t, z_k, (v_{il} - u_p(\tilde{z})) - 2n(n \cdot (v_{il} - u_p(\tilde{z}))))), & \text{for } (v_{il} - u_p(\tilde{z})) \cdot n > 0 \\ f(t, z_k, v_{il}), & \text{else} \end{cases}, \quad (6.40)$$

where $u_p(\tilde{z})$ is the velocity of the piston projected on the point \tilde{z} , see equation 6.37.

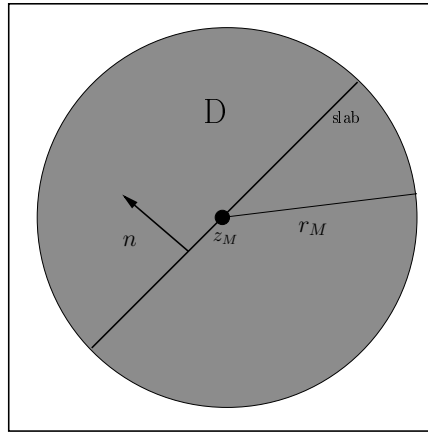


Fig. 6.19: Illustration of set D for the rotating slab problem.

Remark 6.10. The implementation of diffusive boundary conditions works similar to the previous case, see Section 6.2.2. The main difference lies in the treatment of \tilde{z} belonging to characteristics starting outside the computational domain.

6.3.2 Results

In Figures 6.20 and 6.21 we can see results for the rotating slab problem with $\tau = 10^{-1}$ (left) $\tau = 10^{-4}$ (right), $\Delta t = 20^{-3}$ and $\Delta v = 0.25$. The results are gained by using specular reflection conditions at the piston. As expected the density in front of the slab, that means in direction of rotation of the slab increases whereas behind the slab it decreases, see Figure 6.20. We also see, that there is a gradient depending on the distance to the slab. Near the slab the density values are higher/lower than far from it.

The velocity vectors are directed into the rotational direction of the slab, see Figure 6.21. At the outside ends of the slab the velocity values are higher than in the middle. The same holds true for the density values. Furthermore, vortexes develop at the ends of the slab, see Figure 6.21 (a) and (b). The contours of the vortexes are more pronounced for $\tau = 10^{-4}$, see Figure 6.21 (b).

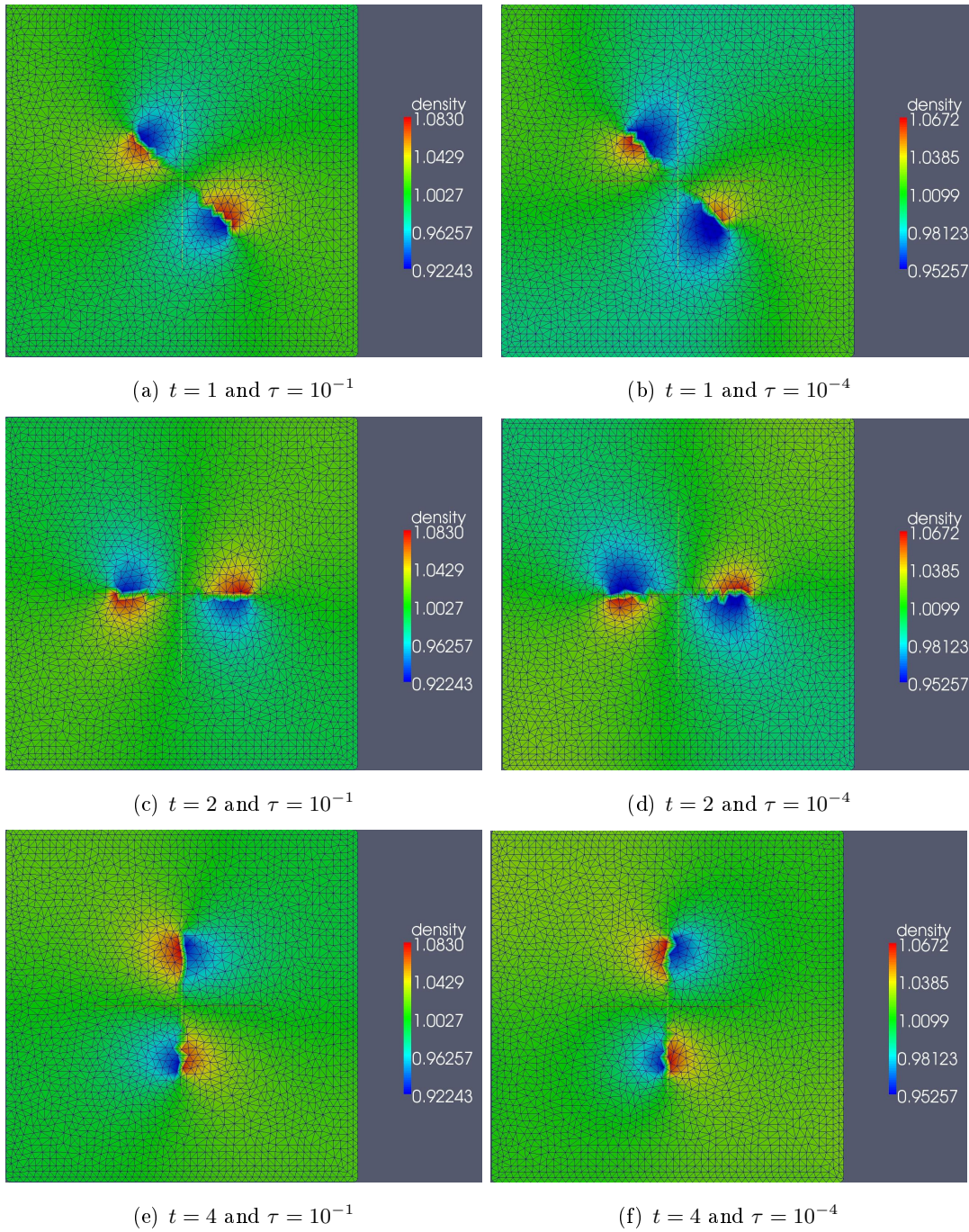


Fig. 6.20: *Density fields for the rotating slab problem*

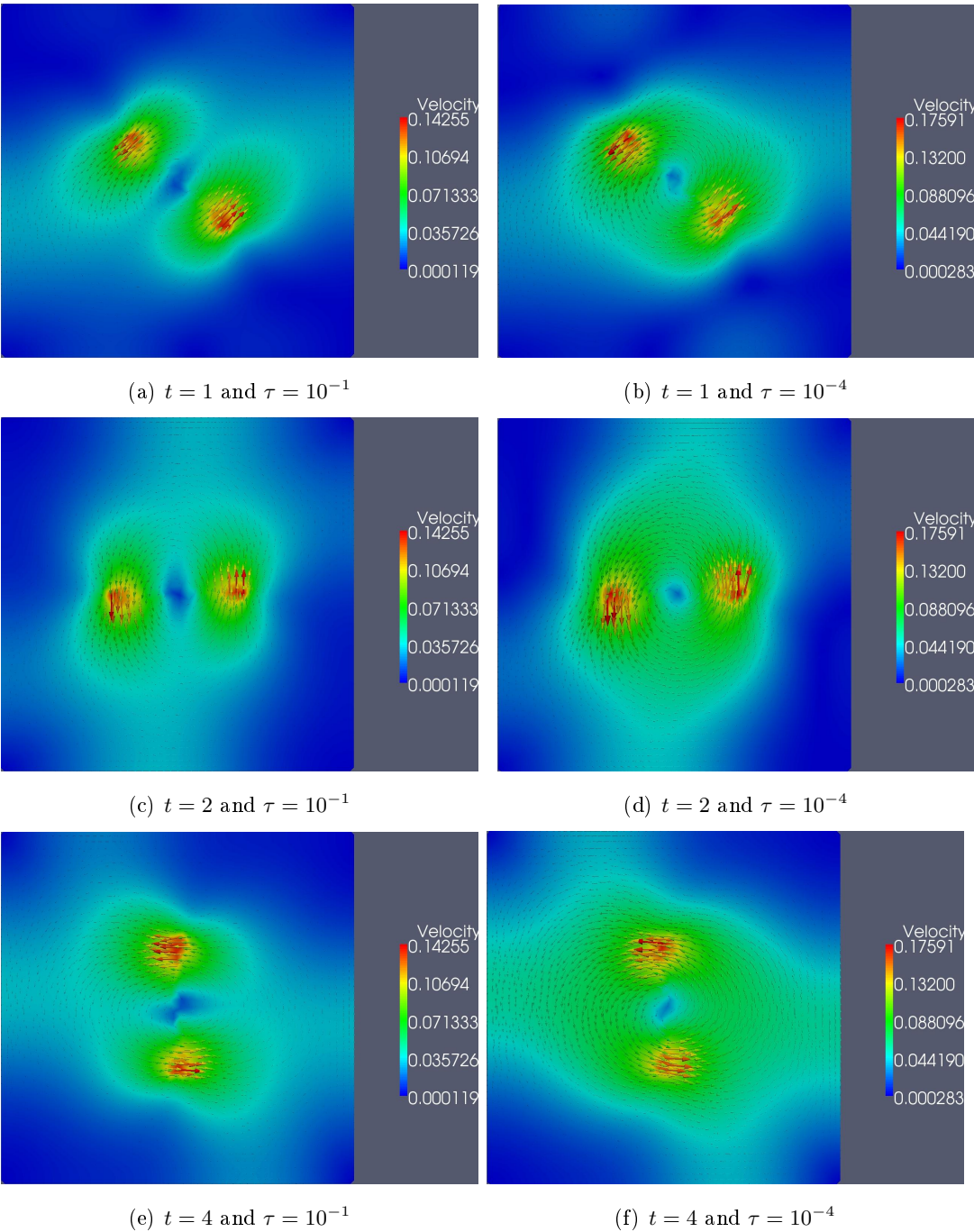


Fig. 6.21: Velocity vector fields for the rotating slab problem

Chapter 7

Conclusions and outlook

In this work a numerical scheme for solving the kinetic BGK model equation in the Finite Pointset Method (FPM) framework is developed. In Chapter 4 different methods for solving the BGK model equation on a one dimensional regular grid are reviewed and evaluated for their performance and accuracy within the FPM framework. The Semi-Lagrangian scheme is found to be suitable for our needs. For this reason, the scheme we develop here, is based on this Semi-Lagrangian scheme, which was established by Russo and Filbet in 2009 [50]. In Section 5.2 the extension of the numerical method to two dimensions in spatial and velocity space is presented.

Furthermore, numerical examples are shown in one and two dimensional physical and velocity space, in sections 5.1.3 and 5.2.3, respectively. Chapter 6 details the application of the new, FPM based scheme to problems with moving boundaries. In this context special attention is paid to the implementation of specular and diffusive boundary conditions. We see, that in particular related to these kind of problems, our method is superior to the already existing regular grid methods. The main advantage lies in the implementation of boundary conditions, which requires a less complicated interpolation in the grid free framework than on a regular grid. In fact, some interpolations required for the regular grid methods can be avoided entirely.

The numerical results are computed for different ranges of the Knudsen number. In the continuous flow regime, that is, for small Knudsen numbers, results are compared to the exact solution of the Euler equations, where available. Here, a comparison with solutions of the Navier-Stokes equations would also be interesting and could be done as future work. In the molecular flow regime, they are compared to the results of the

DSMC method. We see that in both cases our solutions are comparable to solutions obtained from the respective reference methods. In the case of molecular flow our solution is smooth whereas the solution of the DSMC method tends to fluctuate. For some examples, there is no comparison to reference solutions is given in the current work. This could be done as future work, as well.

The present scheme is only first order accurate in time and third order accurate in space. To get a higher order scheme in time one can use higher order Runge-Kutta schemes, as it was done in [44]. For increasing the accuracy in space one can easily adapt higher order interpolation with the WLSQ method. In the context of interpolation one has also to account for discontinuities in the distribution function. In the examples used in this work, the solution is expected to be smooth inside the computational domain. If this were not the case, non-oscillating reconstructions should be applied.

One aspect, this work is not concerned with, is that the treatment of moving boundary problems using deterministic methods may create discontinuities of the distribution function in velocity space near the moving parts. Aoki and Tsuji recently raised this statement in [66]. Thus, a natural thing to do, is to study this claim in our context.

Moreover, in this work the FPM particles remain fixed during the computations. A natural next step is, to develop a scheme for solving kinetic models in a moving mesh framework.

Furthermore, we choose a regular velocity space discretization in this work. Thus, a natural extension is, to distribute the points in velocity space irregularly, as well. This can be useful for cases with complicated velocity distributions and also can help to reduce computational effort.

Another obvious extension is, to upgrade the mesh-free scheme first to three dimensions in velocity space and then to three dimensions in physical and velocity space, which allows more realistic simulations of physical objects. In this context, the implementation of mixed boundary conditions can be considered, too. However, computational times are expected to increase drastically with each increase in dimension. An efficient parallelization scheme or a less time consuming method, such as the solution of

13 Moment Equations, seem to be required to avoid unrealistically long computational times.

Furthermore, a considerable extension is the application to more realistic Boltzmann collision terms, such as for example the variable hard sphere molecules.

Bibliography

- [1] V. V. Aristov. *Direct Methods for Solving the Boltzmann Equation and Study of Nonequilibrium Flows*. Kluwer Academic Publishers, 2001.
- [2] L. Arkeryd. On the Boltzmann equation. *Archive for Rational Mechanics and Analysis*, 45:1–16, January 1972.
- [3] L. Arkeryd. On the Boltzmann equation part II: The full initial value problem. *Archive for Rational Mechanics and Analysis*, 45:17–34, January 1972.
- [4] Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. Implicit-Explicit Runge-Kutta Methods for Time-Dependent Partial Differential Equations. *Appl. Numer. Math.*, 25:151–167, 1997.
- [5] Menwer M. Attarakih, Christian Drumm, and Hans-Jörg Bart. Solution of the Population Balance Equation using the Sectional Quadrature Method of Moments (SQMOM). *Chemical Engineering Science*, 64(4):742–752, 2009. 3rd International Conference on Population Balance Modelling.
- [6] P. L. Bhatnagar, E. P. Gross, and M. Krook. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.*, 94:511–525, May 1954.
- [7] G.A. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford Engineering Science Series vol 42, 1994.
- [8] Robert C. Blanchard, Richard G. Wilmoth, and James N. Moss. Aerodynamic Flight Measurements and Rarefied-Flow Simulations of Mars Entry Vehicles. *Journal of Spacecraft and Rockets*, 34(5):687–690, 1997.

- [9] Stéphane Brull. Gas-surface interaction and boundary conditions for the Boltzmann equation. *arXiv e-prints*, 2013.
- [10] C. Buet. A discrete-velocity scheme for the Boltzmann operator of rarefied gas dynamics. *Transport Theory and Statistical Physics*, 25(1):33–60, 1996.
- [11] C. Cercignani. *Theory and application of the Boltzmann equation*. Elsevier, 1976.
- [12] C. Cercignani and G. Tironi. Some applications of a linearized kinetic model with correct Prandtl number. *Il Nuovo Cimento B Series 10*, 43(1):64–78, 1966.
- [13] Carlo Cercignani. *Slow Rarefied Flows: Theory and Application to Micro-Electro-Mechanical Systems*. 41. Birkhäuser Basel, 2006.
- [14] S. Chapman and T. G. Cowling. *The mathematical Theory of Non-uniform Gases*. Cambridge University Press, 1970.
- [15] F. Coron and B. Perthame. Numerical Passage from Kinetic to Fluid Equations. *SIAM Journal on Numerical Analysis*, 28(1):pp. 26–42, 1991.
- [16] E. L. Cussler. *Diffusion: Mass Transfer in Fluid Systems*. Cambridge Series in Chemical Engineering, 1997.
- [17] G. Dechristé and L. Mieussens. A moving mesh approach for the numerical simulation of gas micro flows. In M. Mareschal and A. Santos, editors, *American Institute of Physics Conference Series*, volume 1501 of *American Institute of Physics Conference Series*, pages 366–372, November 2012.
- [18] G Dechristé and L Mieussens. Numerical simulation of micro flows with moving obstacles. *Journal of Physics: Conference Series*, 362(1):012030, 2012.
- [19] O. Desjardins, R.O. Fox, and P. Villedieu. A quadrature-based moment method for dilute fluid-particle flows. *Journal of Computational Physics*, 227(4):2514–2539, 2008.
- [20] C. Drum, S. Tiwari, J. Kuhnert, and H.-J. Bart. Finite Pointset Method for Simulation of the Liquid-Liquid Flow Field in an Extractor. *Comput. Chem. Eng.*, 32, 2008.

- [21] J. H. Ferziger and H. G. Kaper. *Mathematical Theory of Transport Processes in Gases*. North-Holland, Amsterdam, 1972.
- [22] Francis Filbet and Chang Yang. An inverse Lax–Wendroff method for boundary conditions applied to Boltzmann type models. *Journal of Computational Physics*, 245(0):43–61, 2013.
- [23] R.O. Fox. A quadrature-based third-order moment method for dilute gas-particle flows. *Journal of Computational Physics*, 227(12):6313–6350, 2008.
- [24] Maria Friedrich. Application of the Finite Pointset Method to a microfluidical problem. Master’s thesis, TU Kaiserslautern, 2010.
- [25] Vicente Garzó and Andrés Santos. Comparison between the Boltzmann and BGK equations for uniform shear flow. *Physica A: Statistical Mechanics and its Applications*, 213(3):426–434, 1995.
- [26] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.
- [27] Harold Grad. On the kinetic theory of rarefied gases. *Communications on Pure and Applied Mathematics*, 2(4):331–407, 1949.
- [28] Amy Henderson, Jim Ahrens, and Charles Law. *The ParaView Guide*. Kitware Inc., 2004.
- [29] H.L. Frisch J.L. Lebowitz and E. Helfand. Theory and Simulation of Hard-Sphere Fluids and Related Systems. *Phys. Fluids*, 3:325, 1960.
- [30] Y. Sone K. Aoki and T. Yamada. Numerical analysis of gas flows condensing on tis plane condensed phase on the basis of kinetic theory. *Physics of Fluid A*, 2:1867–1878, 1990.
- [31] Axel Klar. An Asymptotic Preserving Numerical Scheme for Kinetic Equations in the Low Mach Number Limit. *SIAM J. Numer. Anal.*, 36(5):1507–1527, July 1999.

- [32] Maria Kobert, Joerg Kuhnert, and Axel Klar. Application of the Finite Pointset Method (FPM) to the kinetic BGK model. *PAMM*, 12(1):657–658, 2012.
- [33] M. Jaradat H.-J. Bart J. Kuhnert C. Drumm S. Tiwari V.K. Sharma A. Klar M. Attarakih. *A multivariate sectional quadrature method of moments for the solution of the population balance*. in G.B. Ferrais S. Pierucci, editor, Proceed. 20th ECSAPE. Elsevier, Amsterdam, 2010.
- [34] G. Russo M. Groppi and G. Stracquadanio. High order semi-Lagrangian methods for the BGK equation. *ArXiv e-prints*, 2014.
- [35] J. Clerk Maxwell. On Stresses in Rarefied Gases Arising from Inequalities of Temperature. *Phil. Trans. R. Soc. Lond*, 170:231–256, 1879.
- [36] Luc Mieussens. Discrete velocity model and implicit Scheme for the BGK equation of rarefied gas dynamics. *Mathematical Models and Methods in Applied Sciences*, 10(08):1121–1149, 2000.
- [37] M.Krook. Continuum equations in the dynamics of rarefied gases. *J. Fluid Mech.*, 6:593–604, 1959.
- [38] A.A. Mohamad. *Lattice Boltzmann Method, Fundamentals and Engineering Applications with Computer Codes*. Springer, 2011.
- [39] E. P. Muntz. Rarefied gas Dynamics. *Annual Reviews of Fluid Mechanics*, 21:387–417, 1989.
- [40] Taku Ohwada. Higher Order Approximation Methods for the Boltzmann Equation. *Journal of Computational Physics*, 139(1):1–14, 1998.
- [41] Lorenzo Pareschi and Russel E. Caflisch. An implicit Monte Carlo method for rarefied gas dynamics. I: The space homogeneous case. *J. Comput. Phys*, pages 90–116, 1999.
- [42] Cem Pekardan, Sruti Chigullapalli, Lin Sun, and Alina Alexeenko. Immersed boundary method for Boltzmann model kinetic equations. *AIP Conference Proceedings*, 1501(1):358–365, 2012.

- [43] B. Perthame. Global existence to the BGK model of Boltzmann equation. *Journal of Differential Equations*, 82(1):191–205, 1989.
- [44] S. Pieraccini and G. Puppo. Implicit - Explicit schemes for BGK kinetic equations. *Journal of Scientific Computing*, 2007.
- [45] Phoolan Prasad. P.L. Bhatnagar and the BGK Model.
- [46] W.H. Press. *Numerical recipes in Fortran 90 : the art of parallel scientific computing*. Cambridge University Press, Cambridge, 1996.
- [47] Thorsten Pöschel and T. Schwager. Direct Simulation Monte Carlo. In *Computational Granular Dynamics*, pages 191–210. Springer Berlin Heidelberg, 2005.
- [48] Francois Rogier and Jacques Schneider. A direct method for solving the Boltzmann equation. *Transport Theory and Statistical Physics*, 23(1-3):313–338, 1994.
- [49] E. J. Rosenbourn. The Mathematical Theory of Non-Uniform Gases (Chapman, S.; Cowling, T. G.). *Journal of Chemical Education*, 18(1):48, 1941.
- [50] G. Russo and F. Filbet. Semilagrangian Schemes applied to moving boundary problems for the BGK model of rarefied gas dynamics. *Kinetic and Related Models*, 2:231–250, 2009.
- [51] G. Russo, P. Santagati, and S.-B. Yun. Convergence of a Semi-Lagrangian Scheme for the BGK Model of the Boltzmann Equation. *ArXiv e-prints*, July 2010.
- [52] S. Manservisi S. Tiwari. Incompressible Navier-Stokes Flows by Least Squares Approximation. *The Nepali Math. Sci. Report*, 20, 2002.
- [53] P. Santagati and G. Russo. A New Class of Conservative Large Time Step Methods for the BGK Models of the Boltzmann Equation. *ArXiv e-prints*, March 2011.
- [54] M.Sh. Shavaliyev. Super-Burnett corrections to the stress tensor and the heat flux in a gas of Maxwellian molecules. *Journal of Applied Mathematics and Mechanics*, 57(3):573–576, 1993.

- [55] Chi-Wang Shu. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, volume 1697 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg.
- [56] H. Struchtrup. Stable Transport equations for rarefied gases at high orders in the Knudsen number. *Physics of Fluids*, 16:3921–3934, 2004.
- [57] H. Struchtrup. Derivation of 13 Moment equations for rarefied gas flow to second order accuracy for arbitrary interaction potentials. *Multiscale Model Simul.*, 3:221–243, 2005.
- [58] Henning Struchtrup and Manuel Torrilhon. Regularized 13 moment equations for rarefied gas flows. In Salvatore Rionero and Giovanni Romano, editors, *Trends and Applications of Mathematics to Mechanics*, pages 247–267. Springer Milan, 2005. 10.1007/88-470-0354-719.
- [59] P Taheri, M Torrilhon, and H Struchtrup. Couette and Poiseuille microflows: analytical solutions for regularized 13-moment equations. *Physics of Fluids*, 2009.
- [60] Mario Theriault. *Great Maritime Inventions 1833-1950*. Goose Lane Editions, 2001.
- [61] V.A. Titarev and E.M. Shakhov. Computational study of a rarefied gas flow through a long circular pipe into vacuum. *Vacuum*, 86(11):1709–1716, 2012.
- [62] S. Tiwari and J. Kuhnert. Grid free method for solving the Poisson equation. In *Wavelet Analysis And Applications*, 2004.
- [63] Sudarshan Tiwari and Joerg Kuhnert. Finite pointset method based on the projection method for simulations of the incompressible Navier-Stokes equations. *Meshfree methods for partial differential equations, Band 1*, 2004.
- [64] Brian V. Smith Tom Sato. Xfig user manual @ONLINE, 2013.
- [65] Manuel Torrilhon and Henning Struchtrup. Boundary conditions for regularized 13-moment-equations for micro-channel-flows. *J. Comput. Phys.*, 227(3):1982–2011, January 2008.

- [66] Tetsuro Tsuji and Kazuo Aoki. Moving boundary problems for a rarefied gas: Spatially one-dimensional case. *Journal of Computational Physics*, 250(0):574–600, 2013.
- [67] W. Vogt. Adaptive Verfahren zur numerischen Quadratur und Kubatur. *Preprint No. M*, 2006.
- [68] John Wendt. *Computational Fluid Dynamics: An Introduction*. Springer, 2009.
- [69] Thomas Williams, Colin Kelley, and many others. Gnuplot 4.4: an interactive plotting program, March 2010.
- [70] W. R. Wolf and J. L. F. Azevedo. High-order ENO and WENO schemes for unstructured grids. *International Journal for Numerical Methods in Fluids*, 55(10):917–943, 2007.
- [71] J.Y. Yang and J.C. Huang. Rarefied Flow Computations Using Nonlinear Model Boltzmann Equations. *Journal of Computational Physics*, 120(2):323–339, 1995.
- [72] W. Yomsatieankul. *High-order Non-oscillatory Schemes Using Meshfree Interpolating Moving Least Squares Reconstruction for Hyperbolic Conservation Laws*. 2010.

Curriculum Vitae

- 10/2010 - 12/2014: PhD studies, Fraunhofer Institute for Industrial Mathematics (ITWM), supervised by Dr. Jörg Kuhnert (ITWM) and Prof. Dr. Axel Klar (TU Kaiserslautern), title of the dissertation:
Application of the Finite Pointset Method to moving boundary problems for the BGK model of rarefied gas dynamics
- 06/2010: Diploma in Mathematics, Technical University (TU) Kaiserslautern, specialization:
Modeling and Scientific Computing
- 10/2009 - 06/2010: Diploma thesis, TU Kaiserslautern, title of the thesis:
Application of the Finite Pointset Method to a microfluidical problem
- 09/2008 - 02/2009: Studies abroad,
Saint Petersburg State University,
Saint Petersburg, Russia
- 10/2005 - 06/2010: Study of Mathematics with Economics as minor, TU Kaiserslautern
- 09/2002 - 07/2005: Grammar School, Walldürn
- 09/1996 - 07/2002: Junior High School, Buchen
- 09/1995 - 07/1996: Elementary School, Buchen
- 09/1991 - 06/1995: Elementary School, Scheschart (Russia)

Akademischer Werdegang

- 10/2010 - 12/2014: Promotion am Fraunhofer - Institut für Techno- und Wirtschaftsmathematik (ITWM), betreut durch Dr. Jörg Kuhnert (ITWM) und Prof. Dr. Axel Klar (TU Kaiserslautern), Titel der Dissertation:
Application of the Finite Pointset Method to moving boundary problems for the BGK model of rarefied gas dynamics
- 06/2010: Diplom in Mathematik, Technische Universität (TU) Kaiserslautern, Schwerpunkt:
Modellierung und Wissenschaftliches Rechnen
- 10/2009 - 06/2010: Diplomarbeit, TU Kaiserslautern, Titel der Arbeit:
Application of the Finite Pointset Method to a microfluidical problem
- 09/2008 - 02/2009: Auslandssemester,
Staatliche Universität St. Petersburg,
St. Petersburg, Russland
- 10/2005 - 06/2010: Studium der Mathematik mit Nebenfach Wirtschaftswissenschaften, TU Kaiserslautern
- 09/2002 - 07/2005: Wirtschaftsgymnasium, Walldürn
- 09/1996 - 07/2002: Haupt- und Werkrealschule, Buchen
- 09/1995 - 07/1996: Grundschule, Buchen
- 09/1991 - 06/1995: Grundschule, Scheschart (Russland)