Monte Carlo methods for kinetic equations

Lecture 2: Monte Carlo simulation methods

Lorenzo Pareschi

Department of Mathematics & CMCS
University of Ferrara
Italy

http://utenti.unife.it/lorenzo.pareschi/
lorenzo.pareschi@unife.it

KT2009: Tutorials
IPAM, March 10-13, 2009

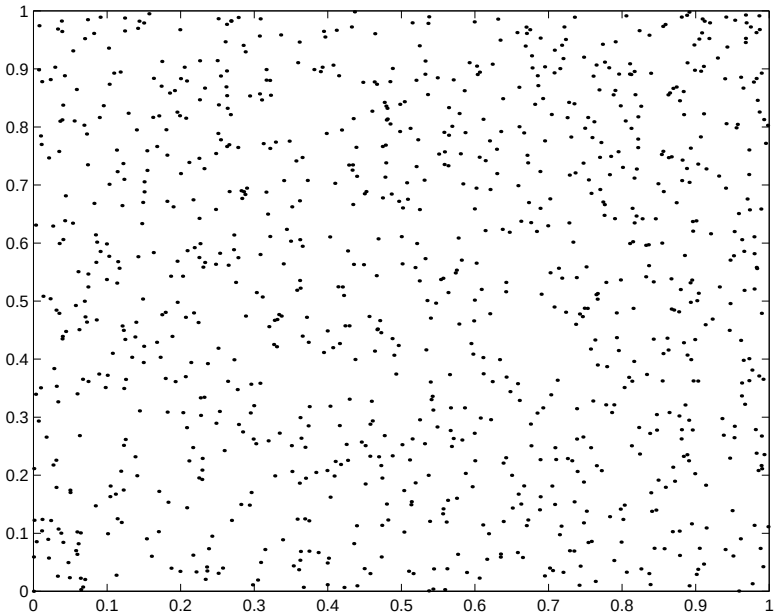
Outline

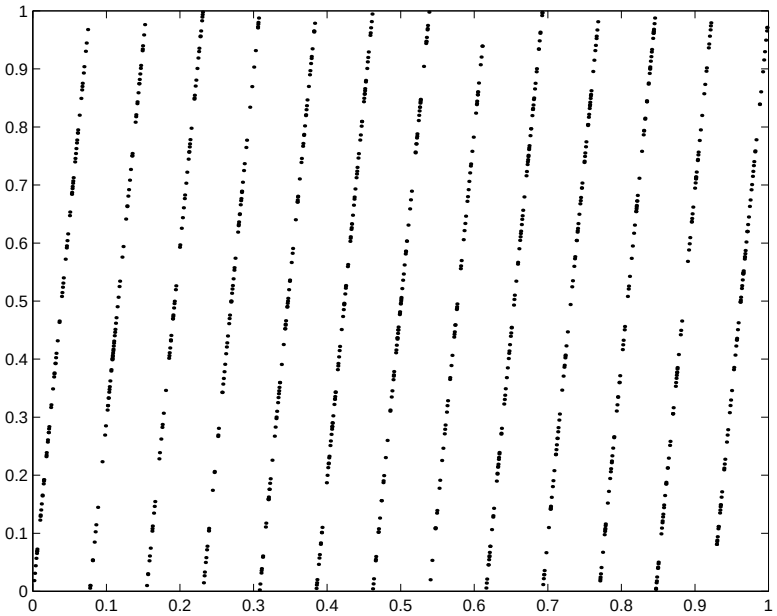
Pseudo-Random numbers

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 pseudo random number between 0 and 1.

- Real random numbers can not be generated because
  - floating points are used as approximation or real numbers
  - a really random sequence can not be generated even at a discrete level (it would require an infinite memory)
- Random number generators produce a sequence of numbers which satisfy some properties of random sequences. In particular, one wishes to generate a sequence ξ_n which is
  - uniformly distributed (approximate Lebesgue measure in $[0,1]$)
  - the elements of the sequence are uncorrelated (for example absence of pairwise correlation means that (ξ_n, ξ_{n+1}) should approximate Lebesgue measure in $[0,1]^2$)
  - they have to be computed quickly.

Good and bad generators





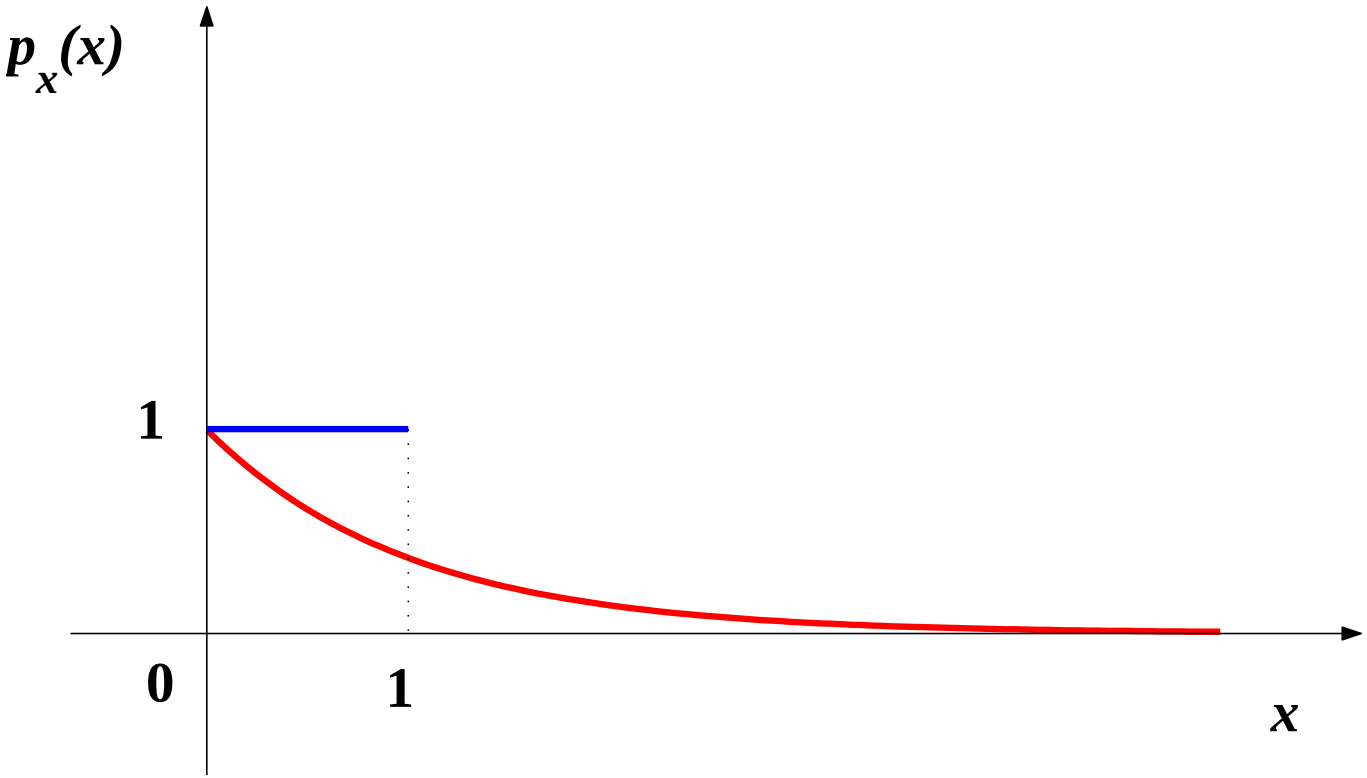
Example: *Linear Congruential Generators* (LCG):

$$x_{n+1} = (ax_n + c) \bmod m, \quad n \geq 0$$

with $a, c, m \in \mathbb{N}$.
Dividing x_n by m one obtains an approximation of the uniform distribution in $[0, 1]$.
x_n is a sequence with period at most m, therefore m has to be large enough. The quality of the result depends on the choice of a, c, m. A "good" choice, used by Matlab 4.0, is $m = 2^{31} - 1, a = 7^5, c = 0$.

Monovariate distributions

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 ξ be a uniformly distributed random variable (number) in $[0, 1]$.



Inverse transform methods

Then the relation between x and ξ can be found using inverse transform methods. We have

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

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 ξ, 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 $[0, 1]$.

Acceptance-rejection methods

To compute the inverse function, in general, a nonlinear equation has to be solved. This can be computationally expensive. 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

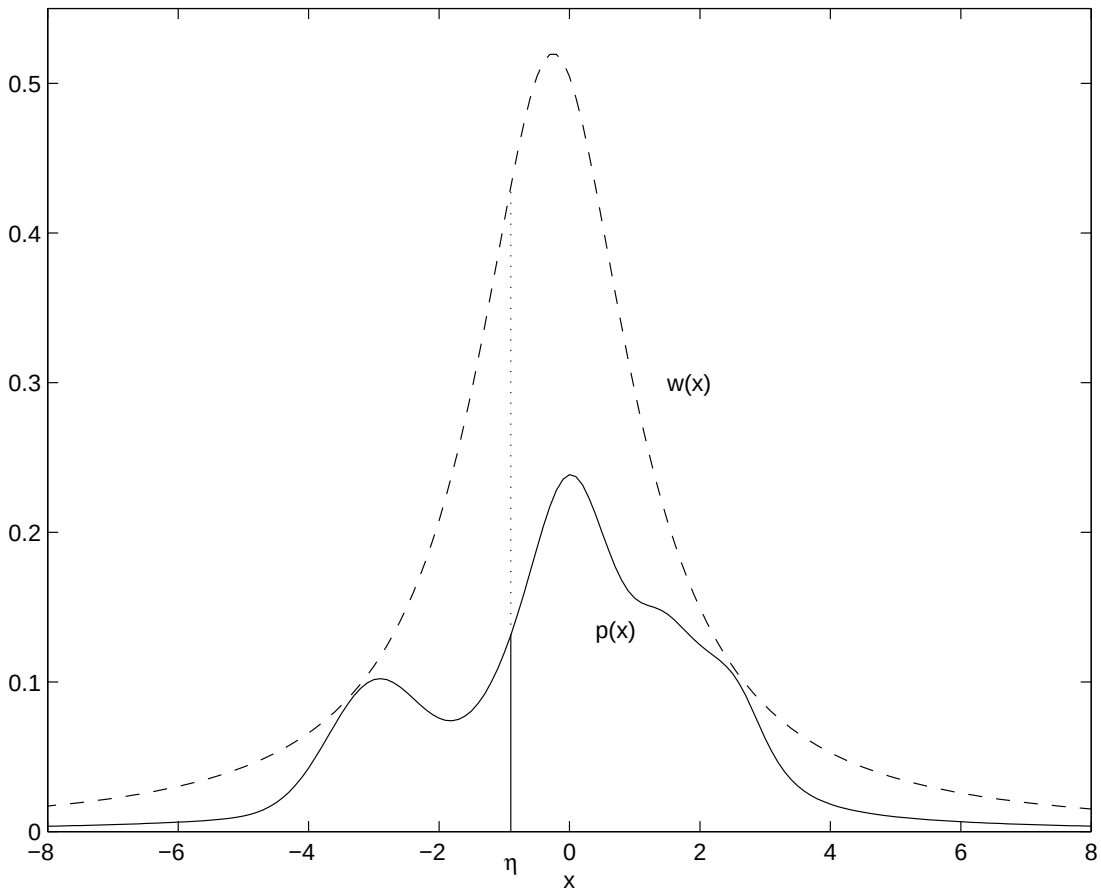
$$A = \int_{-\infty}^{\infty} w(x) \, dx$$

and denote with ξ_1 and ξ_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.*

Acceptance-rejection



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 $1/A$.

Convex combinations

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.

$$w_i \geq 0, \quad \sum_{i=1}^{M} w_i = 1,$$

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* $p_i(x)$*.*

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 $T : x \to \eta = T(x)$ such that in the new variables the probability density is factorized, i.e.

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

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

Normal distributions

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

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\, dt,$$

denotes the error function. However the inversion of the error function may be expensive.

Box-Muller method

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

$$x = \rho\cos\theta,\ y = \rho\sin\theta,$$

then we have

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

Box-Muller method

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, from these x and y are easily obtained.
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

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

Surface of a sphere

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

$$\varphi = 2\pi\xi_1, \quad \theta = \arccos(1 - 2\xi_2).$$

Monte Carlo integration

Consider the simple integral

$$I[f] = \int_{[0,1]^d} f(x)dx, \quad d \geq 1,$$

then if x is a random vector uniformly distributed in $[0,1]^d$ we have $I[f] = E[f(x)]$, where $E[\cdot]$ denotes the *expectation*.
If $\{x_n\}$ is a sequence of pseudo-random vectors uniform in $[0,1]^d$ then

$$I_N[f] = \frac{1}{N}\sum_{n=1}^{N} f(x_n), \quad E[I_N[f]] = I[f].$$

For the law of large numbers it converges in probability[1]

$$\lim_{N\to\infty} I_N[f] = I[f],$$

and

$$I[f] - I_N[f] \approx \sigma_f N^{-1/2} w, \quad E[(I[f] - I_N[f])^2] = \sigma_f N^{-1/2},$$

where σ_f^2 is the variance of f and w is a normal random variable. Note that there is no dependence on the dimension.

[1] W.Feller '71, R.E.Caflisch '98

Moments

If $g(x)$ is a non uniform probability density function in $[0,1]^d$ and we consider the integral

$$I[f] = \int_{[0,1]^d} f(x)g(x)dx, \quad d \geq 1,$$

then if $\{x_n\}$ is a sequence of pseudo-random vectors distributed as $g(x)$ in $[0,1]$ we have

$$I_N^k[f] = \frac{1}{N}\sum_{n=1}^{N} f(x_n), \quad E[I_N^k[f]] = I_k[f],$$

and again convergence rate goes like $O(N^{-1/2})$.
A typical situation of this type is when we evaluate moments of $g(x)$

$$I^k[f] = \int_{[0,1]^d} g(x)x^k dx, \quad d, k \geq 1,$$

as

$$I_N^k[f] = \frac{1}{N}\sum_{n=1}^{N}(x_n)^k.$$

Remark: The convergence rate for a deterministic grid based quadrature is $O(N^{-k/d})$ for an order k method. Thus Monte Carlo is "better" if $k/d \leq 1/2$.

Variance reduction strategies

- One of the main drawback of the acceptance-rejection techniques described above is the large variance of the samples we obtain. This is mainly due to the fact that we sample from the whole interval of interest of the distribution function.
- *Stratified sampling* The basic principle stratified sampling is to divide the sampling interval up into subintervals (cells). You then perform an inverse transform sampling or an acceptance-rejection approach on each subinterval.
- *Importance sampling* You rewrite the integral as

$$I[f] = \int_{[0,1]^d} \frac{f(x)}{g(x)} g(x)\, dx$$

and given $\{x_n\}$ pseudo-random numbers distributed as $g(x)$ estimate

$$I_N[f] = \frac{1}{N} \sum_{n=1}^{N} \frac{f(x_i)}{g(x_i)}.$$

Reconstruction

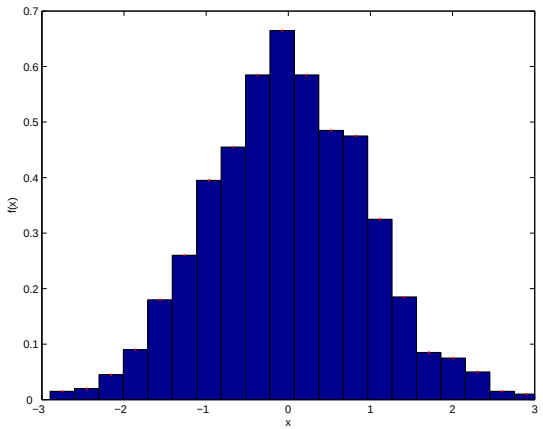
Given a set of N samples $\xi_1, \xi_2, \ldots, \xi_N$ the probability density is defined by

$$f(x) = \frac{1}{N} \sum_{k=1}^{N} \delta(x - \xi_k).$$

The simplest method, which produces a piecewise constant reconstruction, is based on evaluating the histogram of the samples at the cell centers of a grid

$$f(x_{j+1/2}) = \frac{1}{N} \sum_{k=1}^{N} \Psi(\xi_k - x_{j+1/2}), \quad j = \ldots, -2, -1, 0, 1, 2, \ldots$$

where $\Psi(x) = 1/\Delta x$ if $|x| \leq \Delta x/2$ and $\Psi(x) = 0$ elsewhere.



Splitting approach

A common approach to solve a kinetic equation is operator splitting. The solution in one time step Δt may be obtained by the sequence of two steps.
First integrate the space homogeneous equation for all $x \in \Omega$,

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

for a time step Δt (*collision step*) to obtain $\tilde{f} = C_{\Delta t}(f_0)$.
Then solve the transport equation using the output of the previous step as initial condition,

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

for a time step Δt (*transport step*) to get $f = T_{\Delta t}(\tilde{f}) = T_{\Delta t}(C_{\Delta t}(f_0))$.
After this splitting the major numerical difficulties are in the collision step. Note that the transport step corresponds to simple free flow of particles.

Splitting approach

- The splitting scheme described above is first order accurate in time. The accuracy in time may be improved by a more sophisticated splitting. For example Strang splitting[2] is second order accurate (provided both steps are at least second order). It can be written as

$$f = C_{\Delta t/2}(T_{\Delta t}(C_{\Delta t/2}(f_0))),$$

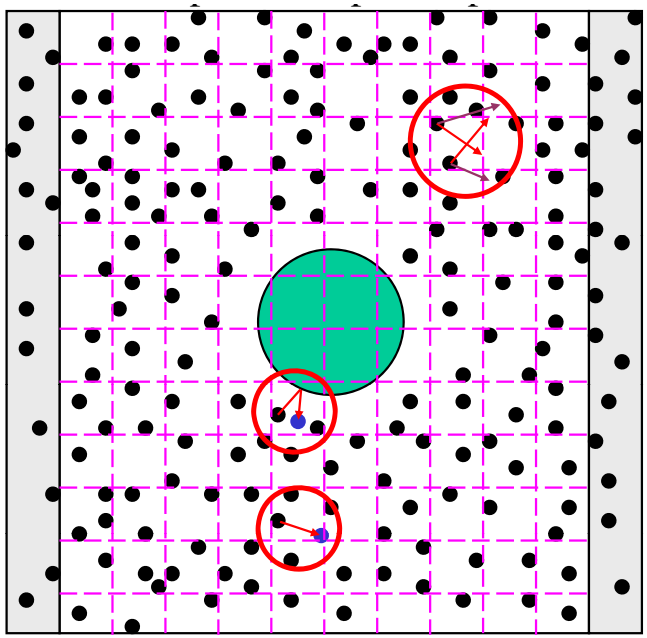
or equivalently as

$$f = T_{\Delta t/2}(C_{\Delta t}(T_{\Delta t/2}(f_0))).$$

- Note that, if the initial data is in local equilibrium and both steps are solved exactly, then simple splitting and Strang splitting does not differ. So simple splitting becomes second order accurate.
- Both splitting methods for vanishingly small values of ε becomes a first order *kinetic scheme* for the underlying fluid dynamic limit. The collision step becomes a projection towards the local Maxwellian $C_{\Delta t}(f_0) = M(f_0)$ which is then transported by the transport step $f = T_{\Delta t}(M(f_0))$. Thus Strang splitting reduces its accuracy to first order in time in this regime.

[2] G.Strang, 1968

DSMC basics



- Initialize system with particles (x_i, v_i), $i = 1, \ldots, N$ (*sampling*).
- Loop over time steps of size Δt.
- Create particles at open boundaries.
- Move all the particles $x_i = x_i + v_i \Delta t$ (*transport step*.
- Process any interactions of particle and boundaries (*Maxwell's b.c.*).
- Sort particles into cells.
- Select and execute random collisions (*collision step*).
- Compute average statistical values.

DSMC for the collision step

In this paragraph we will describe the *classical DSMC methods* due to Bird and Nanbu in the case of spatially homogeneous Boltzmann equations[3].
We assume that the kinetic equations 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 s.t.

$$\frac{1}{\mu}\int_{\mathbb{R}} P(f,f)(v)\phi(v)\,dv = \int_{\mathbb{R}} f(v)\phi(v)\,dv, \quad \phi(v) = 1, v, v^2.$$

For the BGK equation $P(f,f) = \mu M(\rho,u,T)(v)$, for the Boltzmann equation in the Maxwellian case

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

and $\mu = 4\pi\rho$. The case of general VHS kernels will be discussed later.

[3] G.Bird '63, K.Nanbu '83

Nanbu's method (DSMC no time counter)

We assume that f is a probability density, i.e. $\rho = \int_{-\infty}^{+\infty} f(v,t)\,dv = 1$.
Consider a time interval $[0, t_{\max}]$, and discretize it in n_{TOT} intervals of size Δt.
Let $f^n(v)$ be an approximation of $f(v, n\Delta t)$. The forward Euler scheme writes

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

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.

- *Physical level*: 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 $P(f^n, f^n)(v)$.
- *Monte Carlo level*: 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 $P(f^n, f^n)(v)/\mu$.

Note that $\Delta t \le \epsilon/\mu$ to have the probabilistic interpretation. For the BGK model the algorithm is straightforward since sampling from $P(f,f)/\mu$ is simply sampling from a Maxwellian.

Maxwellian case

First we consider the case where the collision kernel does not depend on the relative velocity.

Algorithm[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_{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

Nanbu's algorithm is not conservative, i.e. momentum and energy are conserved only in the mean, but not at each collision. A conservative algorithm is obtained selecting independent particle pairs, instead of single particles.

Nanbu-Babovsky for the Maxwellian case

The expected number of collision pairs in a time step Δt is $N\mu\Delta t/(2\epsilon)$.

Algorithm[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_{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 $\mathrm{Iround}(x)$ we denote

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

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

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 ξ denotes a random number, uniformly distributed in $[0,1]$.

For the Boltzmann equation 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 ω is chosen uniformly in the unit sphere.

More precisely we have:

Two-dimension:

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

Three-dimension:

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

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

Variable Hard Sphere case

The extend the algorithm to non constant scattering cross section we shall assume that the collision kernel satisfies some *cut-off hypothesis*.
We will denote by $Q_\Sigma(f,f)$ the collision operator with kernel

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

and, for a fixed Σ, 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

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

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

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

In this case, a simple scheme is obtained by using the acceptance-rejection technique to sample the post collisional velocity according to $P(f,f)/\mu$.

Nanbu-Babovsky for VHS

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,* $\{v_i^0, i = 1, \dots, N\}$, *by sampling them from the initial density* $f_0(v)$
2. for $n = 1$ to n_{TOT}
   given $\{v_i^n, i = 1, \dots, N\}$
   - *compute an upper bound* Σ *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* $B_{ij} = B(|v_i - v_j|)$
     - if $\Sigma\,\mathrm{Rand} < B_{ij}$
       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 Σ

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

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

since this computation would require an $O(N^2)$ operations.
An upper bound of $B_{\max}$ 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:

- The probabilistic interpretation breaks down if $\Delta t/\epsilon$ is too large. This implies that the time step becomes extremely small when approaching the fluid dynamic limit.
- 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 Σ (like in the Maxwellian case).

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 (as a simplified molecular dynamics) for the simulation of particle collisions.
Let us consider first the Maxwellian case. The number of collisions in a short time step Δt is given by

$$N_c = \frac{N\mu\Delta t}{2\varepsilon}, \quad \mu = 4\pi\rho.$$

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

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

The method is then based on selecting randomly a particle pair, compute the collision result and update the local time counter by Δt_c.

Bird for Maxwellian case

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,* $\{v_i^0, i = 1, \dots, N\}$, *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_{TOT}
   - repeat
     - *select a random pair* (i, j) *uniformly within all possible 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

The algorithm is similar to the Nanbu-Babovsky (NB) scheme for Maxwellian molecules. 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.

Variable Hard Sphere case

For a more general kernel, Bird's scheme is 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

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

where $\overline{B}$ denotes the *average collision frequency*.
Then the mean collision time can be computed as

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

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 $\overline{B}$ is too expensive. To avoid this one computes a local time counter as follows. First select a collision pair (i, j) using rejection. Then compute

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

Bird for VHS

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,* $\{v_i^0, i = 1, \dots, N\}$, *by sampling them from the initial density* $f_0(v)$
2. *set time counter* $t_c = 0$
3. for $n = 1$ to n_{TOT}
   - *compute an upper bound* Σ *of the cross section*
   - repeat
     - *select a random pair* (i, j) *uniformly*
     - *compute the relative cross section* $B_{ij} = B(|v_i - v_j|)$
     - if $\Sigma\,\xi < B_{ij}$
       - *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* $\Delta t_{ij} = 2\varepsilon/(N\rho B_{ij})$
       - *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

Final remarks

- The presence of *multiple collisions* per time step in Bird's method has a profound influence on the time accuracy. While Nanbu scheme converges in probability to the time discrete Boltzmann equation, Bird's method converges to the space homogeneous Boltzmann equation[4].
- Numerical tests confirm that in space non homogeneous situations Bird's method with simple splitting can achieve *second order accuracy* in time whereas Nanbu's is always first order[5].
- In the original nonconservative form one can show that Nanbu's method gives the wrong expectation for the temperature[6].
- Exact conservation of moments forces the velocity domain to remain bounded during relaxation $|v_i| \leq \sqrt{2EN}$. As a consequence steady state particles are never "true" Maxwellian samples.
- Similarly to Nanbu's method also Bird's method becomes very expensive and practically unusable near the *fluid regime*. Infact, 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] R.Illner, H.Babovski '89, W.Wagner '92
[5] A.Garcia, W.Wagner '00
[6] C.Greengard, L.G.Reyna '91