ESAIM: Proceedings CEMRACS 1999
URL: http://www.emath.fr/Maths/Proc/Vol.10

AN INTRODUCTION TO MONTE CARLO METHODS FOR THE BOLTZMANN EQUATION*

Lorenzo PARESCHI[1] and Giovanni RUSSO[2]

Abstract. The purpose of this note is to give an introductory overview of the different direct simulation Monte Carlo (DSMC) methods for the numerical solution of the Boltzmann equation. Particular emphasis is given to some recently developed time relaxed Monte Carlo (TRMC) schemes for the acceleration of DSMC computations near continuum regimes. Several numerical examples are also included. The note is far from being complete and only some aspects are treated of such a vast theme.

AMS Subject Classification. 34A65, 65C05, 76P05, 82C80 .

.

Contents

Keywords and phrases: Boltzmann equation, Monte Carlo methods, direct simulation schemes, time relaxed schemes, sampling algorithms, boundary conditions, fluid-dynamic limit.

* *This work was partially supported by TMR project “Asymptotic Methods in Kinetic Theory”, Contract Number ERB FMRX CT97 0157, and partially by MURST project “Numerical methods for evolution problems”.*

[1] Mathematics Department, University of Ferrara, Via Machiavelli 35, 44100 Ferrara, ITALY.
e-mail: pareschi@dm.unife.it, Web: http://www.unife.it/∼prl

[2] Mathematics Department, University of Catania, Viale Andrea Doria 6, 95125 Catania, ITALY.
e-mail: russo@univaq.it, Web: http://univaq.it/∼russo

1. Introduction

In many engineering applications, from the space shuttle to vacuum pumps [27, 28, 39], the Euler or Navier-Stokes equations of fluid dynamics do not give a satisfactory description of the physical system and an accurate kinetic description through the Boltzmann equation of rarefied gas dynamics (RGD) is required.

From a computational point of view the numerical solution of the Boltzmann equation is much more expensive then the equations of fluid dynamics and represents a real challenge for numerical methods. This is mainly due to the large number of variables in the problem and to the multi-dimensional nonlinear integral that defines the collision operator. In addition, this integration has to be handled carefully since it is at the basis of the conservation properties of the Boltzmann equation.

As a consequence since the early 1970's the dominant methods for computations of RGD are based on probabilistic Monte-Carlo techniques at different levels [1, 3, 4, 10, 12, 22, 23, 29, 30, 36, 40, 41]. Pioneers of these methods are the direct simulation Monte Carlo method (DSMC) by Bird [3, 4] and later the modified DSMC method proposed by Nanbu [29]. The common feature of these methods is to evolve a finite set of particles by moving them according to their velocity and by performing collisions between randomly chosen particles.

Monte Carlo methods present several advantages and have been very successful in a wide range of applications. First, if compared with deterministic approaches, the computational cost is strongly reduced and can be considered of the order of the number of particles used in the velocity space. Second, these methods do not need any artificial boundary in the velocity space. In fact, particles can have any velocity and thus the discretization points are always well defined independently of the physical problem.

At variance, in addition to the computational complexity, a major drawback associated with deterministic methods that use a fixed discretization in the velocity domain is that the velocity space is approximated by a finite region. Hence, a large region with a huge number of discretization points is required in problems with very high Mach numbers, which greatly increase the computing effort.

On the other hand, particle methods yield low accurate and fluctuating results with respect to deterministic methods and the convergence in general is quite low. In particular, there is one important situation where DSMC methods lose its effectiveness: flows where the Knudsen number varies over several orders of magnitude. In such problems there are regions in which the Knudsen number is small enough that the collision rate is large, but not small enough that the flow is well described by fluid mechanics.

For these *nearly continuum regions* a standard kinetic treatment would be too expensive since accuracy of DSMC methods depends on the resolution of the collisional length and time scales and a pure fluid dynamic approach inappropriate.

Domain decomposition methods have been proposed for this problem. In these methods the computational domain is divided into a *fluid region* in which the system is treated by the Euler or Navier-Stokes fluid dynamic equations, and a *kinetic region*, where the Boltzmann equation is used. Suitable matching condition are then used to couple the two regions [6, 18, 24, 42]. The main difficulties of this approach is the detection of the two regions and the design of boundary conditions to couple the fluid and the kinetic regions.

In some recently developed Monte Carlo methods a different approach is taken, with the goal of constructing simple and efficient numerical methods for the solution of the Boltzmann equation in regions with a large variation in the mean free path [8, 33, 32, 34, 35]. These were the first Monte Carlo methods of this type. Previous implicit versions of Monte Carlo methods in transport problems were developed for the linear equation of photon transport in [15].

These algorithms, hereafter called time relaxed Monte Carlo (TRMC) methods, are based on a a suitable truncation of a power series expansion [47] for the Boltzmann equation first introduced in [16]. This truncated series has the following properties: for large Knudsen number it is a time approximation of the Boltzmann equation of a given order, and when the Knudsen number vanishes, it projects the solution towards the local Maxwellian. In addition, the truncated series preserves mass, momentum and energy.

In the limit of very small Knudsen number, the collision step replaces the distribution function by a local Maxwellian with the same moments. A Monte Carlo method based on such truncated series will behave as a sort of kinetic scheme for the underlying Euler equations of gas dynamics [11, 37, 38].

This approach of coupling the fluid and kinetic equations in the Monte Carlo method represents a complementary strategy to domain decomposition methods, i.e., in order to obtain an efficient coupling, one can use these methods to extend the use of the Monte Carlo solver from the pure kinetic regions to the near continuum regions.

The plan of the rest of the paper is the following. In the next section we give a short introduction on the Boltzmann equation and some of its relevant properties. Section 3 is devoted to a description of Bird's and Nanbu's classical DSMC methods. TRMC methods are discussed in Section 4. Finally in Section 5 and 6 numerical results for space homogeneous and non homogeneous problems are presented.

2. The Boltzmann equation

We consider the Boltzmann equation [9, 10]

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = \frac{1}{\varepsilon} Q(f,f) \tag{1}$$

with the initial condition

$$f(x,v,t=0) = f_0(x,v), \tag{2}$$

where $f = f(x,v,t)$ is a non negative function describing the time evolution of the distribution of particles which move with velocity $v \in I\!R^3$ in the position $x \in \Omega \subset I\!R^3$ at time $t > 0$. The parameter $\varepsilon > 0$ is the Knudsen number and is proportional to the mean free path between collisions. The bilinear collision operator $Q(f,f)$ describes the binary collisions of the particles and is given by

$$Q(f,f)(v) = \int_{I\!R^3} \int_{S^2} \sigma(|v-v_*|,\omega)[f(v')f(v'_*) - f(v)f(v_*)]\, d\omega\, dv_*. \tag{3}$$

In the above expression, ω is a unit vector of the sphere S^2, so that $d\omega$ is an element of area of the surface of the unit sphere S^2 in $I\!R^3$. Moreover (v',v'_*) represent the post-collisional velocities associated with the pre-collisional velocities (v,v_*) and the collision parameter ω; i.e.,

$$v' = \frac{1}{2}(v + v_* + |v-v_*|\omega), \quad v'_* = \frac{1}{2}(v + v_* - |v-v_*|\omega). \tag{4}$$

The kernel σ is a nonnegative function which characterizes the details of the binary interactions. In the case of inverse k-th power forces between particles the kernel has the form

$$\sigma(|v-v_*|,\theta) = b_\alpha(\theta)|v-v_*|^\alpha, \tag{5}$$

where $\alpha = (k-5)/(k-1)$. For numerical purposes, a widely used model is the Variable Hard Sphere (VHS) model [4], corresponding to take $b_\alpha(\theta) = C_\alpha$ where C_α is a positive constant. The case $\alpha = 0$ is referred to as the Maxwellian gas, whereas the case $\alpha = 1$ yields the Hard Sphere gas.

For the Maxwellian gas the scattering kernel σ is a constant. This case has been widely studied theoretically, since it has several interesting mathematical properties. In particular, exact analytical solutions can be found [5, 14].

A simplified model of the space homogeneous Boltzmann equation was derived by Kac [20]. This model is often used because its simplicity allows a more rigorous mathematical analysis than the full Boltzmann equation.

2.1. **Boundary conditions**

Equation (1-2) usually is complemented with the boundary conditions for $v \cdot n \geq 0$ and $x \in \partial\Omega$, where n denotes the unit normal, pointing inside the domain Ω. Usually the boundary represents the surface of a solid object (an obstacle or a container). The particles of the gas that hit the surface interact with the atoms of the object and are reflected back into the domain Ω. Mathematically, such boundary conditions are modeled by an expression of the form

$$|v \cdot n| f(x,v,t) = \int_{v_* \cdot n < 0} |v_* \cdot n(x)| K(v_* \to v, x, t) f(x, v_*, t)\, dv_*. \tag{6}$$

This is the so-called reflective condition on $\partial\Omega$. The ingoing flux is defined in terms of the outgoing flux modified by a given boundary kernel K according to the integral in (6). This boundary kernel is such that positivity and mass conservation at the boundaries are guaranteed,

$$K(v_* \to v, x, t) \geq 0, \qquad \int_{v \cdot n(x) \geq 0} K(v_* \to v, x, t)\, dv = 1. \tag{7}$$

Commonly used reflecting boundary conditions are the so-called Maxwell's conditions. From a physical point of view, one assumes that at the solid boundary a fraction α of molecules is absorbed by the wall and then re-emitted with the velocities corresponding to those in a still gas at the temperature of the solid wall, while the remaining portion $(1-\alpha)$ is specularly reflected. This is equivalent to impose for the ingoing velocities

$$f(x,v,t) = (1-\alpha) Rf(x,v,t) + \alpha Mf(x,v,t), \tag{8}$$

in which $x \in \partial\Omega, \quad v \cdot n(x) \geq 0$. The coefficient α, with $0 \leq \alpha \leq 1$, is called the *accommodation coefficient* and

$$Rf(x,v,t) = f(x, v - 2n(n \cdot v), t), \tag{9}$$

$$Mf(x,v,t) = \mu(x,t) M_w(v). \tag{10}$$

In (10), if we denote by T_w the temperature of the solid boundary, M_w is given by

$$M_w(v) = \exp(-\frac{v^2}{2T_w}),$$

and the value of μ is determined by mass conservation at the surface of the wall

$$\mu(x,t) \int_{v \cdot n \geq 0} M_w(v) |v \cdot n| dv = \int_{v \cdot n < 0} f(x,v,t) |v \cdot n| dv. \tag{11}$$

We note that according to (8), for $\alpha = 0$ (specular reflection) the re-emitted molecules have the same flow of mass, temperature and tangential momentum of the incoming molecules, while for $\alpha = 1$ (full accommodation) the re-emitted molecules have completely lost memory of the incoming molecules, except for conservation of the number of molecules.

In the case of inflow boundary conditions, one assumes that the distribution function of the particles entering the domain is known, i.e.

$$f(x,v,t) = g(v,t), \quad x \in \partial\Omega, \quad v \cdot n > 0,$$

A typical example of such condition is used in shock wave calculations, where one assumes that the distribution function at the boundary of the computational domain is a Maxwellian. In this case the particles that enter the domain in a time step Δt have a distribution proportional to $v \cdot n\, g(v,t)$, $v \cdot n > 0$. In fact the number of particles that enter the domain in a time interval Δt with velocity between v and $v + dv$ through a small area ΔS is given by

$$dN = \Delta S \Delta t\, v \cdot n\, g(v,t)\, d^3v.$$

Other physical boundary conditions are modeled by different expressions (see for example [9]).

In section 6 an implementation of the previous boundary conditions in the context of Monte Carlo methods will be described.

2.2. Fluid dynamic limit

During the evolution process, the collision operator preserves mass, momentum and energy, i.e.,

$$\int_{\mathbb{R}^3} Q(f,f)\phi(v)\, dv = 0, \quad \phi(v) = 1, v, v^2, \tag{12}$$

and in addition it satisfies Boltzmann's well-known H-theorem

$$\int_{\mathbb{R}^3} Q(f,f) \log(f) dv \leq 0. \tag{13}$$

From a physical point of view, Boltzmann's H-theorem implies that any equilibrium distribution function, i.e. any function f for which $Q(f,f) = 0$, has the form of a locally Maxwellian distribution

$$M(\rho,u,T)(v) = \frac{\rho}{(2\pi T)^{3/2}} \exp\left(-\frac{|u-v|^2}{2T}\right), \tag{14}$$

where ρ, u, T are the density, mean velocity and temperature of the gas defined by

$$\rho = \int_{\mathbb{R}^3} f\, dv, \qquad u = \frac{1}{\rho}\int_{\mathbb{R}^3} vf\, dv, \qquad T = \frac{1}{3\rho}\int_{\mathbb{R}^3} [v-u]^2 f\, dv. \tag{15}$$

As $\varepsilon \to 0$ the distribution function approaches the local Maxwellian (14). In this case the higher order moments of the distribution f can be computed as function of ρ, u, and T, by using (14) and we obtain to the leading order the closed system of compressible Euler equations of gas dynamics

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \nabla_x \cdot (\rho u) = 0 \\ &\frac{\partial \rho u}{\partial t} + \nabla_x \cdot (\rho u \otimes u + p) = 0 \qquad (16) \\ &\frac{\partial E}{\partial t} + \nabla_x \cdot (Eu + pu) = 0 \\ &p = \rho T, \quad E = \frac{3}{2}\rho T + \frac{1}{2}\rho u^2. \qquad (17) \end{aligned}$$

where p is the gas pressure.

2.3. **Splitting approach**

The most common approach to solve the full Boltzmann equation is based on an operator splitting. Consider the following initial value problem (1-2).

The solution after one time step Δt may be obtained by the sequence of two steps. First integrate the space homogeneous equation (collision step) for all $x \in \Omega$,

$$\begin{aligned} \frac{\partial \tilde{f}}{\partial t} &= \frac{1}{\varepsilon} Q(\tilde{f}, \tilde{f}), \\ \tilde{f}(x,v,0) &= f_0(x,v), \end{aligned} \tag{18}$$

for a time step Δt, and then the transport equation (convection step), using the output of the previous step as initial condition,

$$\begin{aligned} \frac{\partial f}{\partial t} + v \cdot \nabla_x f &= 0, \\ f(x,v,0) &= \tilde{f}(x,v,\Delta t). \end{aligned} \tag{19}$$

After computing an approximation of the solution at time Δt, the process may be iterated to obtain the numerical solution at later times. Clearly, after this splitting almost all the main difficulties are contained in the collision step since the nonlinear collision operator becomes highly stiff near the fluid regime ($\varepsilon \ll 1$) [17, 16, 43].

The discretization of these equations can be performed in a variety of ways (particle methods, finite volume, and so on). The choice of the discretization mainly depends on the method that is used for the solution of the space homogeneous Boltzmann equation. In this paper we illustrate the use of particle methods, because we assume that we use a Monte Carlo method for the integration of the space homogeneous equation.

The splitting scheme (simple splitting) described above is first order accurate in space and time. Note that the order of accuracy of this simple splitting does not improve even if we solve with great accuracy both collision and convection steps. The accuracy in time may be improved by a more sophisticated splitting. For example Strang splitting is second order accurate, provided both steps are at least second order [44] (an application to deterministic methods for the Boltzmann equation is given for example, in [31]).

For the Monte Carlo method we shall use only the simple splitting. The motivation is that it is not clear, at present stage, if the order of accuracy of the space discretization used in Monte Carlo methods is higher than first order. This subject is a presently under investigation by the authors.

3. Direct Simulation Monte Carlo (DSMC) Methods

In this section we describe two of the most commonly used Monte Carlo methods for the simulation of collision step (18), namely Nanbu-Babovsky's and Bird's schemes. Numerical results for these methods will be presented in Section 5 for space homogeneous computations and in Section 6 for a stationary shock problem.

Before entering the description of the methods, we give a brief review of random sampling, which is at the basis of several Monte Carlo methods.

3.1. **Random sampling**

In this section we revise some basic features common to many applications of the Monte Carlo method. We assume that our computer is able to generate a uniformly distributed random number between 0 and 1. A simple reference on such topics and on Monte Carlo techniques is the book by Kalos [21].

3.1.1. *Monovariate distribution*

Let $x \in I\!\!R$ be a random variable with density $p_x(x)$, i.e. $p_x(x) \geq 0$, $\int_\Omega p_x(x)\,dx = 1$, and let ξ be a uniformly distributed random number in $[0,1]$. Then the relation between x and ξ can be found by equating the infinitesimal probabilities

$$p_x(x)\,dx = d\xi.$$

By integration one has

$$P_x(x) = \xi, \tag{20}$$

where $P_x(x) = \int_{-\infty}^x p_x(y)\,dy$ is the distribution function corresponding to the random variable x, i.e. the primitive of $p_x(x)$. Then the random variable x can be sampled by sampling a uniformly distributed variable ξ, and then solving Eq. (20) for x,

$$x = P_x^{-1}(\xi). \tag{21}$$

Example 3.1. Let $p_x(x) = \exp(-x)$, $x \geq 0$. Then

$$P_x(x) = \int_0^x \exp(-y)\,dy = 1 - \exp(-x) = \xi,$$

and therefore

$$x = -\ln(1-\xi)$$

or $x = -\ln\xi$, because $1-\xi$ is also a uniformly distributed $[0,1]$ random number.

Remark 3.2. We consider density functions defined on the whole real line. If the support of the density is strictly contained in $I\!\!R$, as in the previous example, the density function can be defined by using the Heaviside Θ-function. In the example above one could define $p_x(x) = \exp(-x)\Theta(x), x \in I\!\!R$.

Sometimes it may be expensive to the compute the inverse function, since in general a nonlinear equation has to be solved. In this case a different technique, called *acceptance-rejection*, can be used to sample a random variables with a given density.

Let x be a random variable with density $p_x(x)$, $x \in I\!\!R$. Then we look for a function $w(x) \geq p_x(x)\,\forall x \in I\!\!R$ whose primitive $W(x)$ is easily invertible. Let $A = \int_{-\infty}^{\infty} w(x)\,dx$. Then the algorithm works as follows:

Algorithm 3.3.

```
1. sample from $w(x)/A$ by solving the equation $W(x) = A\xi_1$,
2. if $w(x)\xi_2 < p_x(x)$ then accept the sample, else reject the sample and repeat step 1.
```

Here ξ_1 and ξ_2 represent, as usual, uniformly $[0,1]$ random numbers. It is clear that the efficiency of the scheme depends on how easy it is to invert the function $W(x)$ and how frequently we accept the sample. The fraction of accepted samples equals the ratio of the areas below the two curves $p_x(x)$ and $W(x)$ and it is therefore equal to $1/A$ (see Fig. 1).

Sometimes a density function is given as a convex combination of simpler density functions,

$$p(x) = \sum_{i=1}^M w_i p_i(x)$$

where w_i are probabilities (i.e. $w_i \geq 0, \sum_{i=1}^M w_i = 1$), and $p_i(x)$ are probability densities. In that case the sampling can be performed as follows

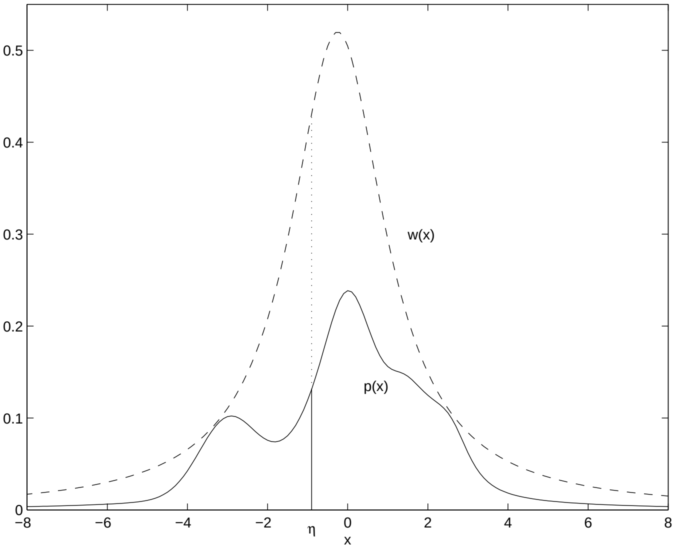

FIGURE 1. Acceptance-rejection procedure. η has been sampled from $w(x)$. It will be accepted with probability equal to the ratio $p(\eta)/w(\eta)$.

Algorithm 3.4.

1. select an integer $i \in \{1, \dots, M\}$ with probability w_i
2. sample x from a random variable with density $p_i(x)$.

Because of the relevance in several applications, step 1 of the previous algorithm deserves an extended discussion.

3.1.2. *Discrete sampling*

Let us suppose that $k \in \{1, \dots, M\}$ is an integer random number, with probabilities $\{w_k\}$. In order to sample k with probability w_k we can proceed as follows:

divide interval $[0, 1]$ in M intervals, i-th interval being of length w_i, extract a uniform $[0, 1]$ random number ξ, and detect the interval k to which ξ belongs.

This can be done in the following way:

Algorithm 3.5.

1. compute $W_k = \sum_{i=1}^{k} w_k, k = 1, \dots, M, \quad W_0 = 0$
2. find the integer k such that $W_{k-1} \leq \xi < W_k$.

For an arbitrary set of probabilities $\{w_i\}$, once W_i have been computed, step 2 can be performed with a binary search, in $O(\ln M)$ operations.

This approach is efficient if the numbers W_k can be computed once, and then used several times, and if M is not too large. In several circumstances, however, the number M may be very large, and the numbers w_k may change, or, in other words, one is interested in sampling k with probability w_k without explicitly computing all the probabilities w_k. This can be done by using the acceptance-rejection technique, if at least an estimate $\overline{w}$ such that $\overline{w} \geq w_i, \ i = 1, \dots, M$ is known. The procedure is used as follows:

Algorithm 3.6.

1. select an integer random number uniformly in $[1, \dots, M]$, as

$$k = [\![M\xi_1]\!] + 1$$

   where $[\![x]\!]$ denotes the integer part of x,
2. if $w_k\xi_2 < \overline{w}$ then accept the sampling else reject it and repeat point 1.

Clearly the procedure can be generalized to the case in which the estimate depends on k, i.e. $\overline{w}_k \geq w_k$, $k = 1, \dots, M$.

The above technique is of crucial relevance in the Monte Carlo simulation of the scattering.

3.1.3. *Sampling without repetition*

Sometimes it is useful to extract n numbers, $n \leq N$, from the sequence $1, \dots, N$ without repetition. This sampling is often used in several Monte Carlo simulations. A simple and efficient method to perform the sampling is the following.

Algorithm 3.7.

```
1. set  $\mathtt{ind}_i = i$, $i = 1, \dots, N$
2. $M = N$
3. for $i = 1$ to $n$
       set $j = [\![M\xi]\!] + 1$, $\mathtt{seq}_i = \mathtt{ind}_j$,
       $\mathtt{ind}_j = \mathtt{ind}_M$, $M = M - 1$
   end for
```

At the end the vector `seq` will contain n distinct integers randomly sampled from the first N natural numbers. Of course if $n = N$, the vector `seq` contains a permutation of the sequence $1, \dots, N$.

3.1.4. *Multivariate distributions*

Suppose we want to sample a n-dimensional random variable $x = (x_1, \dots, x_n)$, whose probability density is $p_x(x)$.

If the density can be written as a product of densities of scalar random variables (marginal probability densities), i.e. if

$$p_x(x_1, \dots, x_n) = p_1(x_1)p_2(x_2)\cdots p_n(x_n),$$

then the n scalar random variables $x_1, \dots, x_n$ are independent, and they can be sampled separately, i.e. the problem is equivalent to sampling n monovariate random variables.

If this is not the case, then one may first look for a transformation $T : x \to \eta = T(x)$ such that in the new variables the probability density is factorized, i.e.

$$p_x(x_1, \dots x_n)dx_1dx_2 \dots dx_n = p_{\eta_1}(\eta_1)p_{\eta_2}(\eta_2)\cdots p_{\eta_n}(\eta_n)d\eta_1 d\eta_2 \dots d\eta_n, \tag{22}$$

then sample the variables $\eta_1, \dots \eta_n$, and finally compute x by inverting the map T, i.e. $x = T^{-1}(\eta)$.

In some cases such transformation can be readily found. In other cases it is more complicated. There is a general technique to find a mapping $T : x \to \xi$, where $\xi = (\xi_1, \dots, \xi_n)$ denotes a uniformly random variable in $[0, 1]^n$. Of course such transformation is not unique, since we only impose that its Jacobian determinant $J = |\partial\xi/\partial x|$ is a given function $p_x(x_1, \dots, x_n)$. An explicit transformation is constructed as follows

$$
\begin{aligned}
T_1(x_1) &= \int_{-\infty}^{x_1} d\eta \int_{I\!R^{n-1}} dx_2 \dots dx_n \, p_x(\eta, \dots, x_n), \\
T_2(x_1, x_2) &= \frac{\int_{-\infty}^{x_2} d\eta \int_{I\!R^{n-2}} dx_3 \dots dx_n \, p_x(x_1, \eta, \dots, x_n)}{\int_{-\infty}^{\infty} d\eta \int_{I\!R^{n-2}} dx_3 \dots dx_n \, p_x(x_1, \eta, \dots, x_n)}, \\
T_3(x_1, x_2, x_3) &= \frac{\int_{-\infty}^{x_3} d\eta \int_{I\!R^{n-3}} dx_4 \dots dx_n \, p_x(x_1, x_2, \eta, \dots, x_n)}{\int_{-\infty}^{\infty} d\eta \int_{I\!R^{n-3}} dx_4 \dots dx_n \, p_x(x_1, x_2, \eta, \dots, x_n)}, \\
&\vdots \\
T_n(x_1, \dots, x_n) &= \frac{\int_{-\infty}^{x_n} d\eta \, p_x(x_1, \dots, \eta)}{\int_{-\infty}^{\infty} d\eta \, p_x(x_1, \dots, \eta)}.
\end{aligned}
\tag{23}
$$

It is straightforward to check that $|\partial\xi/\partial x| = p_x(x_1, \dots x_n)$. Furthermore, the computation of the inverse requires the solution of a triangular system: find x_1 by solving the first equation of system (23), substitute x_1 in the second equation, and solve it for x_2, substitute x_1 and x_2 in the third equation and solve for x_3, and so on, therefore the systems of equations is equivalent to n single nonlinear equations for $x_1, \dots, x_n$.

Remark 3.8. The transformation (23) is used to map an arbitrary measure with density $p_x(x)$ into the Lebesgue measure on the unit cube $[0,1]^n$. This can be used, for example, to approximate a continuous measure by a discrete measure, once a good discrete approximation of the Lebesgue measure is known. Let us assume that we have a “good” approximation of the uniform measure obtained by N suitably chosen points $\eta_{(i)} \in [0,1]^n$, $i = 1, \dots, N$,

$$\frac{1}{N} \sum_{i=1}^{N} \delta(\eta - \eta_{(i)}) \approx \chi_{[0,1]^n}(\eta),$$

then

$$\frac{1}{N} \sum_{i=1}^{N} \delta(x - x_{(i)}) \approx p_x(x),$$

where $x_{(i)} = T^{-1}(\eta_{(i)}),\ i = 1, \dots, N$. This technique can be effectively used to obtain good quadrature formulae to compute integrals in highly dimensional space, and is the basis of the so called *quasi-Monte Carlo* integration [7, 26].

If the inverse transform map is too expensive, then the acceptance-rejection technique can be used, exactly as in the case of the monovariate distribution. More precisely, let x be a random variable with density $p_x(x)$, $x \in I\!R^n$. Then we look for a function $w(x) \geq p_x(x) \, \forall x \in I\!R^n$ which is “easy to sample”. Let $A = \int_{I\!R^n} w(x)\, dx$. Then the algorithm works as follows

Algorithm 3.9.

1. sample from x from $w(x)/A$ by any known method,
2. if $w(x)\xi < p_x(x)$ then accept the sample, else reject it and repeat step 1.

Example 3.10. As an example we show how to sample from a Gaussian distribution. Let x be a normally distributed random variable with zero mean and unit variance,

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

In order to sample from p one could invert the distribution function $P(x) = (1 + \mathrm{erf}(x/\sqrt{2}))/2$, where $\mathrm{erf}(x)$ denotes the error function. However the inversion of the error function may be expensive. An alternative procedure is obtained by the so called *Box-Muller method* described below. Let us consider a two dimensional normally distributed random variable. Then

$$p(x, y) = \frac{1}{2\pi} \exp\left(-\frac{x^2 + y^2}{2}\right) = p(x)p(y).$$

If we use polar coordinates

$$x = \rho \cos\theta, \; y = \rho \sin\theta, \tag{24}$$

then we have

$$\frac{1}{2\pi} \exp\left(-\frac{x^2 + y^2}{2}\right) dx\, dy = \frac{1}{2\pi} \exp\left(-\frac{\rho^2}{2}\right) \rho\, d\rho\, d\theta,$$

therefore in polar coordinates the density function is factorized as $p_\rho\, d\rho\, p_\theta\, d\theta$, with

$$\begin{aligned} p_\rho &= \exp\left(-\frac{\rho^2}{2}\right) \rho, \quad \rho \geq 0 \\ p_\theta &= \frac{1}{2\pi}, \quad 0 \leq \theta < 2\pi \end{aligned}$$

The random variables ρ and θ are readily sampled by inverting p_ρ and p_θ, i.e.

$$\rho = \sqrt{-2 \ln \xi_1}, \; \theta = 2\pi\xi_2,$$

and, finally, x and y are computed from Eq.(24).

At the end of the procedure we have two points sampled from a Normal(0,1) distribution (i.e. a Gaussian distribution with zero mean and unit variance). Of course, if the random variable has mean μ and standard deviation σ, then x and y will be scaled accordingly as

$$x = \mu_x + \sigma_x \rho \cos\theta, \; y = \mu_y + \sigma_y \rho \sin\theta.$$

Example 3.11. Here we show how to sample a point uniformly from the surface of a sphere. A point on a unit sphere is identified by the two polar angles (φ, θ) ,

$$\begin{aligned} x &= \sin\theta \cos\varphi, \\ y &= \sin\theta \sin\varphi, \\ z &= \cos\theta. \end{aligned}$$

Because the distribution is uniform, the probability of finding a point in a region is proportional to the solid angle

$$dP = \frac{d\omega}{4\pi} = \frac{\sin\theta\, d\theta}{2} \cdot \frac{d\varphi}{2\pi},$$

and therefore

$$\begin{aligned} \frac{d\varphi}{2\pi} &= d\xi_1, \\ \frac{\sin\theta\, d\theta}{2} &= d\xi_2. \end{aligned}$$

Integrating the above expressions we have

$$\begin{aligned} \varphi &= 2\pi\xi_1, \\ \theta &= \arccos(1-2\xi_2). \end{aligned}$$

3.2. Direct simulation schemes

In this paragraph we will describe the classical DSMC methods [1, 29, 3, 4] in the case of the spatially homogeneous Boltzmann equation (18).

We assume that the kinetic equations we are considering can be written in the form

$$\frac{\partial f}{\partial t} = \frac{1}{\epsilon}[P(f,f) - \mu f], \tag{25}$$

where $\mu \neq 0$ is a constant and $P(f,f)$ is a non negative bilinear operator. In particular, for both Kac equation and Boltzmann equation for Maxwellian molecules we have $P(f,f) = Q^+(f,f)$ where $Q^+(f,f)$ denotes the gain part of the collision operator.

The Kac equation in fact reads [20]

$$\frac{\partial f}{\partial t} = \frac{1}{\varepsilon}\int_0^{2\pi}\int_{-\infty}^{+\infty}\frac{1}{2\pi}[f(v')f(v'_*) - f(v)f(v_*)]\,dv_*\,d\theta, \tag{26}$$

where

$$v' = v\cos\theta - v_*\sin\theta, \qquad v'_* = v\sin\theta + v_*\cos\theta. \tag{27}$$

Hence, the equation is of the form (25) with

$$P(f,f) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} f(v')f(v'_*)\,dv_*\,d\theta, \tag{28}$$

and $\mu = \rho = \int_{I\!R} f(v)\,dv$. The only conserved quantities are the total mass and the energy, i.e. the collision invariants are $\phi(v) = 1, v^2$.

For the Boltzmann equation in the Maxwellian case, similarly we have

$$Q^+(f,f)(v) = \int_{I\!R^3}\int_{S^2}\sigma(\omega)f(v')f(v'_*)\,d\omega\,dv_*. \tag{29}$$

The case of general kernels with cut-off will be discussed later.

3.2.1. *Nanbu-Babovsky scheme*

We assume that f is a probability density, i.e.

$$\rho = \int_{-\infty}^{+\infty} f(v,t)\,dv = 1.$$

Let us consider a time interval $[0, t_{\max}]$, and let us discretize it in n_{TOT} intervals of size Δt. Let us denote by $f^n(v)$ an approximation of $f(v, n\Delta t)$. The forward Euler scheme applied to Eq. (25) writes

$$f^{n+1} = (1 - \frac{\mu\Delta t}{\epsilon})f^n + \frac{\mu\Delta t}{\epsilon}\frac{P(f^n,f^n)}{\mu}. \tag{30}$$

This equation has the following probabilistic interpretation: a particle with velocity v_i will not collide with probability $(1 - \mu\Delta t/\epsilon)$, and it will collide with probability $\mu\Delta t/\epsilon$, according to the collision law described by $P(f^n, f^n)(v)$.

Let us consider first kinetic equations for which $P(f,f) = Q^+(f,f)$, i.e. the collision kernel does not depend on the relative velocity of the particles.

An algorithm based on this probabilistic interpretation was proposed by Nanbu [29] and can be described as follows

Algorithm 3.12 (Nanbu for Maxwell molecules).

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N\}$,
   by sampling them from the initial density $f_0(v)$
2. for $n = 1$ to $n_{\mathrm{TOT}}$
     for $i = 1$ to $N$
         with probability $1 - \mu\Delta t/\epsilon$
                ∘ set $v_i^{n+1} = v_i^n$
         with probability $\mu\Delta t/\epsilon$
                ∘ select a random particle $j$
                ∘ compute $v_i'$ by performing the collision
                  between particle $i$ and particle $j$
                ∘ assign $v_i^{n+1} = v_i'$
     end for
   end for
```

In its first version, Nanbu's algorithm was not conservative, i.e. energy was conserved only in the mean, but not at each collision. A conservative version of the algorithm was introduced by Babovsky [1]. Instead of selecting single particles, independent particle pairs are selected, and conservation is maintained at each collision. The expected number of particles that collide in a small time step Δt is $N\mu\Delta t/\epsilon$, and the expected number of collision pairs is $N\mu\Delta t/(2\epsilon)$. The algorithm is the following

Algorithm 3.13 (Nanbu-Babovsky for Maxwell molecules).

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N\}$,
   by sampling them from the initial density $f_0(v)$
2. for $n = 1$ to $n_{\mathrm{TOT}}$
     given $\{v_i^n, i = 1, \dots, N\}$
          ∘ set $N_c = \mathrm{Iround}(\mu N\Delta t/(2\epsilon)$
          ∘ select $N_c$ pairs $(i, j)$ uniformly among all possible pairs,
            and for those
            - perform the collision between $i$ and $j$, and compute
              $v_i'$ and $v_j'$ according to the collision law
            - set $v_i^{n+1} = v_i'$, $v_j^{n+1} = v_j'$
          ∘ set $v_i^{n+1} = v_i^n$ for all the particles that have not been selected
   end for
```

Here by Iround(x) we denote a suitable integer rounding of a positive real number x. In our algorithm we choose

$$\mathrm{Iround}(x) = \begin{cases} [\![x]\!] + 1 & \text{with probability} \quad x - [\![x]\!] \\ [\![x]\!] & \text{with probability} \quad [\![x]\!] + 1 - x \end{cases}$$

where $[\![x]\!]$ denotes the integer part of x.

Remark 3.14. As we said before, the algorithm just described can be applied to the Kac equation and to the homogeneous Boltzmann equation with Maxwellian molecules. The only difference in the two cases consists in the computation of the post-collisional velocities.

When the above scheme is applied to the Kac equation, the new velocities v_i' and v_j' are computed as

$$v_i' = v_i \cos\theta - v_j \sin\theta, \quad v_j' = v_i \sin\theta + v_j \cos\theta, \tag{31}$$

where $\theta = 2\pi\,\text{Rand}$ and Rand denotes a random number, uniformly distributed in $[0,1]$.

For Maxwell molecules one has

$$v_i' = \frac{v_i + v_j}{2} + \frac{|v_i - v_j|}{2}\omega, \quad v_j' = \frac{v_i + v_j}{2} - \frac{|v_i - v_j|}{2}\omega, \tag{32}$$

where ω is chosen uniformly in the unit sphere, according to:

2D:

$$\omega = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}, \quad \theta = 2\pi\,\text{Rand}, \tag{33}$$

3D:

$$\omega = \begin{pmatrix} \cos\phi\sin\theta \\ \sin\phi\sin\theta \\ \cos\theta \end{pmatrix}, \quad \theta = \arccos(2\xi_1 - 1), \quad \phi = 2\pi\xi_2, \tag{34}$$

where ξ_1, ξ_2 are uniformly distributed random variables in $[0,1]$.

The above algorithm has to be modified when the scattering cross section is not constant. To this aim we shall assume that the collision kernel satisfies some cut-off hypothesis, which is essential from a numerical point of view.

We will denote by $Q_\Sigma(f,f)$ the collision operator obtained by replacing the kernel σ with the kernel σ_Σ

$$\sigma_\Sigma(|v - v_*|,\omega) = \min\{\sigma(|v - v_*|,\omega), \Sigma\}, \quad \Sigma > 0.$$

Thus, for a fixed Σ, let us consider the homogeneous problem

$$\frac{\partial f}{\partial t} = \frac{1}{\varepsilon} Q_\Sigma(f,f). \tag{35}$$

Problem (35) can be written in the form (25) taking

$$P(f,f) = Q_\Sigma^+(f,f) + f(v)\int_{\mathbb{R}^3}\int_{S^2} [\Sigma - \sigma_\Sigma(|v - v_*|,\omega)] f(v_*)\,d\omega\,dv_*, \tag{36}$$

with $\mu = 4\pi\Sigma\rho$ and

$$Q_\Sigma^+(f,f) = \int_{\mathbb{R}^3}\int_{S^2} \sigma_\Sigma(|v - v_*|,\omega) f(v') f(v_*')\,d\omega\,dv_*. \tag{37}$$

It is a simple exercise to verify that the functions $\phi(v) = 1, v, v^2$ are the collision invariants.

In this case, a simple scheme is obtained by using dummy collisions, and acceptance-rejection technique. Note that this approach is equivalent to sample the post collisional velocity according to $P(f,f)/\mu$, where $\mu = 4\pi\Sigma$ and Σ is an upper bound of the scattering cross section.

The conservative DSMC algorithm for VHS collision kernels can be written as

Algorithm 3.15 (Nanbu-Babovsky for VHS molecules).

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N\}$,
   by sampling them from the initial density $f_0(v)$
2. for $n = 1$ to $n_{\mathrm{TOT}}$
       given $\{v_i^n, i = 1, \dots, N\}$
           ○ compute an upper bound $\Sigma$ of the cross section
           ○ set $N_c = \mathrm{Iround}(N\rho\Sigma\Delta t/(2\epsilon))$
           ○ select $N_c$ dummy collision pairs $(i, j)$ uniformly
             among all possible pairs, and for those
             - compute the relative cross section $\sigma_{ij} = \sigma(|v_i - v_j|)$
             - if $\Sigma\,\mathrm{Rand} < \sigma_{ij}$
                   perform the collision between $i$ and $j$, and compute
                   $v_i'$ and $v_j'$ according to the collisional law
                   set $v_i^{n+1} = v_i'$, $v_j^{n+1} = v_j'$
           ○ set $v_i^{n+1} = v_i^n$ for all the particles that have not collided
   end for
```

The upper bound Σ should be chosen as small as possible, to avoid inefficient rejection, and it should be computed fast. It would be too expensive to compute Σ as

$$\Sigma = \sigma_{\max} \equiv \max_{ij} \sigma(|v_i - v_j|),$$

since this computation would require an $O(N^2)$ operations. An upper bound of $\sigma_{\max}$ is obtained by taking $\Sigma = \sigma(2\Delta v)$, where

$$\Delta v = \max_i |v_i - \bar{v}|, \quad \bar{v} := \frac{1}{N}\sum_i v_i.$$

The new velocities v_i' and v_j' are computed using Eqs. (32–34).

For general collision kernel, the algorithm is slightly modified by introducing the angular dependence. In this case the collision is always performed, and the new velocities are extracted according to the differential cross section. Then a rejection is used to decide whether the collision is accepted.

We remark here that the probabilistic interpretation of Eq. (30) breaks down if $\Delta t/\epsilon$ is too large, because the coefficient of f^n on the right hand side may become negative. This implies that the time step becomes extremely small when approaching the fluid dynamic limit. Therefore Nanbu-Babovsky method becomes unusable near the fluid regime.

We refer the reader to [2] for a detailed discussion on the convergence of Nanbu-Babovsky method.

Remark 3.16. We observe that in actual implementations, one chooses a given number of *dummy collision pairs*, equal to $\mu N\Delta t/2$. Such pairs are sampled uniformly among all possible pairs. The parameter μ is obtained by an estimate Σ of the maximum cross section between all particle pairs. Then dummy collisions are performed, and each of them is accepted with a probability which depends on the ratio $\sigma(v_i, v_j)/\Sigma$. If we do not had to compute Σ, then the computational cost of such Monte Carlo method would be independent on the time step, since its cost would be proportional to the number of collisions, which is basically independent on the time step, for small time steps. This is the case, for example, of Maxwell molecules. However, if we compute Σ at every time step, then the cost of the computation increases when we decrease the time step, since the estimate of Σ has a cost $\mathcal{O}(N)$ and not $\mathcal{O}(N\Delta t)$.

3.2.2. *Bird's scheme*

In the sixties G. Bird proposed a numerical scheme for the simulation of particle collisions. The relation to the Boltzmann equation has been conjectured since its first appearance, but it has been rigorously formalized only recently [45].

The method can be described as follows. Let us consider first the Maxwellian case. The number of collisions in a short time step Δt is given by

$$N_c = \frac{N\mu\Delta t}{2\varepsilon}.$$

This means that the average time between collisions Δt_c is given by

$$\Delta t_c = \frac{\Delta t}{N_c} = \frac{2\varepsilon}{\mu N}.$$

Now it is possible to set a time counter, t_c, and to perform the calculation as follows

Algorithm 3.17 (Bird for Maxwell molecules)**.**

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N\}$,
   by sampling them from the initial density $f_0(v)$
2. set time counter $t_c = 0$
3. set $\Delta t_c = 2\varepsilon/(\mu N)$
4. for $n = 1$ to $n_{\mathrm{TOT}}$
      ◦ repeat
          - select a random pair $(i, j)$ uniformly within all possible $N(N-1)/2$ pairs
          - perform the collision and produce $v_i'$, $v_j'$
          - set $\tilde{v}_i = v_i', \tilde{v}_j = v_j'$
          - update the time counter $t_c = t_c + \Delta t_c$
        until $t_c \geq (n+1)\Delta t$
      ◦ set $v_i^{n+1} = \tilde{v}_i, i = 1, \dots, N$
   end for
```

p The above algorithm is very similar to the Nanbu-Babovsky (NB) scheme for Maxwellian molecules or for the Kac equation. The main difference is that in NB scheme the particles can collide only once per time step, while in Bird's scheme multiple collisions are allowed. This has a profound influence on the time accuracy of the method. In fact, while the solution of the NB scheme converges in probability to the solution of the time discrete Boltzmann equation, Bird's method converges to the solution of the space homogeneous Boltzmann equation. In this respect it may be considered a scheme of infinite order in time. For the space homogeneous Boltzmann equation, the time step Δt, in fact, can be chosen to be the full time span $t_{\max}$. When the full space non-homogeneous Boltzmann equation is considered, however, then the scheme is used on a time step Δt which is the time discretization used in the splitting scheme. When a more general gas is considered, Bird's scheme has to be modified to take into account that the average number of collisions in a given time interval is not constant, and that the collision probability on all pairs is not uniform. This can be done as follows. The expected number of collisions in a time step Δt is given by[1]

$$N_c = \frac{N\rho\overline{\sigma}\Delta t}{2\varepsilon},$$

[1] In sections 3, 4, and 5 we denote by ρ rather by ρ/m the number density of particles (particles per unit volume).

where $\overline{\sigma}$ denotes the average collision frequency, and can be computed, in principle, by

$$\overline{\sigma} = \frac{2}{N(N-1)} \sum 1 \leq i < j \leq N \sigma(|v_i - v_j|). \tag{38}$$

Once the expected number of collisions is computed, then the mean collision time can be computed as

$$\Delta t_c = \frac{\Delta t}{N_c} = \frac{2\varepsilon}{N\rho\overline{\sigma}}.$$

The N_c collisions have to be performed with probability proportional to $\sigma_{ij} = \sigma(|v_i - v_j|)$. In order to do this efficiently, one can use the same acceptance-rejection technique use in Nanbu-Babovsky scheme. The drawback of this procedure is that expression (38) for $\overline{\sigma}$ is too expensive. A very simple and effective solution to this problem is to compute a local time counter Δt_c as follows. First select a collision pair (i,j) using rejection. Then compute

$$\Delta t_{ij} = \frac{2\varepsilon}{N\rho\sigma_{ij}}.$$

It is immediate to show that this choice gives the correct expected value for the collision time

$$\Delta t_c = \sum_{1\leq i<j\leq N} \Delta t_{ij} p_{ij},$$

where

$$p_{ij} = \frac{\sigma_{ij}}{\sum_{1\leq i<j\leq N} \sigma ij}$$

is the conditional probability that pair (i,j) collides. Performing the calculation one has

$$\Delta t_c = \sum_{1\leq i<j\leq N} \Delta t_{ij} p_{ij} = \frac{2\varepsilon}{N\rho\overline{\sigma}}.$$

Bird's algorithm for general VHS molecules can therefore be summarized in the following algorithm:

Algorithm 3.18 (Bird for VHS molecules).

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N\}$,
   by sampling them from the initial density $f_0(v)$
2. set time counter $t_c = 0$
3. for $n = 1$ to $n_{\mathrm{TOT}}$
        ∘ compute an upper bound $\Sigma$ of the cross section
        ∘ repeat
          - select a random pair $(i,j)$ uniformly within all possible $N(N-1)/2$ pairs
          - compute the relative cross section $\sigma_{ij} = \sigma(|v_i - v_j|)$
          - if $\Sigma$ Rand $< \sigma_{ij}$
              perform the collision between $i$ and $j$, and compute
              $v_i'$ and $v_j'$ according to the collisional law
              set $\tilde{v}_i = v_i', \tilde{v}_j = v_j'$
              set $\Delta t_{ij} = 2\varepsilon/(N\rho\sigma_{ij})$
              update the time counter $t_c = t_c + \Delta t_{ij}$
     until $t_c \geq (n+1)\Delta t$
```

```
        ∘ set $v_i^{n+1} = \tilde{v}_i, i = 1, \dots, N$
    end for
```

Remark 3.19. At variance with NB scheme, there is no restriction on the time step Δt for Bird's scheme. For space homogeneous calculations Δt could be chosen to be the total computation time $t_{\max}$. However, the scheme requires an estimate of $\sigma_{\max}$, and this has to be updated in time. This can be done either by performing the estimate at certain discrete time steps as in NB scheme, or by updating its value at every collision process. A possible solution in $O(1)$ operations is to check whether the new particles v_i', v_j' generated at each collision increase the quantity $\Delta v = \max_j |v_j - \bar{v}|$.

Unfortunately Bird's method too becomes very expensive and practically unusable near the fluid regime because in this case the collision time between the particles Δt_{ij} becomes very small, and a huge number of collisions is needed in order to reach a fixed final time $t_{\max}$.

4. Time Relaxed Monte Carlo (TRMC) methods

We have seen in the previous section that standard DSMC methods lose their efficiency near the fluid regime. We will describe in this section some recently developed Monte Carlo methods for the acceleration of DSMC computations near continuum regimes. Numerical results for space homogeneous problems will be shown in Section 5, whereas applications to shock wave computations will be given in Section 6.

We start the section with a short review of the Wild sum truncation [16] of the Boltzmann equation which is the starting point for the construction of TRMC methods.

4.1. **Asymptotic preserving schemes**

As proposed in [16], a general idea for deriving unconditionally stable and asymptotic preserving numerical schemes for a nonlinear equation like (18) is to replace high order terms of a suitable well-posed power series expansion by the local equilibrium. This class of schemes has been called *Time Relaxed* (TR) schemes.

Here we will briefly recall the schemes presented in [16] and some generalization used in [34].

4.1.1. *Time relaxed (TR) schemes*

Let us consider a differential system of the type (25) with the initial condition $f(v, t = 0) = f_0(v)$. We have seen in the previous section that the Boltzmann equation with a bounded kernel can be written in the form (25).

Let us replace the time variable t and the function $f = f(v,t)$ using the transformations

$$\tau = (1 - e^{-\mu t/\varepsilon}), \qquad F(v,\tau) = f(v,t)e^{\mu t/\varepsilon}.$$

Then F is easily shown to satisfy

$$\frac{\partial F}{\partial \tau} = \frac{1}{\mu} P(F,F) \tag{39}$$

with $F(v, \tau = 0) = f_0(v)$.

Now, the solution to the Cauchy problem for (39) can be sought in the form of a power series

$$F(v,\tau) = \sum_{k=0}^{\infty} \tau^k f_k(v), \qquad f_{k=0}(v) = f_0(v) \tag{40}$$

where the functions f_k are given by the recurrence formula

$$f_{k+1}(v) = \frac{1}{k+1} \sum_{h=0}^{k} \frac{1}{\mu} P(f_h, f_{k-h}), \quad k = 0, 1, \dots \tag{41}$$

Making use of the original variables we obtain the following formal representation of the solution to the Cauchy problem for (25)

$$f(v,t) = e^{-\mu t/\varepsilon} \sum_{k=0}^{\infty} \left(1 - e^{-\mu t/\varepsilon}\right)^k f_k(v). \tag{42}$$

Remark 4.1. The method was originally developed by Wild [47] to solve the Boltzmann equation for Maxwellian molecules. Here we describe the method under more general hypothesis on P, as derived in [16]. We emphasize that the representation (42) is not unique and other well-posed power series expansions can be obtained in a similar way [16].

Finally, note that expansion (42) continues to hold also if μ is a function of v. Unfortunately this choice leads to nonconservative schemes.

From this representation, a class of numerical schemes can be naturally derived.

In [16] the following class of numerical schemes, based on a suitable truncation for $m \geq 1$ of (42), has been constructed

$$f^{n+1}(v) = e^{-\mu \Delta t/\varepsilon} \sum_{k=0}^{m} \left(1 - e^{-\mu \Delta t/\varepsilon}\right)^k f_k^n(v) + \left(1 - e^{-\mu \Delta t/\varepsilon}\right)^{m+1} M(v), \tag{43}$$

where $f^n = f(n\Delta t)$ and Δt is a small time interval. The quantity M (referred as the local Maxwellian associated to f) is the asymptotic stationary solution of the equation.

It can be shown that the schemes obtained in this way are of order m in time. Furthermore, we have [16] the following

Proposition 4.2. *The schemes defined by (43) satisfy*

i) conservation:

If $P(f,g)$ is a non negative bilinear operator such that there exist some functions $\phi(v)$ with the following property

$$\int_{R^3} P(f,f)\phi(v)\, dv = \mu \int_{R^3} f\phi(v)\, dv, \quad \forall f > 0,\ f \in L^1(I\!R^3), \tag{44}$$

and the initial condition f^0 is a non negative function, then f^{n+1} is nonnegative for any $\mu\Delta t/\varepsilon$, and satisfies

$$\int_{R^3} f^{n+1}\phi(v)\, dv = \int_{R^3} f^n \phi(v)\, dv. \tag{45}$$

ii) asymptotic preservation (AP):

For any $m \geq 1$, we have

$$\lim_{\mu\Delta t/\varepsilon \to \infty} f^{n+1} = M(v). \tag{46}$$

When applied to the Boltzmann equation, the first property guarantees exact conservation of mass, momentum, and energy, for the semi-discrete scheme, while the second property guarantees the correct fluid dynamic limit.

4.1.2. *Generalized TR schemes*

The approach we presented above can be generalized using different weight functions to combine the influence of the high order coefficients appearing in the Wild sum (42). In general such schemes can be written as

$$f^{n+1} = \sum_{k=0}^{m} A_k f_k + A_{m+1} M, \tag{47}$$

where the coefficients f_k are given by Eq. (41).

The weights $A_k = A_k(\tau)$ are nonnegative functions that satisfy the following

Proposition 4.3. *Let the weights $A_k = A_k(\tau)$ be nonnegative functions that satisfy*

i) consistency**:**

$$\lim_{\tau\to 0} A_1(\tau)/\tau \quad = \quad 1, \quad \lim_{\tau\to 0} A_k(\tau)/\tau = 0, \quad k = 2,\dots,m+1 \tag{48}$$

ii) conservation**:**

$$\sum_{k=0}^{m+1} A_k = 1 \quad \tau \in [0,1], \tag{49}$$

iii) asymptotic preservation (AP)**:**

$$\lim_{\tau\to 1} A_k(\tau) = 0, \quad k = 0,\dots,m \tag{50}$$

then (47) is a consistent discretization of problem (25) that satisfies proposition 4.2.

The proof is a simple exercise and we leave it to the reader.

A choice of functions which satisfies the previous requirements is given by

$$A_k = (1-\tau)\tau^k,\ k = 0,\dots,m, \quad A_{m+1} = \tau^{m+1}, \tag{51}$$

which corresponds to the scheme (43). A better choice of parameters is [33]

$$A_k = (1-\tau)\tau^k,\ k = 0,\dots,m-1, \quad A_m = 1 - \sum_{k=0}^{m} A_k - A_{m+1}, \quad A_{m+1} = \tau^{m+2}, \tag{52}$$

which corresponds to take $f_{m+1} = f_m$, $f_k = M$, $k \geq m+2$ in (42).

However, other choices are possible and it is an open problem the determination of the *optimal* set of functions A_k that satisfies the previous requirements and guarantees the most accurate approximation.

The stability properties of such schemes is studied in [33].

4.2. **Time Relaxed Monte Carlo methods**

In this section we describe a class of simulation schemes for the Boltzmann equation, the *Time Relaxed Monte Carlo* (TRMC) methods, based on the probabilistic interpretation of the previous time discretization.

Since we deal with the space homogeneous case, we shall make the simplifying assumption that the function $f^n(v)$ is a probability distribution, i.e.

$$\int_{I\!R^3} f^n(v)\,dv = 1.$$

We recall that the probability interpretation of the Nanbu-Babovsky scheme we have seen in the previous chapter is valid as long as $\mu\Delta t \leq 1$. This corresponds to the stability restriction of the explicit Euler scheme.

4.2.1. *TRMC methods*

The first order TRMC algorithm is based on the TR schemes

$$f^{n+1} = A_0 f^n + A_1 f_1 + A_2 M \tag{53}$$

The probabilistic interpretation of the above equation is the following: a particle extracted from f^n will not collide with probability A_0, it will collide with another particle extracted from f^n with probability A_1, or it will be replaced by a particle sampled from a Maxwellian with probability A_2.

In this formulation the probabilistic interpretation holds uniformly in $\mu\Delta t/\epsilon$, at variance with standard Nanbu's method, which requires $\mu\Delta t/\epsilon < 1$. Furthermore, as $\mu\Delta t/\epsilon \to \infty$, the distribution at time $n+1$ is sampled from a Maxwellian. In this limit, the density f^{n+1} relaxes immediately to its equilibrium distribution. In a space non-homogeneous case, this would be equivalent to the particle method for Euler equations proposed by Pullin [37].

The Monte Carlo scheme described above are conservative in the mean. It is possible to make it exactly conservative by selecting collision pairs uniformly, rather than individual particles, and by using a suitable algorithm for sampling a set of particles with prescribed momentum and energy from a Maxwellian [38].

The conservative version of the methods can be formalized in the following algorithm

Algorithm 4.4 (First order TRMC for VHS molecules)**.**

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N\}$,
   by sampling them from the initial density $f_0(v)$
2. for $n = 1$ to $n_{\rm TOT}$
      given $\{v_i^n, i = 1, \dots, N\}$,
         ∘ compute an upper bound $\Sigma$ of the cross section
         ∘ set $\tau = 1 - \exp(-\rho\Sigma\Delta t/\epsilon)$
         ∘ compute $A_1(\tau)$, $A_2(\tau)$
         ∘ set $N_c = {\rm Iround}(NA_1/2)$
         ∘ perform $N_c$ dummy collisions, as in Algorithm 3.15
         ∘ set $N_M = {\rm Iround}(NA_2)$
         ∘ select $N_M$ particles among those that have not collided,
           and compute their mean momentum and energy
         ∘ sample $N_M$ particles from the Maxwellian with the above
           momentum and energy, and replace the $N_M$ selected particles
           with the sampled ones
         ∘ set $v_i^{n+1} = v_i^n$ for all the $N - 2N_c - N_M$ particles that
           have not been selected
   end for
```

A second order Monte Carlo scheme is obtained by the TR scheme

$$f^{n+1} = A_0 f^n + A_1 f_1 + A_2 f_2 + A_3 M, \tag{54}$$

with

$$f_1 = \frac{P(f^n, f^n)}{\mu}, \quad f_2 = \frac{P(f^n, f_1)}{\mu}.$$

Given N particles distributed according to f^n, the probabilistic interpretation of scheme (54) is the following: NA_0 particles will not collide, NA_1 will be sampled from f_1 (as in the first order scheme), NA_2 will be sampled from f_2, i.e. $NA_2/2$ particles sampled from f^n will undergo dummy collisions with $NA_2/2$ particles sampled from f_1, and NA_3 particles will be sampled from a Maxwellian.

Once again, the methods can be made conservative using the same techniques adopted in the first order scheme. The various steps of the method can be summarized in the following algorithm.

Algorithm 4.5 (Second order TRMC for VHS molecules)**.**

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N\}$,
   by sampling them from the initial density $f_0(v)$
2. for $n = 1$ to $n_{\rm TOT}$
      given $\{v_i^n, i = 1, \dots, N\}$,
          ∘ compute an upper bound $\Sigma$ of the cross section
          ∘ set $\tau = 1 - \exp(-\rho\Sigma\Delta t/\epsilon)$
          ∘ compute $A_1(\tau)$, $A_2(\tau)$, $A_3(\tau)$
          ∘ set $N_1 = \mathrm{Iround}(NA_1/2)$ , $N_2 = \mathrm{Iround}(NA_2/4)$
          ∘ select $N_1 + N_2$ dummy collision pairs $(i, j)$ uniformly
            among all possible pairs
          ∘ perform $N_1$ dummy collisions, as in Algorithm 3.15
          ∘ for $N_2$ pairs
            - compute the relative cross section $\sigma_{ij} = \sigma(|v_i - v_j|)$
            - if $\Sigma\,\mathrm{Rand} < \sigma_{ij}$
                  perform the collision between $i$ and $j$,
                  compute $v_i'$ and $v_j'$ according to the collisional law and
                  store them
          ∘ select $2N_2$ particles from $f^n$
          ∘ perform the collision of these selected particles with the
            set of $2N_2$ particles that have collided once and stored
          ∘ update the velocity of the $4N_2$ particles with the outcome
            of the $2N_2$ collisions (of particles that have never collided
            before with particles that collided once)
          ∘ set $N_M = \mathrm{Iround}(NA_3)$
          ∘ replace $N_M$ particles with samples from Maxwellian, as in Algorithm 4.4
          ∘ set $v_i^{n+1} = v_i^n$ for all the $N - 2N_1 - 4N_2 - N_M$ particles that
            have not been selected
   end for
```

Similarly higher order TRMC methods can be constructed. For example a third order scheme is obtained from

$$f^{n+1} = A_0 f^n + A_1 f_1 + A_2 f_2 + A_3 f_3 + A_4 M, \tag{55}$$

with

$$f_1 = \frac{P(f^n, f^n)}{\mu}, \quad f_2 = \frac{P(f^n, f_1)}{\mu}, \quad f_3 = \frac{1}{3\mu}[2P(f^n, f_2) + P(f_1, f_1)].$$

We omit for brevity the details of the resulting Monte Carlo algorithm.

4.2.2. *Hybrid TRMC methods*

The variance of the Monte Carlo solution can be considerably reduced if one represents the equilibrium part of the distribution function analytically by a Maxwellian. Such idea has been recently presented in the paper [32]. One makes the *ansatz* that the distribution function is the sum of a continuous and a discrete part,

$$f^n(v) = (1-\beta^n)g^n(v) + \beta^n M(v).$$

By substituting this expression in the first order scheme (53), one obtains the evolution equations for β^n and g^n

$$\begin{aligned} \beta^{n+1} &= A_0\beta^n + A_1(\beta^n)^2 + A_2 && (56)\\ g^{n+1} &= p_1 g^n + p_2\left[q_1\frac{P(g^n,g^n)}{\mu} + q_2\frac{P(g^n,M)}{\mu}\right], && (57) \end{aligned}$$

where

$$p_1 = \frac{A_0}{A_0 + A_1(1+\beta^n)}, \quad p_2 = \frac{A_1(1+\beta_n)}{A_0 + A_1(1+\beta_n)}, \tag{58}$$

$$q_1 = \frac{1-\beta^n}{1+\beta^n}, \quad q_2 = \frac{2\beta^n}{1+\beta^n}. \tag{59}$$

Note that $p_1, p_2 \geq 0$, $p_1 + p_2 = 1$, $q_1, q_2 \geq 0$, $q_1 + q_2 = 1$.

Equation (56) is an iterated map for the coefficient β and can be rewritten using the conservation property (49) as

$$\beta^{n+1} - \beta^n = A_1(\beta^n - 1)(\beta^n - A_2/A_1). \tag{60}$$

This discrete dynamical system has stationary points $\beta = 1$ and $\beta = A_1/A_2$. Our interest is only in $0 \leq \beta \leq 1$, which is an invariant region for the system. Hence we can state the following results:

Proposition 4.6.

i) *If $A_2/A_1 > 1$, then $\beta = 1$ is an attracting point and $\beta^n \to 1$ as $n \to \infty$ at an exponential rate with coefficient $(A_2 - A_1)$, i.e.*

$$1 - \beta^n \cong \alpha(A_2 - A_1)^n, \quad \alpha \neq 0, \tag{61}$$

for n large, if $A_2/A_1 > 1$.

ii) *If $A_2/A_1 < 1$, then $\beta = 1$ is unstable and $\beta = A_2/A_1$ is attracting. In this case $\beta^n \to A_2/A_1$ as $n \to \infty$ at an exponential rate with coefficient $(A_1 - A_2)$, i.e.*

$$|A_2/A_1 - \beta^n| \cong \alpha(A_1 - A_2)^n, \quad \alpha \neq 0, \tag{62}$$

for n large, if $A_2/A_1 < 1$.

Remark 4.7. The requirement $A_2/A_1 > 1$ cannot be verified uniformly in $\lambda = \mu\Delta t/\epsilon$. However, the fluid regime corresponds to $\lambda \gg 1$ so that $A_2/A_1 \gg 1$ because of the asymptotic preserving property. This shows that β^n increases monotonically to 1 in the fluid region, as desired.

For example, for the first order scheme corresponding to (51), $A_2/A_1 = \tau/(1-\tau)$. It follows that $\beta^n \to 1$ if $\tau > 1/2$. Similarly the scheme characterized by (52) has $A_2/A_1 = \tau^2/(1-\tau^2)$. Hence $\beta^n \to 1$ if $\tau > 1/\sqrt{2}$. Clearly, near the fluid limit $\mu\Delta t/\varepsilon \gg 1$ and hence $\tau = 1 - e^{-\mu\Delta t/\varepsilon} \approx 1$.

The probabilistic interpretation of the evolution equation for g^n is similar to the one for the purely particle scheme.

An acceptance-rejection technique, similar to the one used for DSMC, can be adopted. The conservative algorithm to update β^n and g^n can be written as

Algorithm 4.8 (Hybrid TRMC for VHS molecules).

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N^0\}$,
   by sampling them from the initial density $f_0(v)$
2. for $n = 0$ to $n_{\mathrm{TOT}} - 1$
       given $\{v_i^n, i = 1, \dots, N^n\}$
       ∘ set $\beta = 0, N_c = 0$
       ∘ compute an upper bound $\Sigma$ of the cross section
       ∘ set $\tau = 1 - \exp(-\rho\Sigma\Delta t/\epsilon)$ and
         compute the corresponding quantities $A_0, A_1, A_2, p_1, p_2, q_1, q_2$
       ∘ set $N_{gM} = \mathrm{Iround}(p_2 q_2 N^n/2)$ and $N_{gg} = \mathrm{Iround}(p_2 q_1 N^n/2)$
       ∘ perform $N_{gg}$ dummy collisions between $g$-particles,
           - extract $(i, j)$ without repetition
           - if $\Sigma\,\mathrm{Rand} < \sigma_{ij}$ then perform the collision between $v_i$ and $v_j$
               set $\tau_{ij} = 1 - \exp(-\rho\sigma_{ij}\Delta t/\epsilon)$ and
               compute the corresponding quantities $A_0^{ij}, A_1^{ij}, A_2^{ij}, \beta_{ij}$
               set $\beta = \beta + 2\beta_{ij}$
               set $N_c = N_c + 2$
       ∘ perform $2N_{gM}$ dummy collisions between the $g$-particles and the Maxwellian,
           - extract $i$ without repetition
           - sample one particle, $m$, from the Maxwellian
           - if $\Sigma\,\mathrm{Rand} < \sigma_{ij}$ then perform the collision between $v_i$ and the Maxwellian
               compute $A_0^{im}, A_1^{im}, A_2^{im}, \beta_{im}$
               set $\beta = \beta + 2\beta_{im}$
               set $N_c = N_c + 1$
       ∘ set $\beta^{n+1} = \beta/N_c$
       ∘ update $N$: $N^{n+1} = \mathrm{Iround}(N^0(1 - \beta^{n+1}))$
       ∘ correct $\beta^{n+1}$ in order to preserve mass
       ∘ correct $u_M$ and $T_M$ to maintain conservation of momentum and energy.
   end for
```

The above scheme conserves momentum and energy only on the average, but not exactly. This is because the collisions with the Maxwellian M, if performed independently from each other, do not maintain exact conservation of momentum and energy. By taking this into account, a conservative algorithm can be constructed by changing momentum u_M and energy E_M of the Maxwellian fraction after each collision (*local conservation*), according to

$$u'_M = u_M - \frac{v'_j - v_j}{N^0\beta^n}, \quad E'_M = E_M - \frac{(v'_j)^2 - (v_j)^2}{2N^0\beta^n}, \quad n > 0, \tag{63}$$

where

$$E_M = \frac{1}{2}(dT_M + u_M^2),$$

d is the dimension of the velocity space and T_M is the temperature of the Maxwellian.

Alternatively, conservation can be restored by modifying the moments associated to the Maxwellian fraction at the end of each time step (*global conservation*).

This is obtained by imposing

$$(1-\beta^{n+1})E_p + \beta^{n+1}E_M = E^0, \quad (1-\beta^{n+1})u_p + \beta^{n+1}u_M = u^0, \quad n > 0,$$

where u_p and E_p are the mean velocity and energy of the particles.

The two approaches give very similar results and can be used if β^n is not too small.

If the distribution is very far from equilibrium, i.e. if $\beta^n \ll 1$, then because of fluctuations, it may happen that the energy decreases too much, and it is impossible to change the parameters of the Maxwellian to impose conservation. On the other hand, in this case, only a very small fraction of collisions will be non conservative, and therefore the lack of exact conservation will not affect the quality of the result.

Note that with this approach the number of particles is a function of time (it is $N^n \approx N^0(1-\beta^n)$).

The advantage of this approach with respect to the purely particle method is twofold: the variance of the distribution is reduced because a fraction of the distribution function is represented analytically, and the scheme is more efficient, because the number of particles is reduced.

On the other hand the extension of this hybrid method to space non homogeneous problems is non trivial since after convection the continuous part is no longer Maxwellian. Hence it must be approximated (from below) by a Maxwellian, and the excedence should be transformed into particles (see [8] for more details).

4.2.3. *Recursive TRMC methods*

First we note that in the case of Maxwell molecules the Wild sum (42) has a very clear probabilistic interpretation. Now $f(\cdot,\Delta t)$ is the velocity distribution of particles at time Δt. Taking a particle at random from this distribution, it might happen that it has not collided one single time. The probability of this event is $\exp(-\mu\Delta t/\varepsilon)$, and the velocity distribution given this is f_0. The first term f_1 in the sum (42) corresponds to those particles that have been involved in exactly one collision. The probability that a particle has such a history is $(1-\tau)\tau = \exp(-\mu\Delta t/\varepsilon)\big(1-\exp(-\mu\Delta t/\varepsilon)\big)$, and the velocity distribution is exactly $P(f_0,f_0)/\mu$.

In the same way f_n, defined recursively by eq. (41), is the conditional velocity distribution given that exactly $n+1$ particles have been involved in its collision history back to the initial time. To be able to find a sample of f_n, we must assume that the densities f_k, $0 \le n-1$ are already known. Of course the only one of these that is really known is f_0, the initial distribution. All this is well described for Maxwell molecules e.g. in [25, 35].

Thus, for Maxwell molecules we can sample particles directly from the solution $f(\cdot,\Delta t)$ using the following simple recursive algorithm. The method is essentially equivalent to the one given in [13].

Algorithm 4.9 (Recursive sampling for Maxwell molecules).

```
1. choose $n$ from a geometric distribution with parameter $\tau = 1-\exp(-\mu\Delta t/\varepsilon)$,
   the expected is $E[n] = \exp(\mu\Delta t/\varepsilon)$, and hence grows rapidly with $\mu\Delta t/\varepsilon$
2. take a sample from the distribution with density $f_n$
     ∘ if $n = 0$, take $v$ from the initial density $f_0$
     ∘ else proceed as follows
         - choose $k \in \{0,1,\dots,n-1\}$ with equal probability
         - take a sample $v_i$ from the density $f_k$,
           and $v_j$ from the density $f_{n-k-1}$ as in step 2
         - perform the collision between $v_i$ and $v_j$
         - then $v = v'_i$ is distributed according to the density $f_n$
```

Note that the samples from f_k and f_{n-k-1} are computed using the same strategy simply performing step 2 recursively. This process always terminates after at most n steps. The algorithm is easy to program in a recursive way, but for many reasons it is not very useful method for Monte Carlo-simulations of the Boltzmann equation. In fact the expected value of n grows exponentially with $\mu\Delta t$ and the inverse of the Knudsen number, and both the computational time and the storage requirement are proportional to $E[n]$. Moreover the method is non conservative.

The method that we now describe will solve the previous drawbacks by symmetrizing the Wild sum calculation, and using fact that the terms of the Wild sum converge to the Maxwellian. This idea is based on the Wild sum truncation we have introduced in the previous paragraphs.

Given a set of N particles distributed according to f_0, then, at time Δt one tries to split the particles into

i) $N\exp(-\mu\Delta t/\varepsilon)$ particles sampled from the initial distribution f_0,

ii) $N\exp(-\mu\Delta t/\varepsilon)\big(1-\exp(-\mu\Delta t/\varepsilon)\big)^n$ particles sampled from f_n, for each $n=1,..,m$

iii) $N\big(1-\exp(-\mu\Delta t/\varepsilon)\big)^{m+1}$ particles sampled from a Maxwellian.

Note that we have

$$\big(1-\exp(-\mu\Delta t/\varepsilon)\big)^n \geq \big(1-\exp(-\mu\Delta t/\varepsilon)\big)^{n+1},$$

and hence the number of particles sampled form f_n is a non increasing function of n.

A schematic description of the algorithm is the following. At the beginning of the collision time step Δt, there are N particles distributed according to f_0. After the recursive collisions, at time Δt, we expect a certain number of particles distributed according to the initial distribution (called f_0), a certain number of particles distributed according to f_1, and so on. For each of these, the final number of particles is given by N_n.

The very first step in this calculation is to compute a first pair of particles (velocities) distributed according to f_m. For VHS molecules this involves the dummy-collisions of $m+1$ particles, which eventually are drawn from the initial condition. Obviously this sets a limit of the number of terms of the Wild-sum that can be estimated with a finite number of particles at the initial time.

All $m+1$ particles must be kept so that the conserved quantities remain exact. In the particular example shown in 2, a pair of f_5-particles are generated, and on the way of doing this, also two f_1-particles, and two f_2-particles are generated. For each of these pairs one particle is stored and the relative c_1 or c_2 counters incremented by one.

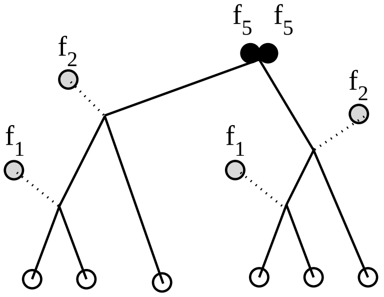

FIGURE 2. Generation of particles in the algorithm

It is clear that if this procedure is continued until the appropriate number N_m of f_m-particles are generated, one risks finding that there are not enough particles at the initial step, and also that the other f_n's are not properly distributed. A way out of this problem is the following. A collision tree as the one shown in figure 2 is evaluated from top to the bottom, just as described in Algorithm 4.9.

The difference here is that each time a particle from a given distribution is needed for a dummy-collision, one first checks whether such a particle has already been stored. In the example above, the upper collision involves two f_2-particles, and if one or two particles already exist, then those are taken as collision partners and the corresponding counter c_2 decreased by one. If there are more than needed, then there is a random choice, and if there are none, or not enough, then the algorithm is called recursively, just as before.

This is repeated until all particles with distribution f_1 up to f_m are generated. The final step is to generate N_{m+1} particles distributed according to a Maxwellian with accurate moments. For this, N_{m+1} particles are taken from the initial distribution, the first moments are computed, and then they are discarded, and replaced by the same number of particles distributed according to a Maxwellian with the corresponding moments. This last step is carried out in a conservative way using Pullin's algorithm [38].

Starting with N particles distributed according to f_0 the method is conservative and can be summarized in the following algorithm.

Algorithm 4.10 (Recursive TRMC for VHS molecules)**.**

```
1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \dots, N\}$,
   by sampling them from the initial velocity $f_0$
2. set $\tau = 1 - \exp(mu\Delta t/\varepsilon)$, $N_0 = \mathrm{Iround}((1-\tau))$, $\tilde{N} = N_0$, $n = 0$
3. repeat
     ∘ set $n = n + 1$
     ∘ compute $N_n = \mathrm{Iround}(N(1-\tau)\tau^n)$
     ∘ set $\tilde{N} = \tilde{N} + N_n$
   until $(\tilde{N} > N)$ or $(N_n < 1)$
4. set $m = n - 1$, $N_{m+1} = N - (N_p - N_n)$.
5. set counters $c_n = 0$ for $n = 1, \dots, m + 1$.
6. compute an upper bound $\Sigma$ of $\sigma_{ij}$ (as in DSMC)
7. for $n = m, \dots, 1$
   take $N_n$ samples from the distribution with density $f_n$, according to
     ∘ repeat
         - choose $k \in \{0, 1, \dots, n-1\}$ with equal probability
         - if $k = 0$ take $v_i$ from the initial density $f_0$
         - else choose $v_i$ from the density $f_k$
               if $c_k > 0$ use a stored particle with a random choice
               set $c_k = c_k - 1$ and $N_k = N_k + 1$
               else sample $v_i$ and $v_i^*$ from $f_k$ (recursively)
               $v_i^*$ is stored and then set $c_k = c_k + 1$, $N_k = N_k - 1$
         - if $n - k - 1 = 0$ take $v_j$ from the initial density $f_0$
         - else choose $v_j$ from the density $f_{n-k-1}$
               if $c_{n-k-1} > 0$ use a stored particle with a random choice
               set $c_{n-k-1} = c_{n-k-1} - 1$ and $N_{n-k-1} = N_{n-k-1} + 1$
               else sample $v_j$ and $v_j^*$ from $f_{n-k-1}$ (recursively)
               $v_j^*$ is stored and then set $c_{n-k-1} = c_{n-k-1} + 1$, $N_{n-k-1} = N_{n-k-1} - 1$
         - if $\Sigma\,\mathrm{Rand} < \sigma_{ij}$ perform the dummy-collision between $v_i$ and $v_j$ (as in DSMC)
         - $v_i'$ and $v_j'$ are random variables distributed according to the density $f_n$
         - set $N_n = N_n - 2$
       until $(N_n > 0)$
   end for
8. Sample $N_{m+1}$ particles from the Maxwellian as in algorithm 4.4
```

Note that, from a practical viewpoint, the number m can be rather large. Thus a maximum allowed value m_{max} of m is fixed (which represent the maximum depth of a possible tree) at the beginning of the computations. Clearly small values of m will provide a faster but in general less accurate algorithm.

5. Space homogeneous numerical results

In this section we test the different TRMC methods presented in the previous section by comparing them with the classical DSMC schemes we have seen in Section 3. Since we are interested in checking the accuracy of the schemes we first consider two initial value problems, respectively for the Kac equation and for the Boltzmann equation for Maxwell molecules, for which we have an exact solution. To simplify notations we denote by TRMC1, TRMC2, TRMC3, TRMCH, TRMCR respectively the first order, second order and third order TRMC methods, the hybrid TRMC method and the recursive TRMC method. Next we will compare the behavior of the schemes for Hard Spheres.

In our tests we use the family of schemes characterized by (52) and perform a single run, with a number of particles sufficiently large to control the effects of the fluctuations. We express the results as a function of the scaled time variable t/ϵ which we denote again by t.

5.1. **Kac equation**

First we compare the Monte Carlo solutions with the exact solution of the Kac equation [20]

$$f(v,t)=\frac{1}{2}\left[\frac{3}{2}(1-C(t))\sqrt{C(t)}+(3C(t)-1)C(t)^{3/2}v^2\right]e^{-C(t)v^2},$$

where

$$C(t)=\left[3-2e^{-\sqrt{\pi}t/16}\right]^{-1}.$$

We reconstruct the density function on a grid, by convolving the particle distribution by a suitable mollifier [12],

$$f(V_I)=\frac{1}{N}\sum_{j=1}^{N}W_H(V_I-v_j),$$

where

$$\{V_I=V_{\min}+I\Delta V, I=1,\dots,N_g\}.$$

The smoothing function W_{H} is given by

$$W_H(x)=\frac{1}{H}W\left(\frac{x}{H}\right),\quad W(x)=\begin{cases} 3/4-x^2 & \text{if } |x|\le 0.5,\\ (x-3/2)^2/2 & \text{if } 0.5<|x|\le 1.5,\\ 0 & \text{otherwise.}\end{cases}$$

The value $H=0.2$ for $\Delta V=0.0625$ has been selected as a good compromise between fluctuations and resolution. The simulations are performed for $t\in[0,8]$, starting with $N=5\times 10^4$ particles.

In Fig. 3 we present the numerical results obtained at time $t=2$ for the different Monte Carlo schemes based on a time discretized equation with the same time step $\Delta t=1.0$ which represents the upper bound for stability of NB. It is evident that all TRMC schemes give a better representation of the solution especially near the local extrema.

In Fig. 4 (left) we report the L^2-norm of the error obtained in the same test with NB, TRMC1, TRMC2 and TRMCH using the same time step $\Delta t=0.5$. It is evident that all the TRMC methods provide a more accurate representation of the solution. In particular, by comparing the solution obtained with NB and TRMC1 one can see the improvement given by the first order TR scheme with respect to the explicit Euler time discretization. It is remarkable that the TRMCH method, as a consequence of its deterministic evolution of the equilibrium part, yields the most accurate solution.

Fig. 4 (right) shows the L^2-norm of the error for the same test where the time step of the TRMC schemes are respectively two times (TRMC1), three times (TRMC2) and four times (TRMCH) larger then the one used by NB. Note that in spite of the different time steps the accuracy of the different schemes is similar.

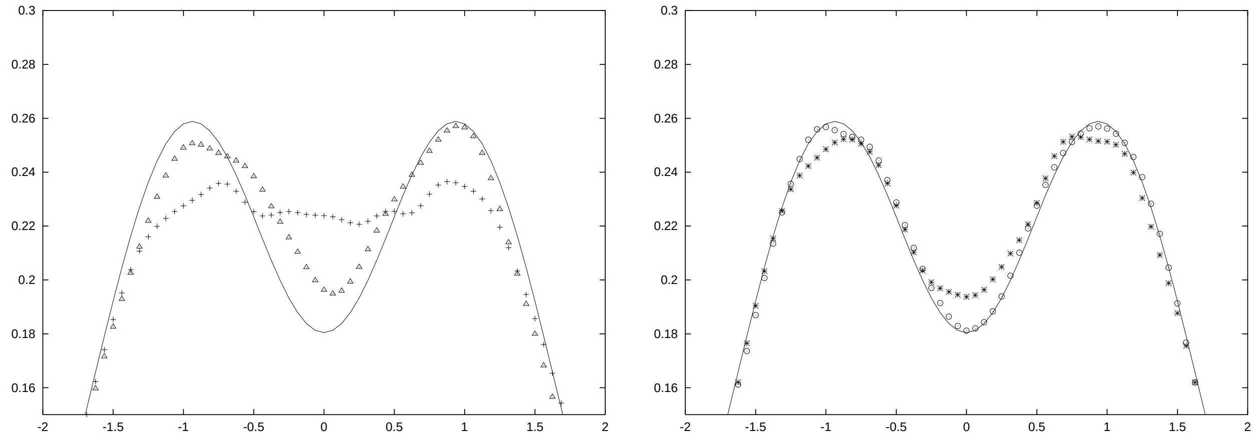

Figure 3. (Kac equation). Details of the distribution function at time $t = 2.0$ for $\Delta t = 1.0$. Exact (line), NB (+), TRMC1 (△), TRMC2 (∗), TRMC3 (∘).

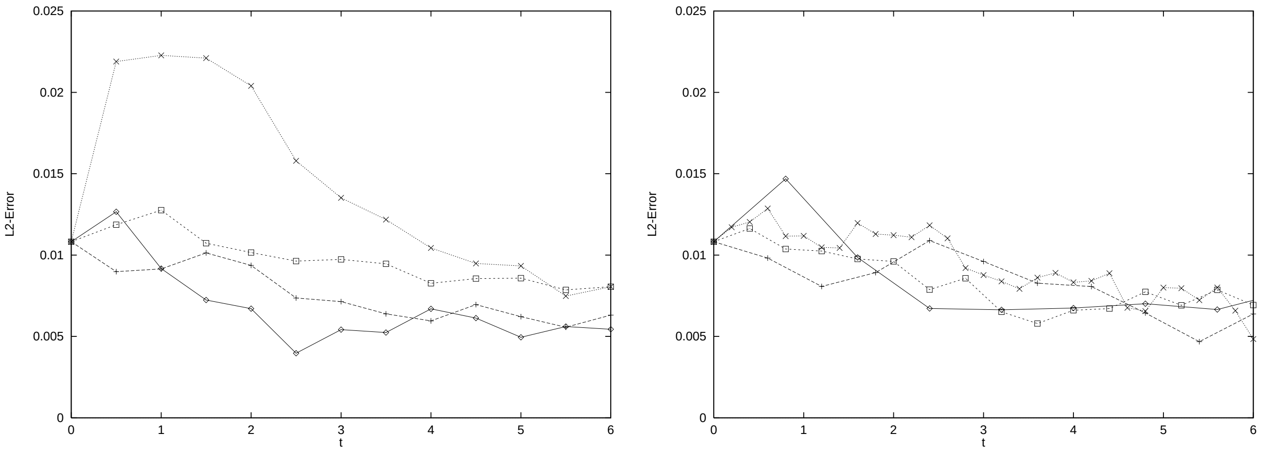

Figure 4. (Kac equation). L^2 error vs time. Left: $\Delta t = 0.5$ for all Monte Carlo schemes. Right: $\Delta t = 0.2$ for NB (×), $\Delta t = 0.4$ for TRMC1 (□), $\Delta t = 0.6$ for TRMC2 (+) and $\Delta t = 0.8$ for TRMCH (⋄).

Next we compare Bird's method with the recursive TRMCR algorithm. We recall that both methods do not contain time discretization errors and involves multiple collisions thus higher accuracy is expected with respect to the previous schemes.

Table 1 shows the behavior of the relative L_2-error for $t' = 2t/\sqrt{\pi}$ ranging from 0 to 15 where the Maxwellian equilibrium state is reached and different values of m_{max} for TRMCR have been used. It is remarkable how uniform accuracy in time is essentially obtained for $m_{\max} = 1000$ (for $t' = 10$ the maximum value is $m = 13819$), but even the values 100 and 25 give reasonable approximations with a slight deterioration of the accuracy at intermediate times. All the three different truncations provide the same result at $t' = 15$ since all particles are sampled from a Maxwellian. Note that in this case the maximum depth of the trees is simply 0 since the first term $N_1 < 1$ and step 3 of algorithm 4.10 stops at $n = 1$.

In Table 2 the corresponding number of collisions is reported. Sampling two particles from the Maxwellian is counted as one collision. The better efficiency of TRMCR with respect to Bird's method is clear (for $t' \geq 7$ the number of collisions in TRMCR methods is much smaller than in Bird's method).

TABLE 1. (Kac equation). Relative L_2 error norm in time for TRMCR with $m_{\max} = 1000$, 100, 25 and Bird's method. We have set $t' = 2t/\sqrt{\pi}$.

time	$m_{\max} = 1000$	$m_{\max} = 100$	$m_{\max} = 25$	Bird's method
$t' = 0$	0.010390	0.010390	0.010390	0.010390
$t' = 1$	0.006633	0.006633	0.006633	0.008301
$t' = 2$	0.005153	0.005153	0.007053	0.010728
$t' = 3$	0.005485	0.005185	0.007532	0.007074
$t' = 5$	0.005933	0.005790	0.021239	0.005914
$t' = 7$	0.005298	0.017166	0.019607	0.007987
$t' = 10$	0.006369	0.010085	0.010356	0.006796
$t' = 15$	0.006202	0.006202	0.006202	0.006293

TABLE 2. (Kac equation). Number of collisions in time for TRMCR with $m_{\max} = 1000$, 100, 25 and Bird's method. We have set $t' = 2t/\sqrt{\pi}$.

time	$m_{\max} = 1000$	$m_{\max} = 100$	$m_{\max} = 25$	m	Bird's method
$t' = 1$	22166	22166	22166	18	22156
$t' = 2$	44224	44224	43607	48	44312
$t' = 3$	66365	66389	53455	112	66468
$t' = 5$	110146	75053	36274	533	110779
$t' = 7$	130895	39305	27218	2280	155090
$t' = 10$	41347	25958	25139	13819	221557
$t' = 15$	25000	25000	25000	0	332336

5.2. Maxwell molecules

Next we consider the numerical solution of the two-dimensional homogeneous Boltzmann equation for Maxwell molecules.

An exact solution for this equation is given by

$$f(v,t) = \frac{1}{2\pi C(t)}\left[1 - \frac{1}{C(t)}(1 - C(t))\left(1 - \frac{v^2}{2C(t)}\right)\right]\exp(-\frac{v^2}{2C(t)}),$$

where $C(t) = 1 - (1/2)e^{-t/8}$.

The density distribution is obtained by reconstructing the function on a regular grid of spacing $\Delta v = 0.25$ by the "weighted area rule" [19] and the simulations have been performed for $t \in [0, 16]$ by starting with $N = 10^5$ particles.

In Fig. 5 we show the L^2 norm of the error in time for both NB and TRMC schemes. In the first picture we report the results obtained with the same time step $\Delta t = 1.0$. The results confirm the gain of accuracy of the TRMC methods on the transient time scale. Next, the results obtained with different time steps, chosen in such a way that the different errors are roughly the same, are also reported.

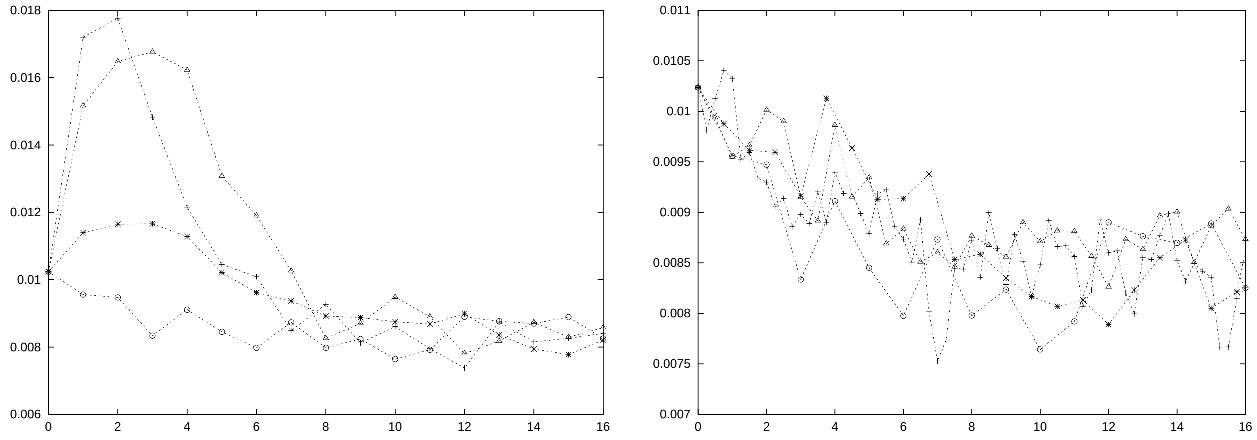

FIGURE 5. (Maxwell molecules). L^2 norm of the error vs time. NB (+), TRMC1 ($\triangle$), TRMC2 ($*$), TRMC3 ($\circ$). Left: time step $\Delta t = 1.0$. Right: NB with $\Delta t = 0.25$, TRMC1 with $\Delta t = 0.5$, TRMC2 with $\Delta t = 0.75$, TRMC3 with $\Delta t = 1.0$.

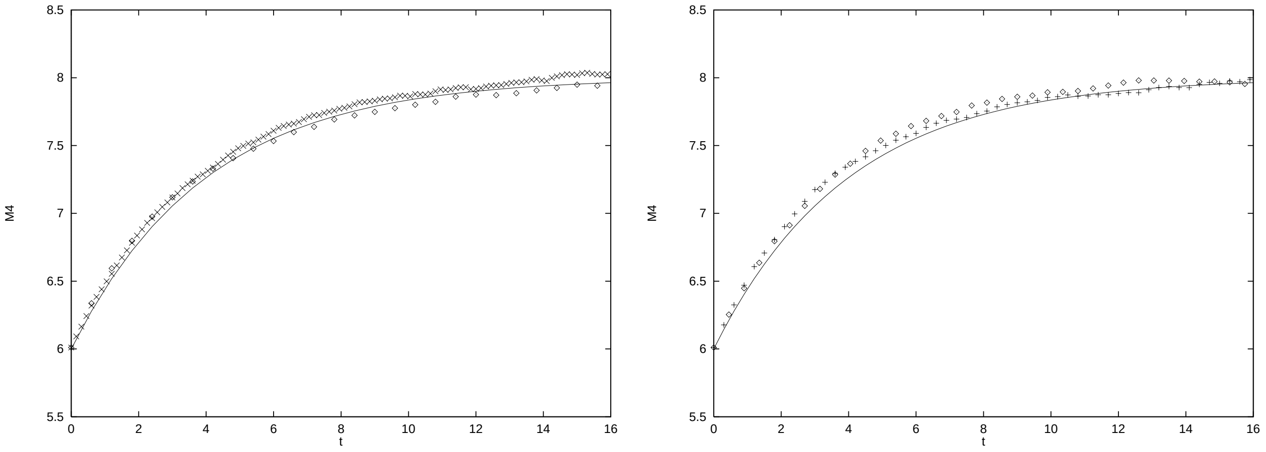

FIGURE 6. (Maxwell molecules). Fourth order moment vs time. Left: $\Delta t = 0.15$ for NB ($\times$) and $\Delta t = 0.6$ for TRMCH ($\diamond$). Right: $\Delta t = 0.3$ for TRMC1 ($\times$) and $\Delta t = 0.45$ for TRMC2 ($\diamond$). The line represents the exact solution.

In Fig. 6 we show the time evolution of the fourth order moment of the solution. The results confirm the gain of accuracy and the reduction of fluctuations of TRMC methods with respect to the NB method for larger time steps.

The comparison of Bird's and TRMCR is reported in Table 3 and 4. The results confirm the superior efficiency of TRMCR with respect to Bird's method (for $t = 10$ we have a factor ten reduction in the computational cost). Note that in this case uniform accuracy in time is obtained for $m_{max} = 100$ whereas $m = 25$ gives an acceptable deterioration of accuracy for intermediate times. In this two-dimensional computations, in fact, fluctuations play a more important rule since we have only twice the number of particles of Kac's case. Again for $t = 15$ all particles are sampled from a Maxwellian.

TABLE 3. (Maxwell molecules). Relative L_2 error norm in time for TRMCR with $m_{\max} = 1000$, 100, 25 and Bird's method

time	$m_{\max} = 1000$	$m_{\max} = 100$	$m_{\max} = 25$	Bird's method
$t = 0$	0.009972	0.009972	0.009972	0.009972
$t = 1$	0.009461	0.009461	0.009461	0.009378
$t = 2$	0.008263	0.008263	0.009039	0.009147
$t = 3$	0.008162	0.009050	0.010919	0.008327
$t = 5$	0.009261	0.010928	0.016342	0.009281
$t = 7$	0.008201	0.011544	0.012152	0.008460
$t = 10$	0.009337	0.008487	0.008978	0.010218
$t = 15$	0.008689	0.008689	0.008689	0.009847

TABLE 4. (Maxwell molecules). Number of collisions in time for TRMCR with $m_{\max} = 1000$, 100, 25 and Bird's method

time	$m_{\max} = 1000$	$m_{\max} = 100$	$m_{\max} = 25$	m	Bird's method
$t = 1$	49975	49975	49975	22	50000
$t = 2$	99879	99879	95880	65	100000
$t = 3$	149907	148497	103956	166	150000
$t = 5$	249668	124034	63378	963	250000
$t = 7$	206706	63784	52037	4946	350000
$t = 10$	61294	50679	50093	33323	500000
$t = 15$	50000	50000	50000	0	750000

5.3. **Hard Sphere molecules**

The last test problem deals with the numerical solution of the Boltzmann equation for Hard Sphere molecules (VHS, for $\alpha = 1$) with $C_\alpha = 1$.

The initial condition is the same used for the Maxwell molecules. The "exact" solution has been computed using NB method with 2×10^6 particles and $\Delta t = 5 \times 10^{-3}$.

As in the previous case, the density distribution is obtained by reconstructing the function on a regular grid of spacing $\Delta v = 0.25$ by the "weighted area rule" and the simulations have been performed for $t \in [0, 16]$ by starting with $N = 10^5$ particles.

In Fig. 7 we show the time evolution of the fourth order moment of the solution. The results confirm the gain of accuracy and the reduction of fluctuations of the TRMCH method with respect to NB method for larger time steps.

Next we report the number of dummy collisions and the number of effective collisions per time step performed by NB and TRMCH (Fig. 8).

Remark 5.1. We conclude this section with some additional considerations on the efficiency of the previous Monte Carlo methods.

First, in spite of the fact that the time step for TRMC1, TRMC2, TRMC3, TRMCH methods can be chosen larger than that of NB, the number of dummy collision is higher for NB. The reason is that this number is proportional to $\mu\Delta t$ for NB, and it is proportional to $1 - \exp(-\mu\Delta t)$ for the other methods.

Second, we emphasize that Bird, TRMC2, TRMC3, and TRMCR take into account multiple collisions of particles during the same time step. Thus, although the direct combination of the previous schemes with a solver for the transport step gives a scheme which is still first order, we expect that the overall accuracy of the solution will be improved also in space non-homogeneous problems (see Section 6).

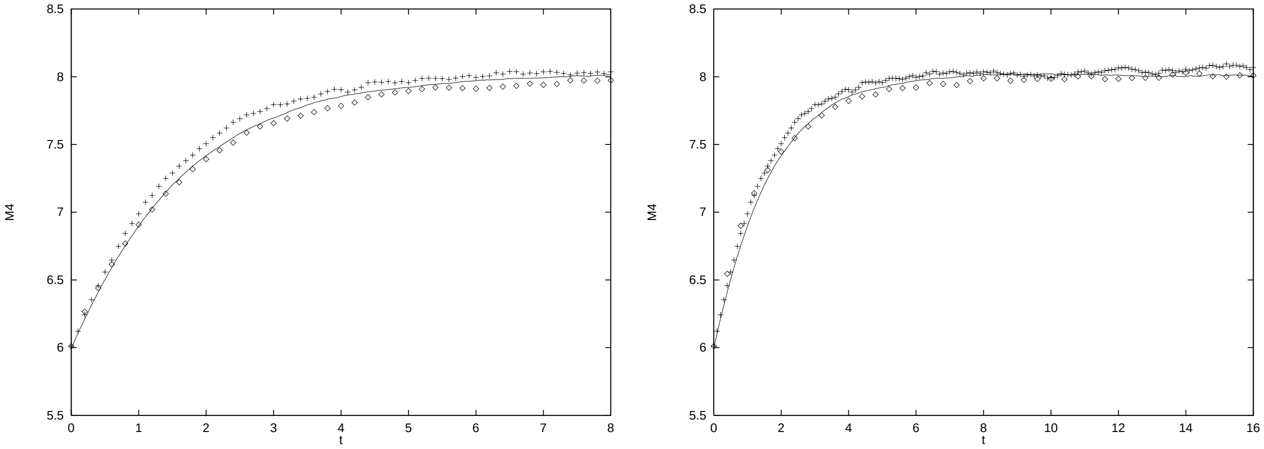

FIGURE 7. (Hard Sphere molecules). Fourth order moment vs time. NB (+), TRMCH (⋄), and "exact" (line) solution. Left: $\Delta t = 0.1$ for NB and $\Delta t = 0.2$ for TRMCH. Right: $\Delta t = 0.1$ for NB and $\Delta t = 0.4$ for TRMCH.

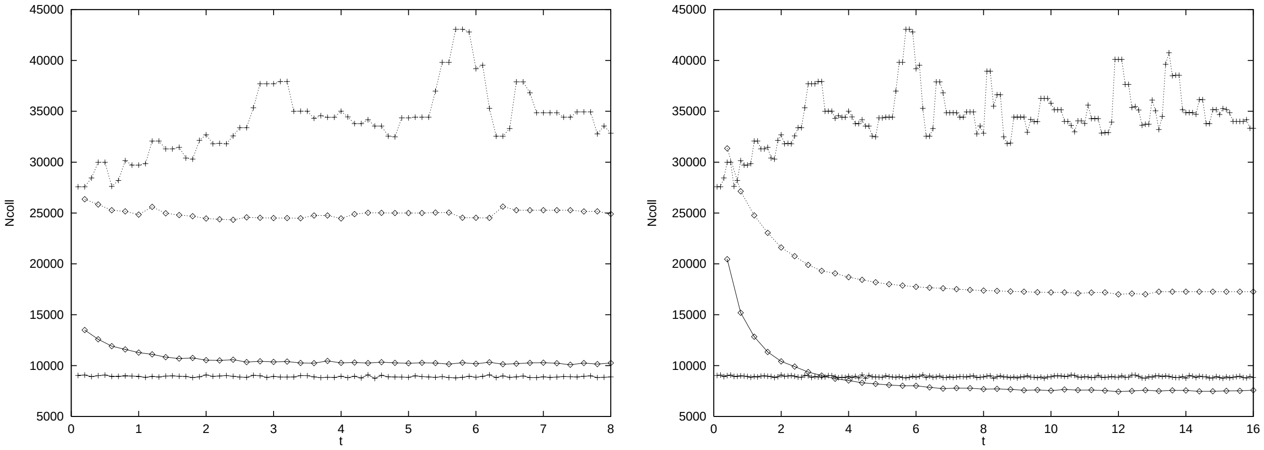

FIGURE 8. (Hard Sphere molecules). Number of effective (line) and dummy (dashed line) collisions vs. time. NB (+), TRMCH (⋄). Left: $\Delta t = 0.1$ for NB and $\Delta t = 0.2$ for TRMCH. Right: $\Delta t = 0.1$ for NB and $\Delta t = 0.4$ for TRMCH.

For space non-homogeneous situations where uniform accuracy in time is a highly desirable property the most promising method seems to be the recursive TRMC. In particular the depth of the truncated trees may be adapted to the space cells accordingly to the problem under study.

Finally, it is remarkable that in the homogeneous test problems we have presented here, the accuracy of the virtually first order TRMCH is higher than the accuracy of the second order TRMC2. In fact, the effect of the reduction of fluctuation given by the deterministic evolution of the Maxwellian part is stronger than the gain in time accuracy given by the multiple collisions of particles.

6. Space non-homogeneous case

In this section we consider the full space non-homogeneous Boltzmann equation. We shall illustrate the method with some simple application in one space dimension.

The splitting approach with particle method is particularly simple. The region Ω is divided into a certain number N_c of cells, and for each cell J one counts the number of particles per cell, N_J. Then during the collision step, the space homogeneous Boltzmann equation is solved by Monte Carlo in each cell, assuming that the density is constant in each cell. Each particle may collide only with particles within the same cell.

The solution of the collision step in the time interval t_n, t_{n+1} can be performed by one of the techniques illustrated in the previous sections (Bird scheme, Nanbu-Babovsky scheme, first or second order TRMC, and so on). In the space homogeneous case we assumed that the density function was a probability density, and therefore its in integral over velocity was one. In the space non-homogeneous case, the integral in velocity gives the number density of the the gas, i.e.

$$\int_{R^3} f(x,v,t)\,\mathrm{d}v = \frac{\rho}{m},$$

where m is the mass of the atom of the gas. The treatment of the previous section can be repeated, with the modification that the parameter μ in Eq. (25) is in this case given by

$$\mu = 4\pi\Sigma\frac{\rho}{m}$$

The density ρ in each cell, in turn, is proportional to the number of particles in the cell. More precisely, the density is given by

$$\rho_J = N_J m^*$$

where m^* is the "mass" of the simulation particles, which in general is much larger than the effective mass of a particle. The reason of this is that one wants to simulate physical systems with a large number of particles, by a much smaller number of simulation particles. As a result, a simulation particle usually represents several physical particles. Detailed discussions about this can be found, for example, in [4].

During the collision step, mass, momentum and energy in each cell remain constant.

At time $t_n = n\Delta t$, each particle is characterized by a position, x_i^n, and a velocity, v_i^n. Before the collision step, one has to determine the particle belonging to each cell. From the point of view of the implementation, one possibility to achieve this is to perform a sweep on all the particles and to determine in which cell each particle lies. This can be obtained, for example, by creating two vectors, $\mathtt{kp}[N]$ and $\mathtt{cp}[N_c]$, where N is the total number of particles, with the following meaning: the particles in cell J are all the particles with index $\mathtt{kp}[k]$, with $\mathtt{cp}[J-1] < k \leq \mathtt{cp}[J]$. Such ordering of the particles can be constructed very quickly in $O(N)$ operations.

After the collision step, the velocity of the particles is updated, and each particle has been assigned a new velocity, let us call it $\tilde{v}_i$, $i = 1, \ldots, N$.

The convection step is performed very easily, with a single loop over all the particles, namely

$$\tilde{x}_i = x_i^n + \tilde{v}_i\Delta t, \quad i = 1, \ldots, N$$

If there are no boundaries, then the new positions and velocities of the particles at the new time t_{n+1} will simply be

$$x_i^{n+1} = \tilde{x}_i,\; v_i^{n+1} = \tilde{v}_i$$

The presence of boundaries will be considered in the next section.

As we shall see, the time step Δt is mainly determined by the requirement that the particles do not travel more than a few cells in one time step. This requirement is related to a stability requirement of upwind schemes for the numerical solution of the single scalar equation.

The time step used in the numerical solution of the space homogeneous Boltzmann equation, on the other hand, is determined by different requirements. For example, in the case of NB scheme, one has to require positivity of the probability, and therefore the collisional time step must satisfy the restriction

$$\frac{\rho}{m}\Sigma\Delta t_{\text{coll}} \leq 1 \tag{64}$$

This condition can be guaranteed by dividing the convection time step, Δt, into an integer number of smaller collision time steps, Δt_{coll}, such that condition (64) is satisfied.

TR schemes do not have such restriction, and therefore can generate more efficient codes. An application to some space non-homogeneous problems will be shown later.

6.1. Implementation of boundary conditions

In this section we shall discuss how to deal with boundaries. Let us suppose, for simplicity, that our problem is one dimensional in space. Let $\Omega = [0, L]$ be the computational domain in space. We shall distinguish between different kinds of boundary conditions. Boundary conditions are applied during the convection step, after the candidate positions $\tilde{x}_i$ have been determined.

6.1.1. *Periodic BC*

In the simple case of periodic boundary conditions, one has to check whether the candidate position is still in Ω, and if not, add or subtract the period. Here is a schematic description of the process

Algorithm 6.1.

```
for $i = 1$ to $N$
    if $0 < \tilde{x}_i < L$
        set $x_i^{n+1} = \tilde{x}_i, \quad v_i^{n+1} = \tilde{v}_i,$
    if $\tilde{x}_i < 0$
        set $x_i^{n+1} = \tilde{x}_i + L, \quad v_i^{n+1} = \tilde{v}_i,$
    if $\tilde{x}_i > L$
        set $x_i^{n+1} = \tilde{x}_i - L, \quad v_i^{n+1} = \tilde{v}_i.$
end for
```

6.1.2. *Reflecting boundary conditions*

In this case the particle is simply reflected by the wall if the candidate space coordinate is out of the domain. This case corresponds to a reflection scattering kernel given by $K(v^* \to v) = \delta(v^* - v + 2n(n \cdot v))$. Algorithmically, this can be realized as follows

Algorithm 6.2.

```
for $i = 1$ to $N$
    if $0 < \tilde{x}_i < L$
        set $x_i^{n+1} = \tilde{x}_i, \quad v_i^{n+1} = \tilde{v}_i,$
    if $\tilde{x}_i < 0$ or $\tilde{x}_i > L$
        set $x_i^{n+1} = b - |\tilde{x}_i - b| \operatorname{sgn}(\tilde{v}_{ix}), \quad v_{ix}^{n+1} = -\tilde{v}_{ix}, \quad v_{iy}^{n+1} = \tilde{v}_{iy}, \quad v_{iz}^{n+1} = \tilde{v}_{iz},$
end for
```

where b denotes either 0 or L. The generalization of the above condition to more dimensions is straightforward.

6.1.3. *Maxwell's boundary conditions*

In this case the particle is either specularly reflected, or absorbed by the boundary, and re-emitted with a Maxwellian distribution corresponding to the thermal equilibrium with the wall. The reflection scattering kernel is given by

$$K(v^* \to v) = \alpha M_w(v)|v\cdot n| + (1-\alpha)\delta(v^* - v - 2n(n\cdot v^*)), \qquad (v^*\cdot n < 0, \quad v\cdot n > 0).$$

Here α is the so called *accommodation coefficient*, and $M_w(v)$ is the Maxwellian distribution corresponding to a gas in thermodynamic equilibrium with the wall. Algorithmically, the boundary condition is implemented in this way. Suppose, for example, that $\tilde{x}_i < 0$. Then, with probability $1-\alpha$ the particle will undergo a specular reflection. With probability α the particle is sampled from the half Maxwellian $M_w(v)$, $v_x > 0$. In this case the new positions and velocities are computed according to

Algorithm 6.3.

```
1. compute $\Delta\tilde{t} = |\tilde{x_i}/\tilde{v}_x|$,
2. sample $\tilde{v}$ from the Maxwellian $M_w(v)$, and assign $v_{ix}^{n+1} = |\tilde{v}_{ix}|$, $v_{iy}^{n+1} = \tilde{v}_{iy}$, $v_{iz}^{n+1} = \tilde{v}_{iz}$,
3. compute $x_i^{n+1} = v_{ix}^{n+1}\Delta\tilde{t}$.
```

An analogue procedure can be used at the wall located at $x = L$.

6.1.4. *Inflow boundary conditions*

These conditions are encountered, for example, when computing the structure of a stationary shock. They are imposed on the left and right boundary, by filling the computational domain with particles sampled from a distribution proportional to the flux $|v_x|M(\rho,u,T)(v)$, where $M(\rho,u,T)$ denotes the Maxwellian distribution with density ρ, mean velocity u and temperature T. Density, velocity and temperature are those assigned at the left and right boundary, and are related by the Rankine-Hugoniot relations [46]

$$\begin{aligned} \rho_L u_L &= \rho_R u_R, \\ \rho_L u_L^2 + p_L &= \rho_R u_R^2 + p_R, \\ u_L(E_L + p_L) &= u_R(E_R + p_R). \end{aligned}$$

The expected number of incoming particles is obtained by integrating the flux on the boundary. On the left boundary, for example, one has

$$< N_L^{\rm new} >= S\Delta x \Delta t \int_0^{+\infty} v_x M(n_L, u_L, T_L)(v)\,dv, \qquad (65)$$

where S is the cross sectional area of the tube, $n_L = \rho_L/m$ is the number density on the left of the computational domain, Δx is the cell size. These parameters are related to the average number of particles per cell on the left boundary by the relation

$$< N_L >= S\,\Delta x\, n_L.$$

Similarly, new particles are created at the right boundary.

The position of the new particles is determined as follows. Suppose a new particle with velocity v^* is generated during the time step, at the boundary. Since the particle is assumed to be generated at a random instant, uniformly distributed in the time interval $[t, t+\Delta t]$, then its position at time $t+\Delta t$ is given by

$$x^* = b + v^*\Delta t\,\xi,$$

where b is either 0 (left boundary) or L (right boundary), and ξ is a uniformly distributed random number in $[0,1]$.

If one creates new particles according to Eq. (65) and the corresponding formula for the particles coming from the right, then the total number of particles in the domain is not constant, but fluctuates around the mean. This effect results in a fluctuation in the shock position. If time average is used to obtains a smoother profile, then the shock thickness will appear much larger than the exact one. In order to avoid this effect, one can impose that the number of particles is constant. This can be done by counting the number of particles $N^{\rm lost}$ which leave the domain in a time step Δt, and replacing them by the same number of new particles. This is obtained, for example, by computing $N_L^{\rm new}$ and $N_R^{\rm new}$ according to the formulas

$$\begin{aligned} N_L^{\rm new} &= \text{Iround}\left(\frac{< N_L^{\rm new} >}{< N_L^{\rm new} > + < N_R^{\rm new} >} N^{\rm lost}\right), \\ N_R^{\rm new} &= N^{\rm lost} - N_L^{\rm new}. \end{aligned}$$

6.2. Shock wave profiles

The test problem deals with the numerical solution of the space non-homogeneous Boltzmann equation for hard sphere molecules (VHS, for $\alpha = 1$) with $C_\alpha = 1$. We present some numerical results for one-dimensional stationary shock profiles. In particular we have computed the structure of the shock for different Knudsen numbers, from the rarefied regime up to the fluid limit. In all our numerical tests the gas is initially at the upstream equilibrium state in the left half-space and in the downstream equilibrium state in the right-half space. The upstream state is determined from the downstream state using the Rankine-Hugoniot relations.

In the present calculations, the downstream state is characterized by

$$\rho = 1.0, \quad T = 1.0, \quad M = 3.0,$$

where M is the Mach number of the shock. The downstream mean velocity is then given by

$$u_x = -M\sqrt{\gamma T}, \quad u_y = 0,$$

with $\gamma = 2$ since we have considered a 2D monatomic gas in velocity space.

The infinite physical space is truncated to the finite region $[-7.5, 7.5]$. The technique previously illustrated has been used to keep the number of particles constant during the time evolution. We report the result obtained with the different schemes using 50 space cells and 500 particles in each downstream cell. Since we are computing a stationary solution after a fixed initial time, we can strongly improve the accuracy of the Monte Carlo solution by averaging in time the solution itself. To this aim we started accumulating statistics at time $t = 5$ and we average up to $t = 20$ which corresponds approximatively to 1500 time steps. The reference solution in the rarefied and intermediate regime is obtained using the Nanbu-Babovsky (NB) method with 200 space cells and 500 particles in each downstream cell and averaging over approximatively 8000 time steps. All the results are reported on the space interval $[-4.5, 4.5]$.

In Fig. 9 we plot the result obtained in the rarefied regime ($\epsilon = 1$) using the different Monte Carlo methods. The computational cost of the methods is comparable since here we are far from the fluid limit. As expected, the results show that essentially the TRMC methods and the NB method are equivalent and provide a good description of the rarefied shock.

Next we consider the intermediate regime ($\epsilon = 0.1$). Two different time steps have been used for NB, a convection time step Δt and a collision time step $\Delta t_{\rm coll} = \Delta t/10$. The collision time step for TRMC is equal to the convection time step. The result show that, despite the larger time step, the accuracy of the solution obtained with TRMC schemes is essentially the same of NB (see Fig. 10).

Finally we give the result of the computations close to the Euler limit ($\epsilon = 10^{-6}$). The profiles obtained with the TRMC methods are reported in Fig. 11. Due to the small Knudsen number the NB method is unusable in practice in this test. Note that the TRMC methods are all equivalent to Pullin's method. Thus we have plotted only one TRMC solution.

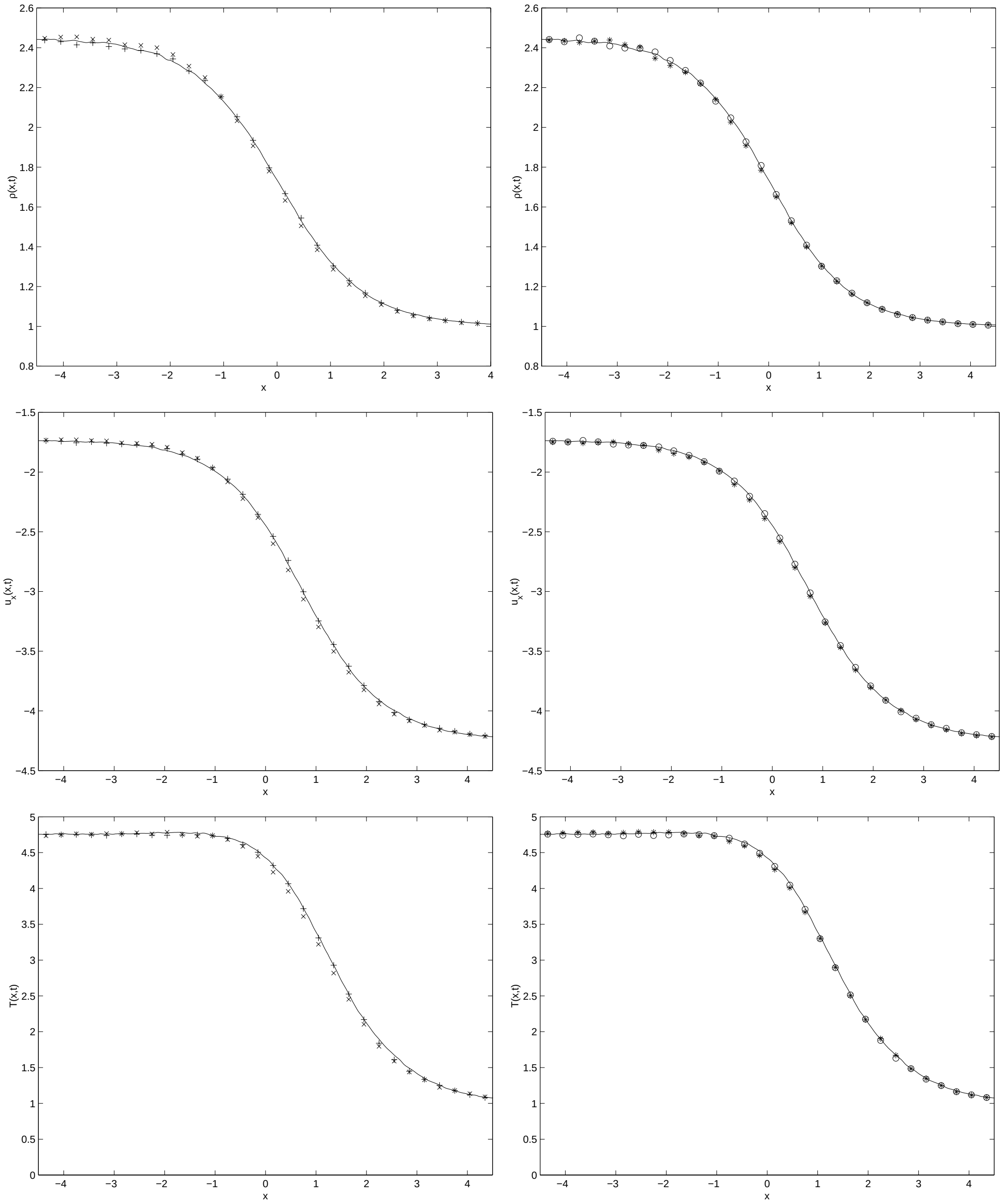

FIGURE 9. Shock wave profiles (rarefied regime): NB(+) and first order TRMC (×) (left column), second order ($*$) and third order ($\circ$) TRMC (right column) for $\epsilon = 1.0$ and $\Delta t = 0.025$. From top to bottom: ρ, u, T. The line is the reference solution.

REFERENCES

[1] H. BABOVSKY, *On a simulation scheme for the Boltzmann equation.* Math. Methods Appl. Sci., 8 (1986), pp. 223-233.

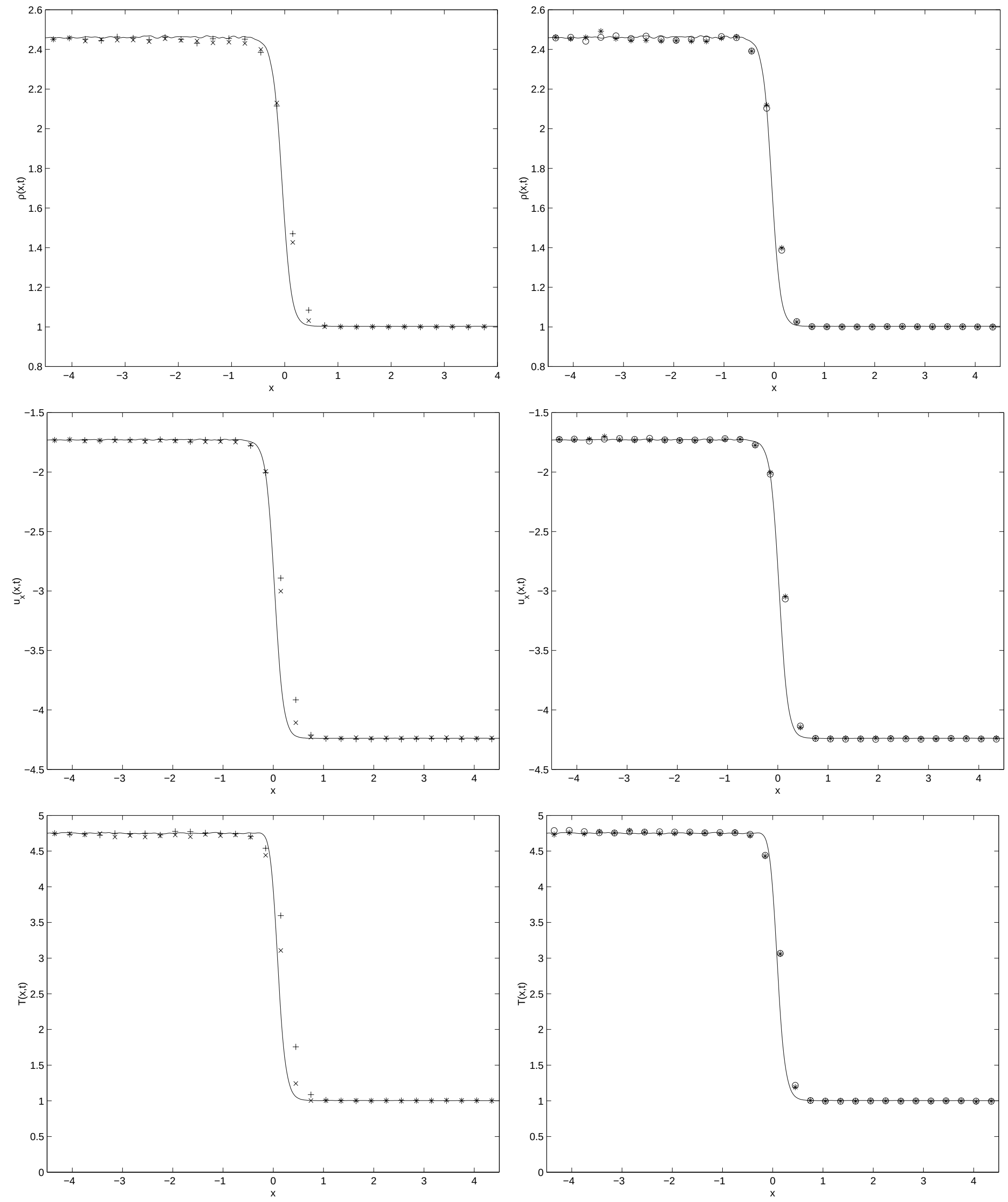

FIGURE 10. Shock wave profiles (intermediate regime): NB(+) and first order TRMC (×) (left column), second order ($*$) and third order ($\circ$) TRMC (right column) for $\epsilon = 1.0$ and $\Delta t = 0.0025$ for NB, $\Delta t = 0.025$ for TRMC. From top to bottom: ρ, u, T. The line is the reference solution.

[2] H. BABOVSKY, R. ILLNER, *A convergence proof for Nanbu's simulation method for the full Boltzmann equation.* SIAM J. Numer. Anal. 26 pp. 45–65, (1989).

[3] G.A. BIRD, *Direct simulation and the Boltzmann equation.* Physics of Fluids A, 13, 2672–2681, 1970.

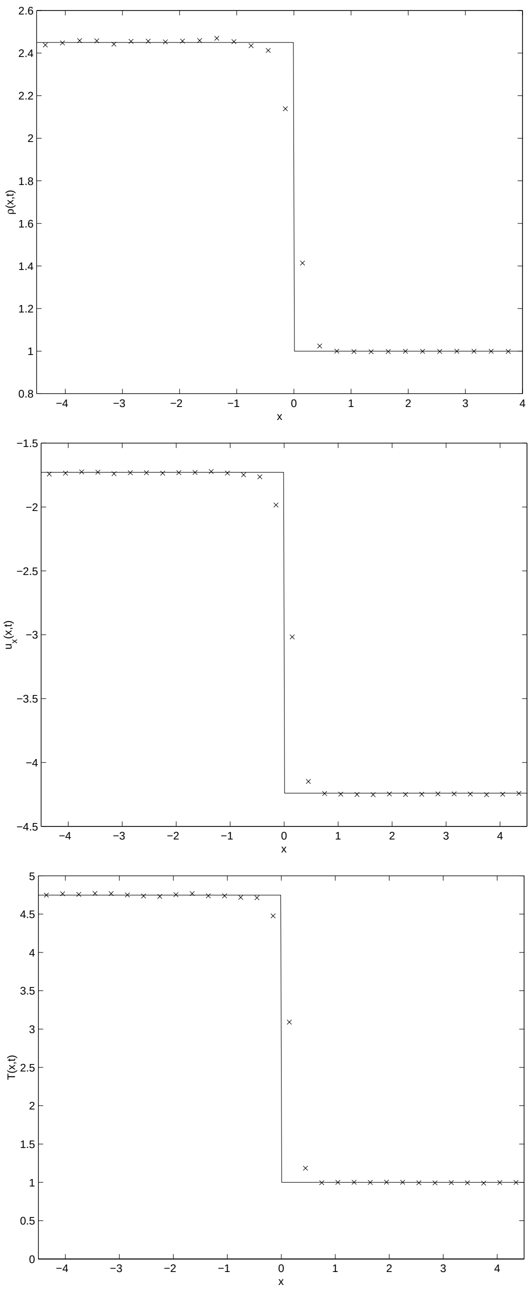

FIGURE 11. Shock wave profiles (fluid regime): first order TRMC (×) for $\epsilon = 10^{-8}$ and $\Delta t = 0.025$. From top to bottom: ρ, u, T.

[4] G.A. BIRD, *Molecular Gas Dynamics*. Oxford University Press, London, 1976.

[5] A.V. BOBYLEV, *Exact solutions of the nonlinear Boltzmann equation and of its models*. Translated from Molecular gas dynamics (Russian), 50–54, "Nauka", Moscow, 1982. Fluid Mech.—Soviet Res. 13 (1984), no. 4, 105–110 (1985).

[6] J.F. BOURGAT, P. LETALLEC, B. PERTHAME, Y. QIU, *Coupling Boltzmann and Euler equations without overlapping*. In "Domain Decomposition", AMS, 1992.

[7] R.E. Caflisch, *Monte Carlo and quasi-Monte Carlo methods.* Acta numerica, 1998, 1–49, Acta Numer., 7, Cambridge Univ. Press, Cambridge, 1998.

[8] R.E. Caflisch, L. Pareschi, *Towards an hybrid scheme for rarefied gas dynamics.* IMA volumes in Appl. Math. (to appear).

[9] C. Cercignani, *The Boltzmann equation and its applications.* Springer-Verlag, Berlin, (1988).

[10] C. Cercignani, R. Illner, M. Pulvirenti, *The mathematical theory of dilute gases.* Springer-Verlag, New York, (1995).

[11] F. Coron, B. Perthame *Numerical passage from kinetic to fluid equations.* SIAM J. Numer. Anal., 28, (1991), pp. 26-42.

[12] L. Desvillettes, R.E. Peralta Herrera, *A vectorizable simulation method for the Boltzmann equation.* Mathematical Modeling and Numerical Analysis, 28 (1994), pp. 745–760.

[13] S.M. Ermakov, V.V. Nekrutkin, A.S. Sipin, *Random processes for classical equations of mathematical physics.* Kluwer Academic Publishers, Dordrecht, 1989.

[14] M.H. Ernst, *Exact solutions of the nonlinear Boltzmann equation and related kinetic models.* Nonequilibrium Phenomena I. The Boltzmann equation, North-Holland, (1983), pp. 52-119.

[15] J.A. Fleck Jr., J.D. Cummings, *An implicit Monte Carlo scheme for calculating time and frequency dependent nonlinear radiation transport.* Journal of Computational Physics, 8, 313–342, 1971.

[16] E. Gabetta, L. Pareschi, G. Toscani, *Relaxation schemes for nonlinear kinetic equations.* SIAM J. Numer. Anal., 34 (1997), pp. 2168-2194.

[17] E. Hairer, G. Wanner, *Solving ordinary differential equations, Vol.2 Stiff and differential-algebraic problems.* Springer-Verlag, New York, 1987.

[18] D.B. Hash, H.A. Hassan *Assessment of schemes for coupling Monte Carlo and Navier-Stokes solution methods.* J. Thermophys. Heat Transf., 10, 242-249, 1996.

[19] R.W. Hockney, J.W. Eastwood, *Computer simulation using particles.* Mc-Graw Hill International Book Co., 1981.

[20] M. Kac, *Probability and related topics in physical sciences.* Lectures in Applied Mathematics, Interscience Publishers, London-New York, 1957.

[21] M.H. Kalos, P.A. Whitlock, *Monte Carlo Methods, Volume I: Basics.* John Wiley & Sons, New York, 1986.

[22] A.I. Khisamutdinov, L.L. Sidorenko, *Monte Carlo fictitious collision algorithms for nonlinear Boltzmann equation.* Monte Carlo Methods Appl. 1 (1995), no. 3, 221–240.

[23] C. Lcot, *A quasi-Monte Carlo method for the Boltzmann equation.* Math. Comp. 56 (1991), no. 194, 621–644.

[24] F. Mallinger, P. Le Tallec, *Coupling Boltzmann and Navier-Stokes Equations by Half Fluxes.* J. Comp. Phys., 136, (1997), pp. 51-67.

[25] H.P. McKean, *An exponential formula for solving Boltzmann's equation for a Maxwellian gas.* J. Combinatorial Theory, 2 (1967), pp. 358–382.

[26] W. Morokoff, R.E. Caflisch *Quasi-Monte Carlo Integration.* J. Comput. Phys., . 122 (1995), pp. 218-230.

[27] J.N. Moss, J.M. Price, *Direct simulation of AFE forebody and wake flow with thermal radiation.* In Rarefied Gas Dynamics: Theoretical and Computational Techniques, Progress in Aeronautics and Astronautics, 118, pages 413–431. Proceedings of the 16th International Symposium on Rarefied Gas Dynamics, 1988.

[28] E.P. Muntz, *Rarefied gas dynamics.* Annual Review of Fluid Mechanics, 21, 387–417, 1989.

[29] K. Nanbu *Direct simulation scheme derived from the Boltzmann equation. I. Monocomponent Gases.* J. Phys. Soc. Japan, 52 (1983), pp. 2042-2049.

[30] H. Neunzert, J. Struckmeier, *Particle methods for the Boltzmann equation.* Acta numerica, 1995, 417–457, Cambridge Univ. Press, Cambridge, 1995.

[31] T. Ohwada, *Higher order approximation methods for the Boltzmann equation.* Journal of Computational Physics, 139 (1998), pp. 1-14.

[32] L. Pareschi, R.E. Caflisch, *An implicit Monte Carlo method for rarefied gas dynamics. I. The space homogeneous case.* J. Comput. Phys., 154 (1999), pp. 90-116.

[33] L. Pareschi, G. Russo, *Asymptotic preserving Monte Carlo methods for the Boltzmann equation.* Transp. Theo. Stat. Phys., 29 (2000), pp. 415-430.

[34] L. Pareschi, G. Russo, *Time relaxed Monte Carlo methods for the Boltzmann equation*, SIAM J. Sci. Comp. submitted.

[35] L. Pareschi, B. Wennberg, *A recursive Monte Carlo method for the Boltzmann equation in the Maxwellian case.* Monte Carlo Methods and Applications (to appear).

[36] B. Perthame *Introduction to the theory of random particle methods for Boltzmann equation.* In Advances in Kinetic Theory and Computing, B. Perthame Editor, World Scientific, (1994).

[37] D.I. Pullin *Direct simulation methods for compressible inviscid ideal gas-flow.* J. Comp. Phys., 34, (1980), pp. 231-244.

[38] D.I. Pullin, *Generation of normal variates with given sample.* J. Statist. Comput. Simul., 9 (1979), pp. 303-309.

[39] A.K. Rebrov, *Studies on physical gas dynamics of jets as applied to vacuum pumps.* In V. Boffi and C. Cercignani, editors, Proceedings of the 15th International Symposium on Rarefied Gas Dynamics, pages 455–473, 1986.

[40] S. Rjasanow, W. Wagner, *A stochastic weighted particle method for the Boltzmann equation.* J. Comput. Phys. 124 (1996), no. 2, 243–253.

[41] S.V. Rogasinsky, *Solution of stationary boundary value problems for the Boltzmann equation by the Monte Carlo method.* Monte Carlo Methods Appl. 5 (1999), no. 3, 263–280.

[42] R. Roveda, D.B. Goldstein, P.L. Varghese *Hybrid Euler/Direct Simulation Monte Carlo calculation of unsteady sit flow. preprint*, 2000.

[43] G. Russo, R.E. Caflisch, *Implicit methods for kinetic equations.* In Rarefied Gas Dynamics: Theory and Simulations, Progress in Aeronautics and Astronautics, v. 159, pages 344–352. Proceedings of the 18th International Symposium on Rarefied Gas Dynamics, (1992).

[44] G. Strang, *On the construction and the comparison of difference schemes.* SIAM J. Numer. Anal., 5 (1968), pp. 506-517.

[45] W. Wagner, *A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation.* J. Statist. Phys., 66 (1992), 1011-1044.

[46] G.B. Whitham, *Linear and nonlinear waves.* Wiley-Interscience, (1974).

[47] E. Wild, *On Boltzmann's equation in the kinetic theory of gases.* Proc. Camb. Phil. Soc., 47 (1951), pp. 602-609.