Efficient particle methods for solving the Boltzmann equation

by

Thomas Homolle

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[February 2007]
January 2007

Author ....
Department of Aeronautics and Astronautics
January 19, 2007

Certified by ....
Nicolas G. Hadjiconstantinou
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by ....
Jaime Peraire
Professor of Aeronautics and Astronautics,
Chair, Committee on Graduate Students

Efficient particle methods for solving the Boltzmann equation

by

Thomas Homolle

Submitted to the Department of Aeronautics and Astronautics
on January 19, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

A new particle simulation method for solving the Boltzmann equation is presented and tested. This method holds a significant computational efficiency advantage for low-signal flows compared to traditional particle methods such as the Direct Simulation Monte Carlo (DSMC). More specifically, the proposed algorithm can efficiently simulate arbitrarily small deviations from equilibrium (e.g. low speed flows) at a computational cost that does not scale with the deviation from equilibrium, while maintaining the basic algorithmic structure of DSMC. This is achieved by incorporating the variance reduction ideas presented in [L. L. Baker and N. G. Hadjiconstantinou, Physics of Fluids, vol 17, art. no 051703, 2005] within a collision integral formulation; the latter ensures that the deviation from equilibrium remains finite and thus the calculation remains stable for collision dominated flows, in contrast to previous attempts. The formulation, developed within this thesis, is described in detail. The resulting scheme is validated for a wide range of Knudsen numbers (ratio of molecular mean free path to characteristic flow lengthscale) -ranging from collision-dominated flow to collisionless flow- and a wide range of deviations from equilibrium. Excellent agreement is found with DSMC solutions for linear and weakly non-linear flows.

Thesis Supervisor: Nicolas G. Hadjiconstantinou
Title: Associate Professor of Mechanical Engineering

Acknowledgments

This research experience has been very enriching from an academic perspective and also beneficial to my development.

I would like to express my sincere appreciations to my supervisor, Pr. Nicolas G. Hadjiconstantinou for welcoming me in his research team and involving me in a very exciting project. I also thank him for his advice, guidance and continuous support during the carrying-out of this work. His extensive knowledge and creative thinking have been an invaluable help.

I am gratful to Lowell L. Baker for his detailed explanations and help with the formulation of the problem. The numerous in-depth discussions with him have been a major source of inspiration. I also thank him for giving me his DSMC code.

I would like to thank Husain Al-Mohssen for managing the computer cluster. I would not have been able to run my simulations without this indispensable tool.

Thanks to my fellow research students Lowell, Husain, Ghassan and Ho-Man from room 3-355A. They contributed to making the office a lively place.

Last but not least, I would like to thank my family for supporting me from France.

Contents

Chapter 1

Introduction

1.1 The Boltzmann Equation

When the system characteristic lengthscale L becomes of the order of or smaller than the molecular mean free path λ, the Navier-Stokes description fails. This is due to the fact that transport is no longer diffusive (collision dominated), but rather, ballistic effects become important. This situation is typically quantified by the Knudsen number kn, defined as $kn = \lambda/L$. In general [1, 2, 3], the Navier-Stokes description is no longer reliable for $kn \gtrsim 0.1$, a situation typically encoutered in flows in the upper atmosphere [3], but more recently in nanoscale flow environments. In these cases, a more general model valid for all Knudsen numbers must be used.

The Boltzmann equation [1, 2, 3] constitutes a possible approach. For a gas composed of identical hard sphere molecules the Boltzmann equation is given by:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla f} = \int\int (f'f'_1 - ff_1)|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d^2\Omega d^3\boldsymbol{v_1} \tag{1.1}$$

Here, $f(\boldsymbol{x}, \boldsymbol{v}, t)$ is the distribution function of the molecules in the phase space at time t, where the position in physical space is $\boldsymbol{x} = (x, y, z)$ and the molecular velocity vector is $\boldsymbol{v} = (v_x, v_y, v_z)$. The quantity $\sigma = d^2/4$ is the differential collision cross-section of the molecules of diameter d and mass m. Ω denotes the solid angle of the scattering and is integrated over the unit sphere. In the above equation, the following notation is also used: $f_1 = f(\boldsymbol{x}, \boldsymbol{v_1}, t)$, $f' = f(\boldsymbol{x}, \boldsymbol{v'}, t)$ and $f_1' = f(\boldsymbol{x}, \boldsymbol{v_1'}, t)$ where $\boldsymbol{v'}$ and $\boldsymbol{v_1'}$ are the postcollision velocities resulting from the collision of the pair $\boldsymbol{v}$ and $\boldsymbol{v_1}$ with scattering angle Ω.

For a homogeneous gas at equilibrium at a reference temperature T_0 and with a reference number density n_0, the distribution function is the Maxwellian $f_0(\boldsymbol{v}) = n_0 \pi^{-3/2} v_0^{-3} \exp[-(\boldsymbol{v}/v_0)^2]$. The most probable velocity is given by $v_0 = \sqrt{2k_B T_0/m}$ where k_B is the Boltzmann's constant. For this equilibrium distribution, the mean free path is equal to $\lambda = 1/(\sqrt{2}\pi n_0 d^2)$, and the corresponding molecular collision time is $\tau = \sqrt{\pi}\lambda/(2v_0)$.

In engineering, the design process of devices is increasingly carried out through the use of numerical simulation. The recent development of Micro-Electro-Mechanical Systems and the need for simulating their behaviors has led to increased interest in numerical solutions of the above Boltzmann equation. However, its nonlinear integro-differential stucture and the high-dimensionality of the distribution function make this equation very hard to simulate.

1.2 The DSMC method

One of the most popular methods for solving the Boltzmann equation is the Direct Simulation Monte Carlo (DSMC) [3]. DSMC solves the Boltzmann equation by simulating the motion of a representative set of particles, which can be thought of as sample of the distribution function f. As is typical with particle simulation methods, the intuitive formulation and easy implementation, the fact that it does not require velocity space discretization, and the low memory usage are the main advantages that triggered the success of DSMC.

DSMC solves the Boltzmann equation by using an operator-splitting technique, which amounts to integrating the advection and collision terms separately. In other words, the following equations are successively integrated over a period δt:

$$\begin{aligned} \frac{\partial f}{\partial t} &= -\boldsymbol{v} \cdot \boldsymbol{\nabla f} && (1.2) \\ \frac{\partial f}{\partial t} &= \int\int (f'f_1' - ff_1)|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega dv_1 && (1.3) \end{aligned}$$

The first equation is a free molecular advection. It is integrated by simply moving the particles according to their velocities while their velocities remain constant. The second equation is a spatially homogeneous relaxation governed by the Boltzmann collision operator. It can be solved by colliding a suitable number of pairs of particles, which will change their velocities while their positions remain constant.

The collision step of DSMC relies on the following form of the collision integral [2]:

$$\left[\frac{\partial f}{\partial t}\right]_{coll} = \frac{1}{2}\int\int\int (\delta_1' + \delta_2' - \delta_1 - \delta_2) f(\boldsymbol{v_1}) f(\boldsymbol{v_2}) |\boldsymbol{v_2} - \boldsymbol{v_1}| \sigma d^2\Omega d^3\boldsymbol{v_1} d^3\boldsymbol{v_2}$$

where $\delta_1 = \delta^3(\boldsymbol{v} - \boldsymbol{v_1})$, $\delta_2 = \delta^3(\boldsymbol{v} - \boldsymbol{v_2})$, $\delta_1' = \delta^3(\boldsymbol{v} - \boldsymbol{v_1'})$ and $\delta_2' = \delta^3(\boldsymbol{v} - \boldsymbol{v_2'})$, δ being the dirac distribution. Here, $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ are the precollision velocities and $\boldsymbol{v_1'}$ and $\boldsymbol{v_2'}$ are the corresponding postcollision velocities with scattering angle Ω. This form of the integral suggests that for a given pair of particles, processing the collision consists of deleting those two particles (terms $-\delta_1$ and $-\delta_2$), and creating a pair of particles with the postcollision velocities (terms $+\delta_1'$ and $+\delta_2'$); this is done in practice by simply updating the velocities of the particles. In order to account for the relative velocity factor $|\boldsymbol{v_2} - \boldsymbol{v_1}|$, collisions are processed with a probability proportional to this relative velocity by means of an acceptance/rejection technique.

Although very efficient for high speed flows, the computational cost of DSMC increases *sharply* [11] as the deviation from equilibrium decreases, making noise-free simulation of low-speed –or more generally low-signal– flows very expensive and in some cases intractable. In this latter case, most of the computational time is spent in computing collisions whose net effect is zero. This observation gave rise to a first improvement of DSMC, a variance-reduced DSMC or VRDSMC [5, 6, 7].

1.3 Variance-Reduced DSMC

In [4] Baker and Hadjiconstantinou presented a general variance reduction method which allows Monte Carlo solutions of the Boltzmann equation for low-signal flows. The basic idea amouts to splitting the distribution function into a Maxwellian distribution f_{mb} and a deviationnal distribution f_d:

$$f = f_{mb} + f_d \tag{1.4}$$

The distribution function is then known through f_d. In a particle method such as VRDSMC, f_d can be represented by a set of numerical particles, called deviational particles. Therefore, one difference from DSMC is that now the particles can be either positive or negative depending on whether f_{mb} accounts for too many or too few particles at a given location in phase space. The above authors showed that due to the fact that the Maxwellian is an equilibrium function, meaning that the collision integral is zero when $f = f_{mb}$, the collision operator reduces to:

$$\left[\frac{\partial f}{\partial t}\right]_{coll} = \frac{1}{2}\int\int\int (\delta_1' + \delta_2' - \delta_1 - \delta_2)(2f_{mb1}f_{d2} + f_{d1}f_{d2})|\boldsymbol{v_2} - \boldsymbol{v_1}|\sigma d^2\Omega d^3\boldsymbol{v_1}d^3\boldsymbol{v_2}$$

According to the above form of the collision integral [5, 6] two kinds of collisions need to be considered: those between the underlying Maxwellian (f_{mb}) particles and the deviational (f_d) particles on the one hand, and those between deviational particles one the other hand. The first ones are carried out by sampling an actual deviational particle and drawing a velocity from the Maxwellian distribution, while the second ones are carried out by sampling two actual deviational particles. Unfortunately, contrary to standard DSMC, the factor $\delta_1' + \delta_2' - \delta_1 - \delta_2$ does not amount to updating the velocities of the numerical particles but requires the creation of new deviational particles. Let us illustrate this fact by an example of the first kind of collision. We see that particle 1 drawn from f_{mb} collides with particle 2 drawn from f_d and leads to the creation of 4 particles: $sign(f_{d2})\delta_1'$, $sign(f_{d2})\delta_2'$, $-sign(f_{d2})\delta_1$ and $-sign(f_{d2})\delta_2$. Only the existing particle 2 ($sign(f_{d2})\delta_2$) is cancelled by the new $-sign(f_{d2})\delta_2$ particle, thus leading to the net creation of 3 particles.

This sheme has been found [5, 6] to be very efficient for $kn \gtrsim 1$, where the flow is not collision dominated, and deviational particles are mostly cancelled through collisions with the system walls. For $kn \lesssim 1$, the high rate of intermolecular collisions leads to high net rate of particle creation that results in a divergence in the number of particles [5, 6], unless a particle cancellation scheme is introduced. Such a scheme was shown [5, 6] to be capable of stabilizing the calculation. Unfortunately, it has the disadvantage of requiring a velocity space discretization and leads to high memory usage and high computational cost.

1.4 The Proposed Method

The purpose of the present thesis is to derive an alternative, *rigorous* method for treating the collision operator which removes the necessity for a cancellation routine by avoiding the creation of a large fraction of the deviational particles. We thus keep the capability of simulating low-signal flows while recovering most of the advantage of a standard particle method, namely the absence of velocity space discretization and a low memory usage. The proposed method relies on two basic ideas:

* The use of a mathematical formulation of the collision operator involving convolutions enables us to calculate the net aggregated effect of all collisions between deviational particles and underlying Maxwellian particles and thus, in some sense, analytically cancel the particles before creating them.

* This net action of the collision operator is not only supported by a change of the deviational distribution f_d through the creation of deviational particles as in

previous works, but also, and for a large part, is absorbed through a modification of the local Maxwellian distribution. Instead of keeping this distribution constant over time and space as it was the case in previous works, we can indeed make its intrinsic parameters (number density n_{mb}, mean velocity $\boldsymbol{u_{mb}}$, and most probable velocity v_{mb}) evolve.

Since the advection and collision parts are independent, we will derive and present their corresponding algorithms in separate chapters. The major change from standard DSMC lies in the collision algorithm, since the method was designed to improve this very part. The advection algorithm is an adaptation of the standard advection algorithm (moving the particles) which account for a nonconstant underlying Maxwellian distribution [5, 6, 7].

Chapter 2

Proposed Treatment of the Boltzmann Collision Operator

2.1 Preliminaries

Let $C(f,g)$ refer to the following general collision operator:

$$C(f,g)(\boldsymbol{v}) = \int\int (f'g_1' - fg_1)|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1} \tag{2.1}$$

For hard sphere molecules, the cross-section is constant and equal to $\sigma = d^2/4$, d being the diameter of the molecules. The collision term in the right hand side of the Boltzmann equation is given by:

$$\left[\frac{\partial f}{\partial t}\right]_{coll} = C(f,f)$$

When the distribution f is written as $f = f_{mb} + f_d$, where f_{mb} is a Maxwellian, the bilinearity of C enables us to write:

$$C(f_{mb} + f_d, f_{mb} + f_d) = C(f_{mb}, f_{mb}) + C(f_{mb}, f_d) + C(f_d, f_{mb}) + C(f_d, f_d)$$

It is known that the Maxwellian being an equilibrium distribution $C(f_{mb}, f_{mb}) = 0$. The collision operator is then made of a linear part $C(f_{mb}, f_d) + C(f_d, f_{mb})$, and a nonlinear (quadratic) part $C(f_d, f_d)$.

Coming back to the definition of C (*Equation* 2.1), the linear part of this operator is:

$$C(f_{mb}, f_d) + C(f_d, f_{mb}) = \int\int (f'_{mb}f'_{d1} - f_{mb}f_{d1} + f'_d f'_{mb1} - f_d f_{mb1})|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1}$$

It can be split into three terms as follows:

$$\begin{aligned} C(f_{mb}, f_d) + C(f_d, f_{mb}) &= \int\int (f'_{mb}f'_{d1} + f'_d f'_{mb1})|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1} - \\ &\int\int f_{mb}f_{d1}|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1} - \\ &\int\int f_d f_{mb1}|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1} \end{aligned}$$

Those three terms can be written explicitly as a two convolutions and a product of functions, when f_{mb} is some Maxwellian with mean velocity $\boldsymbol{u_{mb}}$, number density n_{mb} and most probable velocity v_{mb}. The detailed derivation can be found in Appendix A. The result is:

$$\begin{aligned} \int\int (f'_{mb}f'_{d1} + f'_d f'_{mb1})|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1} &= \int K_1(\boldsymbol{w}, \boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \\ \int\int f_{mb}f_{d1}|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1} &= \int K_2(\boldsymbol{w}, \boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \\ \int\int f_d f_{mb1}|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1} &= -f_d(\boldsymbol{v})\nu(\boldsymbol{w}) \end{aligned}$$

where the kernels are given by $K_i(\boldsymbol{v}, \boldsymbol{v_1}) = \mu_1 K_i^*(\frac{\boldsymbol{v}}{v_{mb}}, \frac{\boldsymbol{v_1}}{v_{mb}})$, $i = 1$ or 2, with:

$$K_1^*(\boldsymbol{v}^*, \boldsymbol{v_1}^*) = \frac{2}{|\boldsymbol{v}^* - \boldsymbol{v_1}^*|} \exp\left[-\frac{(\boldsymbol{v}^* \cdot (\boldsymbol{v}^* - \boldsymbol{v_1}^*))^2}{|\boldsymbol{v}^* - \boldsymbol{v_1}^*|^2}\right] \qquad (2.2)$$

$$K_2^*(\boldsymbol{v}^*, \boldsymbol{v_1}^*) = |\boldsymbol{v}^* - \boldsymbol{v_1}^*| \exp[-\boldsymbol{v}^{*2}] \qquad (2.3)$$

and where the function ν, which is the collision frequency of a deviational particle with the entire Maxwellian versus its velocity $\boldsymbol{v}$, is given by $\nu(\boldsymbol{v}) = \mu_2 \nu^*(\frac{\boldsymbol{v}}{v_{mb}})$ where

ν^* is given by:

$$\nu^*(\boldsymbol{v}^*) = \frac{\pi}{2|\boldsymbol{v}^*|}\left[\pi^{1/2}\mathrm{erf}(|\boldsymbol{v}^*|) + 2|\boldsymbol{v}^*|\exp(-|\boldsymbol{v}^*|^2) + \pi^{1/2}2\boldsymbol{v}^{*2}\mathrm{erf}(|\boldsymbol{v}^*|)\right] \tag{2.4}$$

The prefactors are:

$$\mu_1 = \frac{d^2 n_{mb}}{\pi^{1/2} v_{mb}^2}, \qquad \mu_2 = v_{mb}^3 \mu_1 = \frac{d^2 n_{mb} v_{mb}}{\pi^{1/2}} \tag{2.5}$$

Here, we used the notation $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{v}) = \boldsymbol{v} - \boldsymbol{u_{mb}}$ and $\boldsymbol{v^*} = \boldsymbol{v}^*(\boldsymbol{v}) = \boldsymbol{v}/v_{mb}$, where $\boldsymbol{u_{mb}}$ and v_{mb} are the mean velocity and the characteristic velocity of the local Maxwellian.

Finally, the whole collision term can be written as:

$$\left[\frac{\partial f}{\partial t}\right]_{coll} = \int K_1(\boldsymbol{w}, \boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} - \int K_2(\boldsymbol{w}, \boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} - f_d(\boldsymbol{v})\nu(\boldsymbol{w}) + C(f_d, f_d) \tag{2.6}$$

2.2 Discussion

If we neglect the quadratic term $C(f_d, f_d)$ (the collisions of the deviational part with itself), we are left with a linear operator. In this case, K_1 can be interpreted as the gain term of both the f_{mb} and the f_d parts of f, K_2 as the loss term of the f_{mb} part and $f_d(\boldsymbol{v})\nu(\boldsymbol{w})$ as the loss term of the f_d part. In other words, these operators correspond to the aggregated effect of all collisions of a given deviational particle with the entire Maxwellian. More precisely, for a given velocity $\boldsymbol{v_1}$, the function $\boldsymbol{v} \longrightarrow K_1(\boldsymbol{w}, \boldsymbol{w_1})$ is the distribution of the gain rate, corresponding to the postcollision velocities of the deviational and Maxwellian particles, $\boldsymbol{v} \longrightarrow K_2(\boldsymbol{w}, \boldsymbol{w_1})$ is the distribution of the loss rate, corresponding to the precollision velocities of the Maxwellian particles, and $\nu(\boldsymbol{w_1})$ is the collision (loss) rate of the deviational particles at velocity $\boldsymbol{v_1}$.

From now on, $\tilde{K}_1$, $\tilde{K}_2$ and $\tilde{\nu}$ will refer to the operators corresponding to the kernels of the same name. In short, $\tilde{K}_i f(\boldsymbol{v}) = \int K_i(\boldsymbol{w}, \boldsymbol{w_1}) f(\boldsymbol{v_1}) d^3\boldsymbol{v_1}$ and $\tilde{\nu} f(\boldsymbol{v}) = \nu(\boldsymbol{w}) f(\boldsymbol{v})$

The collision algorithm of the proposed method will be based on the above formulation (*Equation* 2.6).

2.3 Proposed Treatment

In previous variance reduction algorithm [5, 6], the collision part was done by changing only the deviational distribution, while the Maxwellian remained constant through all the calculation. Now, we consider changing the local Maxwellian in order to absorb part of the deviational particles generated by the action of the collision operator. This means that,

$$\delta t \left[\frac{\partial f}{\partial t}\right]_{coll} = \delta f_{mb} + \delta f_d$$

The effect of changing the parameters of the local Maxwellian will be investigated in detail later. However, we can already formally write that if we change its total number density n_{mb} by δn_{mb}, its mean velocity $\boldsymbol{u_{mb}}$ by $\delta \boldsymbol{u_{mb}}$ and its characteristic velocity v_{mb} by δv_{mb}, we are in effect changing the distribution function by

$$\delta f_{mb} = \left(\frac{\partial f_{mb}}{\partial n_{mb}}\right) \delta n_{mb} + \left(\frac{\partial f_{mb}}{\partial v_{mb}}\right) \delta v_{mb} + \left(\frac{\partial f_{mb}}{\partial \boldsymbol{u_{mb}}}\right) \cdot \delta \boldsymbol{u_{mb}}$$

The challenge will then consist of finding δf_{mb} so as to make δf_d as small as possible. This is achieved by rewritting the collision operator $\left[\frac{\partial f}{\partial t}\right]_{coll} = [\tilde{K}_1 - \tilde{K}_2] f_d - \tilde{\nu} f_d + C(f_d, f_d)$ as:

$$\delta t\left[\frac{\partial f}{\partial t}\right]_{coll} = \underbrace{\delta t\left[\tilde{K}_1 - \tilde{K}_2\right] f_d - \delta f_{mb} +}_{\text{generation of particles}} \quad \underbrace{\delta f_{mb}}_{\substack{\text{changing of} \\ \text{the Maxwellian}}} \quad - \quad \underbrace{\tilde{\nu} f_d \delta t}_{\text{deletion of particles}} \quad + \underbrace{C(f_d, f_d)\delta t}_{\text{collision of particles}}$$

As can be seen above, $\delta t\left[\frac{\partial f}{\partial t}\right]_{coll}$ is the sum of four terms which can be treated independently of each other in four distinct routines. Three of these terms act directly on the distribution function of the deviational particles. The fourth one acts on the equilibrium distribution. We briefly outline their structure below.

* The term $\delta t\left[\tilde{K}_1 - \tilde{K}_2\right] f_d - \delta f_{mb}$ is a function of $\boldsymbol{v}$, in other words, it is distribution. As a consequence, we can naturally generate some particles drawn from this function. The only difficulty is that we do not know explicitly its functional form. The evaluation of $\left(\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}\right)(\boldsymbol{v})$ will require some effort. The essence of the algorithm lies in this very term, since we will try to find a δf_{mb} so as to generate as few particles as possible. In addition, the particles that we will generate will be 'precancelled', in the sense that we will not create a positive and a negative particle at the same velocity and position which could then be cancelled.

* The term δf_{mb} corresponds to the shifting of the Maxwellian. In practice, it is just a matter of calculating n_{mb}, $\boldsymbol{u_{mb}}$ and v_{mb}. The method for determining them will be addressed below.

* Due to its structure, the term $-\tilde{\nu} f_d \delta t$ can be implemented by a deletion of some numerical particles with a probability proportional to $\tilde{\nu}$. In addition, it contributes to the efficiency of the algorithm since it reduces the number of numerical particles.

* The quadratic term $C(f_d, f_d)\delta t$ can be implemented by colliding some numerical particles, as in standard DSMC. The difference is that the deviational distribution function f_d may be negative, which means that we may have negative particles, as

explained in the Introduction. For computations near to equilibrium, the contribution from this term will be negligible. As a consequence, we will neglect it in the linear version of the algorithm. In contrast, in highly nonlinear situation, this term will be important.

2.4 Determining δf_{mb}

The natural question that we now have to answer is what to choose for δf_{mb}. As stated above, we want to choose δf_{mb} in such a way that the term $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}$ be as small as possible, since it corresponds to generation of particles. If the function $[\tilde{K}_1 - \tilde{K}_2]f_d$ could be written as a combination of the derivatives of a Maxwellian with respect to n_{mb}, v_{mb} and $\boldsymbol{u_{mb}}$, we could make $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}$ be zero and avoid generating particles. But in general, the function $[\tilde{K}_1 - \tilde{K}_2]f_d$ is not in the span of the derivatives of a Maxwellian. An obvious counterexample is that δf_{mb} is always a continuous function wheras $[\tilde{K}_1 - \tilde{K}_2]f_d$ can be discontinuous. The space of functions accessible to δf_{mb} is thus limited, but we will see that we can still reduce significantly the number of particles that we need to generate, $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}$ becoming just a small correction.

A first choice was to find δn_{mb}, δv_{mb} and $\delta \boldsymbol{u_{mb}}$ that minimize the L^2-norm of $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}$. This solution leads to an intractable set of equations. Another solution is to choose them so as to make $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}$ have no net mass, no net momentum and no net energy. In other words, those parameters are chosen so that the first three moments of $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}$ be zero. The span of δf_{mb} through the choice of δn_{mb}, δv_{mb} and $\delta \boldsymbol{u_{mb}}$ has indeed enough degrees of freedom to allow us to choose them so that δf_{mb} absorb the first three moments of $\left[\frac{\partial f}{\partial t}\right]_{coll}$ resulting in $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}$ having its first three moments equal to zero.

The reasoning behind this choice is the following. Let us consider a homogenious relaxation problem, described by *Equation* 1.3 . We know that if we start from any distribution, the final distribution will be a Maxwellian, the equilibrium distribution. On the other hand, in the proposed algorithm, we see that we delete particles through the term $\tilde{\nu} f_d$ and generate particles drawn from the "moment-free" distribution $[\tilde{K_1} - \tilde{K_2}]f_d - \delta f_{mb}$. This means that the moments of the deviational part f_d will decrease and ultimately go to zero. The final distribution will then be $f = f_{mb} + f_d$ where f_d is a moment-free distribution and f the final equilibrium Maxwellian. It is then easy to show that $f_d = f - f_{mb}$ being a difference of two Maxwellians and having no net mass, no net momentum and no net energy, it is necessarily equal to zero and $f_{mb} = f$. In other words, we can hope that in a general multidimensional problem, this method will make f_{mb} go to a Maxwellian near to the local one and make f_d be small.

Chapter 3

Implementation of the New Collision Operator Treatment

In the following sections, we will investigate in detail how we perform each of the four stages described in the previous chapter. Let us introduce now a few notations and concepts that we will use in these following four sections when deriving the various algorithms.

First of all, performing the Boltzmann collision operator and also computing the outputs requires a physical space discretization. In fact, two particles should ideally be allowed to collide together only if their positions are identical. However, since we have a finite number of particles and the position is a continuous parameter, this condition has a zero probability of occuring. DSMC handles this difficulty by splitting the physical domain into cells and allowing particles to collide with each other provided that they are in the same cell. The proposed method uses the same technique. The volume of a cell will be refered to as V_{cell}, and the corresponding physical domain as $\mathcal{C}$.

In compliance with the splitting of operator technique, the collision part can be treated independently in each cell, as if we had an independent homogenious relaxation in each cell for a period of δt. From now on, f_{mb} or "the local Maxwellian" will refer to the f_{mb} of a given cell which will itself be refered to as the "current cell".

In the current cell, the deviational distribution f_d is respresented by a set of N_d particles with velocity $\boldsymbol{v^i}$, position $\boldsymbol{x^i}$ and sign $s^i \in \{-1, +1\}$. The superscript i is the index of the particles. The set of numerical deviational particles that reside in the current cell will be refered to as $\mathbf{C}$. In mathematical terms, we have $\mathbf{C} = \left\{ i | \boldsymbol{x^i} \in \mathcal{C} \right\}$.

The numerical deviational distribution in the current cell is then given by:

$$f_d(\boldsymbol{x}, \boldsymbol{v}) = \sum_{i \in \mathbf{C}} N_{eff} s^i \delta(\boldsymbol{v} - \boldsymbol{v^i}) \delta(\boldsymbol{x} - \boldsymbol{x^i}) \tag{3.1}$$

where N_{eff} is the effective number of molecules that a numerical particle represents and δ is the dirac distribution. A particle with velocity $\boldsymbol{v^i}$, position $\boldsymbol{x^i}$ and sign s^i acts like a distribution $N_{eff} s^i \delta(\boldsymbol{v} - \boldsymbol{v^i}) \delta(\boldsymbol{x} - \boldsymbol{x^i})$.

Following these considerations and the results from the previous chapter, the general outline of the collision algorithm is:

```
Do separately for all cells

  * Change the Maxwellian by the appropriate amount to account for $\delta f_{mb}$

  * Delete numerical particles with probability proportional to $\nu(\boldsymbol{w})\delta t$

  * Generate particles drawn from $\delta t \left[ \int K_1 f_d - \int K_2 f_d \right] - \delta f_{mb}$

  * Perform hard sphere collisions to account for the quadratic term $C(f_d, f_d)$

End Do
```

3.1 Changing the Maxwellian

The purpose of this section is to derive the analytical expressions of δn_{mb}, δv_{mb} and $\delta \boldsymbol{u_{mb}}$ that make the function $\delta t[\tilde{K_1} - \tilde{K_2}]f_d - \delta f_{mb}$ have no net mass, no net momentum and no net energy, and propose an algorithm to compute them in practice. In essence, we first compute the moments of δf_{mb} as functions of δn_{mb}, δv_{mb} and $\delta \boldsymbol{u_{mb}}$, then compute the moments of $[\tilde{K_1} - \tilde{K_2}]f_d$ as a function of f_d, and solve for the values of δn_{mb}, δv_{mb} and $\delta \boldsymbol{u_{mb}}$ that make the two sets of moments equal.

3.1.1 Changing a Maxwellian distribution

A Maxwellian is a function of $\boldsymbol{v}$ which depends on 3 parameters, namely the mean velocity $\boldsymbol{u_{mb}}$, the number density n_{mb}, and the characteristic velocity v_{mb}. In other words,

$$f_{mb}(\boldsymbol{v}, n_{mb}, v_{mb}, \boldsymbol{u_{mb}}) = \frac{n_{mb}}{\pi^{3/2} v_{mb}^3} \exp\left[\frac{-(\boldsymbol{v} - \boldsymbol{u_{mb}})^2}{v_{mb}^2}\right]$$

A small change of n_{mb} by δn_{mb}, $\boldsymbol{u_{mb}}$ by $\delta \boldsymbol{u_{mb}}$ and v_{mb} by δv_{mb} results in changing the distribution by:

$$\delta f_{mb} = \frac{\partial f_{mb}}{\partial n_{mb}} \delta n_{mb} + \frac{\partial f_{mb}}{\partial v_{mb}} \delta v_{mb} + \frac{\partial f_{mb}}{\partial \boldsymbol{u_{mb}}} \cdot \delta \boldsymbol{u_{mb}} \tag{3.2}$$

where we have, by differentiating the above expression of the Maxwellian:

$$\frac{\partial f_{mb}}{\partial n_{mb}} = \frac{f_{mb}}{n_{mb}} \tag{3.3}$$

$$\frac{\partial f_{mb}}{\partial \boldsymbol{u}} = 2\frac{\boldsymbol{v} - \boldsymbol{u_{mb}}}{v_{mb}^2} f_{mb} \tag{3.4}$$

$$\frac{\partial f_{mb}}{\partial v_{mb}} = \frac{1}{v_{mb}}\left(2\frac{(\boldsymbol{v} - \boldsymbol{u_{mb}})^2}{v_{mb}^2} - 3\right) f_{mb} \tag{3.5}$$

Each derivative is calculated while the two other parameters remain constant. So for example, the derivative with respect to v_{mb} is done at constant number density. This means that $(\partial f_{mb}/\partial n_{mb})$ has the same mean velocity and standard deviation as f_{mb}, $(\partial f_{mb}/\partial \boldsymbol{u_{mb}})$ has no net mass and no net energy, and $(\partial f_{mb}/\partial v_{mb})$ has no net mass and the same mean velocity as f_{mb}.

Denoting $\boldsymbol{w}^* = (\boldsymbol{v} - \boldsymbol{u_{mb}})/v_{mb}$, recalling that $f_{mb}(\boldsymbol{v}) = \pi^{-3/2} v_{mb}^{-3} \exp[-\boldsymbol{w^{*2}}]$, and inserting the expressions of the derivatives given by *Equations* 3.3, 3.4 and 3.5 into *Equation* 3.2 we get:

$$\delta f_{mb}(\boldsymbol{v}) = \left[\delta n_{mb} + n_{mb} v_{mb}^{-1}(2\boldsymbol{w^{*}}^{2} - 3)\delta v_{mb} + 2 n_{mb} v_{mb}^{-1} \boldsymbol{w^*} \cdot \delta \boldsymbol{u_{mb}}\right] v_{mb}^{-3} \pi^{-3/2} \exp(-\boldsymbol{w^{*2}}) \tag{3.6}$$

This is a polynomial times a Maxwellian. We can compute explicitly the first three moments of this distribution as a function of δn_{mb}, δv_{mb} and $\boldsymbol{u_{mb}}$. We make here the arbitrary choice of computing all the moments in the frame moving at velocity $\boldsymbol{u_{mb}}$. This convention has no impact on the final expression provided that we do not change it and remain consistent. The moments of δf_{mb} are:

$$\int \delta f_{mb}(\boldsymbol{v}) d^3\boldsymbol{v} = \delta n_{mb} \tag{3.7}$$

$$\int \boldsymbol{w} \delta f_{mb}(\boldsymbol{v}) d^3\boldsymbol{v} = n_{mb} \delta \boldsymbol{u_{mb}} \tag{3.8}$$

$$\int |\boldsymbol{w}|^2 \delta f_{mb}(\boldsymbol{v}) d^3\boldsymbol{v} = \frac{3}{2} v_{mb}^2 \delta n_{mb} + 3 n_{mb} v_{mb} \delta v_{mb} \tag{3.9}$$

3.1.2 Moments of $[\tilde{K}_1 - \tilde{K}_2] f_d$

The purpose of this section is to derive a general expression of the moments of the distribution $[\tilde{K}_1 - \tilde{K}_2] f_d$ as a function of the distribution f_d.

Let K_i denote K_1 or K_2 and $M(\boldsymbol{w})$ be either 1, $\boldsymbol{w_1}$, or $|\boldsymbol{w_1}^2|$. The moments of $\tilde{K}_i f_d$ are $\int M(\boldsymbol{w}) \tilde{K}_i f_d(\boldsymbol{v}) d^3\boldsymbol{v}$, where we recall that $\tilde{K}_i f_d(\boldsymbol{v}) = \int K_i(\boldsymbol{w}, \boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1}$. Combining these two formulae leads to:

$$\int M(\boldsymbol{w}) \tilde{K}_i f_d(\boldsymbol{w}) d^3\boldsymbol{w} = \int M(\boldsymbol{w}) \left(\int K_i(\boldsymbol{w}, \boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \right) d^3\boldsymbol{w}$$

Performing the change of variable $\boldsymbol{v} \to \boldsymbol{w}$, and switching the order of the integrals, we can express it as:

$$\int M(\boldsymbol{w})\tilde{K}_i f_d(\boldsymbol{w})d^3\boldsymbol{w} = \int f_d(\boldsymbol{v_1})\left(\int M(\boldsymbol{w})K_i(\boldsymbol{w},\boldsymbol{w_1})d^3\boldsymbol{w}\right)d^3\boldsymbol{v_1} \tag{3.10}$$

We see that if we can compute the moments of the kernels, meaning the functions $\boldsymbol{w_1} \longrightarrow \int M(\boldsymbol{w})K_i(\boldsymbol{w},\boldsymbol{w_1})d^3\boldsymbol{w}$ the moments of the distribution $\tilde{K}_i f_d$ will be obtained by integrating the product of the function $\boldsymbol{w_1} \longrightarrow \int M(\boldsymbol{w})K_i(\boldsymbol{w},\boldsymbol{w_1})d^3\boldsymbol{w}$ with f_d.

Computing the first three moments of the kernels can be done explicitly. A summary of the algebra is given in Appendix A. We obtain:

$$\begin{aligned}
\int [K_1(\boldsymbol{w},\boldsymbol{w_1}) - K_2(\boldsymbol{w},\boldsymbol{w_1})]d^3\boldsymbol{w} &= \nu(\boldsymbol{w_1})\\
\int \boldsymbol{w}[K_1(\boldsymbol{w},\boldsymbol{w_1}) - K_2(\boldsymbol{w},\boldsymbol{w_1})]d^3\boldsymbol{w} &= \boldsymbol{w_1}\nu(\boldsymbol{w_1})\\
\int |\boldsymbol{w}|^2[K_1(\boldsymbol{w},\boldsymbol{w_1}) - K_2(\boldsymbol{w},\boldsymbol{w_1})]d^3\boldsymbol{w} &= |\boldsymbol{w_1}|^2\nu(\boldsymbol{w_1})
\end{aligned}$$

We could have expected these results since we know that the collision operator conserves mass, momentum and kinetic energy.

Inserting these latter expressions in the general formula for the moments of $[\tilde{K}_1 - \tilde{K}_2]f_d$ given by *Equation* 3.10 leads finally to:

$$\int [\tilde{K}_1 f_d(\boldsymbol{v}) - \tilde{K}_2 f_d(\boldsymbol{v})]d^3\boldsymbol{v} = \int \nu(\boldsymbol{w_1})f_d(\boldsymbol{v_1})d^3\boldsymbol{v_1} \tag{3.11}$$

$$\int \boldsymbol{w}[\tilde{K}_1 f_d(\boldsymbol{v}) - \tilde{K}_2 f_d(\boldsymbol{v})]d^3\boldsymbol{v} = \int \boldsymbol{w_1}\nu(\boldsymbol{w_1})f_d(\boldsymbol{v_1})d^3\boldsymbol{v_1} \tag{3.12}$$

$$\int |\boldsymbol{w}|^2[\tilde{K}_1 f_d(\boldsymbol{v}) - \tilde{K}_2 f_d(\boldsymbol{v})]d^3\boldsymbol{v} = \int |\boldsymbol{w_1}|^2\nu(\boldsymbol{w_1})f_d(\boldsymbol{v_1})d^3\boldsymbol{v_1} \tag{3.13}$$

3.1.3 Expressions for δn_{mb}, δv_{mb} and $\delta \boldsymbol{u_{mb}}$

To make the moments of $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d - \delta f_{mb}$ be zero, we must make the moments of $\delta t[\tilde{K}_1 - \tilde{K}_2]f_d$ given by *Equations* 3.11, 3.12 and 3.13 equal to those of δf_{mb} given by *Equations* 3.7, 3.8 and 3.9. This leads to the following system in δn_{mb}, δv_{mb} and $\delta \boldsymbol{u_{mb}}$:

$$\begin{aligned}
\delta n_{mb} &= \delta t \int \nu(\boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \\
n_{mb}\delta \boldsymbol{u} &= \delta t \int \boldsymbol{w_1}\nu(\boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \\
\frac{3}{2}v_{mb}^2\delta n_{mb} + 3n_{mb}v_{mb}\delta v_{mb} &= \delta t \int |\boldsymbol{w_1}|^2\nu(\boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1}
\end{aligned}$$

The solution is:

$$\begin{aligned}
\delta n_{mb} &= \delta t \int \nu(\boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \\
\delta \boldsymbol{u_{mb}} &= \frac{\delta t}{n_{mb}} \int \boldsymbol{w_1}\nu(\boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \\
\delta v_{mb} &= \frac{\delta t}{3n_{mb}v_{mb}} \int \left(|\boldsymbol{w_1}|^2 - \frac{3}{2}v_{mb}^2\right) \nu(\boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1}
\end{aligned}$$

or more conveniently for implementation purposes:

$$\delta n_{mb} = \delta t\mu_2 \int \nu^*(\boldsymbol{w_1^*}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \tag{3.14}$$

$$\delta \boldsymbol{u_{mb}} = \frac{\delta t\mu_2 v_{mb}}{n_{mb}} \int \boldsymbol{w_1^*}\nu^*(\boldsymbol{w_1^*}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \tag{3.15}$$

$$\delta v_{mb} = \frac{\delta t\mu_2 v_{mb}}{3n_{mb}} \int \left(|\boldsymbol{w_1^*}|^2 - \frac{3}{2}\right) \nu^*(\boldsymbol{w_1^*}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \tag{3.16}$$

During the simulation we need to compute δn_{mb}, $\delta \boldsymbol{u_{mb}}$ and δv_{mb} from the set of numerical particles. To this effect, we will work with cumulative distribution in the cell, that is to say with the distribution integrated over the current cell. From the expression of $f_d(\boldsymbol{x}, \boldsymbol{v})$ given by *Equation* 3.1, we get:

$$\int_{x\in\mathcal{C}} f_d(\boldsymbol{x}, \boldsymbol{v}) d^3\boldsymbol{x} = \sum_{i\in\mathbf{C}} N_{eff} s^i \delta(\boldsymbol{v} - \boldsymbol{v^i}) \tag{3.17}$$

and

$$\int_{x\in\mathcal{C}} \delta f_{mb}(\boldsymbol{v}) d^3\boldsymbol{x} = V_{cell}\delta f_{mb}(\boldsymbol{v})$$

since the local Maxwellian is constant over the cell.

By integrating over the cell *Equations* 3.14, 3.15 and 3.16 (the above expressions of δn_{mb}, $\delta \boldsymbol{u_{mb}}$ and δv_{mb}), and replacing $\int_{x\in C} f_d d^3\boldsymbol{x}$ by its expression given in *Equation* 3.17 (in terms of a sum of dirac distributions), we get:

$$\begin{aligned}
\delta n_{mb} V_{cell} &= N_{eff}\delta t \mu_2 \int \nu^*(\boldsymbol{w_1}^*) \sum_{i\in \mathbf{C}} s^i \delta(\boldsymbol{v_1} - \boldsymbol{v^i}) d^3\boldsymbol{v_1} \\
\delta \boldsymbol{u_{mb}} V_{cell} &= N_{eff}\frac{\delta t \mu_2 v_{mb}}{n_{mb}} \int \boldsymbol{w_1^*} \nu^*(\boldsymbol{w_1^*}) \sum_{i\in \mathbf{C}} s^i \delta(\boldsymbol{v_1} - \boldsymbol{v^i}) d^3\boldsymbol{v_1} \\
\delta v_{mb} V_{cell} &= N_{eff}\frac{\delta t \mu_2 v_{mb}}{3 n_{mb}} \int \left(|\boldsymbol{w_1^*}|^2 - \frac{3}{2} \right) \nu^*(\boldsymbol{w_1^*}) \sum_{i\in \mathbf{C}} s^i \delta(\boldsymbol{v_1} - \boldsymbol{v^i}) d^3\boldsymbol{v_1}
\end{aligned}$$

After switching the sum and integral sign, we obtain:

$$\begin{aligned}
\delta n_{mb} &= \frac{N_{eff}}{V_{cell}} \delta t \mu_2 \sum_{i\in \mathbf{C}} s^i \nu^*(\boldsymbol{w^{i*}}) \\
\delta \boldsymbol{u_{mb}} &= \frac{N_{eff}}{V_{cell}} \frac{\delta t \mu_2 v_{mb}}{n_{mb}} \sum_{i\in \mathbf{C}} s^i \boldsymbol{w^{i*}} \nu^*(\boldsymbol{w^{i*}}) \\
\delta v_{mb} &= \frac{N_{eff}}{V_{cell}} \frac{\delta t \mu_2 v_{mb}}{3 n_{mb}} \sum_{i\in \mathbf{C}} s^i \left(|\boldsymbol{w^{i*}}|^2 - \frac{3}{2} \right) \nu^*(\boldsymbol{w^{i*}})
\end{aligned}$$

where we recall that $\boldsymbol{w^{i*}} = (\boldsymbol{v^i} - \boldsymbol{u_{mb}})/v_{mb}$ and $\boldsymbol{v^i}$ is the velocity and s^i the sign of the i^{th} particle.

3.1.4 Sketch of algorithm for changing the local Maxwellian

From the above formulae, the algorithm for shifting the Maxwellian comes naturally. In the scheme, we compute the values of δn_{mb}, $\delta \boldsymbol{u_{mb}}$ and δv_{mb} for the current cell and the current timestep, and change the Maxwellian of the current cell.

```
Set $\delta n_{mb} = 0$, $\delta v_{mb} = 0$ and $\delta \boldsymbol{u_{mb}} = \mathbf{0}$

For all particles in the cell:

    get the particle velocity $\boldsymbol{v_1}$ and sign $s$.
    Compute: $\boldsymbol{w_1^*} = (\boldsymbol{v_1} - \boldsymbol{u_{mb}})/v_{mb}$

    $\delta n_{mb} = \delta n_{mb} + \delta t \mu_2 \frac{N_{eff}}{V_{cell}} s \nu^*(\boldsymbol{w_1^*})$
    $\delta \boldsymbol{u_{mb}} = \delta \boldsymbol{u_{mb}} + \delta t \mu_2 \frac{N_{eff}}{V_{cell}} \frac{v_{mb}}{n_{mb}} s \boldsymbol{w_1^*} \nu^*(\boldsymbol{w_1^*})$
    $\delta v_{mb} = \delta v_{mb} + \delta t \mu_2 \frac{N_{eff}}{V_{cell}} \frac{v_{mb}}{n_{mb}} s \frac{1}{3} \left(|\boldsymbol{w_1}^*|^2 - \frac{3}{2}\right) \nu^*(\boldsymbol{w_1}^*)$

End For

$n_{mb} = n_{mb} + \delta n_{mb}$
$\boldsymbol{u_{mb}} = \boldsymbol{u_{mb}} + \delta \boldsymbol{u_{mb}}$
$v_{mb} = v_{mb} + \delta v_{mb}$
```

The values of δn_{mb}, $\delta \boldsymbol{u_{mb}}$ and δv_{mb}, are also useful as a characterization of δf_{mb}, which we will need for the generation of particles.

3.2 Deletion of particles

In this part, we treat the term $-\nu(\boldsymbol{w}) f_d(\boldsymbol{v}) \delta t$.

$\nu(\boldsymbol{w})\delta t$ is the number of collisions of a particle at velocity $\boldsymbol{v} = \boldsymbol{u_{mb}} + \boldsymbol{w}$ during δt. If the timestep δt is small enough, this number of collisions will be much smaller than 1.

This enables us to consider deleting the particle with a probability $\nu(\boldsymbol{w})\delta t$.

Among a large number of particles around the velocity $\boldsymbol{w}$, we will have the ratio $\nu(\boldsymbol{w})\delta t$ of those particles that will be deleted which corresponds to changing the distribution function by $-\nu(\boldsymbol{w})f_d(\boldsymbol{v})\delta t$.

The particles need not, strictly speaking, be deleted at this stage. In our implementation, they are marked for deletion, and are actually deleted at the very end of the collision routine. We indeed want to compute all the parts of the collision term based on the same f_d distribution.

In summary, the scheme for the deletion of particles is:

```
For all particles in the cell:

    get its velocity $\boldsymbol{v}$ and its sign $s$.
    compute: $\boldsymbol{w}^* = (\boldsymbol{v} - \boldsymbol{u_{mb}})/v_{mb}$
    Compute $\mu_2\nu^*(\boldsymbol{w}^*)\delta t$.
    Draw a random number $\mathcal{R}$ between 0 and 1.
    Mark the particle for deletion if $\mu_2\nu^*(\boldsymbol{w}^*)\delta t \geq \mathcal{R}$

End For
```

3.3 Generation of particles

In this part, we want to generate particles to account for the term

$$\delta t \int [K_1(\boldsymbol{w}, \boldsymbol{w_1}) - K_2(\boldsymbol{w}, \boldsymbol{w_1})] f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} - \delta f_{mb}(\boldsymbol{v})$$

To get the distribution of particles that we need to generate for the entire cell, we can as previously integrate over the cell. Using the expression of $\int_{x\in C} f_d d^3\boldsymbol{x}$, given in *Equation* 3.17, and the fact that $K_i(\boldsymbol{w}, \boldsymbol{w_1}) = \mu_1 K_i^*(\boldsymbol{w}^*, \boldsymbol{w_1}^*)$ we get:

$$\delta t \mu_1 N_{eff} \int [K_1^*(\boldsymbol{w}^*, \boldsymbol{w_1}^*) - K_2^*(\boldsymbol{w}^*, \boldsymbol{w_1}^*)] \sum_{i \in \mathbf{C}} \delta(\boldsymbol{v_1} - \boldsymbol{v}^i) d^3\boldsymbol{v_1} - V_{cell} \delta f_{mb}(\boldsymbol{v})$$

which is equal to

$$\delta t \mu_1 N_{eff} \sum_{i \in \mathbf{C}} [K_1^*(\boldsymbol{w}^*, \boldsymbol{w}^{i*}) - K_2^*(\boldsymbol{w}^*, \boldsymbol{w}^{i*})] - V_{cell} \delta f_{mb}(\boldsymbol{v})$$

In summary, we want to generate a set of particles that will represent the distribution

$$\boldsymbol{v} \longrightarrow \delta t \mu_1 N_{eff} \sum_{i \in \mathbf{C}} [K_1^*(\frac{\boldsymbol{v} - \boldsymbol{u_{mb}}}{v_{mb}}, \frac{\boldsymbol{v^i} - \boldsymbol{u_{mb}}}{v_{mb}}) - K_2^*(\frac{\boldsymbol{v} - \boldsymbol{u_{mb}}}{v_{mb}}, \frac{\boldsymbol{v^i} - \boldsymbol{u_{mb}}}{v_{mb}})] - V_{cell} \delta f_{mb}(\boldsymbol{v})$$

The particles are then generated by an acceptance rejection technique. The difficulty lies in that we want to generate particles drawn from a distribution which we do not know explicitly but need to compute on the fly. Since evaluating this distribution function will have to be done a large number of times, we approximate the sum by looping over N_2 randomly choosen particles instead of all N_d particles in the cell. We thus need to premultiply the sum by N_d/N_2.

The essence of the acceptance-rejection algorithm is then as follows. We choose randomly N_1 velocities $\boldsymbol{v}$, which will be the velocities of candidate particles to be, possibly, created. These velocities are drawn from an arbitrary distribution $g(\boldsymbol{v})$, which has to be greater than $|\delta t \left[\int K_1 f_d - \int K_2 f_d\right] - \delta f_{mb}|$. For a given $\boldsymbol{v}$, we loop over N_2 numerical particles in order to compute $\delta t \left[\int K_1 f_d - \int K_2 f_d\right]$ as discussed above. At the end of this loop, we create the particle of velocity $\boldsymbol{v}$ if $|\delta t \left[\int K_1 f_d - \int K_2 f_d\right] - \delta f_{mb}| > \mathcal{R} g(\boldsymbol{v})$, where $\mathcal{R}$ is a uniform random number between 0 and 1.

So, the scheme is:

```
Do $N_1$ times:

  Choose $\boldsymbol{v}$ from a distribution proportional to $g(\boldsymbol{v})$.
  Compute the velocity: $\boldsymbol{w}^* = (\boldsymbol{v} - \boldsymbol{u_{mb}})/v_{mb}$
  set sum=0;

  Do $N_2$ times:

    Pick a numerical particle, and get its velocity $\boldsymbol{v_1}$ and its sign $s$.
    Compute the velocity: $\boldsymbol{w_1}^* = (\boldsymbol{v_1} - \boldsymbol{u})/v_{mb}$
    sum = sum + $s\mu_1\delta t\frac{N_d}{N_2}\left[K_1^*(\boldsymbol{w}^*,\boldsymbol{w}_1^*) - K_2^*(\boldsymbol{w}^*,\boldsymbol{w}_1^*)\right]$

  End Do $N_2$ times

  sum = sum - $\frac{V_{cell}}{N_{eff}}\delta f_{mb}(\boldsymbol{v})$
  pick a random number $\mathcal{R}$ between 0 and 1.
  Create a particle of sign sign(sum) and velocity $\boldsymbol{v}$ if sum $\geq \mathcal{R}g(\boldsymbol{v})$

End Do $N_1$ times
```

We recall that N_d denotes the total number of numerical particles in the current cell, N_{eff} is the number of particles that a numerical particle stands for and V_{cell} is the volume of the cell.

Some further comments on the implementation of this part of the algorithm are given below.

3.3.1 Parameters N_1 and N_2

In order to generate the correct number of particles, we must have $N_1 = \int g d^3\boldsymbol{v}$.

N_2 can be any number. However, if it is chosen too small, the results are likely to be more noisy, since we will generate particles from a more noisy distribution. Ideally, we would like to choose $N_2 = N_d$, but it makes the algorithm less efficient and does not increase accuracy substantially. A recommended way to proceed is to choose N_2 as being a fraction of N_d, for example one fifth.

3.3.2 Importance sampling

The distribution g can simply be a uniform distribution on a box. For more efficiency, we can take g to be a radial piecewise constant distribution centered on the local mean velocity. Since it has to be greater than the distribution we are sampling, we have to update g by increasing the concerned piece of g whenever we come across an occurence of $sum \geq g(\boldsymbol{v})$.

In an implementation, we use $g(\boldsymbol{v}) = 4\pi|\boldsymbol{w}|^2 g_r(|\boldsymbol{w}|)$, where the function g_r is a piecewise constant function:

$$g_r(|\boldsymbol{w}|) = g_r^i \text{ if } r_i \leq |\boldsymbol{w}| \leq r_{i+1}$$

The $r_i's$ are chosen to be proportinal to the standard deviation of the local Maxwellian: $r_i = iv_{mb}/\sqrt{2}$, and $g_r(|\boldsymbol{w}|) = 0$ for $|\boldsymbol{w}| \geq 7v_{mb}/\sqrt{2}$. If we come accross $sum > g_r^i$, then we update the function by doing $g_r^i = sum$. Let's note that the function g_r is different for each cell.

To initialize the function g, we carry out at the begining, and from time to time, a voided run of the generation routine, that is to say of the exact same routine exept that we do not create any particles but only update the function g.

3.3.3 Cutoff of the Kernel K_1

It has to be noted that the first kernel (K_1) is singular at the point $\boldsymbol{v} = \boldsymbol{v_1}$ and diverges as $1/|\boldsymbol{v}-\boldsymbol{v_1}|$. This singularity raises a problem when it comes to generating particles with an acceptance/rejection procedure, since this method is designed for bounded distributions. To deal with this difficulty, we set a cutoff relative velocity v_c and define a modified kernel. The modified kernel is defined such that it is constant $\forall \boldsymbol{v}$ for which $|\boldsymbol{v} - \boldsymbol{v_1}| < v_c$, as sketched in *Figure* 3-1. The constant value is taken to be the mean value of the kernel over the sphere centered on $\boldsymbol{v_1}$, and of radius v_c. The mean value of the kernel over this sphere is $\mu_1 \sqrt{\pi} \frac{3}{2} \frac{\mathrm{erf}(\boldsymbol{v_1}^*)}{\boldsymbol{v_1}^*} \frac{1}{v_c^*}$. The derivation of this mean can be found in Appendix A.

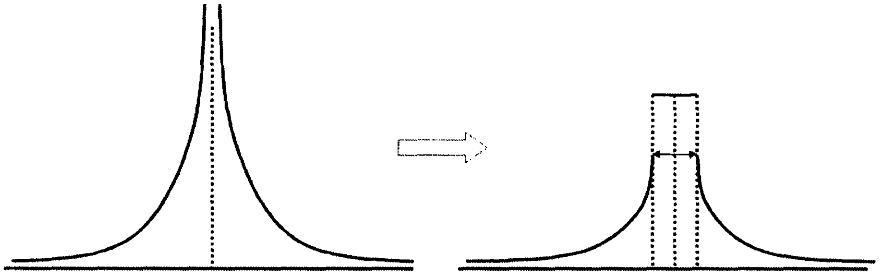


Figure 3-1:

3.4 The Nonlinear Collision Term

The nonlinear collision term $C(f_d, f_d)$ cannot be optimized and is treated exactly in the same way as it was in VRDSMC [5, 6]. As emphasized in Introduction, the proposed implementation requires an average creation of two particles per collision leading to instability issues. This will affect the efficiency of the present algorithm for nonlinear collision-dominated flows. Unstable behaviors were observed for strongly nonlinear collision-dominated flows. In contrast, for low-signal flows, this term is of second order and the number of particles that will be created will remain very small. For linear to weakly nonlinear flows, the creation of particles is balanced by the particle sink discussed in section 3.2.

The method for computing $C(f_d, f_d)$ was briefly outlined in introduction and is explained in detail in [5, 6, 7]. The design of the scheme stems from the following form of the nonlinear collision term:

$$C(f_d, f_d) = \frac{1}{2}\int\int\int(\delta_1' + \delta_2' - \delta_1 - \delta_2)f_{d1}f_{d2}|\boldsymbol{v_2} - \boldsymbol{v_1}|\sigma d^2\Omega d^3\boldsymbol{v_1}d^3\boldsymbol{v_2}$$

As in standard DSMC, collisions are processed by sampling a number of pairs of numerical particles. The operations to perform will then depend on the sign s^1 and s^2 of the particles, as follows:

* If $s^1 = +1$ and $s^2 = +1$, update the velocities as in standard dsmc.

* If $s^1 = +1$ and $s^2 = -1$, create $+\delta_1$, $-\delta_1'$ and update $-\delta_2$ into $-\delta_2'$.

* If $s^1 = -1$ and $s^2 = +1$, create $+\delta_2$, $-\delta_2'$ and update $-\delta_1$ into $-\delta_1'$.

* If $s^1 = -1$ and $s^2 = -1$, create $-\delta_1$, $-\delta_2$, $+\delta_1'$ and $+\delta_2'$.

In the above scheme, postcollision velocities are computed according to the hard sphere scattering angle distribution (uniform over the unit sphere). The derivation is explained in detail in [7]. The final result is:

$$\begin{aligned}
v'_{1,x} &= \frac{1}{2}(v_{1,x}+v_{2,x}) + \frac{1}{2}|\boldsymbol{v_1}-\boldsymbol{v_2}|\sin\theta\sin\phi \\
v'_{1,y} &= \frac{1}{2}(v_{1,y}+v_{2,y}) + \frac{1}{2}|\boldsymbol{v_1}-\boldsymbol{v_2}|\sin\theta\cos\phi \\
v'_{1,z} &= \frac{1}{2}(v_{1,z}+v_{2,z}) + \frac{1}{2}|\boldsymbol{v_1}-\boldsymbol{v_2}|\cos\theta
\end{aligned}$$

$$\begin{aligned}
v'_{2,x} &= \frac{1}{2}(v_{1,x}+v_{2,x}) - \frac{1}{2}|\boldsymbol{v_1}-\boldsymbol{v_2}|\sin\theta\sin\phi \\
v'_{2,y} &= \frac{1}{2}(v_{1,y}+v_{2,y}) - \frac{1}{2}|\boldsymbol{v_1}-\boldsymbol{v_2}|\sin\theta\cos\phi \\
v'_{2,z} &= \frac{1}{2}(v_{1,z}+v_{2,z}) - \frac{1}{2}|\boldsymbol{v_1}-\boldsymbol{v_2}|\cos\theta
\end{aligned}$$

The number of collision candidate to process is $\frac{N_{eff}\pi d^2 \delta t N_d^2 v_r^{MAX}}{2V_{cell}}$, where v_r^{MAX} is the maximum relative velocity usually set to a few times the most probable velocity. Collisions are accepted with probability $|\boldsymbol{v_1}-\boldsymbol{v_2}|/v_r^{MAX}$. To generate scattering angles θ and ϕ uniformly on a unit sphere, we generate:

$$\begin{aligned}
\cos\theta &= 2\mathcal{R}_1 - 1 \\
\sin\theta &= \sqrt{1-\cos^2\theta} \\
\phi &= 2\pi\mathcal{R}_2
\end{aligned}$$

where $\mathcal{R}_1$ and $\mathcal{R}_2$ are two different random numbers sampled from the uniform distribution in $[0,1]$.

Chapter 4

The Free-Molecular Advection Operator

According to the splitting of operators discussed before, the free-molecular advection is described by:

$$\partial f/\partial t + \boldsymbol{v} \cdot \boldsymbol{\nabla f} = 0$$

Contrary to the collision routine, f_{mb} will remain constant all over the advection procedure. This means that we account for the change by modifying f_d only, that is to say by creating or moving deviational particles ($df = df_d$). Since $f = f_{mb} + f_d$, the term $\boldsymbol{v} \cdot \boldsymbol{\nabla f}$ is made up of two parts: $\boldsymbol{v} \cdot \boldsymbol{\nabla f} = \boldsymbol{v} \cdot \boldsymbol{\nabla f_d} + \boldsymbol{v} \cdot \boldsymbol{\nabla f_{mb}}$.

We thus need to solve $\partial f_d/\partial t + \boldsymbol{v} \cdot \boldsymbol{\nabla f_d} = -\boldsymbol{v} \cdot \boldsymbol{\nabla f_{mb}}$

The general solution of an equation of the form $\partial f/\partial t + \boldsymbol{v} \cdot \boldsymbol{\nabla f} = g(\boldsymbol{x}, \boldsymbol{v}, t)$ is $f(\boldsymbol{x}, \boldsymbol{v}, t) = \int_0^t g(\boldsymbol{x} - \boldsymbol{v}t', \boldsymbol{v}, t - t')dt' + f(\boldsymbol{x} - \boldsymbol{v}t, \boldsymbol{v}, 0)$, where $f(\boldsymbol{x}, \boldsymbol{v}, 0)$ is the initial state. (See the Proof in Appendix B).

For convenience and without loss of generality, we can set $t = 0$ at the beginning of the current timestep. The state at $t = \delta t$, which is what is required, is then: $f_d(\boldsymbol{x}, \boldsymbol{v}, \delta t) = \int_0^{\delta t} g(\boldsymbol{x} - \boldsymbol{v}t', \boldsymbol{v}, \delta t - t')dt' + f_d(\boldsymbol{x} - \boldsymbol{v}\delta t, \boldsymbol{v}, 0)$, where in our case

$g(\boldsymbol{x}, \boldsymbol{v}, t) = -\boldsymbol{v} \cdot \boldsymbol{\nabla f_{mb}}$.

$f_d(\boldsymbol{x}, \boldsymbol{v}, \delta t)$ is made of two parts which will be treated separatly. The first part results from the source term $-\boldsymbol{v} \cdot \boldsymbol{\nabla f_{mb}}$ and corresponds to the advection of f_{mb}, that is to say to the advection of some underlying particles not represented by numerical particles. This advection will be done by creation of particles and is refered to as the advection of the f_{mb} part. The second part is a simple advection of the numerical particles and is refered to as the advection of the f_d part.

4.1 Advection of the f_d part

In this part, we perform the following operation:

$$f^d(\boldsymbol{x}, \boldsymbol{v}, \delta t) = f^d(\boldsymbol{x} - \boldsymbol{v}\delta t, \boldsymbol{v}, 0)$$

This is done by moving all particles according to their velocities, as in standard DSMC. In other words, we perform:

```
For all particles in the domain
  $\boldsymbol{x} \rightarrow \boldsymbol{x} + \boldsymbol{v}\delta t$.
End For
```

4.2 Advection of the f_{mb} part

In this part, we want to add the term $\int_0^{\delta t} -\boldsymbol{v} \cdot \boldsymbol{\nabla f_{mb}}(\boldsymbol{x} - \boldsymbol{v}t', \boldsymbol{v}, \delta t - t')dt'$ to the distribution function. This is done by creating deviational particles. We first need to compute $-\boldsymbol{v} \cdot \boldsymbol{\nabla f_{mb}}(\boldsymbol{x}, \boldsymbol{v}, t)$ in order to compute the integral which will provide us with the distribution from which the particles have to be drawn.

For a fixed velocity $\boldsymbol{v}$, the function $\boldsymbol{x} \rightarrow f_{mb}(\boldsymbol{x}, \boldsymbol{v})$ is piecewise constant. As a consequence, the function $\boldsymbol{x} \rightarrow \boldsymbol{\nabla f_{mb}}(\boldsymbol{x}, \boldsymbol{v})$ is zero everywhere, except at the interface of the cells, where it is a dirac (each component is a dirac).

Let S_n be the surface separing the two cells, and $\boldsymbol{n}$ be the unit normal vector of this surface. Without loss of generality, we can assume that S_n contains the origin $\boldsymbol{x} = \boldsymbol{0}$, so that $\boldsymbol{n} \cdot \boldsymbol{x}$ is the distance between $\boldsymbol{x}$ and S_n. Let also f^r_{mb} and f^l_{mb} refer to the local Maxwellian in the right and the left cell respectively.

The detailed derivation of the function ∇f_{mb} is located in the Appendix B. The result is that the gradient is $\boldsymbol{\nabla f_{mb}}(\boldsymbol{x}, \boldsymbol{v}) = (f^r_{mb} - f^l_{mb})\delta(\boldsymbol{n} \cdot \boldsymbol{x})\boldsymbol{n}$, where δ is the dirac distribution. The integral $\delta f_d(\boldsymbol{x}, \boldsymbol{v}, \delta t) = \int_0^{\delta t} g(\boldsymbol{x} - \boldsymbol{v}t', \boldsymbol{v}, \delta t - t')dt'$ computed when $g(\boldsymbol{x}, \boldsymbol{v}, t) = -\boldsymbol{v} \cdot \boldsymbol{\nabla f_{mb}}(\boldsymbol{x}, \boldsymbol{v}) = (f^l_{mb} - f^r_{mb})\delta(\boldsymbol{n} \cdot \boldsymbol{x})\boldsymbol{v} \cdot \boldsymbol{n}$ is:

$$\delta f_d(\boldsymbol{x}, \boldsymbol{v}, \delta t) = (f^l_{mb} - f^r_{mb})\boldsymbol{v} \cdot \boldsymbol{n} \int_0^{\delta t} \delta(\boldsymbol{n} \cdot \boldsymbol{x} - \boldsymbol{n} \cdot \boldsymbol{v}t')dt'$$

The value of the integral is $|\boldsymbol{n} \cdot \boldsymbol{v}|^{-1}$ when $(\boldsymbol{n} \cdot \boldsymbol{x})/(\boldsymbol{n} \cdot \boldsymbol{v}) \in [0\ \delta t]$ and zero otherwise. This finally leads to:

$$\delta f_d(\boldsymbol{x}, \boldsymbol{v}, \delta t) = \begin{cases} \left[f^l_{mb}(\boldsymbol{v}) - f^r_{mb}(\boldsymbol{v})\right] sign(\boldsymbol{v} \cdot \boldsymbol{n}) & \text{if } \boldsymbol{x} \cdot \boldsymbol{n} \leq \boldsymbol{v} \cdot \boldsymbol{n}\delta t \\ 0 & \text{if } \boldsymbol{x} \cdot \boldsymbol{n} > \boldsymbol{v} \cdot \boldsymbol{n}\delta t \end{cases}$$

Let's note that contrary to what we could think at first glance, this does correspond to the difference of two fluxal Maxwellian distribution. Indeed, for a fixed velocity $\boldsymbol{v}$, the total mass that we need to create is $S_n[f^l_{mb}(\boldsymbol{v}) - f^r_{mb}(\boldsymbol{v})]|\boldsymbol{v} \cdot \boldsymbol{n}|\delta t$, where we recall that S_n is the cross-sectional area of the cells, and $\boldsymbol{n}$ is the normal of the surface separing two cells. We obtain this by just integrating over space the above distribution.

In practice, we generate particles drawn from $S_n[f^l_{mb}(\boldsymbol{v}) - f^r_{mb}(\boldsymbol{v})]\boldsymbol{v} \cdot \boldsymbol{n}\delta t$ and spread them out over a domain of length $|\boldsymbol{v} \cdot \boldsymbol{n}|\delta t$, which does correspond to having a density equal to $\left[f^l_{mb}(\boldsymbol{v}) - f^r_{mb}(\boldsymbol{v})\right] sign(\boldsymbol{v} \cdot \boldsymbol{n})$.

This generation of particles has to be carried out wherever there is a discontinuity of f_{mb}, that is to say at all interfaces between cells.

The generation of particles is done by acceptance rejection. The velocities are first drawn uniformly from a cubic velocity domain $[-v_{max}, v_{max}]^3$, where v_{max} is a large velocity compared to the most probable velocity. The computational cost of this part of the whole algorithm is not very high, so we can afford to do a simple acceptance rejection, without resorting to an importance-sampling-type of approach as in Section 3.3.

Finally, the scheme for the advection of the f_{mb} part of the distribution is as follows:

```
For all intefaces:

  Do $N$ times:

    Choose $v_x$ from a uniform distribution over $[-v_{max}, v_{max}]$.
    Choose $v_y$ from a uniform distribution over $[-v_{max}, v_{max}]$.
    Choose $v_z$ from a uniform distribution over $[-v_{max}, v_{max}]$.
    Compute $f^r_{mb}(\boldsymbol{v})$ and $f^l_{mb}(\boldsymbol{v})$
    Draw random numbers $\mathcal{R}_1$ and $\mathcal{R}_2$ between 0 and 1.

    If $|\frac{\delta t S_n}{N_{eff} V_v}[f^l_{mb}(\boldsymbol{v}) - f^r_{mb}(\boldsymbol{v})]\boldsymbol{n} \cdot \boldsymbol{v}| > \mathcal{R}_1 M$

      Create a particle:
      * of velocity $\boldsymbol{v}$
      * of sign $sign([f^r_{mb}(\boldsymbol{v}) - f^l_{mb}(\boldsymbol{v})]\boldsymbol{n} \cdot \boldsymbol{v})$
      * at a position such that $\boldsymbol{n} \cdot \boldsymbol{x} = \delta t \boldsymbol{n} \cdot \boldsymbol{v} \cdot \mathcal{R}_2$

    End If

  End Do

End For
```

In the above scheme, S_n is the cell cross-sectional area, v_{max} is a parameter used to specify the size of the velocity domain where the velocities are drawn from and $V_{\boldsymbol{v}}$ is the volume of this velocity box: $V_{\boldsymbol{v}} = 8v^3_{max}$. The number of loops has to be $N = MV_{\boldsymbol{v}}$ (that is to say the integral of the importance sampling function which is a constant function over the box in this case).

Chapter 5

Boundary Conditions

In most applications, the boundaries of the domain contain solid walls. Within kinetic theory, walls are typically treated [1, 2, 3] as infinite reservoirs of an equilibrium gas (homogeneous Maxwellian distribution) at the wall temperature and velocity and whose number density is calculated so that the mass flux into the wall is the same as the one going out from the wall. The boundary conditions are then treated by simply carrying out a free molecular advection between the reservoir and the actual physical domain [2, 3].

The approach outlined here mirrors the method developed in [5, 6]. Let N^+_{wall} and N^-_{wall} denote the number of respectively positive and negative deviational particles that crossed the wall during the timestep, $N_{wall} = N^+_{wall} + N^-_{wall}$ the total number of those particles and $\Delta N_{wall} = N^+_{wall} - N^-_{wall}$ the net number. Let S_n be the cross-sectional area of the wall-cell interface and $\boldsymbol{n}$ the unit normal of this surface. For convenience and without loss of generality, the following calculations will be carried out for the particular case where $\boldsymbol{n} = \boldsymbol{e_x} = (1\ 0\ 0)$, and where the surface contains the point $\boldsymbol{x} = \boldsymbol{0}$. In other words, the wall is chosen to be the yz-plane. Let us denote the wall distribution $f_{wall} = n_{wall}\tilde{f}_{wall}$ with:

$$\tilde{f}_{wall}(\boldsymbol{v}) = \frac{1}{\pi^{3/2}v^3_{wall}} \exp\left[\frac{-(\boldsymbol{v} - \boldsymbol{u_{wall}})^2}{v^2_{wall}}\right]$$

Due to the free molecular advection, there are particles crossing the wall. The flux $\mathcal{F}$ of particles going out of the domain is the sum of the flux $\mathcal{F}_d$ of outgoing numerical deviational particles (f_d), and the flux $\mathcal{F}_{mb}$ of outgoing underlying Maxwellian particles (f_{mb}). The flux of particles entering the domain is denoted $\mathcal{F}_{wall}$. By definition, we have:

$$\begin{aligned}\mathcal{F}_{mb} &= \int_{v_x<0} v_x f_{mb}(\boldsymbol{v}) d^3\boldsymbol{v} \\ \mathcal{F}_{wall} &= \int_{v_x>0} v_x n_{wall} \tilde{f}_{wall}(\boldsymbol{v}) d^3\boldsymbol{v}\end{aligned}$$

and

$$\delta t S_n \mathcal{F}_d = \delta t S_n \int_{v_x<0} v_x f_d(\boldsymbol{v}) d^3\boldsymbol{v} = N_{eff} \Delta N_{wall} \tag{5.1}$$

The boundary condition must ensure conservation of mass or equivalently the conservation of the number of molecules (represented by deviational and underlying Maxwellian particles) in the physical domain. This conservation of mass is equivalent to having the incoming flux of particles equal to the outgoing one, or in our notation: $\mathcal{F}_{wall} = \mathcal{F}_{mb} + \mathcal{F}_d$. This latter relation is achieved by simply setting n_{wall} to an appropriate value.

However, in order to keep the contributions of the Maxwellian and deviational distribution separate, we write

$$n_{wall}\tilde{f}_{wall} = (n^{mb}_{wall} + n^{d}_{wall})\tilde{f}_{wall}$$

such that

$$\mathcal{F}_{wall} = \mathcal{F}^{mb}_{wall} + \mathcal{F}^{d}_{wall}$$

with $\mathcal{F}^{mb}_{wall}$ being the flux associated with $n^{mb}_{wall}\tilde{f}_{wall}$ and $\mathcal{F}^{d}_{wall}$ the one associated with

$n^d_{wall}\tilde{f}_{wall}$. The condition of mass conservation $\mathcal{F}_{wall} = \mathcal{F}_{mb} + \mathcal{F}_d$ then becomes

$$\begin{aligned} \mathcal{F}^{mb}_{wall} &= \mathcal{F}_{mb} \\ \mathcal{F}^{d}_{wall} &= \mathcal{F}_{d} \end{aligned}$$

We then observe that the situation is very similar to what we had in the previous chapter, where we dealt with the free molecular advection of the Maxwellian contribution. From Chapter 4, we know that the solution amounts to generating particles that sample the following distribution:

$$\begin{cases} \left[n_{wall}\tilde{f}_{wall}(\boldsymbol{v}) - f_{mb}(\boldsymbol{v})\right] sign(\boldsymbol{v}\cdot\boldsymbol{n}) & \text{if } \boldsymbol{x}\cdot\boldsymbol{n} \le \boldsymbol{v}\cdot\boldsymbol{n}\delta t \\ 0 & \text{if } \boldsymbol{x}\cdot\boldsymbol{n} > \boldsymbol{v}\cdot\boldsymbol{n}\delta t \end{cases}$$

Using the fact that we are only interested in sampling particles entering the physical domain ($sign(\boldsymbol{v}\cdot\boldsymbol{n}) > 0$) and after inserting the splitting $n_{wall}\tilde{f}_{wall} = n^{mb}_{wall}\tilde{f}_{wall} + n^d_{wall}\tilde{f}_{wall}$, we get

$$\begin{cases} n^d_{wall}\tilde{f}_{wall}(\boldsymbol{v}) + \left[n^{mb}_{wall}\tilde{f}_{wall}(\boldsymbol{v}) - f_{mb}(\boldsymbol{v})\right] & \text{if } x \le v_x\delta t \\ 0 & \text{if } x > v_x\delta t \end{cases}$$

The two parts, $n^d_{wall}\tilde{f}_{wall}(\boldsymbol{v})$ and $\left[n^{mb}_{wall}\tilde{f}_{wall}(\boldsymbol{v}) - f_{mb}(\boldsymbol{v})\right]$ respectively, are treated in separate routines which are explained in the following two sections.

5.1 The f_d part

This section deals with the treatment of the deviational contribution $n^d_{wall}\tilde{f}_{wall}(\boldsymbol{v})$. As seen above, we have to generate particles to sample the following distribution:

$$\begin{cases} n^d_{wall}\tilde{f}_{wall}(\boldsymbol{v}) & \text{if } x \le v_x\delta t \\ 0 & \text{if } x > v_x\delta t \end{cases}$$

As discussed above, the condition that determines n^d_{wall} is $\mathcal{F}^d_{wall} = \mathcal{F}_d$. The net number of numerical particles that we have to generate is then given by $S_n \delta t N_{eff}^{-1} \mathcal{F}_{wall}^{d} = \delta t N_{eff}^{-1} \mathcal{F}^{d} = \Delta N$. Note that ΔN can be negative. This means that the equivalent of the net mass of the particles that hit the wall is sent back into the domain with the wall velocity distribution. In pratice, we draw their velocities from $\tilde{f}_{wall}(\boldsymbol{v})v_x$ with $v_x > 0$ and spread them out uniformly over a domain of length $v_x \delta t$. To generate the velocities, we use an acceptance/rejection algorithm. Howerver, when f_{wall} has no mean velocity along the normal of the wall (the wall is moving in its plane) there exists more efficient methods for generating such velocities.

In summary, the scheme is:

Get the N_{wall} particles that hit the wall.

Delete $\min(N_{wall}^{+}, N_{wall}^{-})$ of them

Send back the abs(ΔN) remaining particles:

* velocity drawn from $\tilde{f}_{wall}(\boldsymbol{v})v_x$

* with the sign $sign(\Delta N)$

* at position $\delta t v_x \mathcal{R}_1$

5.2 The f_{mb} part

This section deals with the treatment of the Maxwellian contribution $\left[n_{wall}^{mb}\tilde{f}_{wall}(\boldsymbol{v}) - f_{mb}(\boldsymbol{v})\right]$. The number density n_{wall}^{mb} is determined by imposing $\mathcal{F}_{wall}^{mb} = \mathcal{F}_{mb}$. These fluxes can be calculated analytically for any local and wall Maxwellian:

$$\begin{aligned}
\mathcal{F}_{wall}^{mb} &= \int_{v_x>0} n_{wall}^{mb}\tilde{f}_{wall} v_x d^3\boldsymbol{v} = \frac{n_{wall}^{mb} v_{wall}}{2}\left(\frac{1}{\pi^{1/2}}\exp[-u_{wall,x}^{*2}] + u_{wall,x}^{*}\mathrm{erf}[u_{wall,x}^{*}] + u_{wall,x}^{*}\right) \\
\mathcal{F}_{mb} &= \int_{v_x<0} f_{mb} v_x d^3\boldsymbol{v} = \frac{n_{mb} v_{mb}}{2}\left(\frac{1}{\pi^{1/2}}\exp[-u_{mb,x}^{*2}] + u_{mb,x}^{*}\mathrm{erf}[u_{mb,x}^{*}] - u_{mb,x}^{*}\right)
\end{aligned}$$

We choose then n_{wall}^{mb} as follows:

$$n_{wall}^{mb} = n_{mb} \left(\frac{v_{mb}}{v_{wall}} \right) \frac{\pi^{-1/2} \exp[-u_{mb,x}^{*2}] + u_{mb,x}^{*} \mathrm{erf}[u_{mb,x}^{*}] - u_{mb,x}^{*}}{\pi^{-1/2} \exp[-u_{wall,x}^{*2}] + u_{wall,x}^{*} \mathrm{erf}[u_{wall,x}^{*}] + u_{wall,x}^{*}}$$

From this, we can generate particles at the boundary to account for the f_{mb} particles that hit the wall. The scheme is almost the same as the one used for the convection of the f_{mb} distribution. The difference is that it is carried out only at the interface between the domain and the wall (at the boundaries of the domain), and we only need to create particles that go in the domain, that is to say such that $\boldsymbol{v} \cdot \boldsymbol{n} > 0$, where $\boldsymbol{n}$ is the inward normal of the wall. In the special case where $\boldsymbol{n} = \boldsymbol{e_x}$, the scheme is:

```
Do $N$ times:

  Choose $v_x$ from a uniform distribution over $[0, v_{max}]$.
  Choose $v_y$ from a uniform distribution over $[-v_{max}, v_{max}]$.
  Choose $v_z$ from a uniform distribution over $[-v_{max}, v_{max}]$.
  Compute $f_{mb}(\boldsymbol{v})$ and $f_{wall}(\boldsymbol{v})$
  Draw random numbers $\mathcal{R}_1$ and $\mathcal{R}_2$ between 0 and 1.

  If $|\frac{\delta t S_n}{N_{eff} V_{\boldsymbol{v}}}[n_{wall}^{mb}\tilde{f}_{wall}(\boldsymbol{v}) - f_{mb}(\boldsymbol{v})]v_x| > \mathcal{R}_1 M$

    Create a particle:
    * of velocity $\boldsymbol{v}$
    * of sign $sign(n_{wall}^{mb}\tilde{f}_{wall}(\boldsymbol{v}) - f_{mb}(\boldsymbol{v}))$
    * at position $x = \delta t v_x \cdot \mathcal{R}_2$

  End If
End Do
```

Here again, v_{max} is a parameter used to specify the size of the box where the velocity are drawn from and $V_{\boldsymbol{v}}$ is the volume of this velocity box, which is now: $V_{\boldsymbol{v}} = 4v_{max}^3$. The number of loops still has to be $N = MV_{\boldsymbol{v}}$. The distribution $n_{wall}^{mb}\tilde{f}_{wall}$ is the Maxwellian of the wall. Its mean velocity is the velocity of the wall $\boldsymbol{u_{wall}}$, its characteristic velocity v_{wall} is the one that correspond to the temperature of the wall ($v_{wall} = \sqrt{2k_B T_{wall}/m}$), and its number density is the one computed above that ensure the equality of the incoming and outcoming mass flux.

Chapter 6

Computing the Outputs

By output we mean macroscopic hydrodynamic quantities like the velocity of the fluid, the stess, or the temperature. These outputs depend on the local Maxwellian distribution (that is to say the mean velocity, the number density and the standard deviation of the local Maxwellian) and on the deviational particles. They are computed by a separate routine. Due to the splitting of the operator (convection and collision), we get a higher order of accuracy if we compute the outputs twice per step, that is to say before and after the collision routine.

In the following section, N^+ and N^- refer to the number of positive and negative particles in the current cell, $N_d = N^+ + N^-$ is the total number of particles and $\Delta N = N^+ - N^-$ is the net number of particles. N_{eff} is the number of physical molecules that a numerical particle represent. We will also refer to a component of a vector through a subscript as follows: $\boldsymbol{v} = (v_x, v_y, v_z)$, $\boldsymbol{u_{mb}} = (u_{mb,x}, u_{mb,y}, u_{mb,z})$ and $\boldsymbol{u_f} = (u_{f,x}, u_{f,y}, u_{f,z})$. We also recall that the superscript i refers to the index of a particle.

6.1 The Flow Velocity

The velocity of the fluid $\boldsymbol{u_f}$ is the average velocity of the particles within a control volume. We will be using the cells as control volume. This velocity does not coincide with the mean velocity $\boldsymbol{u_{mb}}$ of the local Maxwellian because the mean velocity of deviational particles could be different from $\boldsymbol{u_{mb}}$. Even though the collision routine generates deviational particles with a zero mean velocity in a frame moving at $\boldsymbol{u_{mb}}$, the convection introduces a nonzero mean velocity. By definition. the velocity of the fluid is:

$$\boldsymbol{u_f} = \frac{\int_{C\times\mathbb{R}^3} \boldsymbol{v} f(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v}}{\int_{C\times\mathbb{R}^3} f(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v}}$$

Using the fact that:

$$\begin{aligned}
\int_{C\times\mathbb{R}^3} \boldsymbol{v} f_d(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= N_{eff} \sum_{i\in\mathbf{C}} s^i \boldsymbol{v^i} \\
\int_{C\times\mathbb{R}^3} \boldsymbol{v} f_{mb}(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= n_{mb} V_{cell} \boldsymbol{u_{mb}} \\
\int_{C\times\mathbb{R}^3} f_d(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= N_{eff} \Delta N \\
\int_{C\times\mathbb{R}^3} f_{mb}(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= n_{mb} V_{cell}
\end{aligned}$$

we get:

$$\boldsymbol{u_f} = \frac{N_{eff} \sum_{i\in\mathbf{C}} \boldsymbol{v^i} + n_{mb} V_{cell} \boldsymbol{u_{mb}}}{N_{eff} \Delta N + n_{mb} V_{cell}}$$

where we recall that V_{cell} is the volume of the cell.

6.2 The Stress

By definition, the stress (averaged over a cell) is:

$$p_{kl} = \frac{m}{V_{\text{cell}}} \int_{C\times\mathbb{R}^3} (v_k - u_{f,k})(v_l - u_{f,l}) f(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v}$$

where m is the mass of a molecule, and k and l can be x, y or z. As previously, we use the fact that $f = f_d + f_{mb}$:

$$\begin{aligned}
\int_{C\times\mathbb{R}^3} (v_k - u_{f,k})(v_l - u_{f,l}) f_d(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= mN_{eff} \sum_{i\in\mathbf{C}} s^i (v_k^i - u_{f,k})(v_l^i - u_{f,l}) \\
\int_{C\times\mathbb{R}^3} (v_k - u_{f,k})(v_l - u_{f,l}) f_{mb}(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= mn_{mb} V_{cell} (u_{mb,k} - u_{f,k})(u_{mb,l} - u_{f,l})
\end{aligned}$$

We thus get:

$$p_{kl} = mN_{eff} V_{cell}^{-1} \sum_{i\in\mathbf{C}} s^i (v_k^i - u_{f,k})(v_l^i - u_{f,l}) + mn_{mb}(u_{mb,k} - u_{f,k})(u_{mb,l} - u_{f,l})$$

6.3 The Kinetic Temperature

The kinetic temperature is the mean kinetic energy in a frame moving at the local velocity of the fluid, divided by $\frac{3}{2}k_B$. By definition, it is:

$$T_k = \frac{m}{3k_B} \frac{\int_{C\times\mathbb{R}^3} (\boldsymbol{v} - \boldsymbol{u_f})^2 f(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v}}{\int_{C\times\mathbb{R}^3} f(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v}}$$

Using the fact that $f = f_d + f_{mb}$:

$$\begin{aligned}
\int_{C\times\mathbb{R}^3} (\boldsymbol{v} - \boldsymbol{u_f})^2 f_d(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= N_{eff} \sum_{i\in\mathbf{C}} s^i (\boldsymbol{v^i} - \boldsymbol{u_f})^2 \\
\int_{C\times\mathbb{R}^3} (\boldsymbol{v} - \boldsymbol{u_f})^2 f_{mb}(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= \frac{3}{2} V_{cell} n_{mb} v_{mb}^2 + n_{mb} V_{cell} (\boldsymbol{u_{mb}} - \boldsymbol{u_f})^2
\end{aligned}$$

The kinetic temperature is then:

$$T_k = \frac{m}{3k_B} \frac{N_{eff} \sum_{i\in\mathbf{C}} s^i (\boldsymbol{v^i} - \boldsymbol{u_f})^2 + V_{cell} n_{mb} \left((3/2) v_{mb}^2 + (\boldsymbol{u_{mb}} - \boldsymbol{u_f})^2\right)}{N_{eff} \Delta N + n_{mb} V_{cell}}$$

6.4 The Fourth Moment along v_x

In the homogenious relaxation testcase, we will be using this fourth moment of the distribution as criterion to check the accuracy of the code.

The definition is:

$$< v_x^4 >= \frac{\int_{C\times\mathbb{R}^3} v_x^4 f(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v}}{\int_{C\times\mathbb{R}^3} f(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v}}$$

As previously, we have:

$$\begin{aligned}
\int_{C\times\mathbb{R}^3} v_x^4 f_d(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= N_{eff} \sum_{i\in\mathbf{C}} s^i (v_x^i)^4 \\
\int_{C\times\mathbb{R}^3} v_x^4 f_{mb}(\boldsymbol{x},\boldsymbol{v}) d^3\boldsymbol{x} d^3\boldsymbol{v} &= \frac{3}{4} n_{mb} V_{cell} v_{mb}^4 + 3 n_{mb} V_{cell} v_{mb}^2 u_{mb,x}^2 + n_{mb} V_{cell} u_{mb,x}^4
\end{aligned}$$

So, the total fourth moment is:

$$< v_x^4 >= \frac{N_{eff} \sum_{i\in\mathbf{C}} s^i (v_x^i)^4 + V_{cell} n_{mb} \left(\frac{3}{4} v_{mb}^4 + 3 v_{mb}^2 u_{mb,x}^2 + u_{mb,x}^4\right)}{N_{eff}\Delta N + n_{mb} V_{cell}}$$

Note that in homogenious relaxation cases, the total number of molecules $N_{eff}\Delta N + n_{mb}V_{cell}$ should be conserved in average but might fluctuate due to the noise. In spacially dependent cases, it is not constant because of the advection.

Chapter 7

A Linearized Version of the Algorithm

Except from numerical discretizations involved in the implementation described above, the algorithm presented so far relies on no assumption and solves the full nonlinear Boltzmann equation. On the other hand, we discussed in the Introduction that the present method holds an advantage over DSMC in cases where the deviation from equilibrium is small, and thus when a linearization is valid. A linearized version of the algorithm has thus been developed from the previous nonlinear one. Although it did not display any improvement in terms of computational efficiency, it has the advantage of requiring a much easier implementation.

7.1 Linearization of the Collision Operator

We consider a homogeneous gas at rest at a reference temperature T_0 and with a reference number density n_0, the distribution function being a reference Maxwellian $f_0(\boldsymbol{v}) = n_0\pi^{-3/2}v_0^{-3}\exp[-(\boldsymbol{v}/v_0)^2]$. We recall that the most probable velocity is $v_0 = \sqrt{2k_BT_0/m}$ where k_B is the Boltzmann's constant. We then consider small perturbations from this equilibrium. The distribution function $f - f_0$ is thus defined as being of order one. The distribution function f is still written as $f = f_d + f_{mb}$ where the local Maxwelian f_{mb} will deviate from the reference Maxwellian f_0. Although it is not proven a priori, we can expect -and it will be observed in practice- that the local Maxwellian f_{mb} will deviate in a very small amount from the reference Maxwellian. The small deviation $f - f_0 = f_d + f_{mb} - f_0$ is then the sum of f_d and $f_{mb} - f_0$, both small deviations of order one. We argued that this was not guaranteed because it is mathematically possible that $f - f_0$ be small while f_d and $f_{mb} - f_0$ are large but cancel out. However, as discussed previously, the algorithm was designed in such a way that the local Maxwellian moves toward one that makes f_d small through the absorbtion of the first moments of f_d. This ensures that if $f - f_0$ is a small deviation of order one, so will be f_d and $f_{mb} - f_0$.

We recall that the complete nonlinear collision operator can be written as $C(f, f) = 2C(f_{mb}, f_d) + C(f_d, f_d)$. The part $2C(f_{mb}, f_d)$ gave rise to the three terms involving the two kernels and the collision frequency function, while the part $C(f_d, f_d)$ was left as is and treated as in VRDSMC. Using its billinearity, this collision operator can be written as:

$$C(f, f) = 2C(f_0, f_d) + 2C(f_{mb} - f_0, f_d) + C(f_d, f_d)$$

The distribution $f_{mb} - f_0$ and f_d being deviation of first order, the terms $2C(f_{mb} - f_0, f_d)$ and $C(f_d, f_d)$ are second order. At first order, we are left with $C(f, f) \approx 2C(f_0, f_d)$. The derivation of the kernels detailed in Appendix A suggests that $2C(f_0, f_d)$ will lead to the same kernels but where we have to replace n_{mb}, $\boldsymbol{u_{mb}}$ and v_{mb} by respectively n_0, $\mathbf{0}$ and v_0. The linearized collision operator will then have the same structure as before exept that whenever we had $\boldsymbol{w^*} = (\boldsymbol{v} - \boldsymbol{u_{mb}})/v_{mb}$, we can replace it by $\boldsymbol{v}/v_0$ which does not depend on the local Maxwellian. In summary, the linearized collision integral is

$$\left[\frac{\partial f}{\partial t}\right]_{coll} = \mu_1 \int K_1^*(\frac{\boldsymbol{v}}{v_0}, \frac{\boldsymbol{v_1}}{v_0}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} - \mu_1 \int K_2^*(\frac{\boldsymbol{v}}{v_0}, \frac{\boldsymbol{v_1}}{v_0}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} - f_d(\boldsymbol{v}) \mu_2 \nu^*(\frac{\boldsymbol{v}}{v_0})$$

where the function K_1^*, K_2^* and ν^* are the same as before, given by *Equations* 2.2, 2.3 and 2.4. Note that the prefactors are now

$$\mu_1 = \frac{d^2 n_0}{\pi^{1/2} v_0^2}, \qquad \mu_2 = v_0^3 \mu_1 = \frac{d^2 n_0 v_0}{\pi^{1/2}}$$

and do no longer depend on the local Maxwellian, in contrast to their expressions for the nonlinear case (*Equation* 2.5).

7.2 Linearization of the Change of the Local Maxwellian

As done in chapter 2, we can calculate the moments of the kernels involved in the linearized collision operator and get the corresponding expressions for δn_{mb}, $\delta \boldsymbol{u_{mb}}$ and δv_{mb}. It is straighforward and leads to:

$$\begin{aligned}
\delta n_{mb} &= \delta t \int \nu(\boldsymbol{v_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \\
\delta \boldsymbol{u_{mb}} &= \frac{\delta t}{n_0} \int \boldsymbol{v_1} \nu(\boldsymbol{v_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1} \\
\delta v_{mb} &= \frac{\delta t}{3 n_0 v_0} \int \left(|\boldsymbol{v_1}|^2 - \frac{3}{2} v_0^2 \right) \nu(\boldsymbol{v_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1}
\end{aligned}$$

Note that these expressions also tell us that δn_{mb}, $\delta \boldsymbol{u_{mb}}$ and δv_{mb} being in some sense an integral of f_d, are of order one with respect to "the small parameter" f_d. This means that any expression such that $(\delta n_{mb}/n_{mb})(f_{mb} - f_0)$ is of order two and can be neglected. Using this, we can show that the expression for δf_{mb} (*Equation* 3.6) becomes in the linearized version:

$$\delta f_{mb}(\boldsymbol{v}) = \left[n_0^{-1} \delta n_{mb} + v_0^{-1}(2 v_0^{-2} \boldsymbol{v}^2 - 3) \delta v_0 + 2 v_0^{-2} \boldsymbol{v} \cdot \delta \boldsymbol{u_{mb}} \right] f_0(\boldsymbol{v})$$

7.3 Linearization of the Algorithm

From the following considerations, we see that switching to a linearized version is straighforward. It simply consists in replacing every variable with subscript "mb" by its counterpart with subscript "0". It makes the implementation much simpler because a number of prefactors and expressions will not depend on the parameters of the local Maxwellian.

Note however that for the computation of the outputs we do need to use the actual parameters of the local Maxwellian since they carry important information (of order 1). Therefore, in the linearized version, the routine that computes the outputs is not changed.

The linearization of the advection routine has not been investigated.

Chapter 8

Spacially Homogeneous Relaxation Testcase

The purpose of this part is to validate the algorithm for the collision operator only and investigate the impact of the various parameters that need to be set by hand. To isolate the collision operator, we consider the spacially homogeneous relaxation version of the Boltzmann equation:

$$\frac{\partial f}{\partial t} = \int\int (f'f'_1 - ff_1)|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega d^3\boldsymbol{v_1}$$

Simulations have been run for gaseous Argon at temperatures around $T_0 = 300K$. The initial distribution was:

$$f(\boldsymbol{v}, t = 0) = \frac{1}{2}\frac{n_0}{\pi^{3/2}v_0^3}\exp\left[-\left(\frac{\boldsymbol{v} - v_s\boldsymbol{e_x}}{v_0}\right)^2\right] + \frac{1}{2}\frac{n_0}{\pi^{3/2}v_0^3}\exp\left[-\left(\frac{\boldsymbol{v} + v_s\boldsymbol{e_x}}{v_0}\right)^2\right]$$

n_0 is the number density corresponding to Argon at 300 K at 1 atm, v_0 is the corresponding most probable velocity ($v_0 = \sqrt{2k_BT_0/m}$), and v_s is an arbitrary shift.

The following various data were all compared to the results of a standard DSMC code run with the same numerical and physical properties.

The validation was carried out through the comparison of three relevant quantities:

* the distribution functions themselves at various times,
* the evolution of the fourth moment $< v_x^4 >$ versus time,
* the known final state.

The comparision of the distribution functions is not strictly speaking a comparison of the whole functions but of the density per unit velocity along the x-axis: $v_x \longrightarrow \int_{v_y=-\infty}^{+\infty} \int_{v_y=-\infty}^{+\infty} f(v_x, v_y, v_z, t) dv_y dv_z$. In practice, the quantities plotted are based on the following discretization of the v_x axis:

$$i \longrightarrow \int_{v_x=(i-1/2)q}^{(i+1/2)q} \int_{v_y=-\infty}^{+\infty} \int_{v_y=-\infty}^{+\infty} f(v_x, v_y, v_z, t) dv_y dv_z dv_x$$

q is the parameter that serves to discretize the v_x axis, and was set to be equal to 30 m/s. This function of the integer i can be calculated analyticaly for a Maxwellian, whereas the code computes the deviational part by counting the number of particles that satisfy $v_x \in [(i-1/2)q, (i+1/2)q]$.

The comparison of the distribution functions being essentially visual, it is relevant to compare some moment of the distributions which are very sensitive not only to subtle change of the shape of the distribution that one cannot perceive by eye, but also to the tail of the distribution, meaning the behavior of the distribution at infinity, where it is almost zero. However, because the collision operator conserves mass, momentum of energy, the first moment with an interesting behavior in this relaxation problem is the fourth order moment. In light of the initial distribution function, it appears relevant to focus on the fourth moment along the v_x-axis, whose definition is $\int v_x^4 f(\boldsymbol{v}) d^3\boldsymbol{v}$.

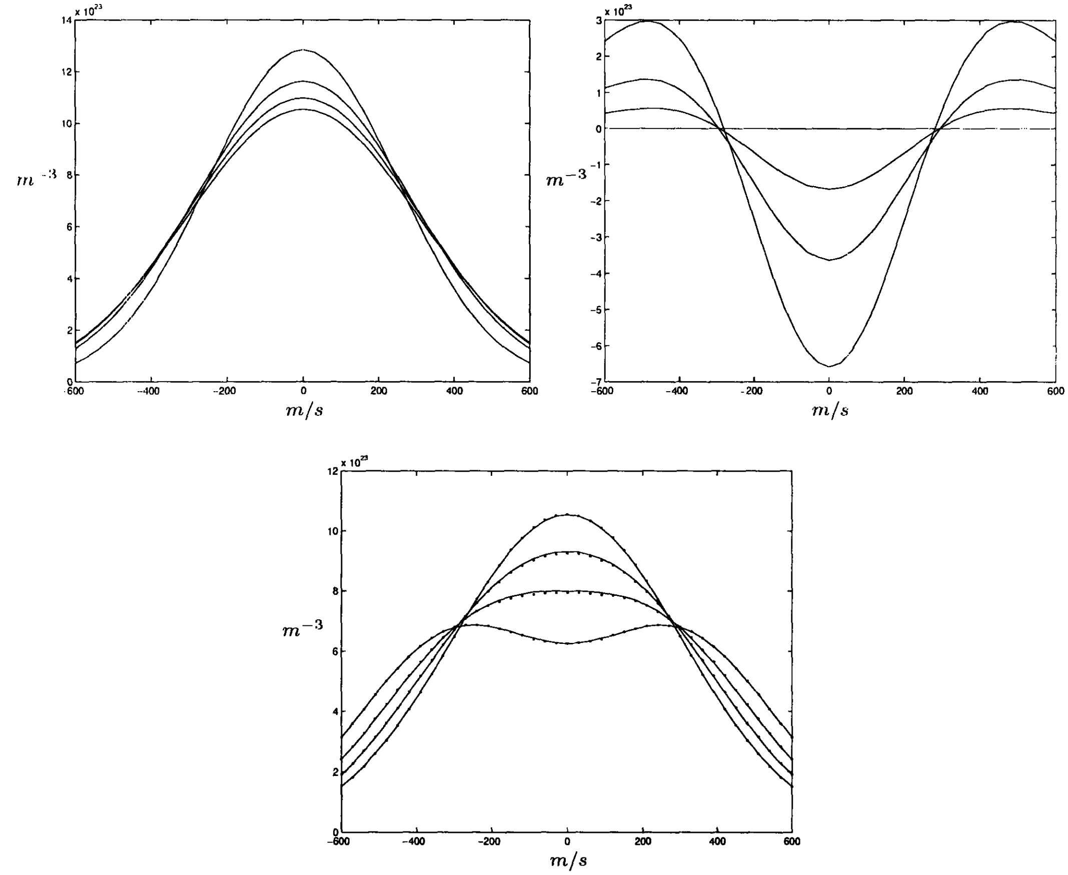
Regarding the final state, it can be computed using the fact that the collision operator conserves mass, momentum and kinetic energy, and that the final distribution is a Maxwellian. The total number density of the initial distribution is n_0, its momentum is zero and the density of kinetic energy is $(3/2)n_0mv_0^2 + n_0mv_s^2$. The density of kinetic energy of the final Maxwellian is $(3/2)n_0mv_\infty^2$ which means that $v_\infty = \sqrt{v_0^2 + (2/3)v_s^2}$. Let us mention that a Taylor expansion of this expression versus v_s, show that the problem will be linear if $v_s^2/v_0^2 << 1$. In this case, the following property indeed hold: $v_\infty \approx v_0$. The value of the asymptotic fourth moment is $< v_x^4 >_\infty = (3/4)v_\infty^4$.

Various sets of simulations were carried out. The full non-linear algorithm was tested on a highly non-linear case, obtained by setting v_s to a value of the same order of magnitude as v_0. This version was also tested on a different initial condition, namely a distribution with a nonzero mean velocity. The purpose of this latter test is to check that the algorithm satisfies the condition that the relaxation does not depend on the frame in which it takes place. The linear version of the algorithm was also tested. To that effect, v_s is set to a small value compared to v_0. The purpose is to validate the ability of the proposed method to capture low-signal flows.

8.1 Nonlinear Case Without Mean Velocity

In this section we test the algorithm on a highly nonlinear case. The parameter v_s is set to $v_s = 300\ m/s$ which is about $0.85v_0$. The characteristic velocity of the asymptotic Maxwellian is then $v_\infty = 1.2166v_0 = 430.05\ m/s$. The corresponding asymptotic value of $< v_x^4 >$ is $< v_x^4 >_\infty = 2.565 \cdot 10^{10}\ m^4/s^4$.

As discussed before, the charactersitic parameters of the local Maxwellian $n_{mb}(t)$, $v_{mb}(t)$, and $\boldsymbol{u_{mb}}(t)$, enables us to construct the local Maxwellian and plot its evolution. We also stored similar data for the deviational distribution. By summing these two distributions, we obtain the total distribution that we can compare to our reference, namely a DSMC solution. These three distributions are gathered in *Figure* 8-1. The distribution function is very close to the reference at all times. Moreover, we observe that the deviational distribution function is zero everywhere after a few relaxation times. This means that the rationale on which this algorithm was designed is valid. The local Maxwellian converges to the distribution that makes the deviational part as small as possible.



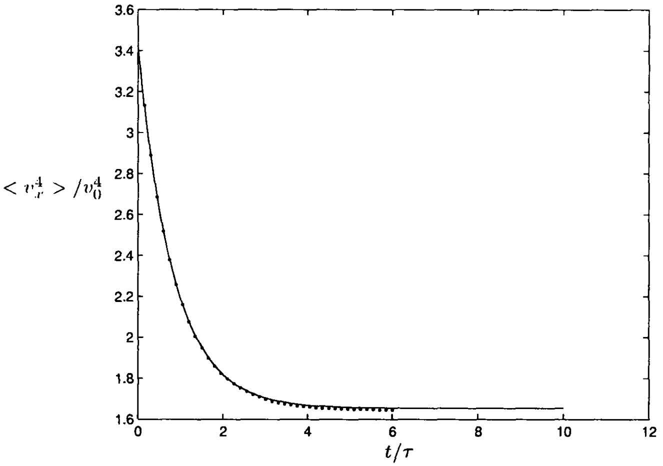
Distribution functions at various time:

Top left: f_{mb}, *Top right:* f_d *Bottom :* f

continuous line: present method, dots: standard DSMC

Figure 8-1:

Figure 8-2 displays the evolution of $< v_x^4 >$. Here again, the agreement with DSMC is excellent. In addition, we can check that the asymptotic value ($< v_x^4 >_\infty = 1.6541 \cdot v_0^4$) matches the analytical prediction ($< v_x^4 >_\infty = 1.643 \cdot v_0^4$). The gap is only 0.6%.

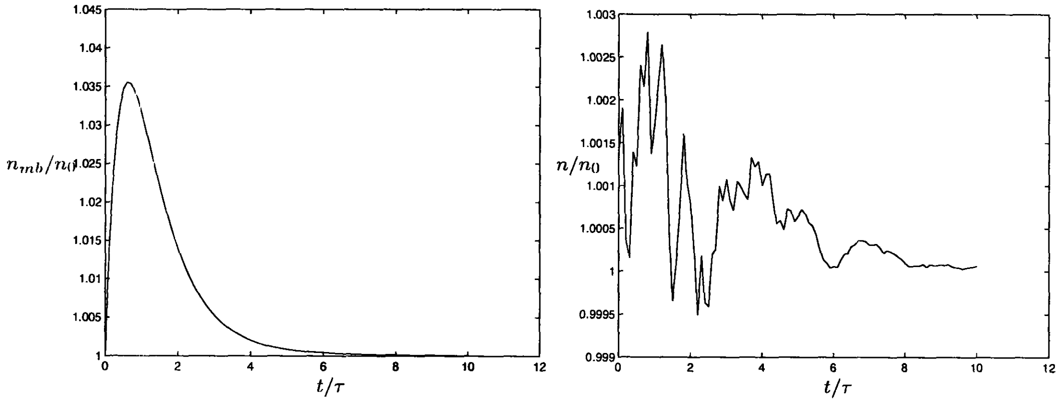


$< v_x^4 > /v_0^4$ versus time

continuous line: present method, dots: standard DSMC

Figure 8-2:

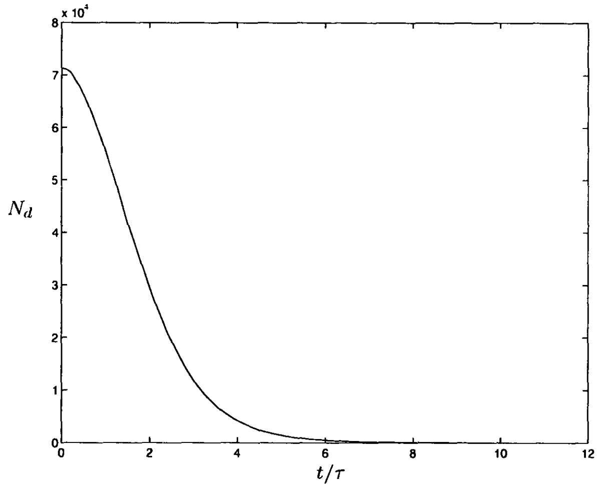
We also may want to check that some basic quantities are conserved. In fact, contrary to DSMC, we recall that the present algorithm does not guarantee the conservation of mass, momentum and energy but is expected to conserve them on avergage. As shown in *Figure* 8-3 $n_{mb}(t)$ undergoes some significant variations, but the total density made up of $n_{mb}(t)$ plus the net deviational density is constant. There is a very small loss of mass of about 0.1%.



Evolution of the number density of the Maxwellian (left), and of the total distribution (right)

Figure 8-3:

Finally, the evolution of the number of numerical particles is plotted in *Figure* 8-4. We observe that it decreases quite fast and ultimatly goes to zero. Let us emphasize that it is not a basic implication of the fact that the deviational part ends up being zero. In previous variance reduction algorithm [5, 6], we could have a distribution equal to zero while still having a lot of numerical particles whose net effect cancelled out. Here, the use of the kernels enables us to create particles that account only for the *net* effect of the collisions and which are therefore *precancelled.*



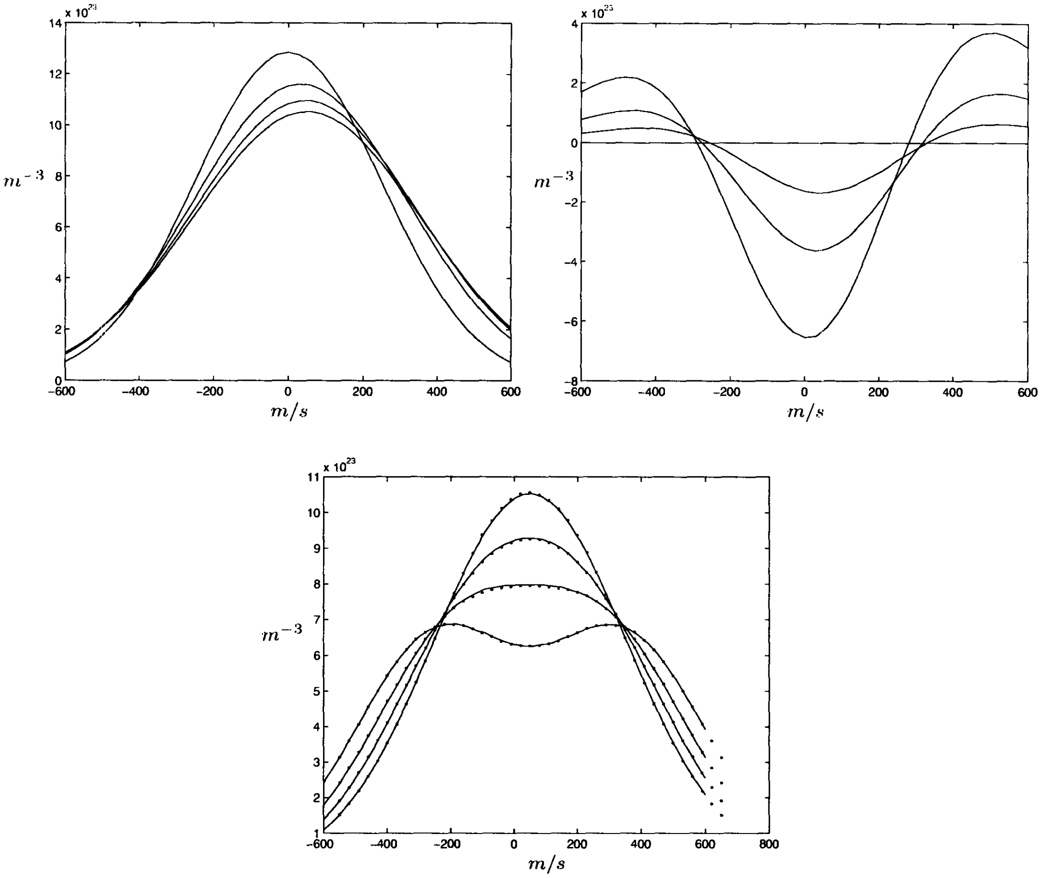
Evolution of the number of numerical particles (N_d)

Figure 8-4:

8.2 Nonlinear Case With Mean Velocity

This testcase differs from the previous one only in the fact that the initial distribution function has a nonzero mean velocity while the initial Maxwellian is the same as before, namely without mean velocity. We thus can validate the adaptation of the local Maxwellian to changes in mean velocity. Physically, the evolution of the distribution function is exactly the same as before exept that it happens in a frame moving at the mean velocity. We can then compute the quantities of interest of the final state analytically as before.

The distribution functions are plotted in *Figure* 8-5.



Distribution functions at various time:

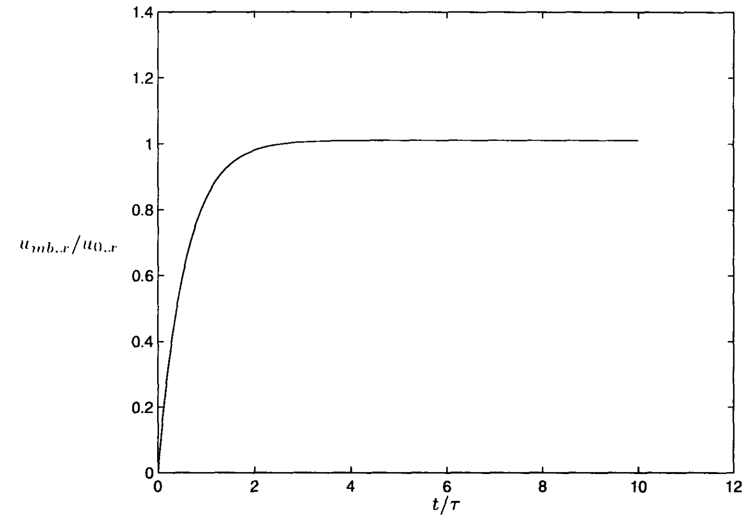
Top left:f_{mb}, Top right:f_d Bottom :f

continuous line: present method, dots: standard DSMC

Figure 8-5:

We observe that the local Maxwellian distribution moves toward a Maxwellian distribution centered on the mean velocity of the system. As before, the deviational distribution converges to zero, meaning that at the end, the local Maxwellian distribution is equal to the actual final distribution. Here again, the agreement with results from a standard DSMC code are excellent.

Figure 8-6 shows the evolution of $u_{mb,x}$ versus time. We can thus check that the mean velocity of the local Maxwellian converges quickly and acurately toward the mean velocity of the system.



Evolution of $u_{mb,x}$, the mean velocity of the local Maxwellian along the x-axis

Figure 8-6:

8.3 Linear Case Without Mean Velocity

In this section we test the algorithm on a linear spacially homogeneous relaxation case. The parameter v_s is now set to $v_s = 30\ m/s$ which is about $0.085 v_0$. The most probable velocity of the asymptotic Maxwellian is then $v_\infty = 1.0024 v_0 = 354.32\ m/s$. The final equilibrium will then be very close to the initial distribution. The corresponding asymptotic value of $< v_x^4 >$ is $< v_x^4 >_\infty = 1.1821 \cdot 10^{10}\ m^4/s^4$.

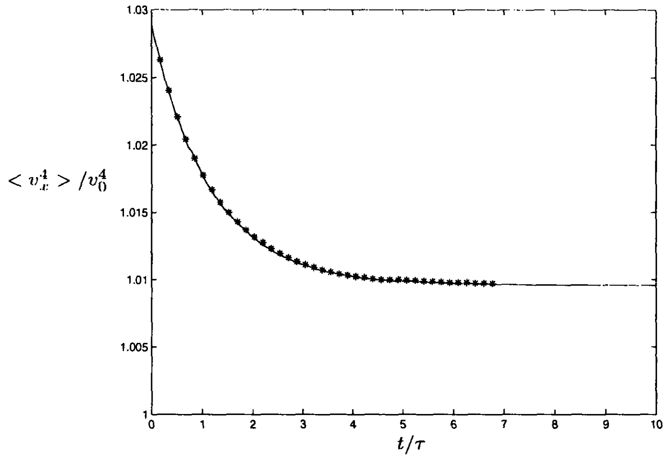
The purpose of this test is two-fold:

* Demonstrate the ability of the code to calculate low-signal flows and compare its efficiency with DSMC.

* Test the linear version of the algorithm.

The comparison of the evolution of $< v_x^4 >$ is plotted in *Figure* 8-7. We observe that the agrement with DSMC remains excellent with the linear version of the algorithm in such situations where the linearization of the Boltzmann collision operator is valid. The same behavior as in previous case is observed. The number of numerical particles goes to zero, and the local Maxwellian moves toward the final one. In this case, the changes in the distribution function are very small and a visual inspection of the evolution is not possible.

In this testcase, the proposed method appeared to be more than two orders of magnitude faster than standard DSMC.



$< v_x^4 >$ versus time

continuous line: present method,   dots: standard DSMC

Figure 8-7:

Chapter 9

Couette Flow Testcase

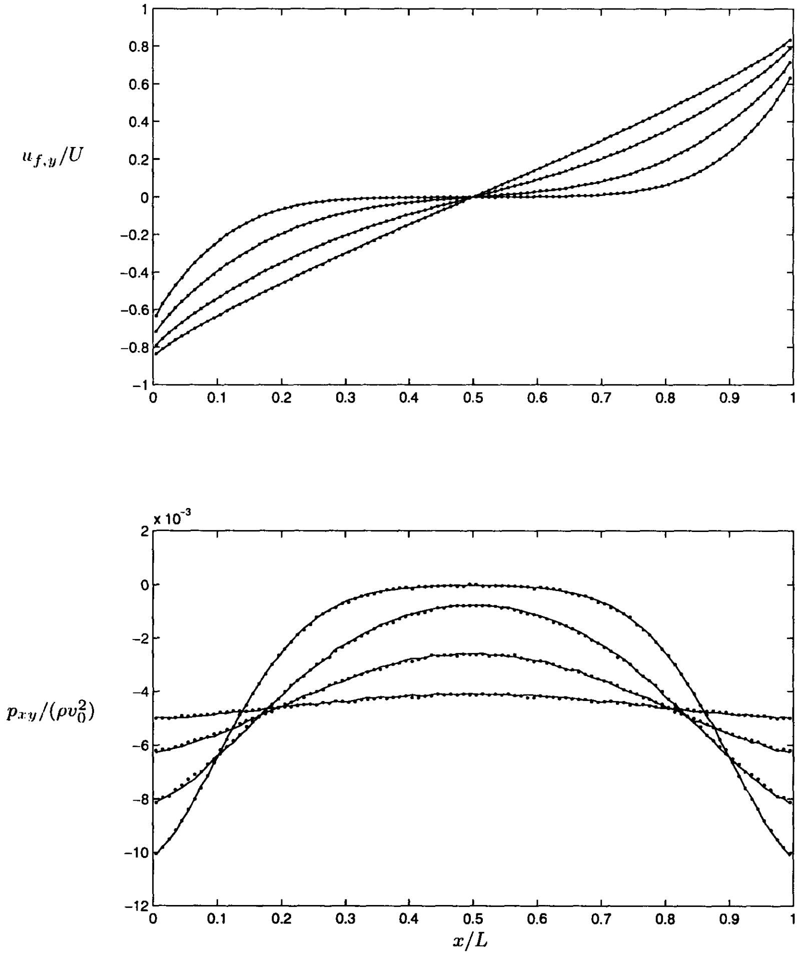
A number of validation tests for spatially dependent problems have been performed. Here, we show the results for an impulsively started shear flow, in which at time $t = 0$ the two (diffuse) walls bounding the system start moving in opposite directions in their plane with velocity $\pm U$. The gas is Argon at an initial temperature $T_0 = 300\ K$. The distance between the two walls L is choosen such that the knudsen number varies between 0.1 and 10. Two sets of simulation were performed:

* One set of simulations was performed with $U = 0.05v_0$. This represents a good compromise between low speed flow (for testing the linear version and the ability of the method to catch low-signal flow) and sufficient signal (such that a DSMC solution is feasible). We performed simulations at $kn = 10$ (advection dominated flow), $kn = 1$ (intermediate) and $kn = 0.1$ (collision dominated flow).

* One set of simulations with $U = v_0$ and $kn = 0.1,\ 1,\ 10$ (the last one is not shown here) to test the ability of the algorithm to deal with nonlinear flows.

9.1 Linear Shear Flow

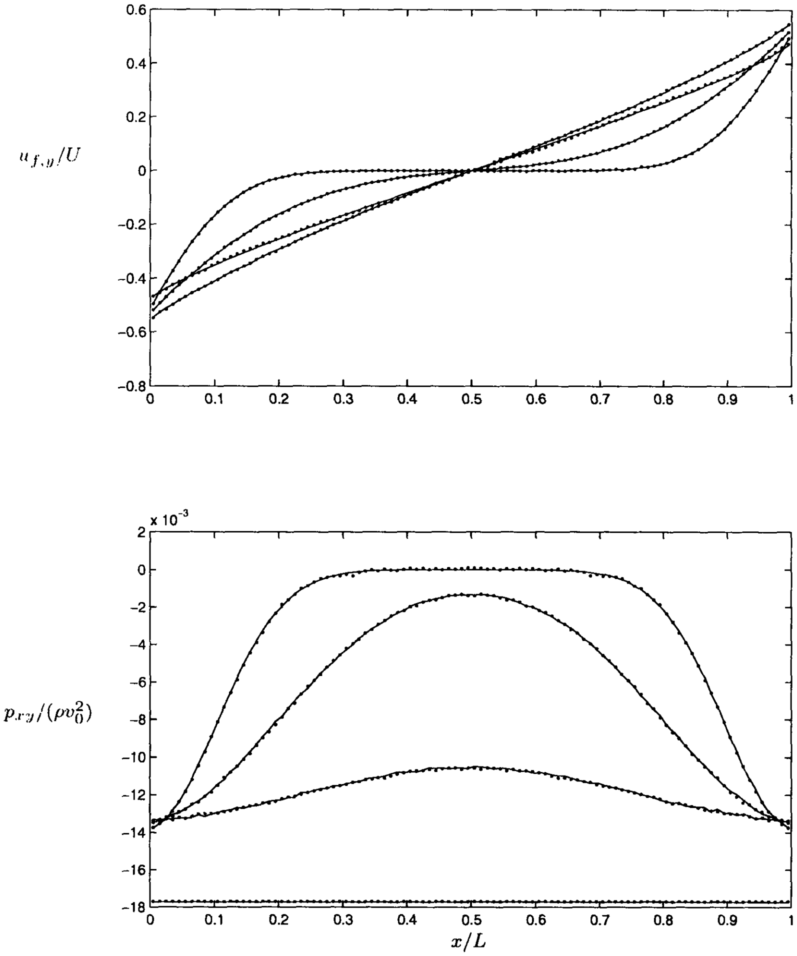
The linear version of the algorithm was tested by simulating the shear flow described above, with $u_{wall,y} = U = 0.05v_0$. This value is sufficiently small to expect a linear behavior and good agreement with DSMC. The outputs used to carry out the comparison are the transverse velocity profiles $u_{f,y}(x)$ and shear stress profiles $p_{xy}(x)$ (defined in Chapter 6) at various times and for $kn = 0.1$, $kn = 1$ and $kn = 10$. Results are given in *Figures* 9-1, 9-2 and 9-3.



Velocity profiles (top) and shear stress profiles (bottom) at various times, for $kn = 0.1$ and $U = 0.05v_0$.

Continuous line: present method, dots: standard DSMC

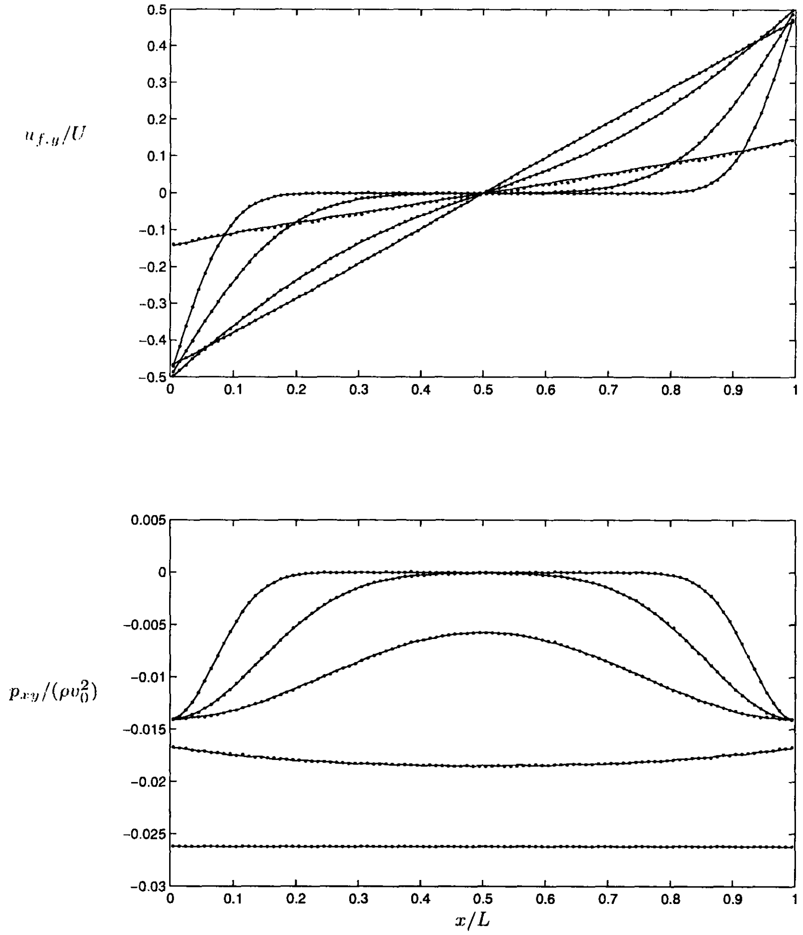
Figure 9-1:



Velocity profiles (top) and shear stress profiles (bottom) at various times, for $kn = 1$ *and* $U = 0.05v_0$.

Continuous line: present method, dots: standard DSMC

Figure 9-2:



Velocity profiles (top) and shear stress profile (bottom) at various times, for $kn = 10$ and $U = 0.05v_0$.

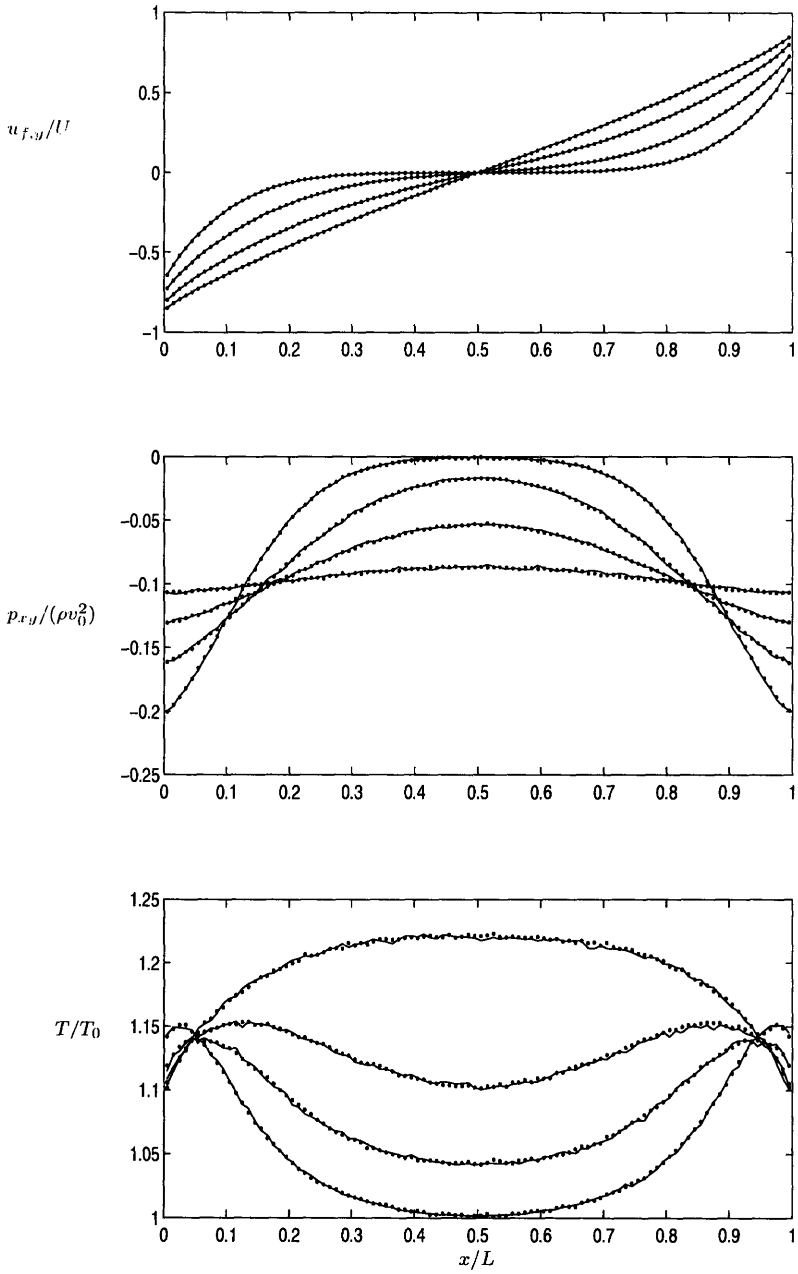
Continuous line: present method, dots: standard DSMC

Figure 9-3:

We notice a very good agreement with DSMC for all Knudsen numbers and for the transient regime as well as for the steady state. Here, the proposed method appeared to be about 1 order of magnitude faster than standard DSMC.

9.2 Non-Linear Shear Flow

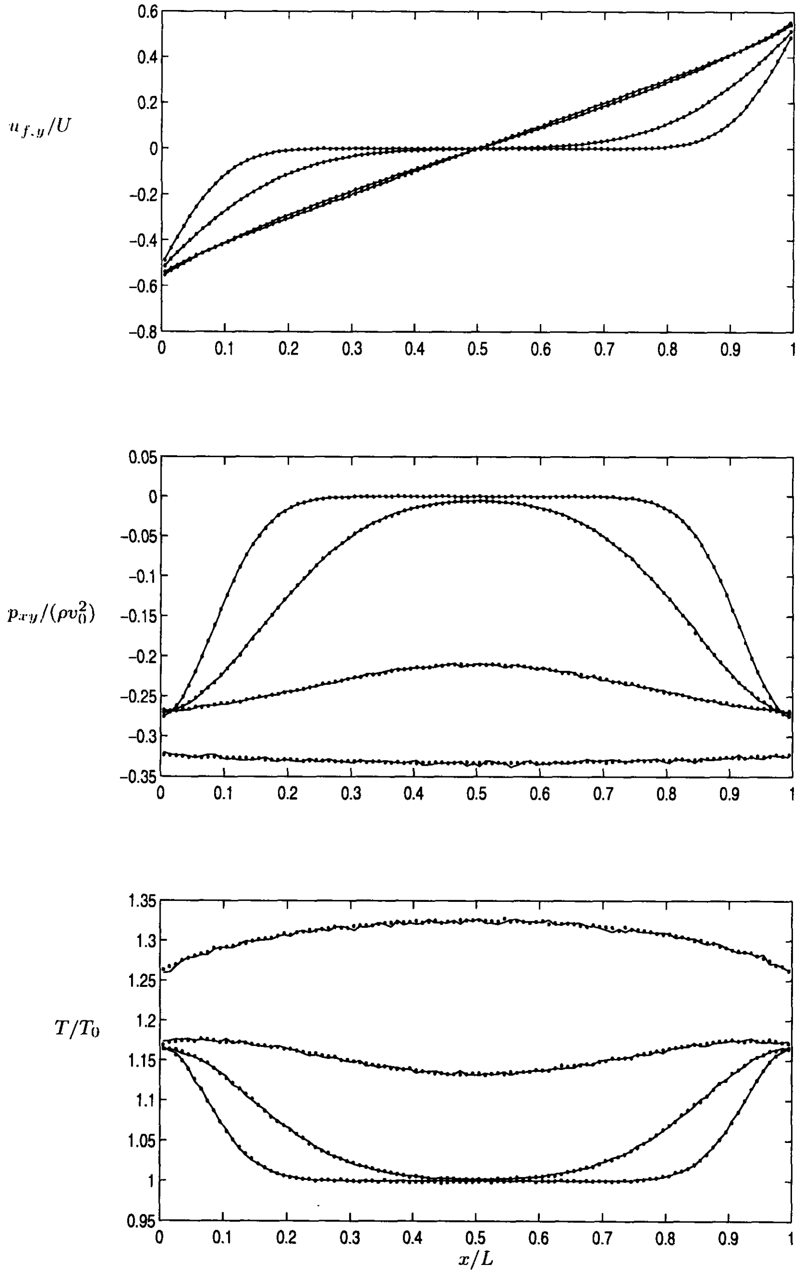
In this section, the algorithm is tested by simulating the same shear flow as before, except that $u_{wall,y} = U = v_0$. The validation is made through the comparison of the same outputs. Results are shown in *Figures* 9-4 and 9-5.



Velocity profiles (top), shear stress profiles (middle) and temperature profiles (bottom) at various times, for $kn = 0.1$ *and* $U = v_0$.

Continuous line: present method, dots: standard DSMC

Figure 9-4:



Velocity profiles (top), shear stress profiles (middle) and temperature profiles (bottom) at various times, for $kn = 1$ and $U = v_0$.

Continuous line: present method, dots: standard DSMC

Figure 9-5:

Flows with $U/v_0 > 1$ lead to a divergence in the number of particles.

Chapter 10

Conclusions

A new particle method for solving the Boltzmann equation has been developed. This method can capture arbitrarily small deviations from equilibrium at a computational cost that does not scale with this deviation. This is achieved by simulating only the deviation from equilibrium, as originally proposed by Baker and Hadjiconstantinou [4]. The proposed method is closely related to DSMC and differs only in the ways necessary to consider the deviation from equilibrium. The most important feature of this method, is that it requires no particle cancellation and thus no discretization in velocity space. This is achieved by allowing the local equilibrium distribution function to change as a result of the action of the collision integral and thus enabling the generation of a minimal irreductible set of deviational particles around an "optimal" local Maxwellian at each time step.

The resulting algorithm is significantly faster than DSMC in the limit of low-sigal flow. For example, in the low-speed validation problems of Chapter 8, the proposed method is at least one order of magnitude faster than a mature and optimized DSMC code. Moreover, the proposed method can simulate flows with smaller characteristic velocities at the same cost, while the cost of DSMC increases quadratically [11] as the signal decreases. Our results also indicated that with the addition of the nonlinear term $C(f_d, f_d)$, nonlinear flows up to Mach number $M_a \approx 1$ can be simulated, even though the proposed method no longer holds an efficiency advantage over DSMC. Beyond $M_a \approx 1$, the number of deviational particles grows without bound.

Possible future work may focus on number of improvements. First, various parts of the algorithm can be optimized; this is particularly true for the acceptance/rejection routines which can be made more efficient or replaced by more sophisticated algorithms. Second, a serious investigation of the impact of the various parameters introduced (e.g the cutoff v_c for regularizing the singularity of kernel K_1) remains to be carried out. Finally, a better understanding of the stability limits of the method for large deviations from equilibrium is also of interest, albeit mostly academic.

Appendix A

Collision Integral Kernels

A.1 Derivation of the Kernels for a centered Maxwellian

The collision term of the Boltzmann equation is:

$$C(f,f)(\boldsymbol{v}) = \int\int (f'f'_1 - ff_1)|\boldsymbol{v} - \boldsymbol{v_1}|\sigma d\Omega dv_1$$

We know [3] that it can be rewriten as:

$$C(f,f)(\boldsymbol{v}) = \frac{1}{2}\int\int\int (\delta'_1 + \delta'_2 - \delta_1 - \delta_2)f_1 f_2|\boldsymbol{v_2} - \boldsymbol{v_1}|\sigma d\Omega dv_1 dv_2$$

When $f = f_{mb} + f_d$, it becomes:

$$C(f,f)(\boldsymbol{v}) = \frac{1}{2}\int\int\int (\delta'_1 + \delta'_2 - \delta_1 - \delta_2)(f_{d1}f_{d2} + f_{mb1}f_{d2} + f_{d1}f_{mb2})|\boldsymbol{v_2} - \boldsymbol{v_1}|\sigma d\Omega dv_1 dv_2$$

Due to its symmetry, the linear part is

$$\frac{1}{2}\int\int\int (\delta'_1 + \delta'_2 - \delta_1 - \delta_2)(f_{mb1}f_{d2} + f_{d1}f_{mb2})|\boldsymbol{v_2} - \boldsymbol{v_1}|\sigma d\Omega dv_1 dv_2 =$$
$$\int\int\int (\delta'_1 + \delta'_2 - \delta_1 - \delta_2)f_{d1}f_{mb2}|\boldsymbol{v_2} - \boldsymbol{v_1}|\sigma d\Omega dv_1 dv_2$$

For a given pair of precollision velocities $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$, the postcollision velocities $\boldsymbol{v'_1}$ and $\boldsymbol{v'_2}$ run on a sphere centered on the mean velocity $(\boldsymbol{v_1} + \boldsymbol{v_2})/2$ and of radius the relative velocity $|\boldsymbol{v_2} - \boldsymbol{v_1}|$. This means that $\boldsymbol{v_1'}$ and $\boldsymbol{v_2'}$ can be switched without changing the value of the integral. From this, the linear part of the collision integral becomes:

$$\int\int\int (2\delta_1' - \delta_1 - \delta_2) f_{d1} f_{mb2} |\boldsymbol{v_2} - \boldsymbol{v_1}| \sigma d\Omega dv_1 dv_2$$

This integral can be split into 3 integrals which will give rise to the kernels K_1 and K_2 and to the collision frequency ν.

A.1.1 Kernel K_2

The second kernel comes from the term $\int\int\int \delta_2 f_{d1} f_{mb2} |\boldsymbol{v_2} - \boldsymbol{v_1}| \sigma d\Omega dv_1 dv_2$. In this expression, the scattering angle does not play any part since the postcollision velocities do not appear. The integral over the scattering angle can be taken out and simply give 4π. Because of the dirac $\delta(\boldsymbol{v} - \boldsymbol{v_2})$, the integral over $\boldsymbol{v_2}$ will simply be the value of the intregrant for $\boldsymbol{v} = \boldsymbol{v_2}$. We are left with $4\pi \int f_d(\boldsymbol{v_1}) f_{mb}(\boldsymbol{v}) |\boldsymbol{v} - \boldsymbol{v_1}| \sigma dv_1$. The kernel appears naturally and is:

$$K_2(\boldsymbol{v}, \boldsymbol{v_1}) = \frac{\pi d^2 n_{mb}}{\pi^{3/2} v_{mb}^3} \exp(-\frac{\boldsymbol{v}^2}{v_{mb}^2}) |\boldsymbol{v} - \boldsymbol{v_1}| = \frac{d^2 n_{mb}}{\pi^{1/2} v_{mb}^2} \exp(-\boldsymbol{v}^{*2}) |\boldsymbol{v}^* - \boldsymbol{v_1}^*|$$

since $\sigma = d^2/4$, where d is the diameter of the hard sphere molecules and because $f_{mb}(\boldsymbol{v}) = \frac{n_{mb}}{\pi^{3/2} v_{mb}^3} \exp(-\boldsymbol{v}^2/v_{mb}^2)$.

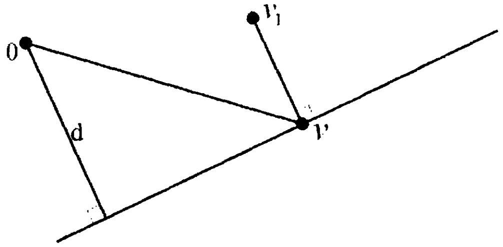
For convenience, we introduce the prefactor $\mu_1 = \frac{d^2 n_{mb}}{\pi^{1/2} v_{mb}^2}$. The kernel is then $K_2(\boldsymbol{v}, \boldsymbol{v_1}) = \mu_1 \exp(-\boldsymbol{v}^{*2}) |\boldsymbol{v}^* - \boldsymbol{v_1}^*|$.

A.1.2 Kernel K_1

The first kernel comes from the term $\int\int\int 2\delta_1' f_{d1} f_{mb2} |\boldsymbol{v_2} - \boldsymbol{v_1}| \sigma d\Omega dv_1 dv_2$. In [8], it is shown that this term is equal to:

$$2\int_{\boldsymbol{v_1}\in\mathbb{R}} f_d(\boldsymbol{v_1})\frac{4\sigma}{|\boldsymbol{v}-\boldsymbol{v_1}|}\int_{\boldsymbol{v_2}\in\mathcal{P}} f_{mb}(\boldsymbol{v_2})dv_1 dv_2$$

where $\mathcal{P}$ is the plane orthogonal to $\boldsymbol{v}-\boldsymbol{v_1}$ and that passes through $\boldsymbol{v_1}$. The second integral is thus an integral of a 3 dimensional Maxwellian over a plane. By means of appropriate change of variable, it is easy to show that it is simply $\pi v_{mb}^2 f_{mb}\left[\frac{\boldsymbol{v}\cdot(\boldsymbol{v}-\boldsymbol{v_1})}{|\boldsymbol{v}-\boldsymbol{v_1}|}\right]$. The factor πv_{mb}^2 comes from the integration of the Maxwellian over the plane and the expression $d=\frac{\boldsymbol{v}\cdot(\boldsymbol{v}-\boldsymbol{v_1})}{|\boldsymbol{v}-\boldsymbol{v_1}|}$ is nothing but the distance between the origin of the axis $\mathbf{0}$ and the plane, which we get by projecting $\boldsymbol{v}$ (a point that stands on the plane) on the unit normal $\frac{\boldsymbol{v}-\boldsymbol{v_1}}{|\boldsymbol{v}-\boldsymbol{v_1}|}$.



Following this, the whole term is:

$$2\int_{\boldsymbol{v_1}\in\mathbb{R}} f_d(\boldsymbol{v_1})\frac{4\sigma}{|\boldsymbol{v}-\boldsymbol{v_1}|}\pi v_{mb}^2 f_{mb}\left[\frac{\boldsymbol{v}\cdot(\boldsymbol{v}-\boldsymbol{v_1})}{|\boldsymbol{v}-\boldsymbol{v_1}|}\right] dv_1$$

This integral is a convolution of f_d against a kernel K_1 which is:

$$K_1(\boldsymbol{v},\boldsymbol{v_1}) = 2\frac{4\sigma}{|\boldsymbol{v}-\boldsymbol{v_1}|}\pi v_{mb}^2 f_{mb}\left[\frac{\boldsymbol{v}\cdot(\boldsymbol{v}-\boldsymbol{v_1})}{|\boldsymbol{v}-\boldsymbol{v_1}|}\right]$$

where we have $f_{mb}\left[\frac{\boldsymbol{v}\cdot(\boldsymbol{v}-\boldsymbol{v_1})}{|\boldsymbol{v}-\boldsymbol{v_1}|}\right] = \frac{n_{mb}}{\pi^{3/2}v_{mb}^3}\exp\left[-\left(\frac{\boldsymbol{v^*}\cdot(\boldsymbol{v^*}-\boldsymbol{v_1^*})}{|\boldsymbol{v^*}-\boldsymbol{v_1^*}|}\right)^2\right]$, $\sigma = d^2/4$ for hard sphere molecules, and where $1/|\boldsymbol{v}-\boldsymbol{v_1}|$ can be written as $1/(v_{mb}|\boldsymbol{v^*}-\boldsymbol{v_1^*}|)$, which finally leads to:

$$K_1(\boldsymbol{v},\boldsymbol{v_1}) = \frac{2\mu_1}{|\boldsymbol{v^*}-\boldsymbol{v_1^*}|}\exp\left[-\left(\frac{\boldsymbol{v^*}\cdot(\boldsymbol{v^*}-\boldsymbol{v_1^*})}{|\boldsymbol{v^*}-\boldsymbol{v_1^*}|}\right)^2\right]$$

where we recall that we denote $\mu_1 = \frac{d^2 n_{mb}}{\pi^{1/2}v_{mb}^2}$.

A.1.3 Collision frequency function

The collision frequency function comes from the term $\int\int\int \delta_1 f_{d1} f_{mb2} |\boldsymbol{v_2} - \boldsymbol{v_1}| \sigma d\Omega dv_1 dv_2$. As previously, the integral over the scattering angle can be taken out and results in a factor of 4π. The dirac lies now on $\boldsymbol{v_1}$: $\delta(\boldsymbol{v} - \boldsymbol{v_1})$. The integral over $\boldsymbol{v_1}$ is equal to the integrand taken in $\boldsymbol{v_1} = \boldsymbol{v}$. The whole term reduces to $4\pi\sigma f_d(\boldsymbol{v}) \int f_{mb}(\boldsymbol{v_2}) |\boldsymbol{v_2} - \boldsymbol{v}| dv_2$.

It can be rewritten as:

$$\begin{aligned} 4\pi\sigma f_d(\boldsymbol{v}) \int f_{mb}(\boldsymbol{v_2}) |\boldsymbol{v_2} - \boldsymbol{v}| d^3 v_2 &= \frac{4\pi\sigma n_{mb}}{\pi^{3/2} v_{mb}^3} f_d(\boldsymbol{v}) \int \exp(-\boldsymbol{v_2}^2 / v_{mb}^2) v_{mb} |\frac{\boldsymbol{v_2}}{\bar{v}} - \frac{\boldsymbol{v}}{v_{mb}}| d^3 v_2 \\ &= \frac{4\pi\sigma n_{mb}}{\pi^{3/2} v_{mb}^2} v_{mb}^3 f_d(\boldsymbol{v}) \int \exp(-\boldsymbol{v_2^{*2}}) |\boldsymbol{v_2^*} - \boldsymbol{v}^*| d^3 v_2^* \end{aligned}$$

The term $\nu(\boldsymbol{v}^*) = \int \exp(-\boldsymbol{v_2^{*2}}) |\boldsymbol{v_2^*} - \boldsymbol{v}^*| d^3 v_2^*$ is just a function of $\boldsymbol{v}^*$. It can be calculated by the change of variables $\tilde{\boldsymbol{v}}_{\boldsymbol{2}}^* = \boldsymbol{v_2^*} - \boldsymbol{v}^*$:

$$\nu(\boldsymbol{v}^*) = \int \exp(-\boldsymbol{v_2^{*2}}) |\boldsymbol{v_2^*} - \boldsymbol{v}^*| d^3 v_2^* = \int \exp(-(\boldsymbol{v}^* + \tilde{\boldsymbol{v}}_{\boldsymbol{2}}^*)^2) |\tilde{\boldsymbol{v}}_{\boldsymbol{2}}^*| d^3 \tilde{v}_2^*$$

We can then switch to spherical coordinates with $\boldsymbol{v}^*/|\boldsymbol{v}^*|$ as z-axis. the integral becomes:

$$\begin{aligned} \nu(\boldsymbol{v}^*) &= \iiint \exp(-(|\boldsymbol{v}^*| \boldsymbol{e_z} + r\boldsymbol{e_r})^2) r^3 sin\theta dr d\theta d\phi \\ &= \iiint \exp(-|\boldsymbol{v}^*|^2 - r^2 - 2|\boldsymbol{v}^*| r \cos\theta) r^3 sin\theta dr d\theta d\phi \end{aligned}$$

The integral versus ϕ results in a prefactor of 2π. The integral versus θ can be done explicitely and leads to $\frac{2\pi r^2}{2|\boldsymbol{v}^*|}(\exp(-|\boldsymbol{v}^*|^2 - r^2 + 2|\boldsymbol{v}^*|r) - \exp(-|\boldsymbol{v}^*|^2 - r^2 - 2|\boldsymbol{v}^*|r))$ The integral of this latter function versus r is carried out by parts. The integrand can be written as $\frac{2\pi r^2 \exp(-|\boldsymbol{v}^*|^2)}{2|\boldsymbol{v}^*|}(\exp(-r^2 + 2|\boldsymbol{v}^*|r) - \exp(-r^2 - 2|\boldsymbol{v}^*|r))$ The integration by parts leads to:

$$\nu(\boldsymbol{v}^*) = \frac{2\pi\exp(-|\boldsymbol{v}^*|^2)}{2|\boldsymbol{v}^*|}\left(\frac{1}{2}\sqrt{\pi}\exp(\boldsymbol{v}^{*2})\mathrm{erf}(|\boldsymbol{v}^*|) + |\boldsymbol{v}^*| + \boldsymbol{v}^{*2}\sqrt{\pi}\exp(\boldsymbol{v}^{*2})\mathrm{erf}(|\boldsymbol{v}^*|)\right)$$

After some cancellations, we get the final expression:

$$\nu(\boldsymbol{v}^*) = \frac{\pi}{2|\boldsymbol{v}^*|}\left(\sqrt{\pi}\mathrm{erf}(|\boldsymbol{v}^*|) + 2|\boldsymbol{v}^*|\exp(-\boldsymbol{v}^{*2}) + 2\sqrt{\pi}\boldsymbol{v}^{*2}\mathrm{erf}(|\boldsymbol{v}^*|)\right)$$

Coming back to the initial expression, the whole term is

$$\frac{4\pi\sigma n_{mb}v_{mb}}{\pi^{3/2}}f_d(\boldsymbol{v})\nu(\boldsymbol{v}^*) = \frac{d^2 n_{mb}v_{mb}}{\pi^{1/2}}f_d(\boldsymbol{v})\nu(\boldsymbol{v}^*) = \mu_2 f_d(\boldsymbol{v})\nu(\boldsymbol{v}^*)$$

where we used the fact that $\sigma = d^2/4$ and where we denote $\mu_2 = v_{mb}^3\mu_1 = \frac{d^2 n_{mb}v_{mb}}{\pi^{1/2}}$. μ_2 has the dimension of a frequency. $\mu_2\nu(\boldsymbol{v}^*)$ is the collision frequency of a particle at velocity $\boldsymbol{v}$ with the entire Maxwellian.

In summary, the collision term can be written as:

$$C(f,f)(\boldsymbol{v}) = \int K_1(\boldsymbol{v},\boldsymbol{v_1})f_d(\boldsymbol{v_1})d^3\boldsymbol{v_1} - \int K_2(\boldsymbol{v},\boldsymbol{v_1})f_d(\boldsymbol{v_1})d^3\boldsymbol{v_1} - \mu_2 f_d(\boldsymbol{v})\nu(\boldsymbol{v}^*) + C(f_d,f_d)$$

For this case where the local Maxwellian has no mean velocity, this expression was proposed under a slighty different form in [2].

A.2 Kernels for a shifted Maxwellian ($\boldsymbol{u_{mb}} \neq 0$)

All the above derivations have been carried out while assuming that the maxwellian was simply $f_{mb} = n_{mb}/(\pi^{3/2}v_{mb}^3)\exp(-\boldsymbol{v}^2/v_{mb}^2)$. In this part, we will derive the kernel when the local maxwellian is not centered on zero but is equal to $f_{mb} = n_{mb}/(\pi^{3/2}v_{mb}^3)\exp(-(\boldsymbol{v}-\boldsymbol{u_{mb}})^2/v_{mb}^2)$.

To deal with the fact that the Maxwellian is not centered on zero, we will be doing the change of variables $\boldsymbol{w} = \boldsymbol{v} - \boldsymbol{u_{mb}}$, where $\boldsymbol{u_{mb}}$ is the mean velocity of the local

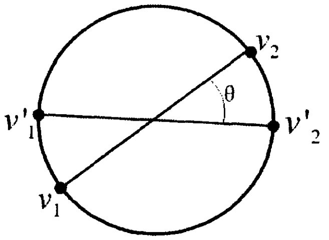
Maxwellian. For a given pair of precollision velocities $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$. the corresponding postcollision velocities can formally be expressed as:

$$\left\{\begin{array}{lcl} v_1' & = & \frac{\boldsymbol{v_1}+\boldsymbol{v_2}}{2} + \boldsymbol{\mathcal{R}}(\theta,\phi)\cdot\frac{\boldsymbol{v_1}-\boldsymbol{v_2}}{2} \\ v_2' & = & \frac{\boldsymbol{v_1}+\boldsymbol{v_2}}{2} - \boldsymbol{\mathcal{R}}(\theta,\phi)\cdot\frac{\boldsymbol{v_1}-\boldsymbol{v_2}}{2} \end{array}\right.$$

where $\boldsymbol{\mathcal{R}}(\theta,\phi)$ is the matrix of a rotation with a scattering angle equal to θ. Replacing $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ by respectively $\boldsymbol{u_{mb}}+\boldsymbol{w_1}$ and $\boldsymbol{u_{mb}}+\boldsymbol{w_2}$, we see that the postcollision velocities expressed in terms of the shifted velocities $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ are:

$$\left\{\begin{array}{lcccl} v_1' & = & \boldsymbol{u_{mb}}+\frac{\boldsymbol{w_1}+\boldsymbol{w_2}}{2} + \boldsymbol{\mathcal{R}}(\theta,\phi)\cdot\frac{\boldsymbol{w_1}-\boldsymbol{w_2}}{2} & = & \boldsymbol{u_{mb}}+\boldsymbol{w_1'} \\ v_2' & = & \boldsymbol{u_{mb}}+\frac{\boldsymbol{w_1}+\boldsymbol{w_2}}{2} - \boldsymbol{\mathcal{R}}(\theta,\phi)\cdot\frac{\boldsymbol{w_1}-\boldsymbol{w_2}}{2} & = & \boldsymbol{u_{mb}}+\boldsymbol{w_2'} \end{array}\right.$$

where $\boldsymbol{w_1'}$ and $\boldsymbol{w_2'}$ are the postcollision velocities corresponding to the precollision velocities $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$, that is to say, the same operator applied to $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$. In short, the postcollision velocities of the shifted velocities is equal to the shifted postcollision velocity of the nonshifted velocities.



A.2.1 Kernel K_2

As seen before, K_2 comes from the term $\int\int\int \delta_2 f_{d1} f_{mb2} |\boldsymbol{v_2}-\boldsymbol{v_1}|\sigma d\Omega dv_1 dv_2$. Now, the Maxwellian is $f_{mb}(\boldsymbol{v}) = n_{mb} v_{mb}^{-3}\pi^{-3/2}\exp[-(\boldsymbol{v}^* - \boldsymbol{u}_{mb}^*)^2]$, where we used again the notation $\boldsymbol{v}^* = \boldsymbol{v}/v_{mb}$ with v_{mb} being the most probable velocity of the Maxwellian. We can then rewrite this expression as $\int\int\int \delta(\boldsymbol{v}-\boldsymbol{v_2}) f_d(\boldsymbol{v_1}) n_{mb} v_{mb}^{-3}\pi^{-3/2}\exp[-(\boldsymbol{v_2^*}-\boldsymbol{u}^*)^2]|\boldsymbol{v_2}-\boldsymbol{v_1}|\sigma d\Omega dv_1 dv_2$. Doing the following change of variable:

$$\begin{cases} \boldsymbol{w_1} &= \boldsymbol{v_1} - \boldsymbol{u_{mb}} \\ \boldsymbol{w_2} &= \boldsymbol{v_2} - \boldsymbol{u_{mb}} \\ \boldsymbol{w} &= \boldsymbol{v} - \boldsymbol{u_{mb}} \\ d^3\boldsymbol{v_1} &= d^3\boldsymbol{w_1} \\ d^3\boldsymbol{v_2} &= d^3\boldsymbol{w_2} \end{cases}$$

the above integral can be turned into:

$$\tilde{K_2}f_d(\boldsymbol{v}) = \int\int\int \delta(\boldsymbol{w} - \boldsymbol{w_2})f_d(\boldsymbol{u_{mb}} + \boldsymbol{w_1})\frac{n_{mb}}{v_{mb}^3\pi^{3/2}}\exp(-\boldsymbol{w_2^{*2}})|\boldsymbol{w_2} - \boldsymbol{w_1}|\sigma d\Omega dw_1 dw_2$$

Note that $n_{mb}v_{mb}^{-3}\pi^{-3/2}\exp(-\boldsymbol{w_2^{*2}})$ as a function of the dummy variable w_2 is the nonshifted Maxwellian used in the previous section to derive the kernels. If we rename the dummy variables $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ into $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$, we have the same expression as the one that serves to derive the kernel, exept that it is applied to the function $\boldsymbol{v} \longrightarrow f_d(\boldsymbol{u_{mb}}+\boldsymbol{v})$ instead of $\boldsymbol{v} \longrightarrow f_d(\boldsymbol{v})$ and that the dirac bring into play the velocity $\boldsymbol{w}$ instead of $\boldsymbol{v}$. Following the results of the previous section, the whole integral is then equal to:

$$\tilde{K_2}f_d(\boldsymbol{v}) = \int K_2(\boldsymbol{w}, \boldsymbol{w_1})f_d(\boldsymbol{u_{mb}} + \boldsymbol{w_1})d^3\boldsymbol{w_1}$$

We can apply again the change of variable, but in the other way. This leads to

$$\tilde{K_2}f_d(\boldsymbol{v}) = \int K_2(\boldsymbol{v} - \boldsymbol{u_{mb}}, \boldsymbol{v_1} - \boldsymbol{u_{mb}})f_d(\boldsymbol{v_1})d^3\boldsymbol{v_1}$$

For convenience, we will use the notation:

$$\tilde{K_2}f_d(\boldsymbol{v}) = \int K_2(\boldsymbol{w}, \boldsymbol{w_1})f_d(\boldsymbol{v_1})d^3\boldsymbol{v_1}$$

A.2.2 Kernel K_1

The derivation for K_1 is almost the same as the one for K_2. The only difference is that it brings into play the postcollision velocities.

The initial term is $\tilde{K_1} f_d = 2 \int \int \int \delta(\boldsymbol{v} - \boldsymbol{v_1'}) f_d(\boldsymbol{v_1}) f_{mb}(\boldsymbol{v_2}) |\boldsymbol{v_2} - \boldsymbol{v_1}| \sigma d\Omega dv_1 dv_2$. Here again, we have $f_{mb}(\boldsymbol{v}) = n_{mb} v_{mb}^{-3} \pi^{-3/2} \exp[-(\boldsymbol{v^*} - \boldsymbol{u_{mb}^*})^2]$. By doing the same change of variable as previously, and using the fact that $\boldsymbol{w_1'}(\boldsymbol{w_1}, \boldsymbol{w_2}) = \boldsymbol{v_1'}(\boldsymbol{v_1}, \boldsymbol{v_2}) - \boldsymbol{u_{mb}}$, we get:

$$\tilde{K_1} f_d(\boldsymbol{v}) = \int \int \int \delta(\boldsymbol{w} - \boldsymbol{w_1'}) f_d(\boldsymbol{u_{mb}} + \boldsymbol{w_1}) \frac{n_{mb}}{v_{mb}^3 \pi^{3/2}} \exp(-\boldsymbol{w_2^{*2}}) |\boldsymbol{w_2} - \boldsymbol{w_1}| \sigma d\Omega dw_1 dw_2$$

As before, we get the term used to defined the kernel K_1 for a nonshifted Maxwellian applied to the function $\boldsymbol{v} \longrightarrow f_d(\boldsymbol{u_{mb}} + \boldsymbol{v})$ instead of $\boldsymbol{v} \longrightarrow f_d(\boldsymbol{v})$. Doing the same operations as before, the final expression is:

$$\tilde{K_1} f_d(\boldsymbol{v}) = \int K_1(\boldsymbol{w}, \boldsymbol{w_1}) f_d(\boldsymbol{v_1}) d^3\boldsymbol{v_1}$$

A.2.3 Collision frequency function (ν)

The function ν was derived from the term
$\tilde{\nu} f_d = \int \int \int \delta(\boldsymbol{v} - \boldsymbol{v_1}) f_d(\boldsymbol{v_1}) f_{mb}(\boldsymbol{v_2}) |\boldsymbol{v_2} - \boldsymbol{v_1}| \sigma d\Omega dv_1 dv_2$

Writting f_{mb} as a shifted Maxwellian, doing the same change of variables as before, and identifying with the definition of the function ν, we finally get:

$$\begin{aligned} \tilde{\nu} f_d(\boldsymbol{v}) &= \int \int \int \delta(\boldsymbol{w} - \boldsymbol{w_1}) f_d(\boldsymbol{u_{mb}} + \boldsymbol{w_1}) \frac{n_{mb}}{v_{mb}^3 \pi^{3/2}} \exp(-\boldsymbol{w_2^{*2}}) |\boldsymbol{w_2} - \boldsymbol{w_1}| \sigma d\Omega dw_1 dw_2 \\ &= \nu(\boldsymbol{w}) f_d(\boldsymbol{v}) \end{aligned}$$

A.3 Computation of the moments of the Kernels

In this section, we will derive the expressions of the moments of the kernels used to calculate the shifting of the Maxwellian. All those moments will be calculated with resprect to the mean velocity of the Maxwellian. In other words, we choose the frame moving at $\boldsymbol{u_{mb}}$ as reference.

Let K_i denote K_1 or K_2 and $M(\boldsymbol{w})$ be either 1, $\boldsymbol{w_1}$, or $|\boldsymbol{w_1}^2|$. The moments of $\tilde{K}_i f_d$ are $\int M(\boldsymbol{w})\tilde{K}_i f_d(\boldsymbol{v})d^3\boldsymbol{v}$, where $\tilde{K}_i f_d(\boldsymbol{v}) = \int K_i(\boldsymbol{w}, \boldsymbol{w_1}) f_d(\boldsymbol{v_1})d^3\boldsymbol{v_1}$

In Chapter 3, it is shown that computing $\int M(\boldsymbol{w})\tilde{K}_i f_d(\boldsymbol{v})d^3\boldsymbol{v}$ requires the compuation of the functions $\boldsymbol{w_1} \rightarrow \int M(\boldsymbol{w})K_i(\boldsymbol{w}, \boldsymbol{w_1})d^3\boldsymbol{w}$. The moments of $\tilde{K}_i f_d$ are then obtained by integrating the product of these functions with f_d.

We can start by simplifying this expression. We know that $K_i(\boldsymbol{w}, \boldsymbol{w_1}) = \mu_1 K_i^*(\frac{\boldsymbol{w}}{v_{mb}}, \frac{\boldsymbol{w_1}}{v_{mb}}) = \mu_1 K_i^*(\boldsymbol{w}^*, \boldsymbol{w_1}^*)$. After the change of variable $\boldsymbol{w}^* = \boldsymbol{w}/v_{mb}$, $d^3\boldsymbol{w}^* = d^3\boldsymbol{w}/v_{mb}^3$, we can write the moments of the kernel as $\mu_1 v_{mb}^3 \int M(v_{mb}\boldsymbol{w}^*)K_i^*(\boldsymbol{w}^*, \boldsymbol{w_1}^*)d^3\boldsymbol{w}^*$, where the K_i^* are now nondimensionnal functions.

These integrals are then computed separately for K_1 and K_2. This is the object ot the following two sections. The algebra is long and not very interesting. Therefore, the derivations will be restricted to the salient points of the derivation.

A.3.1 Moments of the Kernel K_2

The moments of K_2 are $\mu_1 v_{mb}^3 \int M(v_{mb}^3\boldsymbol{w}^*)|\boldsymbol{w}^* - \boldsymbol{w}_1^*| \exp[-\boldsymbol{w}^{*2}]d^3\boldsymbol{w}^*$ By doing the change of variable $\tilde{\boldsymbol{w}}^* = \boldsymbol{w}^* - \boldsymbol{w_1}^*$, it can be rewritten as $\mu_1 v_{mb}^3 \int M(v_{mb}\boldsymbol{w}_1^* + v_{mb}\tilde{\boldsymbol{w}}^*)|\tilde{\boldsymbol{w}}^*| \exp[-(\boldsymbol{w}_1^* + \tilde{\boldsymbol{w}}^*)^2]d^3\tilde{\boldsymbol{w}}^*$. We now switch to spherical coordinates with an orientation choosen so that the main axis be $\boldsymbol{w_1}^*/|\boldsymbol{w_1}^*|$. We then write $\tilde{\boldsymbol{w}}^* = r\boldsymbol{e_r}$ The expression of the moments becomes $\mu_1 v_{mb}^3 \int M(v_{mb}\boldsymbol{w_1}^* + v_{mb}r\boldsymbol{e_r}) \exp[-|\boldsymbol{w_1}^*|^2 - r^2 - 2r|\boldsymbol{w_1}^*| \cos\theta]r^3 \sin\theta drd\theta d\phi$

A.3.2 Moments of the Kernel K_1

The moments of K_1 are $2\mu_1 v_{mb}^3 \int M(v_{mb}\boldsymbol{w}^*)\frac{1}{|\boldsymbol{w}^* - \boldsymbol{w}_1^*|} \exp[-(\boldsymbol{w}^* \cdot (\boldsymbol{w}^* - \boldsymbol{w_1}^*)/|\boldsymbol{w}^* - \boldsymbol{w_1}^*|)^2]d^3\boldsymbol{w}^*$ By doing the same change of variable as previously and using the same spherical coordinate system, we can rewrite it as $\mu_1 v_{mb}^3 \int M(v_{mb}\boldsymbol{w_1}^* + v_{mb}r\boldsymbol{e_r}) \exp[-(r + \cos\theta)^2]r \sin\theta drd\theta d\phi$

A.3.3 Results

The above integrations can then be performed analytically (by integrating by part). The algebra is not interesting and leads to complicated expressions which happens to simplify when we substract them to get the moments of $K_1 - K_2$. We will just provide the results, which were obtained with a formal calculus software.

$$\begin{aligned}
\int [K_1(\boldsymbol{w}, \boldsymbol{w_1}) - K_2(\boldsymbol{w}, \boldsymbol{w_1})] d^3\boldsymbol{w} &= \nu(\boldsymbol{w_1}) \\
\int \boldsymbol{w}[K_1(\boldsymbol{w}, \boldsymbol{w_1}) - K_2(\boldsymbol{w}, \boldsymbol{w_1})] d^3\boldsymbol{w} &= \boldsymbol{w_1}\nu(\boldsymbol{w_1}) \\
\int |\boldsymbol{w}|^2[K_1(\boldsymbol{w}, \boldsymbol{w_1}) - K_2(\boldsymbol{w}, \boldsymbol{w_1})] d^3\boldsymbol{w} &= |\boldsymbol{w_1}|^2\nu(\boldsymbol{w_1})
\end{aligned}$$

A.4 Mean value of the Kernel K_1 over a ball

We will derive in this part the mean value of the kernel K_1 over a ball centered on $\boldsymbol{v_1}$ and of radius v_c. Let $\mathcal{B}$ denote this ball.

The nondimensional kernel is:

$$K_1^*(\boldsymbol{v}^*, \boldsymbol{v_1}^*) = \frac{2}{|\boldsymbol{v}^* - \boldsymbol{v_1}^*|} \exp\left[-\frac{(\boldsymbol{v}^* \cdot (\boldsymbol{v}^* - \boldsymbol{v_1}^*))^2}{|\boldsymbol{v}^* - \boldsymbol{v_1}^*|^2}\right]$$

Let's note $\boldsymbol{V}^* = \boldsymbol{v}^* - \boldsymbol{v_1}^*$. Let's define the spherical coordinates with the z-axis parallel to $\boldsymbol{v_1}^*$. We then have $\boldsymbol{v_1}^* = |\boldsymbol{v_1}^*|e_z$. $\boldsymbol{V}^*$ run on the sphere centered on $\mathbf{0}$ and of radius v_c^*. In the spherical coordinates, we then have $\boldsymbol{V}^* = r\boldsymbol{e_r}$. With these notation, we have $\boldsymbol{v}^* = \boldsymbol{v_1}^* + r\boldsymbol{e_r}$.

We then have:

$$
\begin{aligned}
\int_{\mathcal{B}} K_1^*(\boldsymbol{v}^*, \boldsymbol{v_1}^*) d^3\boldsymbol{v}^* &= \int \frac{2}{r} \exp\left[-((\boldsymbol{v_1}^* + r\boldsymbol{e_r}) \cdot \boldsymbol{e_r})^2\right] r^2 \sin(\theta) dr d\theta d\phi \\
&= 2\int r \exp\left[-(\boldsymbol{v_1}^* \cdot \boldsymbol{e_r} + r)^2\right] \sin(\theta) dr d\theta d\phi \\
&= 2\int r \exp\left[-(|\boldsymbol{v_1}^*| \cos(\theta) + r)^2\right] \sin(\theta) dr d\theta d\phi \\
&= 4\pi \int r \exp\left[-(|\boldsymbol{v_1}^*| \cos(\theta) + r)^2\right] \sin(\theta) dr d\theta \\
&= 2\pi^{3/2} \int \frac{r}{|\boldsymbol{v_1}^*|} \left[-\mathrm{erf}(-|\boldsymbol{v_1}^*| + r) + \mathrm{erf}(|\boldsymbol{v_1}^*| + r)\right] dr
\end{aligned}
$$

Since v_c^* will be much smaller than 1, we can proceed to a Taylor expansion of $-\mathrm{erf}(-|\boldsymbol{v_1}^*| + r) + \mathrm{erf}(|\boldsymbol{v_1}^*| + r)$ around $|\boldsymbol{v_1}^*|$. The first order cancel out, and we are left with $2\mathrm{erf}(|\boldsymbol{v_1}^*|)$. We then have:

$$
\begin{aligned}
\int_{\mathcal{B}} K_1^*(\boldsymbol{v}^*, \boldsymbol{v_1}^*) d^3\boldsymbol{v}^* &= \frac{4\pi^{3/2}\mathrm{erf}(|\boldsymbol{v_1}^*|)}{|\boldsymbol{v_1}^*|} \int_0^{v_c^*} r dr \\
&= 2\frac{\mathrm{erf}(|\boldsymbol{v_1}^*|)}{|\boldsymbol{v_1}^*|} \pi^{3/2} v_c^{*2}
\end{aligned}
$$

To get the mean value, we divide by the volume of the sphere $(4/3)\pi v_c^{*3}$ and get

$$
\frac{3}{2} \frac{\mathrm{erf}(|\boldsymbol{v_1}^*|)}{|\boldsymbol{v_1}^*|} \pi^{1/2} \frac{1}{v_c^*}
$$

The corresponding dimensional value is then

$$
\mu_1 \frac{3}{2} \frac{\mathrm{erf}(|\boldsymbol{v_1}^*|)}{|\boldsymbol{v_1}^*|} \pi^{1/2} \frac{1}{v_c^*}
$$

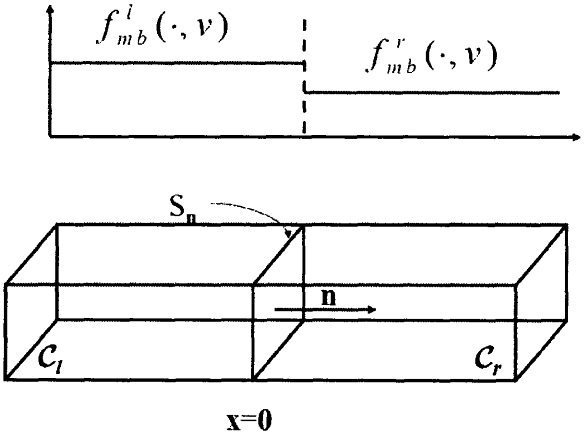
where $\boldsymbol{v_1}^* = \boldsymbol{v_1}/v_{mb}$.

Appendix B

Derivations of Advection Formulae

B.1 Derivation of ∇f_{mb}

In this section, we want to derive the expression of ∇f_{mb}, given that for a fixed velocity $\boldsymbol{v}$, the function $\boldsymbol{x} \longrightarrow f_{mb}(\boldsymbol{x}, \boldsymbol{v})$ is piecewise constant. The function ∇f_{mb} will then be zero everywhere, exept at some locations where it will be a dirac distribution.



We can find this dirac by writing f_{mb} is terms of an heaviside function H whose definition is:

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

It is known that the derivative of this function is $H'(x) = \delta(x)$, where δ is the dirac distribution.

On the other hand, for a fixed velocity $\boldsymbol{v}$, the function $f_{mb}(\cdot, \boldsymbol{v})$ over the two adjacent cells under consideration can be written as

$$f_{mb}(\boldsymbol{x}, \boldsymbol{v}) = f^r_{mb}(\boldsymbol{v})H(\boldsymbol{n} \cdot \boldsymbol{x}) + f^l_{mb}(\boldsymbol{v})H(-\boldsymbol{n} \cdot \boldsymbol{x})$$

Simple differentiation rules tell us that $\nabla H(\boldsymbol{n} \cdot \boldsymbol{x}) = \nabla(\boldsymbol{n} \cdot \boldsymbol{x})H'(\boldsymbol{n} \cdot \boldsymbol{x}) = \boldsymbol{n}\delta(\boldsymbol{n} \cdot \boldsymbol{x})$.

Combining the last equations, we obtain that the gradient is

$$\boldsymbol{\nabla f_{mb}}(\boldsymbol{x}, \boldsymbol{v}) = (f^r_{mb} - f^l_{mb})\delta(\boldsymbol{n} \cdot \boldsymbol{x})\boldsymbol{n}$$

B.2 Solution of $\partial f/\partial t + \boldsymbol{v} \cdot \boldsymbol{\nabla f} = g(\boldsymbol{x}, \boldsymbol{v}, t)$

In this equation, $\boldsymbol{v}$ does not show up in differentiations and can thus be considered as a constant parameter. We then do the change of variable:

$$\left\{\begin{array}{rcl} \boldsymbol{s}(\boldsymbol{x}, t) & = & \boldsymbol{x} - \boldsymbol{v}t \\ u(\boldsymbol{x}, t) & = & t \end{array}\right. \Longleftrightarrow \left\{\begin{array}{rcl} \boldsymbol{x}(\boldsymbol{s}, u) & = & \boldsymbol{s} + \boldsymbol{v}u \\ t(\boldsymbol{s}, u) & = & u \end{array}\right.$$

The corresponding derivatives are:

$$\begin{array}{rcl} \dfrac{\partial \boldsymbol{s}}{\partial t} & = & -\boldsymbol{v} \\ \dfrac{\partial \boldsymbol{s}}{\partial \boldsymbol{x}} & = & \mathbf{I} \\ \dfrac{\partial \boldsymbol{u}}{\partial t} & = & 1 \\ \dfrac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}} & = & \mathbf{0} \end{array}$$

Here, $\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{x}}$ denotes the gradient of the "vector field" $\boldsymbol{s}(\boldsymbol{x})$, $\mathbf{I}$ is the identity matrix, and $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}$ is the gradient of the "scalar field" $u(\boldsymbol{x})$.
In this new system of variable, we then have:

$$\begin{cases} \dfrac{\partial f}{\partial t} = \dfrac{\partial f}{\partial \boldsymbol{s}} \cdot \dfrac{\partial \boldsymbol{s}}{\partial t} + \dfrac{\partial f}{\partial u}\dfrac{\partial u}{\partial t} = -\boldsymbol{v} \cdot \dfrac{\partial f}{\partial \boldsymbol{s}} + \dfrac{\partial f}{\partial u} \\ \dfrac{\partial f}{\partial \boldsymbol{x}} = \dfrac{\partial f}{\partial \boldsymbol{s}} \cdot \dfrac{\partial \boldsymbol{s}}{\partial \boldsymbol{x}} + \dfrac{\partial f}{\partial u}\dfrac{\partial u}{\partial \boldsymbol{x}} = \dfrac{\partial f}{\partial \boldsymbol{s}} \end{cases}$$

The left hand side of the equation becomes $\partial f/\partial t + \boldsymbol{v} \cdot \boldsymbol{\nabla f} = \partial f/\partial u$. The equation is then $\partial f/\partial u = g(\boldsymbol{s}, \boldsymbol{v}, u)$. For convenience, we used the not rigorous notation $g(\boldsymbol{s}, \boldsymbol{v}, u) = g(\boldsymbol{x}(\boldsymbol{s}, u), \boldsymbol{v}, t(\boldsymbol{s}, u))$ and the equivalent one for f. Integrating versus u at constant $\boldsymbol{s}$ leads to $f(\boldsymbol{s}, \boldsymbol{v}, u) = \int_0^u g(\boldsymbol{s}, \boldsymbol{v}, u')du' + h(\boldsymbol{s}, \boldsymbol{v})$.

We now go back to the original variables. The constant value of $\boldsymbol{s}$ is $\boldsymbol{s} = \boldsymbol{x} - \boldsymbol{v}t$. Then, $t' = u'$ and $\boldsymbol{x}'(\boldsymbol{s}, u') = \boldsymbol{s} + \boldsymbol{v}u' = \boldsymbol{x} - \boldsymbol{v}t + \boldsymbol{v}t'$. This leads to

$$f(\boldsymbol{x}, \boldsymbol{v}, t) = \int_0^t g(\boldsymbol{x} - \boldsymbol{v}t + \boldsymbol{v}t', \boldsymbol{v}, t')dt' + h(\boldsymbol{x} - \boldsymbol{v}t, \boldsymbol{v})$$

Doing the change of variable $t' \rightarrow t - t'$ in the integral, we get:

$$f(\boldsymbol{x}, \boldsymbol{v}, t) = \int_0^t g(\boldsymbol{x} - \boldsymbol{v}t', \boldsymbol{v}, t - t')dt' + h(\boldsymbol{x} - \boldsymbol{v}t, \boldsymbol{v})$$

Using the initial condition, namely the distribution at $t = 0$ $f(\boldsymbol{x}, \boldsymbol{v}, 0)$, it becomes

$$f(\boldsymbol{x}, \boldsymbol{v}, t) = \int_0^t g(\boldsymbol{x} - \boldsymbol{v}t', \boldsymbol{v}, t - t')dt' + f(\boldsymbol{x} - \boldsymbol{v}t, \boldsymbol{v}, 0)$$

Bibliography

[1] Y. Sone, *Kinetic Theory and Fluid Dynamics*, Birkhauser, 2002.

[2] C. Cercignani *The Boltzmann Equation and its Applications*, Springer-Verlag, New York, 1988.

[3] G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, 1994.

[4] L. L. Baker and N. G. Hadjiconstantinou, "Variance reduction for Monte Carlo solutions of the Boltzmann Equation", *Physics of Fluids*, **17**, art. no. 051703, 2005.

[5] L. L. Baker and N. G. Hadjiconstantinou, "Variance-reduced Particle Methods for Solving the Boltzmann Equation", submitted to *Theoretical Nanoscience*, 2006.

[6] L. L. Baker and N. G. Hadjiconstantinou, "Variance reduction in particle methods for solving the Boltzmann Equation", *Proceedings of the Fourth International Conference on Nanochannels, Microchannels and Minichannels*, Paper ICNMM2006-96089, 2006.

[7] L. L. Baker, "Efficient numerical solutions of the Boltzmann equation for low-speed dilute gas flows", PhD Thesis, Mechanical Engineering Department, MIT, Jan. 2007.

[8] T. Carleman, "Problèmes Mathématiques dans la Théorie Cinétique des Gaz", *Almqvist and Wilsell, Uppsala*, 1957.

[9] W. G. Vincenti and C. H. Kruger, *Introduction to Physical Gas Dynamics*, Krieger, Florida, 1965.

[10] W. Wagner, "A Convergence Proof for Bird's Direct Simulation Monte Carlo Method for the Boltzmann Equation", *Journal of Statistical Physics*, **66**. 1011, 1992.

[11] N. G. Hadjiconstantinou, A. L. Garcia, M. Z. Bazant and G. He, "Statistical Error in Particle Simulations of Hydrodynamic Phenomena," *Journal of Computational Physics*, **187**, 274-297, 2003.