Monte Carlo methods for integro-differential equations

Lecture 1: Monte Carlo methods

Lorenzo Pareschi

Department of Mathematics & CMCS
University of Ferrara
Italy

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

NSPDE2
Malaga, February 8-12, 2010

Course Outline

1. Lecture #1: Monte Carlo methods
   (a) Introduction to Monte Carlo methods
   (b) Integration and applications to PDEs
2. Lecture #2: Kinetic equations
   (a) Direct Simulation Monte Carlo
   (b) Asymptotic Preserving Monte Carlo

Outline

Introduction

We shall present here the motivation and a general description of a method dealing with a class of problems in mathematical physics. The method is, essentially, a statistical approach to the study of differential equations, or more generally, of integro-differential equations that occur in various branches of the natural sciences.

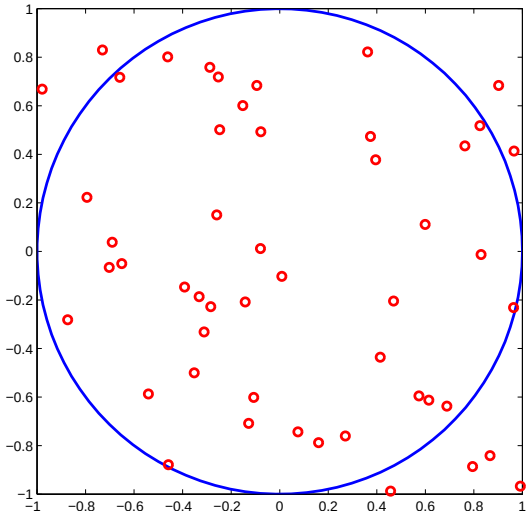
(N.Metropilis, S.Ulam, "The Monte Carlo method", J. Am. Stat. Ass., 1949.)

Monte Carlo methods

- Nowadays Monte Carlo methods find application in a wide field of areas, including many subfields of physics, like statistical physics or high energy physics, and ranging to areas like biology, chemistry, finance, computer graphics and video games.
- Classical mathematical applications of Monte Carlo involves the computation of multidimensional integrals, the solution of partial differential equations, Markov chains and optimization problems.
- Monte Carlo methods are often used when other methods fail, since they are much less sensitive to the course of dimensionality, which plagues deterministic methods in problems with a large number of variables.
- Despite the widespread use of the methods, and numerous descriptions of them in articles and monographs[1], it is virtually impossible to find a unique notion of Monte Carlo method in the literature and the term Monte Carlo is often used to denote any numerical technique based on some kind of stochastic simulation.

[1] J.M.Hammersley, D.C.Handscomb, Monte Carlo Methods, 1964. N.Madras, Lectures on Monte Carlo methods, 2002.

A simple example

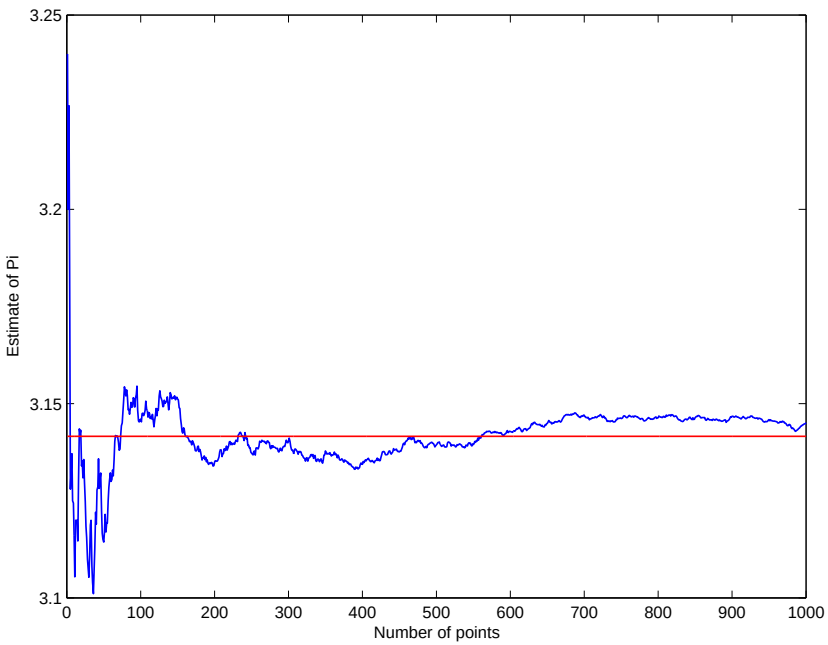


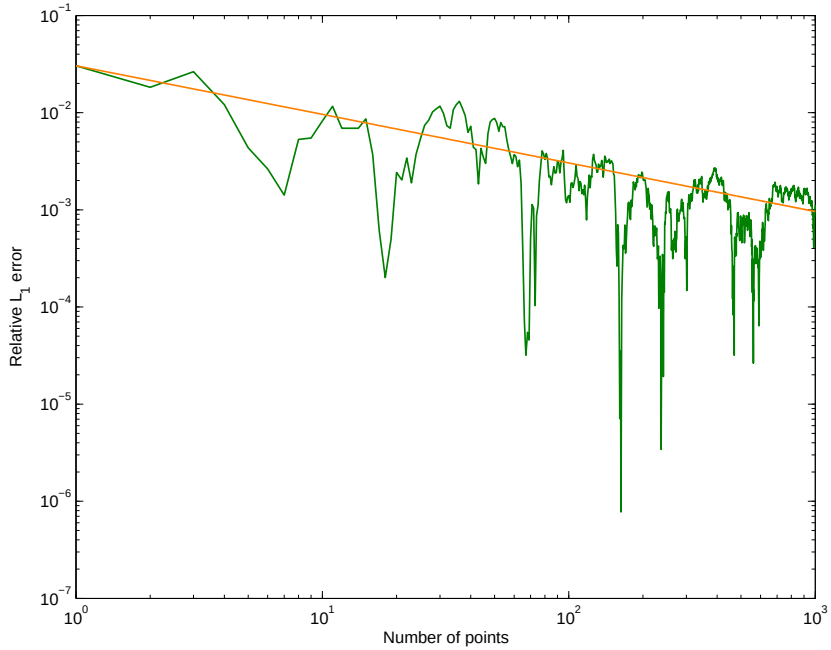
Suppose we want to estimate π with a Monte Carlo method. We can consider a circle of diameter 2 inside the square $[-1,1]\times[-1,1]$. The square has area 4 and the circle π. Suppose we pick up a point P at random uniformly inside the square. The probability that P lies inside the circle is $\pi/4$. If we select N points $P_1,\ldots,P_N$ independently and uniformly in the square we can estimate π from

$$\frac{\pi}{4}\approx\frac{Z}{N}$$

where Z is the number of points inside the circle. More precisely Z has a binomial distribution with parameters N and $\pi/4$ and its expectation is $E(Z)=N\pi/4$.

Convergence





For example, taking $N = 10^5$ points suppose we observed $Z = 78582$ then our estimates of π would be $4 \times 78582/10^5 = 3.143280$. It can be shown that for the law of large numbers

$$\lim_{N \to \infty} P(|\pi - E(4Z/N)| \geq \varepsilon) = 0,$$

namely as N gets large there is a very small probability that our estimates deviates much from π.

Basic issues

By extending this example it is evident that one could use Monte Carlo to estimate areas and volumes and more in general subsets of $\mathbb{R}^d$ (and therefore also integrals).
From the example we can also see the importance of the following basic issues common to most Monte Carlo methods

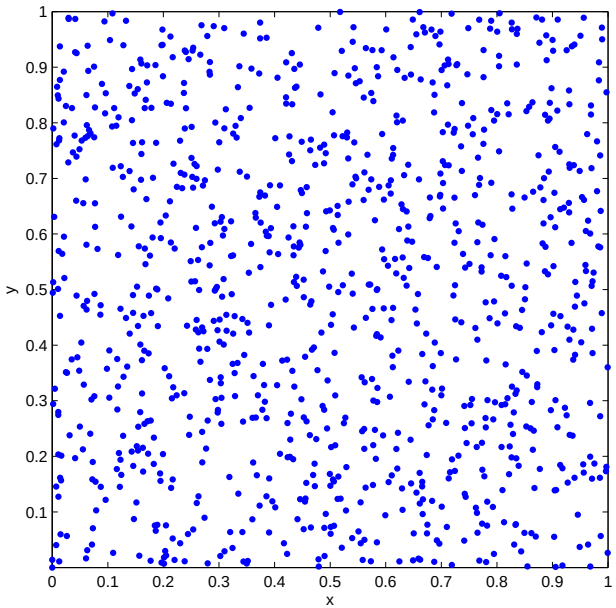
- Generation. How do we generate random numbers with a specific distribution? We have to keep in mind that we need many random numbers and that they must be computed fast.
- Accuracy. How accurate can we expect our answers to be? In the example we found that the accuracy of our estimate was proportional to $1/\sqrt{N}$. Is this a general principle of Monte Carlo methods?
- Efficiency. How efficient is our method? Is there another method that will gives us comparably accurate answers in less time?

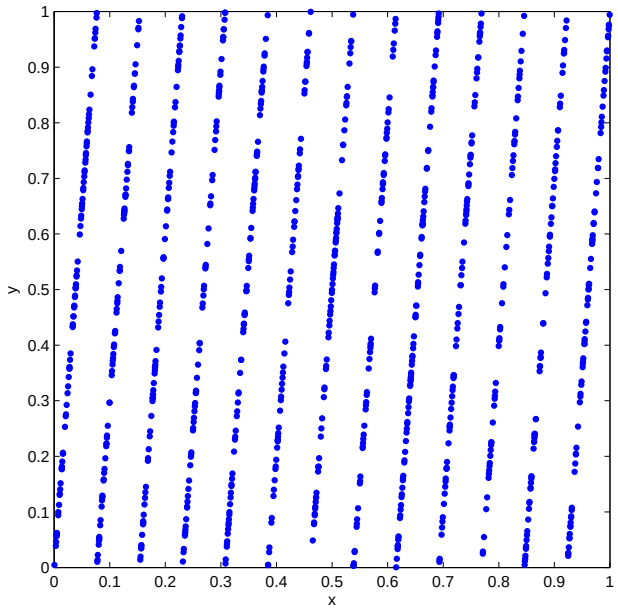
Pseudo-Random numbers

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





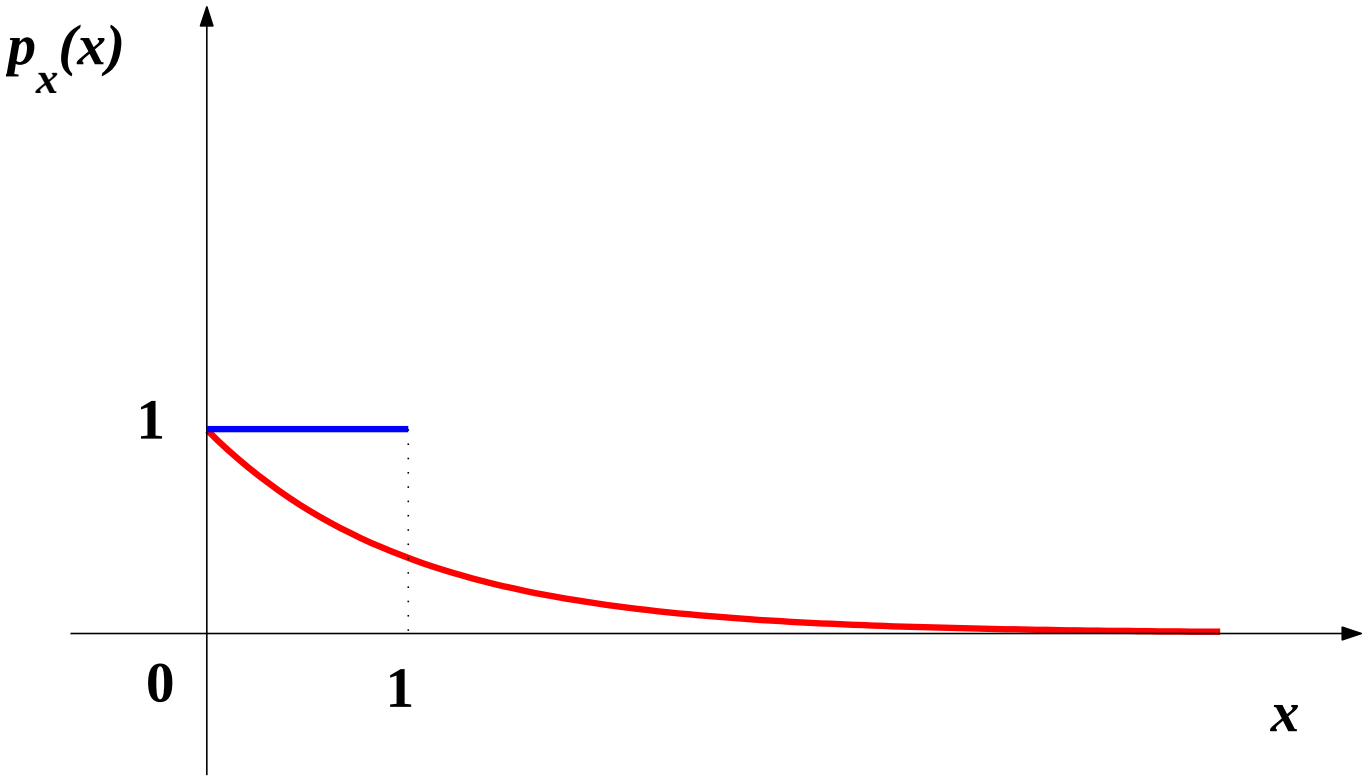
Example: *Linear Congruential Generators* (LCG):

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

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

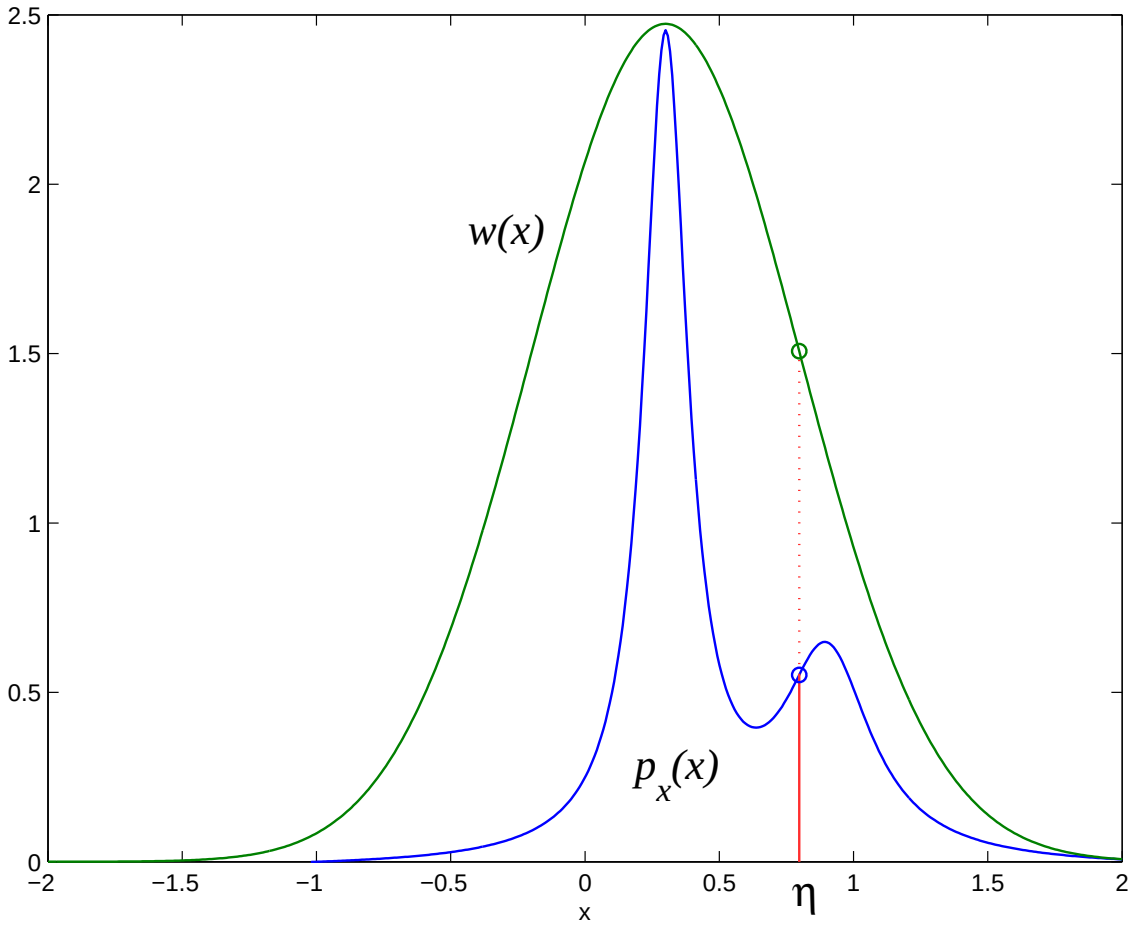
$$A = \int_{-\infty}^{\infty} w(x) \, dx$$

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.

$$p_x(x_1, \ldots x_n)dx_1 dx_2 \ldots dx_n = p_{\eta_1}(\eta_1)p_{\eta_2}(\eta_2)\cdots p_{\eta_n}(\eta_n)d\eta_1 d\eta_2 \ldots d\eta_n,$$

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[2]

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

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 with zero mean and unit variance. Note that there is no dependence on the dimension.

[2] 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

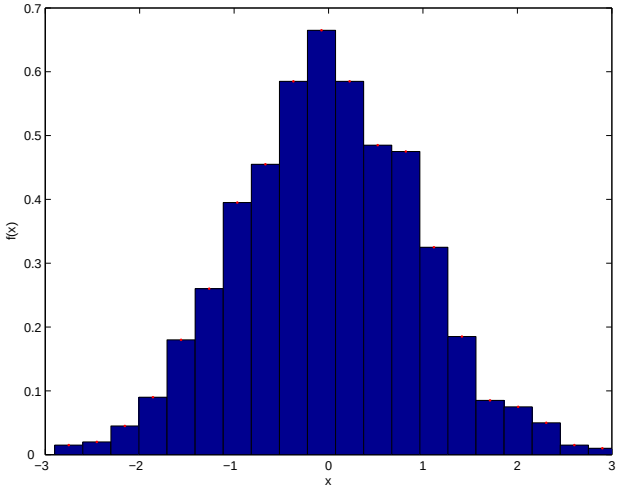
- One of the main drawback of the acceptance-rejection techniques described above is the large variance of the samples we obtain. This is mainly due to the fact that we sample from the whole interval of interest of the distribution function.
- *Stratified sampling* The basic principle stratified sampling 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.

Applications to PDEs

- Several partial differential equations can be solved numerically by probabilistic algorithms such as Monte-Carlo methods, stochastic particle methods, ergodic algorithms, etc.
- Usually these probabilistic methods are not competitive in terms of efficiency with direct deterministic discretizations (finite differences/volumes) of the same differential equations.
- However they become efficient compared to deterministic approaches in problems where a large number of dimensions or complex geometries are involved.
- Moreover, thanks to the natural interpretation of the statistical sample as a physical particle, quite often Monte Carlo methods (or particle methods) allow a better description of the physical properties of the system under study.

Diffusion problems

Diffusion equations are a classical field where Monte Carlo simulations play a relevant rule. Let's illustrate the basic principles of the method with a simple linear example

$$\begin{cases} \dfrac{\partial u}{\partial t} &= D\triangle u, \quad x \in \mathbb{R}^d, \quad D, t > 0 \\ u(x,0) &= u_0(x). \end{cases}$$

Explicit exact solution is given by the integral formula

$$u(x,t) = \int_{\mathbb{R}^d} G_t(x-y)u_0(y)\,dy = G_t * u_0(x),$$

where G_t is the heat kernel

$$G_t(x-y) = \frac{1}{(4\pi Dt)^{d/2}} e^{\frac{|x-y|^2}{4Dt}}.$$

Let us assume $u_0(x)$ a probability density and consider a set of N samples $\xi_1^0, \ldots, \xi_N^0 \in \mathbb{R}^d$ from $u_0(x)$. Note that in this case $u(x,t)$ is also a probability density.

- A first method is based on evaluating the integral solution as an expected value of the heat kernel to get

$$u(x,t) \approx \frac{1}{N} \sum_{i=1}^{N} G_t(x - \xi_i^0).$$

Note however that for each point $x \in \mathbb{R}^d$ we have a computational cost of $O(N)$, so for a grid of M points in space we have a cost of $O(MN)$.

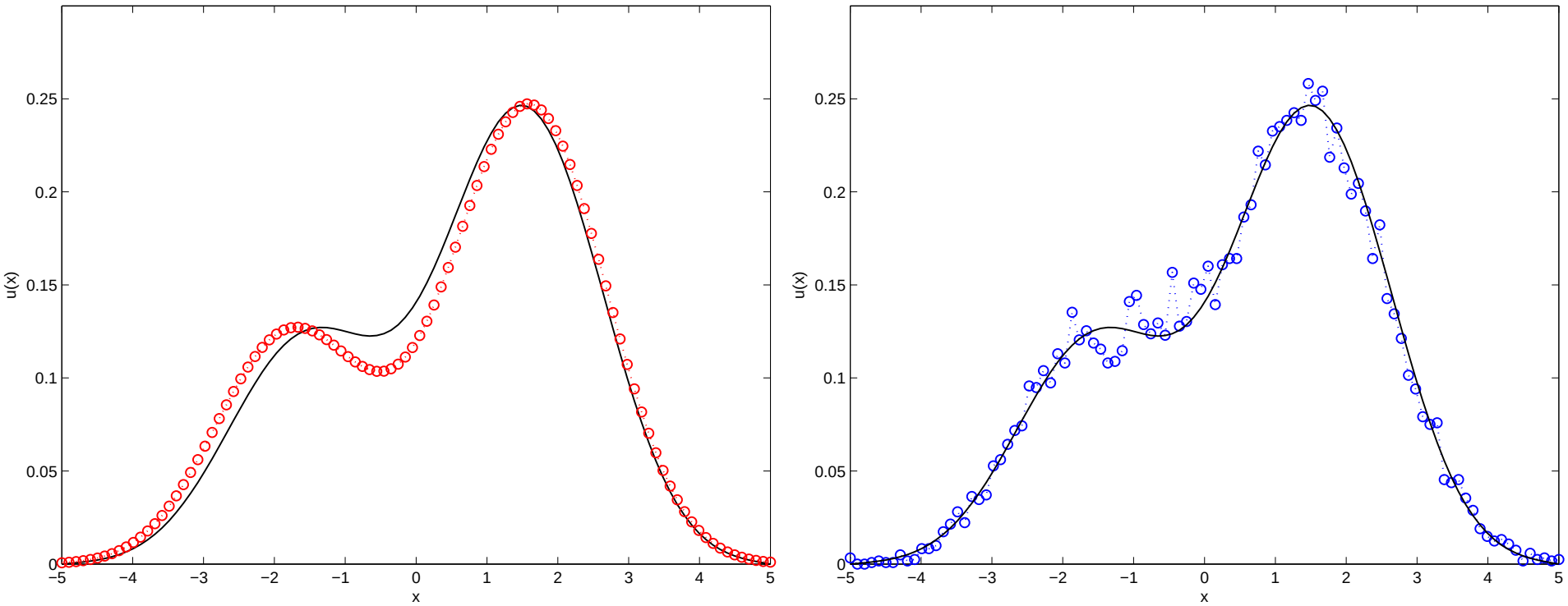
- A different approach is based on sampling directly the solution from the integral representation. This can be easily achieved observing that given two independent random variables X and Y distributed respectively as $G_t(x)$ and $u_0(y)$ then $Z = X + Y$ is distributed as $u(z,t) = G_t * u_0(z)$.
Thus a set of samples $\xi_1, \ldots, \xi_N \in \mathbb{R}^d$ from $u(x,t)$ is obtained as

$$\xi_i = \xi_i^0 + \sqrt{2Dt}\,\eta_i, \quad i = 1, \ldots, N$$

where $\eta_i \in \mathbb{R}^d$ are normally distributed with zero mean and unit variance. Note that this is nothing else then the classical random walk method[3]. In this case the cost of the reconstruction is $O(N)$ independently of M.

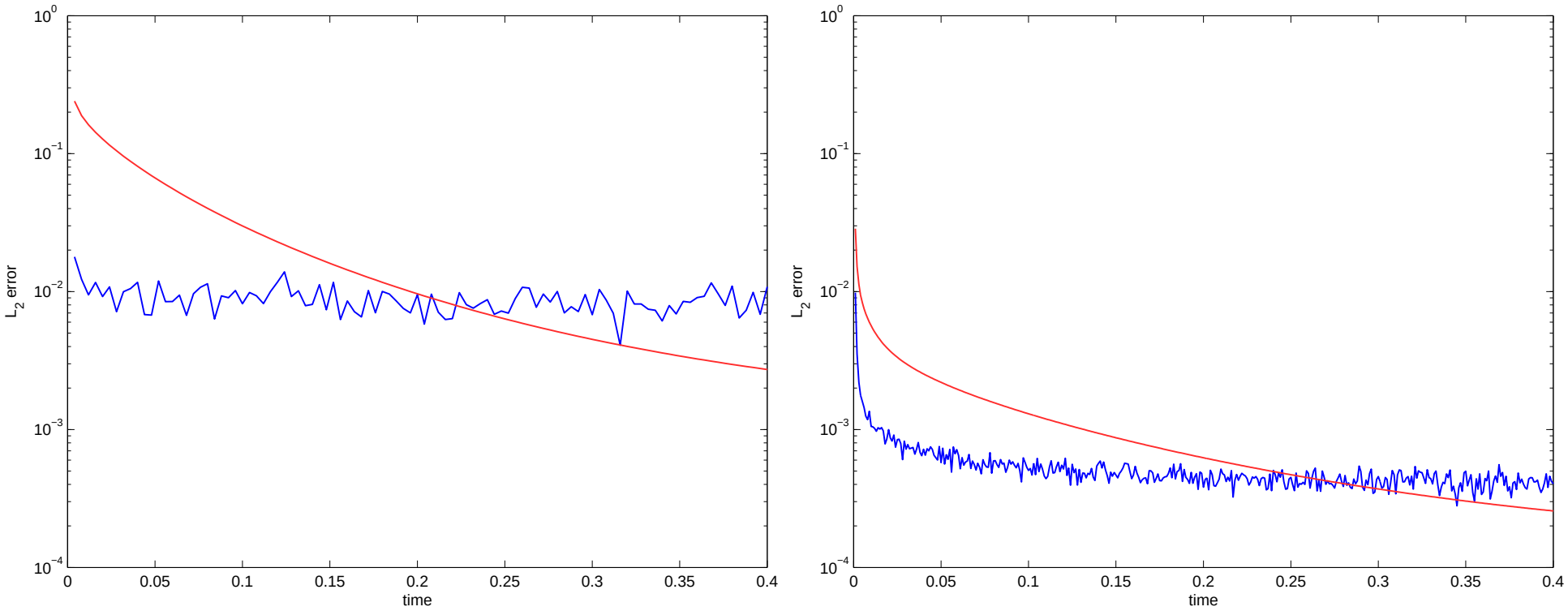
[3] A.Chorin, '70

Heat equation



Solution of the heat equation with $D = 1$ at $t = 0.4$ by estimating the expectation of the heat kernel with $M = 100$, $N = 120$ (left) and by the random walk method with $M = 100$, $N = 12 \times 10^3$ and simple piecewise constant reconstruction (right). The initial data is the sum of two square waves $u_0(x) = 0.5$, $x \in [-2.5, -0.5]$, $u_0(x) = 1$, $x \in [0.5, 2.5]$ and $u_0(x) = 0$ elsewhere.

Heat equation



Plot of the L_2 error in time for the two Monte Carlo methods. Left: $M = 100$ and expectation of the heat kernel with $N = 120$ (red) and by the random walk method with $N = 12 \times 10^3$ (blue). Right: $M = 200$ and expectation of the heat kernel with $N = 6000$ (red) and by the random walk method with $N = 12 \times 10^5$ (blue).

Elliptic equations

Monte-Carlo methods have been widely used for solving elliptic equations such as Laplace equation and Poisson equation. This becomes very economical when you want to solve the equation at a few points or when there is a severe gradient near the boundary. Let us consider the Laplace equation

$$\begin{cases} \triangle u &= 0, \quad x \in \Omega \subset \mathbb{R}^2 \\ u(x) &= g(x), \quad x \in \partial\Omega. \end{cases}$$

Solutions of Laplace equation are called harmonic functions and for such functions we have the mean value property

$$u(x) = \frac{1}{2\pi R} \int_{\partial D} u(x)\, dx,$$

for any disk $D \subset \Omega$ of radius R and center x.
This property can be used to set up a Monte Carlo simulation to compute the value of the solution in a given point $x \in \mathbb{R}^2$ by means of an iterative random walk technique.

The basic random walk method reads as follows.

1. First compute the shortest distance d of the point x to any of the boundaries of your domain Ω.
2. Calculate the next position of your point using relations

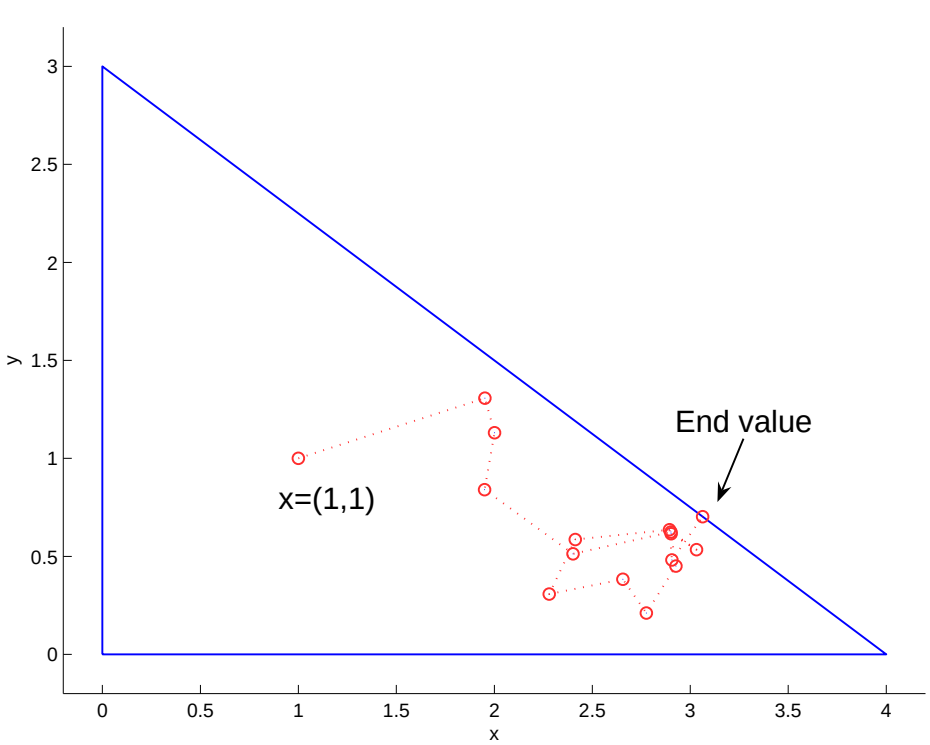
$$x = x + dn, \quad n = (\cos(\theta), \sin(\theta)),$$

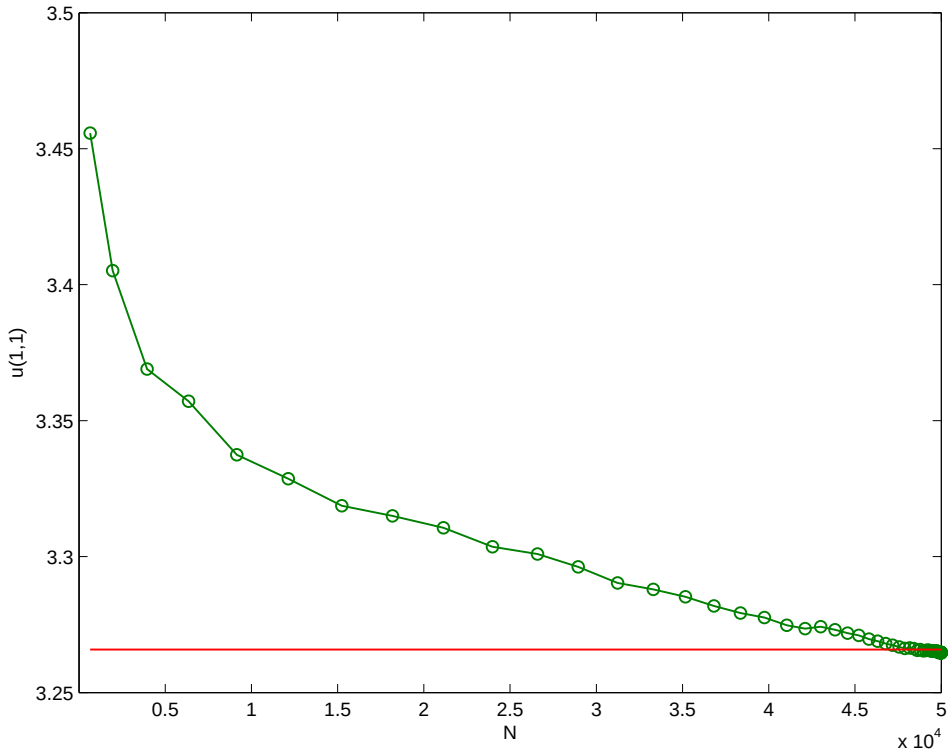
   where θ is a uniformly distributed random variable in $[0, 2\pi]$.
3. Repeat step 1 and 2 until the point reaches the boundary (up to a small fixed tolerance). Record the value of g at that point of the boundary.

If we repeat the above random walk N times starting from the same initial point x and obtain the final values $g(\xi_1), \ldots, g(\xi_N)$ it can be shown that a Monte Carlo estimate of the solution in x is given by

$$u(x) \approx \frac{1}{N} \sum_{i=1}^{N} g(\xi_i)$$

Laplace equation





Random walk solution at the point $(1,1)$. An example of random path (left) and convergence to the exact value $u(1,1) = 3.26584\ldots$ (right). The domain Ω is the triangle of vertices $(0,0)$, $(4,0)$ and $(0,3)$. The exact solution is

$$u(x_1, x_2) = \frac{1}{2} \sum_{i=1}^{3} q_i \log((x_1 - X_i)^2 + (x_2 - Y_i)^2),$$

with $q = (-2, 1, 3)$, $X = (-1, 5, -1)$ and $Y = (-1, -1, 4)$.

Convection equations

Next we consider the linear advection equation

$$\begin{cases} \dfrac{\partial u}{\partial t} + a \cdot \nabla_x u &= 0, \quad a, x \in \mathbb{R}^d, \quad t > 0 \\ u(x,0) &= u_0(x). \end{cases}$$

The exact solution reads

$$u(x,t) = u_0(x - at).$$

Again we assume $u_0(x)$ a probability density and consider a set of N samples $\xi_1^0, \ldots, \xi_N^0 \in \mathbb{R}^d$ from $u_0(x)$. Using this set of samples we can sample directly from the exact solution. In fact, $u(x,t)$ is simply given by a shift of $|at|$ in the a direction of the initial data. Thus a set of samples $\xi_1, \ldots, \xi_N \in \mathbb{R}^d$ from $u(x,t)$ is obtained as

$$\xi_i = \xi_0 + at, \quad i = 1, \ldots, N.$$

Note that except from sampling the initial data no additional source source of randomness is present and the method corresponds to a standard deterministic particle transport.

Convection-diffusion

Linear convection-diffusion problems in the form

$$\begin{cases} \dfrac{\partial u}{\partial t} + a \cdot \nabla_x u &= D \triangle u, \quad a, x \in \mathbb{R}^d, \quad D, t > 0 \\ u(x,0) &= u_0(x), \end{cases}$$

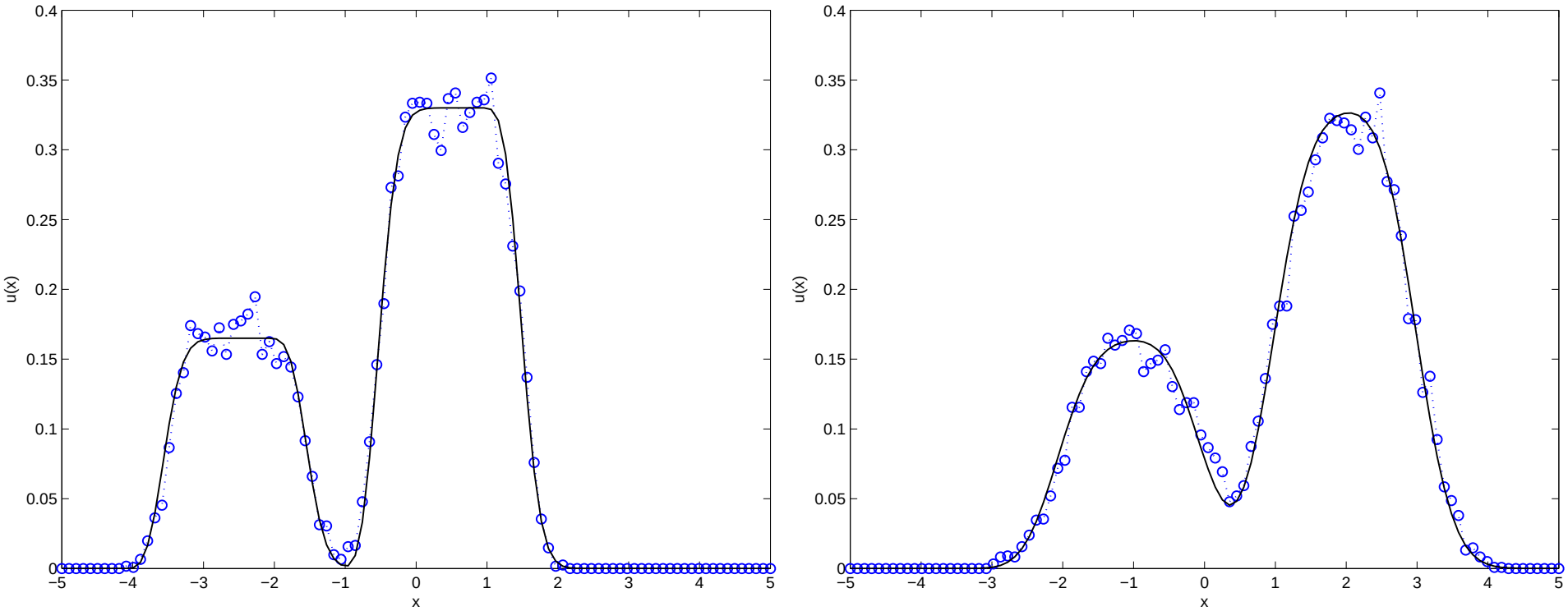
can be treated similarly by coupling the particle transport with the random walk method. Thus starting with a set of N samples $\xi_1^0, \ldots, \xi_N^0 \in \mathbb{R}^d$ from $u_0(x)$ a set of samples $\xi_1, \ldots, \xi_N \in \mathbb{R}^d$ from the solution $u(x,t)$ is obtained as

$$\xi_i = \xi_0 + at + \sqrt{2Dt}\,\eta_i \quad i = 1, \ldots, N,$$

where $\eta_i \in \mathbb{R}^d$ are normally distributed with zero mean and unit variance.

- In time dependent problems such as diffusion, convection and convection-diffusion the corresponding Monte Carlo methods don't have any stability constraints for the choice of the time step.
- In contrast with most deterministic schemes no extra dissipation due to the numerical solution of the transport part is added.

Convection-diffusion



Solution at $t = 0.5$ (left) and $t = 2$ (right) with $a = 1$ and $D = 0.05$ using $N = 12000$ samples and $M = 100$ grid points in the reconstruction. The initial data is the sum of two square waves $u_0(x) = 0.5$, $x \in [-4, -2]$, $u_0(x) = 1$, $x \in [-1, 1]$ and $u_0(x) = 0$ elsewhere.

Conservation laws

Consider now the following scalar conservation law

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial F(u)}{\partial x} &= 0, \quad x \in \mathbb{R}, \quad t > 0 \\ u(x,0) &= u_0(x). \end{cases}$$

Approximations with Monte Carlo are found in the literature for rather specific situations like the Burgers' equation[4]. We show a general way to construct Monte Carlo methods based on the following relaxation approximation[5]

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial v}{\partial x} &= 0, \\ \dfrac{\partial v}{\partial t} + a^2 \dfrac{\partial u}{\partial x} &= -\dfrac{1}{\varepsilon}(v - F(u)). \end{cases}$$

In the limit $\varepsilon \to 0$ we get the local equilibrium $v = F(u)$ and, under the subcharacteristic condition $a^2 > F'(u)^2$, we recover the scalar conservation law.

[4] M.Bossy, D.Talay, '97

[5] S.Jin, Z.Xin '95

If we define with $a > 0$

$$f = \frac{au + v}{2a},\ g = \frac{au - v}{2a},\ E_f(u) = \frac{au + F(u)}{2a},\ E_g(u) = \frac{au - F(u)}{2a},$$

the system can be rewritten in diagonal form as

$$\begin{cases} \dfrac{\partial f}{\partial t} + a\dfrac{\partial f}{\partial x} &= -\dfrac{1}{\varepsilon}(f - E_f(u)) \\ \dfrac{\partial g}{\partial t} - a\dfrac{\partial g}{\partial x} &= -\dfrac{1}{\varepsilon}(g - E_g(u)). \end{cases}$$

The solution is then approximated by means of an operator splitting based on alternating the solution of the convection and homogeneous relaxation steps

$$\begin{cases} \dfrac{\partial f}{\partial t} + a\dfrac{\partial f}{\partial x} &= 0 \\ \dfrac{\partial g}{\partial t} - a\dfrac{\partial g}{\partial x} &= 0. \end{cases} \qquad \begin{cases} \dfrac{\partial f}{\partial t} &= -\dfrac{1}{\varepsilon}(f - E_f(u)) \\ \dfrac{\partial g}{\partial t} &= -\dfrac{1}{\varepsilon}(g - E_g(u)). \end{cases}$$

In the sequel we assume $a \geq |F(u)|/u$ so that E_f and E_g are nonnegative quantities such that $E_f + E_g = u$.

Monte Carlo approximation

Let us assume f_0 and g_0 probability densities. Given a set of samples $(\xi_1^0, v_1^0), \ldots, (\xi_N^0, v_N^0)$, where $v_i \in \{-a, a\}$ characterize the samples of $f_0(x)$ from those of $g_0(x)$, we can sample directly from the exact solutions of the operator splitting steps

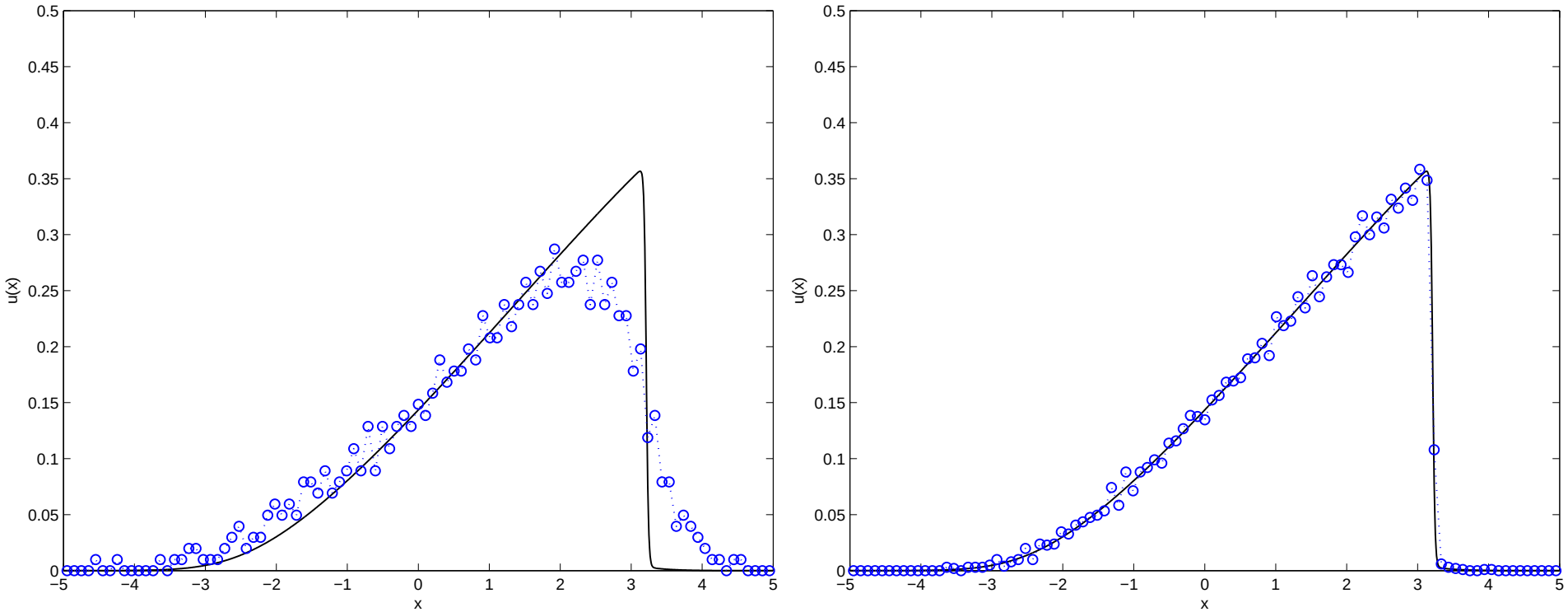
$$f^*(x,t) = f_0(x - at), \qquad f(x,t) = e^{-t/\varepsilon} f^*(x,t) + (1 - e^{-t/\varepsilon}) E_f(u^*(x,t)),$$
$$g^*(x,t) = g_0(x + at), \qquad g(x,t) = e^{-t/\varepsilon} g^*(x,t) + (1 - e^{-t/\varepsilon}) E_g(u^*(x,t)).$$

A new set of samples $(\xi_1, v_1), \ldots, (\xi_N, v_N)$ is obtained as follows[6].

1. First compute $\xi_i = \xi_i^0 + v_i^0 t, \quad i = 1, \ldots, N$
2. On a space grid of M points reconstruct $u^*(x_j, t)$, $j = 1, \ldots, M$.
3. In each space cell j given a sample (ξ_i, v_i^0) with probability $1 - e^{-t/\varepsilon}$ do the following
   - with probability $E_f(u^*(x_j,t))/u^*(x_j,t)$ set $v_i = a$
   - with probability $E_g(u^*(x_j,t))/u^*(x_j,t)$ set $v_i = -a$.
4. otherwise set $v_i = v_i^0$.

[6] L.P.,M.Seaid, '07

Burgers' equation



Solution at $t = 10$ for $F(u) = u^2/2$. Left: $N = 1000$ and $\Delta t = 0.1$. Right: $N = 10000$, $\Delta t = 0.01$. We take the limit case $\varepsilon = 0$ with $a = 1$. The solution is reconstructed on $M = 100$ grid points. The initial data is a Gaussian with zero mean and unit variance.

Final considerations

- Monte Carlo methods are a powerful tool for the approximation of PDEs in presence of a large number of variables or in complicate geometries.
- From the viewpoint of balance between accuracy and efficiency, roughly speaking, they satisfy

$$|\text{Error}| \approx \frac{1}{\sqrt{\text{Computational effort}}}.$$

- They are much less sensitive to many common problems to deterministic solvers like numerical dissipation, loss of physical properties like conservations and positivity, mesh constructions and of course the dimensionality of the problem.
- For non positive solutions it is usually possible to apply the methods introducing negative samples or equivalently particles with negative mass.
- In some circumstances we can speed up the convergence of the method using different techniques like importance sampling, control variates, antithetic variables and quasi Monte Carlo strategies.