Part III: Monte Carlo methods

Lorenzo Pareschi

Department of Mathematics
University of Ferrara, Italy
Outline

• Random sampling
  – Monovariate distributions
  – Multivariate distributions

• Direct simulation Monte Carlo methods
  – Nanbu’s method
  – Bird’s method

• Time Relaxed discretizations
  – Wild sum expansions
  – Time Relaxed schemes

• Time Relaxed Monte Carlo methods
  – First order
  – Second order

• Numerical results

• Basic references
Random Sampling

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.

*We assume that our computer is able to generate a uniformly distributed random number between 0 and 1.*

Monovariate distribution

Let $x \in \mathbb{R}$ 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 $\xi$ be a uniformly distributed random variable (number) in $[0, 1]$. 

![Monovariate distribution graph](image)
Then the relation between $x$ and $\xi$ can be found by equating the infinitesimal probabilities

$$p_x(x) \, dx = 1 \cdot d\xi.$$ 

By integration one has

$$P_x(x) = \int_{-\infty}^{x} p_x(y) \, dy = \xi,$$

where $P_x(x)$ 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 $\xi$, and then solving

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

**Example:** 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 uniformly distributed in $[0, 1]$.

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


It may be expensive to compute the inverse function, since in general a nonlinear equation has to be solved. A different technique is the so-called **acceptance-rejection**.

Let $x$ be a random variable with density $p_x(x)$, $x \in \mathbb{R}$. We look for a function $w(x) \geq p_x(x) \forall x \in \mathbb{R}$ whose primitive $W(x)$ is easily invertible. Let $A = \int_{-\infty}^{\infty} w(x) \, dx$ and denote with $\xi_1$ and $\xi_2$ uniformly $[0, 1]$ random numbers.

**Algorithm [acceptance-rejection]:**

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.**

In Figure $\eta$ has been sampled from $w(x)$. It will be accepted with probability equal to the ratio $p(\eta)/w(\eta)$.
Remark: 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 \( \frac{1}{A} \).

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

Algorithm:

1. select an integer \( i \in \{1, \ldots, M\} \) 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.
Discrete sampling

Let us suppose that \( k \in \{1,\ldots,M\} \) 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 \( \xi \), and detect the interval \( k \) to which \( \xi \) belongs in the following way.

Algorithm [discrete sampling 1]:

1. Compute \( W_k = \sum_{i=1}^{k} w_k, \quad k = 1,\ldots,M, \quad W_0 = 0; \)
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. If the number \( M \) is very large and the numbers \( w_k \) may change a more efficient acceptance-rejection technique can be used, if at least an estimate \( \bar{w} \) such that \( \bar{w} \geq w_i, \quad i = 1,\ldots,M \) is known.

Algorithm [discrete sampling 2]:

1. Select an integer random number uniformly in \([1,\ldots,M]\), as
   \[
   k = \lfloor M \xi_1 \rfloor + 1
   \]
   where \( \lfloor x \rfloor \) denotes the integer part of \( x \);
2. if \( w_k \xi_2 < \bar{w} \) 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. $\bar{w}_k \geq w_k$, $k = 1, \ldots, M$.

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

**Sampling without repetition**

Sometimes it is useful to extract $n$ numbers, $n \leq N$, from the sequence $1, \ldots, N$ 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:**

1. set $\text{ind}_i = i$, $i = 1, \ldots, N$;

2. $M = N$;

3. for $i = 1$ to $n$
   
   - set $j = \lceil M\xi \rceil + 1$, seq$_i = \text{ind}_j$,
   
   - $\text{ind}_j = \text{ind}_M$, $M = M - 1$

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 random permutation of the sequence $1, \ldots, N$. 
Multivariate distributions

Suppose we want to sample a \( n \)-dimensional random variable \( x = (x_1, \ldots, x_n) \), 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

\[
p_x(x_1, \ldots, x_n) = p_1(x_1)p_2(x_2) \cdots p_n(x_n),
\]

then the \( n \) scalar random variables \( x_1, \ldots, x_n \) are independent, and the problem is equivalent to sampling \( n \) monovariate random variables.

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

\[
p_x(x_1, \ldots, x_n)dx_1dx_2 \cdots dx_n = p_{\eta_1}(\eta_1)p_{\eta_2}(\eta_2) \cdots p_{\eta_n}(\eta_n)d\eta_1d\eta_2 \cdots d\eta_n,
\]

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

**Remark**: 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 \( T : x \rightarrow \xi \) where \( \xi = (\xi_1, \ldots, \xi_n) \) 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 \( p_x(x_1, \ldots, x_n) \).
An explicit transformation is constructed as follows

\[
T_1(x_1) = \int_{-\infty}^{x_1} d\eta \int_{\mathbb{R}^{n-1}} dx_2 \ldots dx_n p_x(\eta, \ldots, x_n),
\]

\[
T_2(x_1, x_2) = \frac{\int_{-\infty}^{x_2} d\eta \int_{\mathbb{R}^{n-1}} dx_3 \ldots dx_n p_x(x_1, \eta, \ldots, x_n)}{\int_{-\infty}^{\infty} d\eta \int_{\mathbb{R}^{n-1}} dx_3 \ldots dx_n p_x(x_1, \eta, \ldots, x_n)},
\]

\[
T_3(x_1, x_2, x_3) = \frac{\int_{-\infty}^{x_3} d\eta \int_{\mathbb{R}^{n-1}} dx_4 \ldots dx_n p_x(x_1, x_2, \eta, \ldots, x_n)}{\int_{-\infty}^{\infty} d\eta \int_{\mathbb{R}^{n-1}} dx_4 \ldots dx_n p_x(x_1, x_2, \eta, \ldots, x_n)},
\]

\[\vdots\]

\[
T_n(x_1, \ldots, x_n) = \frac{\int_{-\infty}^{x_n} d\eta p_x(x_1, \ldots, \eta)}{\int_{-\infty}^{\infty} d\eta p_x(x_1, \ldots, \eta)}.
\]

It is straightforward to check that \(|\partial \xi / \partial x| = p_x(x_1, \ldots, x_n)|. Furthermore, the computation of the inverse requires the solution of a triangular system: find \(x_1\) by solving the first equation of the system, 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 \(x_1, \ldots, x_n\).

**Remark:** This transformation 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 $\eta(i) \in [0, 1]^n$, $i = 1, \ldots, N$,

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

then

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

where $x(i) = T^{-1}(\eta(i))$, $i = 1, \ldots, N$. 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.

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 $p_x(x)$, $x \in \mathbb{R}^n$. Then we look for a function $w(x) \geq p_x(x) \forall x \in \mathbb{R}^n$ which is “easy to sample”. Let $A = \int_{\mathbb{R}^n} w(x) \, dx$. Then the algorithm works as follows

Algorithm:

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:** 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 \( P(x) = \frac{1 + \text{erf}(x/\sqrt{2})}{2} \), where

\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} \exp(-t^2) dt,
\]

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

\[
x = \rho \cos \theta, \quad y = \rho \sin \theta,
\]

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

\[
p_\rho = \exp\left(-\frac{\rho^2}{2}\right) \rho, \quad \rho \geq 0
\]

\[
p_\theta = \frac{1}{2\pi}, \quad 0 \leq \theta < 2\pi
\]

The random variables $\rho$ and $\theta$ are readily sampled by inverting $p_\rho$ and $p_\theta$, i.e.

\[
\rho = \sqrt{-2 \ln \xi_1}, \quad \theta = 2\pi \xi_2,
\]

and, from these $x$ and $y$ are easily obtained.

**Remark:** 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 $\mu$ and standard deviation $\sigma$, then $x$ and $y$ will be scaled accordingly as

\[
x = \mu_x + \sigma_x \rho \cos \theta, \quad y = \mu_y + \sigma_y \rho \sin \theta.
\]
Example: 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 \((\varphi, \theta)\),

\[
\begin{align*}
    x &= \sin \theta \cos \varphi, \\
    y &= \sin \theta \sin \varphi, \\
    z &= \cos \theta.
\end{align*}
\]

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

\[
\begin{align*}
    \frac{d\varphi}{2\pi} &= d\xi_1, \\
    \frac{\sin \theta \ d\theta}{2} &= d\xi_2.
\end{align*}
\]

Integrating the above expressions we have

\[
\begin{align*}
    \varphi &= 2\pi \xi_1, \\
    \theta &= \arccos(1 - 2\xi_2).
\end{align*}
\]
Direct simulation Monte Carlo (DSMC) methods

In this paragraph we will describe the classical DSMC methods due to Bird and Nanbu in the case of the spatially homogeneous Boltzmann equation.

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

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

where $\mu > 0$ 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 $P(f, f) = Q^+(f, f)$.

Examples:

The Kac equation is of the above form with

$$P(f, f) = \frac{1}{2\pi} \int_0^{2\pi} \int_{-\infty}^{+\infty} f(v') f(v_*) dv_* d\theta, \quad \mu = \rho = \int_{-\infty}^{\infty} f(v) dv.$$

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

$$Q^+(f, f)(v) = \int_{\mathbb{R}^3} \int_{S^2} b_0(\cos \theta) f(v') f(v'_*) d\omega dv_*.$$

The case of general kernels with cut-off will be discussed later.
Nanbu’s method (DSMC no time counter)

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 \([0, t_{\text{max}}]\), and let us discretize it in \( n_{\text{tot}} \) intervals of size \( \Delta t \). Let us denote by \( f^n(v) \) an approximation of \( f(v, n\Delta t) \). The forward Euler scheme writes
\[
f^{n+1} = \left(1 - \frac{\mu \Delta t}{\epsilon}\right)f^n + \frac{\mu \Delta t}{\epsilon} P(f^n, f^n)
\]

Clearly if \( f^n \) is a probability density both \( P(f^n, f^n)/\mu \) and \( f^{n+1} \) are probability densities. Thus the equation has the following probabilistic interpretation:

**physics**: 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 \( P(f^n, f^n)(v) \).

**statistics**: to sample \( v_i \) from \( f^{n+1} \) with probability \((1 - \mu \Delta t/\epsilon)\) we sample from \( f^n \), and with probability \( \mu \Delta t/\epsilon \) we sample from \( P(f^n, f^n)(v)/\mu \).

**Remark**: Note that \( \Delta t \leq \epsilon/\mu \) to have the probabilistic interpretation.
Maxwellian case

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.

Algorithm[Nanbu for Maxwell molecules]:

1. compute the initial velocity of the particles, \( \{v^0_i, i = 1, \ldots, N\} \), by sampling them from the initial density \( f_0(v) \)
2. for \( n = 1 \) to \( n_{\text{tot}} \)
   for \( i = 1 \) to \( N \)
   with probability \( 1 - \mu \Delta t/\epsilon \)
   \- set \( v^{n+1}_i = v^n_i \)
   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 \( v^{n+1}_i = v'_i \)
end for
end for

Remark: 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. 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 $\Delta t$ 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 [Nanbu-Babovsky for Maxwell molecules]:**

1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \ldots, N\}$, by sampling them from the initial density $f_0(v)$
2. for $n = 1$ to $n_{\text{tot}}$ given $\{v_i^n, i = 1, \ldots, N\}$
   - set $N_c = \text{Iround}(\mu N \Delta t/(2\epsilon))$
   - 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 $\text{Iround}(x)$ we denote a suitable integer rounding of a positive real number $x$. For example

$$\text{Iround}(x) = \begin{cases} \lfloor x \rfloor + 1 & \text{with probability } x - \lfloor x \rfloor \\ \lfloor x \rfloor & \text{with probability } \lfloor x \rfloor + 1 - x \end{cases}$$

where $\lfloor x \rfloor$ denotes the integer part of $x$.

**Remark:** 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.
Post-collisional velocities

When the above scheme is applied to the Kac equation, the new velocities $v'_i$ and $v'_j$ are computed as

$$v'_i = v_i \cos \theta - v_j \sin \theta, \quad v'_j = v_i \sin \theta + v_j \cos \theta,$$

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

For Maxwell molecules one has

$$v'_i = \frac{v_i + v_j}{2} + \frac{|v_i - v_j|}{2} \omega, \quad v'_j = \frac{v_i + v_j}{2} - \frac{|v_i - v_j|}{2} \omega,$$

where $\omega$ is chosen uniformly in the unit sphere, according to:

**2D**

$$\omega = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix}, \quad \theta = 2\pi \xi,$$

**3D**

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

where $\xi_1, \xi_2$ are uniformly distributed random variables in $[0, 1]$. 
Variable Hard Sphere case

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 $B$ with the kernel $B_{\Sigma}$

$$B_{\Sigma}(|v - v_*|) = \min \{ B(|v - v_*|), \Sigma \}, \quad \Sigma > 0.$$ 

Thus, for a fixed $\Sigma$, let us consider the homogeneous problem

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

The operator $Q_{\Sigma}(f, f)$ can be written in the form $P(f, f) - \mu f$ taking

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

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

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

**Remark:** In this case, a simple scheme is obtained by using the acceptance-rejection technique to sample the post collisional velocity according to $P(f, f)/\mu$, where $\mu = 4\pi \Sigma \rho$ and $\Sigma$ is an upper bound of the scattering cross section.
The conservative DSMC algorithm for VHS collision kernels can be written as

Algorithm[Nanbu-Babovsky for VHS molecules]:

1. compute the initial velocity of the particles, \( \{v_i^0, i = 1, \ldots, N\} \), by sampling them from the initial density \( f_0(v) \)
2. for \( n = 1 \) to \( n_{\text{tot}} \) given \( \{v_i^n, i = 1, \ldots, N\} \)
   - compute an upper bound \( \Sigma \) of the cross section
   - set \( N_c = \text{Iround}(N \rho \Sigma \Delta t/(2\epsilon)) \)
   - select \( N_c \) dummy collision pairs \((i, j)\) uniformly among all possible pairs, and for those
     - compute the relative cross section \( B_{ij} = B(|v_i - v_j|) \)
     - if \( \Sigma \text{ Rand} < B_{ij} \)
       - 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
Evaluation of $\Sigma$

The upper bound $\Sigma$ should be chosen as small as possible, to avoid inefficient rejection, and it should be computed fast. It is be too expensive to compute $\Sigma$ as

$$\Sigma = B_{\text{max}} \equiv \max_{i,j} B(|v_i - v_j|),$$

since this computation would require an $O(N^2)$ operations.

An upper bound of $B_{\text{max}}$ is obtained by taking $\Sigma = B(2\Delta v)$, where

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

Remarks:

- For general collision kernel, the algorithm is slightly modified by introducing the angular dependence.

- The probabilistic interpretation 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 method becomes unusable near the fluid regime.

- The cost of the method is proportional to the number of dummy collision pairs, that is $\mu N \Delta t/2$. Thus for a fixed final time $T$ the total cost is independent of the choice of $\Delta t = T/n$. However this is true only if we do not had to compute $\Sigma$ (like in the Maxwellian case). If we compute $\Sigma$ at every time step, then the cost of the computation increases when we decrease the time step, since the estimate of $\Sigma$ has a cost $O(N)$ and not $O(N\Delta t)$. 

22
Bird’s method (DSMC time counter)

The method is currently the most popular method for the numerical solution of the Boltzmann equation. It has been derived accordingly to physical considerations for the simulation of particle collisions.

Maxellian case

Let us consider first the Maxwellian case. The number of collisions in a short time step $\Delta t$ is given by

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

This means that the average time between collisions $\Delta t_c$ 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 [Bird for Maxwell molecules]:**

1. compute the initial velocity of the particles, $\{v_i^0, i = 1, \ldots, N\}$, by sampling them from the initial density $f_0(v)$
2. set time counter $t_c = 0$
3. set $\Delta t_c = 2\epsilon/(\mu N)$
4. for $n = 1$ to $n_{\text{tot}}$
   - 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 $t_c \geq (n + 1)\Delta t$
5. set $v_i^{n+1} = \tilde{v}_i, i = 1, \ldots, N$

**Remark:** 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 $\Delta t$, in fact, can be chosen to be the full time span $t_{\text{max}}$. 


Variable Hard Sphere case

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 $\Delta t$ is given by

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

where $B$ denotes the average collision frequency.

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

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

The $N_c$ collisions have to be performed with probability proportional to $B_{ij} = B(|v_i - v_j|)$. In order to do this one can use the same acceptance-rejection technique as in Nanbu-Babovsky scheme. The drawback of this procedure is that computing $B$ would be too expensive. A solution to this problem is to compute a local time counter $\Delta t_c$ as follows. First select a collision pair $(i, j)$ using rejection. Then compute

$$\Delta t_{ij} = \frac{2\varepsilon}{N \rho B_{ij}}.$$
This choice gives the correct expected value for the collision time

\[ \Delta t_c = \sum_{1 \leq i < j \leq N} \Delta t_{ij} \frac{B_{ij}}{\sum_{1 \leq i < j \leq N} B_{ij}} = \frac{2\varepsilon}{N \rho B}. \]

Bird’s algorithm for general VHS molecules can therefore be summarized as:

**Algorithm [Bird for VHS molecules]:**

1. compute the initial velocity of the particles, \( \{v_i^0, i = 1, \ldots, N\} \), by sampling them from the initial density \( f_0(v) \)
2. set time counter \( t_c = 0 \)
3. for \( n = 1 \) to \( n_{\text{tot}} \)
   - 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 \( B_{ij} = B(|v_i - v_j|) \)
     - if \( \Sigma \text{Rand} < B_{ij} \)
       - 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 \( \Delta t_{ij} = 2\varepsilon/\left( N \rho B_{ij} \right) \)
       - update the time counter \( t_c = t_c + \Delta t_{ij} \)
     until \( t_c \geq (n + 1)\Delta t \)
   - set \( v_i^{n+1} = \tilde{v}_i, i = 1, \ldots, N \)
end for
Remarks:

• At variance with NB scheme, there is no restriction on the time step $\Delta t$ for Bird's scheme. For space homogeneous calculations $\Delta t$ could be chosen to be the total computation time $t_{\text{max}}$. However, the scheme requires an estimate of $B_{\text{max}}$, 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 $\Delta t_{ij}$ becomes very small, and a huge number of collisions is needed in order to reach a fixed final time $t_{\text{max}}$. 

Time Relaxed (TR) discretizations

Wild sum expansion

Starting point for the construction of TR schemes.

Consider the differential system

\[
\frac{\partial f}{\partial t} = \frac{1}{\varepsilon} [P(f, f) - \mu f]
\]

with initial condition

\[f(v, 0) = f_0(v)\]

\(\mu > 0\) is a constant and \(P\) a bilinear operator.

It is possible to show that the function \(f\) satisfies the following formal expansion

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

The functions \(f_k\) are given by the recurrence formula for \(k = 0, 1, \ldots\)

\[
f_{k+1}(v) = \frac{1}{k+1} \sum_{h=0}^{k} \frac{1}{\mu} P(f_h, f_{k-h})
\]
Properties

i) Conservation: If the collision operator preserves some moments, then all the functions $f_k$ will have the same moments, i.e. if, for some function $\phi(v)$

$$\int_{\mathbb{R}^3} P(f,f)\phi(v) \, dv = \mu \int_{\mathbb{R}^3} f \phi(v) \, dv,$$

then the coefficients $f^{(k)}$ defined by (2) are nonnegative and satisfy $\forall k > 0$

$$\int_{\mathbb{R}^3} f^{(k)}\phi(v) \, dv = \int_{\mathbb{R}^3} f^{(0)}\phi(v) \, dv.$$

ii) Asymptotic behavior: If the sequence $\{f^{(k)}\}_{k \geq 0}$ defined by (2) is convergent, then (1) is well defined for any value of $\varepsilon$. Moreover, if we denote by $M(v) = \lim_{k \to \infty} f_k$ then

$$\lim_{t \to \infty} f(v,t) = M(v),$$

in which $M(v)$ is the local (Maxwellian) equilibrium.

Remark: If $\sigma_B(|v-v_1|,\omega) \leq B$ is a bounded scattering cross section then the BE can always be written in the above form, where $\mu = 4\pi \rho B$ is a positive constant s.t.

$$\mu \geq L_B[f](v) \equiv \int_{\mathbb{R}^3} \int_{S^2} \sigma_B(|v-v_1|,\omega)f(v_*) \, d\omega \, dv_*$$

and $P(f,f) = Q_B^+(f,f) + (\mu - L_B[f])f.$
TR schemes

From the previous representation, the following class of numerical schemes is obtained

\[
f^{n+1}(v) = (1 - \tau) \sum_{k=0}^{m} \tau^k f^n_k(v) + \tau^{m+1} M(v),
\]

where \( f^n = f(n\Delta t) \), \( \Delta t \) is a small time interval, and \( \tau = 1 - e^{-\mu \Delta t/\varepsilon} \) (relaxed time).

**Properties:**

i) *conservation*: If the initial condition \( f^0 \) is a non negative function, then \( f^{n+1} \) is nonnegative for any \( \mu \Delta t/\varepsilon \), and satisfies

\[
\int_{\mathbb{R}^3} f^{n+1} \phi(v) dv = \int_{\mathbb{R}^3} f^n \phi(v) dv.
\]

ii) *asymptotic preservation*: For any \( m \geq 1 \), we have \( \lim_{\mu \Delta t/\varepsilon \to \infty} f^{n+1} = M(v) \).

iii) *accuracy*: If \( \sup_{k>m} \{ |f^n_k - M| \} \leq C \) for some constant \( C = C(v) \) then

\[
|f(v,t) - f^{n+1}(v)| \leq C \tau^{m+1}.
\]
Generalized TR schemes

Generalized schemes can be derived, of the following form

\[ f^{n+1}(v) = \sum_{k=0}^{m} A_k f^n_k(v) + A_{m+1} M(v). \]

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

i) **conservation:**

\[ \sum_{k=0}^{m+1} A_k(\tau) = 1 \quad \tau \in [0, 1], \]

ii) **asymptotic preserving:**

\[ \lim_{\tau \to 1} A_k(\tau) = 0, \quad k = 0, \ldots, m \]

iii) **consistency:**

\[ \lim_{\tau \to 0} \frac{A_1(\tau)}{\tau} = 1, \quad \lim_{\tau \to 0} \frac{A_k(\tau)}{\tau} = 0, \quad k = 2, \ldots, m+1 \]
Time Relaxed Monte Carlo Methods

First order TR scheme (TRMC1):

Form $m = 1$ the generalized TR schemes give

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

The probabilistic interpretation of the above equation is the following.

A particle extracted from $f^n$

- does not collide with probability $A_0$, (i.e. it is sampled from $f^n$)
- collides with another particle extracted from $f^n$ with probability $A_1$ (i.e. it is sampled from the function $f_1$)
- is replaced by a particle sampled from a Maxwellian with probability $A_2$.

Remarks: In this formulation the probabilistic interpretation holds uniformly in $\mu \Delta t$, at variance with NB, which requires $\mu \Delta t < 1$. Furthermore, as $\mu \Delta t \to \infty$, the distribution at time $n + 1$ is sampled from a Maxwellian.

In a space non homogeneous case, this would be equivalent to a particle method for Euler equations.
Second order TR scheme (TRMC2):

Form \( m = 2 \) the generalized TR schemes give

\[
f^{n+1} = A_0 f^n + A_1 f_1 + A_2 f_2 + A_3 M,
\]

with \( f_1 = P(f^n, f^n)/\mu, f_2 = P(f^n, f_1)/\mu. \)

The probabilistic interpretation of the scheme is the following. Given \( N \) particles distributed according to \( f^n \):

- \( NA_0 \) particles do not collide,
- \( NA_1 \) are sampled from \( f_1 \), as in the first order scheme,
- \( NA_2 \) are 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 \( f_1 \),
- \( NA_3 \) particles are sampled from a Maxwellian.

Remarks: Previous MC schemes can be made exactly conservative. This goal is achieved by using a suitable algorithm for sampling a set of particles with prescribed momentum and energy from a Maxwellian.

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

\[
f^{n+1} = A_0 f^n + A_1 f_1 + A_2 f_2 + A_3 f_3 + A_4 M
\]

with \( f_1 = P(f^n, f^n)/\mu, f_2 = P(f^n, f_1)/\mu \) and \( f_3 = [2P(f^n, f_2) + P(f_1, f_1)]/3\mu. \)
Numerical results

Space homogeneous case

Comparison between: 

NB, TRMC1, TRMC2, TRMC3.

Test problems:

Exact solution for the Kac equation

Exact solution for Maxwell molecules

Remarks The function is reconstructed on a regular grid by convolving the particle distribution with a suitable mollifier.

Number of particles: $N = 5 \times 10^4$ (1d) $N = 5 \times 10^5$ (2d)
**Kac equation**: Details of the distribution function at time $t = 2.0$ for $\Delta t = 1.0$. Exact (line), DSMC (+), first order TRMC ($\triangle$), second order TRMC ($\ast$), third order TRMC ($\odot$).
**Kac equation:** $L^2$ norm of the error vs time. DSMC (+), first order TRMC (△), second order TRMC (⋆), third order TRMC (○). Left: Time step $\Delta t = 1.0$. Right: DSMC with $\Delta t = 0.25$, first order TRMC with $\Delta t = 0.5$, second order TRMC with $\Delta t = 0.75$, third order TRMC3 with $\Delta t = 1.0$. 
Maxwellian case: $L^2$ norm of the error vs time. DSMC (+), first order TRMC ($\triangle$), second order TRMC (*), third order TRMC (○). Left: Time step $\Delta t = 1.0$. Right: DSMC with $\Delta t = 0.25$, first order TRMC with $\Delta t = 0.5$, second order TRMC with $\Delta t = 0.75$, third order TRMC3 with $\Delta t = 1.0$. 
Space non homogeneous case

1D Shock wave profiles

Comparison between:

NB, TRMC1, TRMC2, TRMC2

Initial data \( f(x,v,t) = M(\rho,u,T) \), with

\[
\rho = 1.0, \quad T = 1.0, \quad Ma = 3.0, \quad x > 0,
\]

where \( Ma \) is the Mach number. The mean velocity is

\[
u_x = -Ma\sqrt{\left(\frac{\gamma}{\gamma^2-1}\right)T}, \quad u_y = 0,
\]

with \( \gamma = 5/3 \)

The values for \( \rho, u \) and \( T \) for \( x < 0 \) are given by the Rankine-Hugoniot conditions.

Test problem:

- **Hard spheres**: 50 – 100 space cells and 500 particles in each cell on \( x > 0 \). The reference solution is obtained with 200 space cells and 500 particles in each cell on \( x > 0 \).

**Remark**: Since we have a stationary shock wave the accuracy of the methods can be increased by computing averages on the solution for \( t \gg \).
**1D shock profile:** DSMC(+) and first order TRMC (×) (top), second order (*) and third order (◦) TRMC (bottom) for $\epsilon = 1.0$ and $\Delta t = 0.025$. From left to right: $\rho$, $u$, $T$. The line is the reference solution.
**1D shock profile**: DSMC(+) and first order TRMC (×) (top), second order (*) and third order (◦) TRMC (bottom) for $\epsilon = 0.1$ and $\Delta t = 0.0025$ for DSMC, $\Delta t = 0.025$ for TRMC. From left to right: $\rho$, $u$, $T$. The line is the reference solution.
1D shock profile: First order TRMC (×) for $\epsilon = 10^{-6}$ and $\Delta t = 0.025$. From left to right: $\rho$, $u$, $T$. 
2D Flow past an ellipse

Euler or Navier-Stokes region

Boltzmann region

ε << 0.01

ε > 0.01

NB, TRMC1 and TRMC2 schemes

\( Ma = 20, \ \rho_{inf} = 0.01, \ T_{inf} = 200, \ T_{obj} = 1000, \ \epsilon = 0.1, 0.01, 0.000001 \)

Test problem:

- **Hard spheres**: 75 × 60 space cells and 100 particles in each cell at 'infinity'. Full accommodation boundary condition.

**Remark**: Since we have a stationary shock wave the accuracy of the methods can be increased by computing averages on the solution for \( t \gg \).
2D flow: $\varepsilon = 0.1$. NB, TRMC1 and TRMC2 solution for the mass $\rho$. 
2D flow: $\varepsilon = 0.1$. Transversal and longitudinal sections for the mass $\rho$ at $y = 6$ and $x = 5$ respectively for $\varepsilon = 0.1$ and $M = 20$; DSMC-NB (○), TRMC I (+), TRMC II (×).
2D flow: $\varepsilon = 0.01$. NB, TRMC1 and TRMC2 solution for the mass $\rho$. 
**2D flow**: $\epsilon = 0.01$. Transversal and longitudinal sections for the mass $\rho$ at $y = 6$ and $x = 5$ respectively for $\epsilon = 0.1$ and $M = 20$; DSMC-NB ($\circ$), TRMC I ($+$), TRMC II ($\times$).
2D flow: $\varepsilon = 0.000001$. NB, TRMC1 and TRMC2 solution for the mass $\rho$. 
2D flow: \( \epsilon = 0.000001 \). Transversal and longitudinal sections for the mass \( \rho \) at \( y = 6 \) and \( x = 5 \) respectively for \( \epsilon = 0.1 \) and \( M = 20 \); DSMC-NB (○), TRMC I (+), TRMC II (×).
2D flow: Number of "Collisions". From left to right $\epsilon = 0.1, 0.01, 0.001$; NB ($\circ$), TRMC1 ($+$), TRMC2 ($\times$).
Basic References


Note: T►Time discretization, MC►Monte Carlo methods.