Introduction to the Monte Carlo Methods

Anne GILLE-GENEST

February 5, 2003

Premia 5

Contents

Introduction

This part contains a general presentation to the Monte Carlo and Quasi-Monte Carlo simulation methods. These two types of methods are used to evaluate an integral as an expected value. We will consider the following problem

$$I = E_\mu[\psi(X)] = \int_E \psi(x) d\mu(x)$$

where ψ is some function on $E \in \mathcal{R}^n$ over $\mathcal{R}$ and $X = (X_1, \ldots, X_n)$ is a n-dimensional vector of random variables with law μ.
The first section is devoted to Monte Carlo Simulation, principle of estimation, variance reduction techniques and efficiency of the simulation.
In a second step, we give an introduction to Quasi-Monte Carlo methods.
The next section deals with the simulation of random variables. We present some algorithms, especially for uniform and gaussian variables. Finally we describe how to simulate diffusion processes. Such algorithms are necessary to price financial options with simulation methods.
The last section details how to use simulation methods.

1 Monte Carlo simulation

This section contains an introduction to Monte Carlo methods. A first point is devoted to the principle of simulation. In a second point, we are interested in variance reduction techniques applied to increase efficiency of the simulation. To compare various methods, we present a criterion of efficiency in a third point.
For a more detailed review on these methods, the reader can refer for instance to [9], [7], [5] or [2]. A reference for applications in Finance is [1].

1.1 Principle of the Monte Carlo Simulation

Monte Carlo Simulation is a general method for evaluating an integral as an expected value. It is based on the *Strong Law of Large Numbers* (LLN) and the *Central Limit Theorem*. It provides an unbiased estimator and the error on the estimate is controlled within a confidence interval.

We first recall these two limit theorems, and then we describe principle of the simulation. The last point is devoted to some properties of the estimate.

1.1.1 Limit theorems

X denotes a n-dimensional random vector, ψ is some function on $E \in \mathcal{R}^n$ over $\mathcal{R}$.

• **Strong Law of Large Numbers:** This theorem shows that the mean of $\psi(x_i)$ for a large sample converges to the expected value of ψ under an integrability condition.
If x_i are $i.i.d$ (independent and identically distributed) to X and if $E[|\psi(X)|] < +\infty$ then

$$\frac{1}{N}\sum_{i=1}^{N}\psi(x_i) \xrightarrow{a.s} E[\psi(X)]$$

• **Central Limit Theorem:** This theorem shows that the mean estimator converges in law to a gaussian standard distribution.
We note $\sigma^2 = Var[\psi(X)]$.
If $\sigma^2 < +\infty$ then

$$\frac{\sqrt{N}}{\sigma}\left(\frac{1}{N}\sum_{i=1}^{N}\psi(x_i) - E[\psi(X)]\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0,1)$$

1.1.2 Estimation principle

• An unbiased estimator of I for N trials with the Standard Monte Carlo method is defined by:

$$\theta_N = \frac{1}{N}\sum_{i=1}^{N}\psi(x_i)$$

with x_i $i.i.d$ to μ.
Simulation of sample $\{x_i\}$ according to μ is explained in the section about random variables simulation. It is important to remember that we need random numbers, generated from pseudo-random numbers generators.

• From the Law of Large Numbers we have that $\theta_N \xrightarrow{a.s} I$.

• Variance of the estimator is given as:

$$\sigma_N^2 = \frac{\sigma^2}{N}$$

and it is estimated by:

$$\widetilde{\sigma}_N^2 = \frac{1}{N-1}\left[\frac{1}{N}\sum_{i=1}^{N}\psi^2(x_i) - \theta_N^2\right]$$

Variance decreases to 0 when $N \to +\infty$. It means that the greater N is, the more accurate the estimator is. The speed of convergence of θ_N to I is $\frac{\sigma}{\sqrt{N}}$. It is not dependent on the dimension n.

• From the Central Limit Theorem, we can define a confidence interval $IC = [A, B]$ for the threshold $1-2\alpha$. It is such that $P(A < I < B) = 1-2\alpha$ and it is built as follows:

$$IC = [\theta_N - z_\alpha \sigma_N; \theta_N + z_\alpha \sigma_N]$$

where $z_\alpha = \Phi^{-1}(1-\alpha)$ and Φ^{-1} is the inverse cumulative distribution function of the standard gaussian law.
For instance, if the threshold is chosen to 95% then $\alpha = 2,5\%$ and $z_\alpha \approx 1.96$.

1.1.3 Properties

We briefly summarize some advantages and disadvantages of the Standard Monte Carlo method.

- **Advantages:**
  - This method does not require regularity or differentiability properties for the function ψ. Thus we can implement it very easily if we are able to generate the variable X according to μ.
  - The estimator is unbiased, that is $E[\theta_N] = I$.
  - Error on the estimate can be controlled by the Central Limit Theorem, and we can built a confidence interval.
  - Speed of convergence is independent on the dimension.

- **Disadvantages:**
  - We have to realize a lot of simulations to obtain an accurate estimator. Therefore computing time can be very high.

1.2 Variance Reduction Techniques

We saw that a disadvantage of the standard Monte Carlo Simulation is its required computing time. Thus we are now interested in Accelerated Monte Carlo Simulation.

To reduce computing time, we can use variance reduction techniques. The principle is to rewrite the parameter I in order to express it in function of a new random variable with smaller variance σ^2. Then we need a smallest number of iterations to obtain the same accuracy on the estimate. In the next section, we present a criterion of efficiency used in order to compare different accelerated methods.
We now briefly describe some of the major variance reduction techniques: importance sampling, stratified sampling, control variate, antithetics variates, conditional sampling. A description of this techniques can be found in [9].
Applying variance reduction techniques requiers some choices of parameters. Then it is not always obvious to determine the better technique, neither to implement it.

1.2.1 Conditional Sampling

This technique is based on the property of the conditional expectation.
If we note $X = (Y, Z)$ then we have:

$$\begin{aligned} I &= \int\int_E \psi(y,z)\mu(y,z)dydz \\ &= \int\int \psi(y,z)\mu_{Z|Y}(z)\mu_Y(y)dydz \\ &= E_Y\left[E_Z[\psi(Y,Z)|Y]\right] \end{aligned}$$

This method can be applied if we are able to calculate the marginal law $\mu_Y(y)$ and the conditional expectation $E_Z[\psi(Y,Z)|Y]$.
Then we just have to simulate y_i $i.i.d$ to Y and the estimator is expressed by:

$$\theta_N = \frac{1}{N}\sum_{i=1}^{N} E[\psi(y_i, Z)|y_i]$$

and

$$\sigma_N^2 = \frac{1}{N} Var\left[E_Z[\psi(Y,Z)|Y]\right]$$

Variance reduction with regard to standard Monte Carlo simulation is garanteed by the conditional Jensen inequality.
Applying this method leads in fact to reduce dimension of the initial model. We can express I as

$$I = E[h(Y)]$$

where h is the function obtained with $E_Z[\psi(Y,Z)|Y]$ and Y is a m-dimensional vector with $m < n$.

1.2.2 Importance Sampling

The basic idea of importance sampling consists in concentrating the distribution of the sample points in the most contributive parts of the space. For that, we introduce an new density ν which changes the initial density of X.

$$I = \int_E \psi(x)\frac{\mu(x)}{\nu(x)}d\nu(x) = E_\nu\left[\psi(X)\frac{\mu(X)}{\nu(X)}\right]$$

- We obtain the following estimator:

$$\theta_N = \frac{1}{N}\sum_{i=1}^{N}\psi(x_i)\frac{\mu(x_i)}{\nu(x_i)}$$

with x_i *i.i.d* to ν.

- Variance of the estimator is expressed as:

$$\sigma_N^2 = \frac{1}{N}Var_\nu\left[\psi(X)\frac{\mu(X)}{\nu(X)}\right]$$

- ν is named the *importance function*. It must verify that $\nu(x) > 0$, $\forall x \in E$ such that $\psi(x)\mu(x) > 0$. Otherwise the estimator we obtain is not unbiased.

Variance reduction with regard to standard Monte Carlo simulation is not garanteed by this method. It depends on the choice of ν, which is not an easy step. However, the minimum of the variance is reached for the following importance density μ^* called the *optimal density*.

$$\mu^* = \frac{|\psi(x)|\mu(x)}{\int|\psi(y)|\mu(y)dy}$$

Usually this density is unknown and it contains the term I as soon as $\psi > 0$.

1.2.3 Stratified Sampling

For stratified sampling, we separate the space E into M disjoints subregions (*the strata*) E_i. Each subintegral on E_i can be estimated separately by the standard Monte Carlo method. The final estimator is obtained by a recomposition of the different estimators.

$$\begin{aligned} I &= \sum_{i=1}^{M}\int_{E_i}\psi(x)\mu(x)dx \\ &= \sum_{i=1}^{M}P_i\int_{E_i}\psi(x)\frac{\mu(x)}{P_i}dx \\ &= \sum_{i=1}^{M}P_iI_i \end{aligned}$$

- Each integral I_i is estimated by a standard simulation with N_i samples in E_i. P_i denotes the weight of the stratum E_i.

$$\theta_N = \sum_{i=1}^{M} P_i \frac{1}{N_i} \sum_{k=1}^{N_i} \psi(x_k^i)$$

with x_k^i $i.i.d$ to $\frac{\mu(x)}{P_i}$ over E_i.

- Variance of the estimator is given by:

$$\sigma_N^2 = \sum_{i=1}^{M} \frac{P_i^2}{N_i} Var\left[\psi(X^i)\right] = \sum_{i=1}^{M} \frac{P_i^2 \sigma_i^2}{N_i}$$

As for importance sampling, variance reduction is not garanteed by stratified sampling. Accuracy of the estimate depends on the choice of the strata and the sample size N_i on each of them.

An optimal choice for the N_i under the condition $\sum_{i=1}^{M} N_i = N$ is given by:

$$N_i = N \frac{P_i \sigma_i}{\sum_{j=1}^{M} P_j \sigma_j}$$

where σ_i denotes the standard error on E_i.
In this case, the variance of the estimator is:

$$\sigma_N^2 = \frac{1}{N} \left[\sum_{j=1}^{M} P_i \sigma_i \right]^2$$

Note that this choice can not be found a priori because parameters σ_i are unknown. But it is possible to estimate these values with a first simulation and to realize a fast optimal choice in a second step.

Remarks:
- Stratified sampling can be seen as a particular case for importance sampling where the importance density is a decomposition of the initial density over its support.
- Strata are usually built from intervals for each variable: bounds are defined with values or quantiles in the distribution. It is necessary to be able to simulate variable according to the law on each stratum.

1.2.4 Control Variables

Principle of this method is to introduce an other model for which we have an explicit solution and to estimate the difference between our first parameter I and the new one.

$$\begin{aligned} I &= E[\psi(X)] \\ &= E[\psi(X) - \xi(X)] + E[\xi(X)] \end{aligned}$$

where ξ is a function such that $E[\xi(X)] = m$ is known.

An unbiased estimator of I with N trials is defined by:

$$\theta_N = \frac{1}{N} \sum_{i=1}^{N} (\psi(x_i) - \xi(x_i)) + m$$

with x_i $i.i.d$ to μ.
Variance of the estimator is given as:

$$\begin{aligned} \sigma_N^2 &= \tfrac{1}{N} Var[\psi(X) - \xi(X)] \\ &= \tfrac{1}{N} \{Var[\psi(X)] + Var[\xi(X)] - 2Cov(\psi(X), \xi(X))\} \end{aligned}$$

Variance reduction with regard to standard Monte Carlo simulation is not garanteed by this method. To decrease the variance, functions ψ and ξ must have a large positive correlation. It implies a specific choice for the control variate ξ.

1.2.5 Antithetics Variables

Principle of antithetic variables is to introduce some correlations between the terms of the estimate.
When simulation is done by the inverse cumulative distribution function (see the section about random variable simulation), we use uniform numbers u_i on $[0, 1]$. For this method, we use each u_i twice, as u_i and $1 - u_i$. These both variables have same law but are not independent. We note x_i and x_i' the variables generated from u_i and $1 - u_i$ respectively.

- An unbiased estimator of I with N trials is defined by:

$$\theta_N = \frac{1}{2N} \sum_{i=1}^{N} (\psi(x_i) + \psi(x_i'))$$

with x_i $i.i.d$ to μ.

- Variance of the estimator is given as:

$$\sigma_N^2 = \frac{1}{2N}\left(Var[\psi(X)] + Cov[\psi(X), \psi(X')]\right)$$

The following theorem give sufficient conditions to obtain a variance reduction with this method.

Theorem: If ψ is a monoton, continue, derivable function then

$$(\sigma_N^{ant})^2 \leq \frac{1}{2}(\sigma_N^{std})^2$$

Factor $1/2$ is due to the sample size for the antithetic method: in fact, the estimator contains $2N$ terms.

1.3 Efficiency of the Monte Carlo methods

We previously saw that Monte Carlo simulation requieres large sample size to obtain an accurate estimate and thus computing time can be very high. To accelerate the simulation, variance reduction techniques can be implemented. But to measure their performance, we need to take into account both the variance of the estimate and its computing time. In fact it is useless to decrease variance while increasing too much computing time because of a more complicated algorithm. What is interesting is which accuracy in what time we have for a given method. So we need a criterion which determines whether a variance reduction technique is better than an other one.

We introduce a criterion to compare the efficiency of the various simulation methods, standard simulation or with variance reduction techniques. This criterion takes into account the computing time required by the simulation for each method.

• Efficiency of the method j with regard to the method i is defined by:

$$\varepsilon(i,j) = \frac{\sigma_{N_i}(i)}{\sigma_{N_j}(j)}\sqrt{\frac{t_{N_i}(i)}{t_{N_j}(j)}}$$

where N, t_N, σ_N respectively denote sample size, computing time and standard error of the estimate from N simulations. This value seems to be dependent on sample sizes N_i and N_j, but it is not if we assume that computing

time is proportional to the sample size, that is there exists a factor k such that $t_{N_i}(i) = k_i N_i$. This hypothesis is very realistic. Then:

$$\varepsilon(i,j) = \frac{\sigma_i}{\sigma_j}\sqrt{\frac{k_i}{k_j}}$$

To obtain this formulation, we just use that $\sigma^2_N(i) = \frac{\sigma^2(i)}{N_i}$ where $\sigma^2(i)$ is the variance of the estimated function for the method i.
k exprimes complexity of the algorithm for the considered method.

• The value of $\varepsilon(i,j)$ can not be calculated, unless we know variances $\sigma^2(i)$ and $\sigma^2(j)$. This is not the case in general: when we estimate $I = E[\psi(X)]$ we don't know the parameter $\sigma^2 = Var[\psi(X)]$ too. But we can obtain an estimation of this efficiency within the Monte Carlo simulation.

$$\widetilde{\varepsilon}(i,j) = \frac{\widetilde{\sigma}_{N_i}(i)}{\widetilde{\sigma}_{N_j}(j)}\sqrt{\frac{t_{N_i}(i)}{t_{N_j}(j)}}$$

and

$$\lim_{N_i, N_j \to +\infty} \widetilde{\varepsilon}(i,j) = \varepsilon(i,j)$$

where $\widetilde{\sigma}_{N_i}$ is the estimated standard error for the estimator with the method i.

• The method j is considered to be more efficient than the method i if $\varepsilon(i,j) \geq 1$. We obviously have $\varepsilon(i,i) = 1$.
For instance, if $\varepsilon(i,j) = 3$, it means that method i requires $9(= 3^2)$ times more time than method j to obtain the same accuracy. In other words, with the same computing time, standard error for method j is 3 times smaller than the one of method i.
The higher $\varepsilon(i,j)$ is, the more efficient is the method j with regard to the method i.

• To compare different methods, the criterion of efficiency can be used as follows:
We can choose a reference method i and compare several other methods with regards to i. For example, choose i as the standard Monte Carlo method.
If $\varepsilon(i,j) < \varepsilon(i,k)$, then the method k is more efficient than the method j.
Note that generally we only have estimations for efficiencies. We have to analyse results carefully, by taking into account sufficient large sample sizes to obtain good convergence of the estimates.

2 Quasi Monte Carlo Simulation

A first method to estimate the integral $\int \psi(x)dx$ by simulation is to use sequences $\{x_i\}$ of random variables obtained from (pseudo) random numbers generators and to give an approximation by $\frac{1}{N}\sum_{i=1}^{N}\psi(x_i)$. This approach is described in the Monte Carlo Simulation part and Pseudo-random numbers generators are discussed in a special part (Pseudo random generator).

We are now interested in another way to realize simulation. It deals with the Quasi-Monte Carlo Methods. We first have to rewrite I by $\int_{[0,1]^d} f(u)du = E[f(U)]$ with U uniform over $[0,1]^d$. Instead of using random number sequences $\{u_i\}$, we introduce the notion of low-discrepancy sequences $\{\xi_i\}$. Such sequences neither are random nor pseudo-random but deterministic and successive values are not independent. However they satisfy good properties of equidistribution on $[0,1]^d$ and we have that

$$\frac{1}{N}\sum_{i=1}^{N} f(\xi_i) \to \int_{[0,1]^d} f(u)du$$

We first present theoretical aspect of those sequences and justification of their use in simulation through the Koksma-Hlawka formula. Various low discrepancy sequences are described in an other part low discrepancy sequences. We explain their construction and discuss some of their properties, especially on their discrepancy.
General references about the Quasi-Monte Carlo simulation are [3], [8], [10], [6] or [4]. This list is not exhaustive. Specific references will be given in the following sections.
Description for the implemented sequences is given in the implemented part. C codes can be found in the source part.

2.1 Discrepancy, Variation, Koksma-Hlawka formula

To present low discrepancy sequences, we need to introduce some notations and definitions, especially the discrepancy and the variation.

Notations:
We note $[|0,x|] = \{y = (y_1,\dots,y_d) \in [0,1]^d, y < x\}$; we consider that $y \leq x$ if and only if for all $j = 1,\dots,d : y_j \leq x_j$.
$\Pi(x)$ denotes the volume of $[|0,x|]$. $\Pi(x) = x_1 \times \cdots \times x_d$.
We note $I^d = [0,1]^d$ the closed d-dimensional unit cube.

For $\xi = (\xi_n)_{n\geq 1}$ a sequence in I^d and $x \in I^d$, we note :

$$D_n(\xi, x) = \frac{1}{n}\sum_{i=1}^{n} 1_{[|0,x|]}(\xi_i) - \Pi(x)$$

Definitions:

- A sequence $(\xi_n)_n \geq 1$ is said to be *equidistributed* on $[0,1]^d$ if

$$\forall x \in [0,1]^d, \lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^{n} 1_{[|0,x|]}(\xi_i) = \Pi(x)$$

- The value $D_n^*(\xi)$ defined as:

$$D_n^*(\xi) = sup_{x\in I^d}|D_n(\xi, x)|$$

  is called the *star discrepancy* for the first n terms of the ξ sequence. The discrepancy is a very important notion for Quasi-Monte Carlo simulation. It measures how a given set of points is distributed in $I^d = [0,1]^d$. It can be viewed as a quantitative measure for the deviation from the uniform distribution.

- A sequence (ξ) is said to be a *low-discrepancy sequence* if its discrepancy satisfies $D_N = O(\frac{(\log N)^d}{N})$ or if it is asymptotically better than the one of a random sequence obtained from the law of iterated logarithm as $O((\frac{\log\log N}{N})^{\frac{1}{2}})$.

- The variation of a function f on I^d in the sens of *Vitali* is defined by :

$$V^d(f) = \sup_{p\in P(I^d)} \sum_{A\in p} |\Delta(f, A)|$$

  where $P(I^d)$ is the set of all partitions of I^d into subintervals, $p \in P(I^d)$ denotes a partition and $A \in p$ a subinterval.
  $\Delta(f, A)$ is the alternative sum of the values of f at the vertices of A.

$$V^d(f) = \int_{[0,1]^d} \left|\frac{\partial^d f}{\partial u_1 \dots \partial u_d}\right| du_1 \dots du_d$$

  if partial derivative is continuous on I^d.

- The variation of f on I^d in the sense of *Hardy and Krause* is defined by :

$$V(f) = \sum_{r=1}^{d} \sum_{1 \leq i_1 < \cdots < i_r \leq d} V^r(f; i_1, \ldots, i_r)$$

where V^r denotes the variation in the sense of Vitali on the restriction of f to the r dimensional face $\{(x_1, \ldots, x_d) \in I^d / x_k = 1 \; if k \notin \{i_1, \ldots, i_r\}\}$

2.2 Estimation

Construction of the estimator is based on the Koksma-Hlawka formula.
• **Koksma-Hlawka inequality**:
If f has bounded variation $V(f)$ on I^d in the sense of Hardy-Krause, then for any $\xi_1, \ldots, \xi_n \in [0,1]^d$, we have:

$$\forall n \geq 1, \left| \frac{1}{n} \sum_{i=1}^{n} f(\xi_i) - \int_{[0,1]^d} f(u)du \right| \leq V(f) D_n^*(\xi)$$

• **Properties of the estimator**:
- This formula gives an a priori deterministic bound for the error in the approximation of $\int_{[0,1]^d} f(x)dx$ by $\frac{1}{n}\sum_{i=1}^{n} f(\xi_i)$.
This error is expressed in term of the discrepancy of the sequence (independent of f) and the variation of the function f (independent of n). Nevertheless it is often difficult to calculate even to estimate the variation of f.
Through the Koksma-Hlawka inequality, we understand the need to have sequences with discrepancy D_N as small as possible.

- In opposition to Monte Carlo simulation, Quasi-Monte Carlo doesn't provide an confidence interval for the estimator. We cannot compute empirical variance of the sample because succesive terms are not independent. This is due to the construction of the low-discrepancy sequence.
- An other difference in comparison with Monte Carlo is that the convergence rate for QMC simulation depends on the dimension n of the considered model through discrepancy D_n. Values of discrepancy are given in the description part about low discrepancy sequences.

3 Simulation of random variables

To realize a Monte Carlo simulation, we need a sample of N random variables x_i i.i.d according to a same density, as described in the first section. In

this section, we present some general methods to obtain this random sample. First, we recall how to generate random numbers and then we detail the inverse method. The third point is devoted to simulation of gaussian variables. Simulation of several other random variables is described in Rubinstein [9].

3.1 Simulation of uniform variables

Simulation of most of the random variables requiers uniform variables over $[0, 1]$, that is random numbers.
Simulation of random numbers is well described in the two parts: Pseudo random generator and low discrepancy sequences.

Remarks:
- Pseudo random generators deal with Monte Carlo simulation, while low discrepancy sequences with Quasi-Monte Carlo Simulation.
- To obtain independent random numbers for a multi-dimensional Quasi-Monte Carlo simulation, it is necessary to consider a multi-dimensional low-discrepancy sequence. In return, you can take successive values from the same pseudo-random generator for a multi-dimensional Monte Carlo estimation, there is theoretically no problem with independence.

An uniform variable over $[a, b]$ is obtained as $a + (b - a)u$ where u is uniform over $[0, 1]$.

3.2 Inverse method

Inverse method is a general approach to generate a variable X according to a given law. For some distributions, it is the easiest way for simulation.
Simulation of a variable X with probability density function f and cumulative distribution function F can be made by the following way:
- First, simulate a variable u uniformly distributed on $[0, 1]$, that is a random number (from one of the pseudo-random numbers generators or one of the quasi-random number generators).
- And finally take $x = F^{-1}(u)$.
Applying this method supposes that the inverse cumulative distribution F^{-1} is well-known.

3.3 Simulation of Gaussian standard variables

First of all, we present two direct methods to generate standard Gaussian variable: Box-Muller transformation and Gauss-Abramowitz method. The

third point is interested in simulating gaussian variable with the inverse method.
These methods are implemented in PREMIA. You can reach algorithms with links C Code.

- **Box-Muller transformation**: C Code
  If (u, v) is uniformly distributed on $[0, 1]^2$ then x and y defined by:

$$x = \sqrt{-2 \log u} \sin(2\pi v)$$
$$y = \sqrt{-2 \log u} \cos(2\pi v)$$

  are distributed as independent standard gaussians.

  From two uniform random numbers, we obtain two simulations of a standard gaussian variable.

- **Gauss Abramowitz method**: C Code
  This simulation of a Gaussian standard variable is based on a rejection method (see Rubinstein).
  If (u, v) is uniformly distributed on $[-1, 1]^2$ such that $R^2 = u^2 + v^2 <$ 1the unit disk then x and y defined by:

$$x = \sqrt{\frac{-2 \log R^2}{R^2}} u$$
$$y = \sqrt{\frac{-2 \log R^2}{R^2}} v$$

  are distributed as independent standard gaussians.

  As for the Box-Muller method, from two uniform random numbers, we obtain two simulations of a standard gaussian variable.

  Remark on the rejection principle: We need to simulate a point uniformly inside the unit circle. The easiest is to simulate (u, v) uniformly on $[-1, 1]^2$ until the point falls in the circle (until we don't reject it), that is (u, v) such that $u^2 + v^2 < 1$.
  This method is efficient because the rejection rate is rather small. It equals the square area over the unit disk area, that is $\frac{4}{\pi} \sim 1, 3$.

- **Remarks:** Such methods require two **independent** random numbers to obtain two gaussian variables.
  - They can be used when random numbers are generated from pseudo-random generators, because there is no problem of independence between u and v. It deals with the case of Monte Carlo simulation.

- These methods must not be used when random numbers u and v are generated from two successive values of a one-dimensional low-discrepancy sequence, because values are not independent in this case. To apply one of these algorithms for Quasi Monte Carlo simulation, you should generate u and v independently, that is necessary from two different sequences or a two-dimensional sequence. In PREMIA, in the case of Quasi-Monte Carlo simulation, gaussian variables are generated with an inverse method (described in a next point).

- **Inverse method for Gaussian standard variables**
The inverse cumulative distribution Φ^{-1} for a gaussian standard variable has not an explicit form. Thus to use the inverse method to simulate gaussian variable we need an approximation of Φ^{-1}. We present two approaches.
- Function Inverse-erf (C Code)
We consider x uniform on $[0,1]$.
If $x > 0.5$, let $t = \sqrt{-2\log(1-x)}$, then we have the aproximation:

$$\Phi^{-1}(x) \simeq t - \frac{c_0 + t(c_1 + tc_2)}{(1 + t(d_1 + t(d_2 + td_3)))}$$

If $x \leq 0.5$, let $t = \sqrt{-2\log(x)}$ and then:

$$\Phi^{-1}(x) \simeq \frac{c_0 + t(c_1 + tc_2)}{(1 + t(d_1 + t(d_2 + td_3)))} - t$$

with the following constants:

$$\begin{aligned} c_0 &= 2.515517; \\ c_1 &= 0.802853; \\ c_2 &= 0.010328; \end{aligned}$$

$$\begin{aligned} d_1 &= 1.432788; \\ d_2 &= 0.189269; \\ d_3 &= 0.001308; \end{aligned}$$

- Moro Method. (Function Inverse-erf-Moro. C Code)
For x uniform on $[0,1]$, $r = \Phi^{-1}(x)$ is obtained as follows:
Let $u = x - 0.5$.
If $|u| < 0.42$ then:

$$r = u\frac{\sum_{n=0}^{3} a_n u^{2n}}{1 + \sum_{n=0}^{3} b_n u^{2(n+1)}}$$

else let $z = \log(-\log(1-x))$, and compute:

$$r = \sum_{n=0}^{8} c_n z^n$$

If $u > 0$ take r.
And if $u < 0$ take $-r$.

The considered constants are given by:

$$\begin{aligned} a_0 &= 2.50662823884, \\ a_1 &= -18.61500062529, \\ a_2 &= 41.39119773534, \\ a_3 &= -25.44106049636; \end{aligned}$$

$$\begin{aligned} b_0 &= -8.47351093090, \\ b_1 &= 23.08336743743, \\ b_2 &= -21.06224101826, \\ b_3 &= 3.13082909833; \end{aligned}$$

$$\begin{aligned} c_0 &= 0.337475482276147, \\ c_1 &= 0.9761690190917186, \\ c_2 &= 0.1607979714918209, \\ c_3 &= 0.0276438810333863, \\ c_4 &= 0.0038405729373609, \\ c_5 &= 0.0003951896511919, \\ c_6 &= 0.0000321767881768, \\ c_7 &= 0.000002888167364, \\ c_8 &= 0.0000003960315187; \end{aligned}$$

Remark: For Quasi-Monte Carlo simulation in PREMIA, we use an inverse method to simulate gaussian variables. See previous remark on independence in direct methods for gaussian simulation.

3.4 Simulation of gaussian vectors

Simulation of a N-dimensional Gaussian vector Γ with zero mean and a covariance matrix C is done by the following way:
- First we compute the lower triangulary matrix A obtained with the Cholesky decomposition of C, that is such that $C = AA^t$. We have :

$$\begin{aligned} A_{ii} &= \sqrt{C_{ii} - \textstyle\sum_{k=1}^{i-1}(A_{ik})^2} \\ A_{ji} &= \frac{C_{ij} - \sum_{k=1}^{i-1} A_{ik}A_{jk}}{A_{ii}} \qquad \text{for} \quad j = i+1, \ldots, N \end{aligned}$$

- Then we generate N independent gaussian standard variables g_i with one of the methods previously described. We note $G = (g_1, \ldots, g_N)$.
- Finally we compute $\Gamma = AG$. Γ is distributed as $\mathcal{N}(0, C)$.

Remark: this method will be used in case of simulation of correlated brownian motions, for instance to price options in a two dimensional Black and Scholes model.

3.5 Approximation of the cumulative distribution function Φ

We now give two approximations for the cumulative distribution function Φ of a gaussian standard variable. It is very often used to compute price or delta for various options.

- **Standard algorithm**
  Let $t = \frac{1}{1+px}$, then we have:

$$\Phi(x) = \begin{cases} 1 - \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})(b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5) & \text{if} \quad x \geq 0 \\ \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})(b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5) & \text{if} \quad x < 0 \end{cases}$$

  with the following constant values:

$$\begin{aligned} p &= 0.2316419; \\ b_1 &= 0.319381530; \\ b_2 &= -0.356563782; \\ b_3 &= 1.781477937; \\ b_4 &= -1.821255978; \\ b_5 &= 1.330274429; \end{aligned}$$

- **Moro algorithm**
  The approximation is given by:
  For $x \geq 0$

$$\Phi(x) = \begin{cases} 0.5 + x \left(\frac{\sum_{n=0}^{2} a_n x^{2n}}{1+\sum_{n=1}^{3} b_n x^{2n}} \right) & \text{when} \quad 0 \leq x \leq 1.87 \\ 1 - \left(\frac{\sum_{n=0}^{2} c_n x^n}{1+\sum_{n=0}^{3} d_n x^n} \right)^{16} & \text{when} \quad 1.87 < x < 6 \\ 1 & \text{when} \quad x \geq 6 \end{cases}$$

  For $x < 0$, use the formula $\Phi(x) = 1 - \Phi(-x)$ where $\Phi(-x)$ is computed with the previous formula.

In the approximation, we use the following constants:

$$\begin{aligned} a_0 &= 0.0398942270991, \\ a_1 &= 0.020133760596, \\ a_2 &= 0.002946756074, \end{aligned}$$

$$\begin{aligned} b_1 &= 0.217134277847, \\ b_2 &= 0.018576112465, \\ b_3 &= 0.000643163695, \end{aligned}$$

$$\begin{aligned} c_0 &= 1.398247031184, \\ c_1 &= -0.360040248231, \\ c_2 &= 0.02271976588, \end{aligned}$$

$$\begin{aligned} d_0 &= 1.460954518699, \\ d_1 &= -0.305459640162, \\ d_2 &= 0.038611796258, \\ d_3 &= -0.003787400686, \end{aligned}$$

The second approximation is more accurate than the first one.

4 Simulation of Diffusions

This section is especially devoted to simulation of diffusion processes. Reference to these de Olivier Faure et livre Lapeyre. It is necessary to know how to simulate such processes, particularly for path-dependent options pricing. We first present the easiest diffusion process: Brownian motion. After that we consider diffusion in Black and Scholes model and then general diffusion processes for which we need an approximation scheme.

4.1 Simulation of Brownian Motion

• **Definition:** A **Brownian motion** is a continuous adapted process $B = \{B_t, \mathcal{F}_t; 0 \leq t < \infty\}$, defined on some probability space $(\Omega, \mathcal{F}, P)$, with the properties that $B_0 = 0$ a.s and for $0 \leq s < t$, the increment $B_t - B_s$ is independent of $\mathcal{F}_s$ and is normally distributed with mean zero and variance $t - s$.

• **Simulation of B_t:**
Simuation of B_t is an easy step because we have that $\mathcal{L}(B_t) = \mathcal{N}(0, t)$
- first generate a gaussian standard variable g
- and then compute B_t as $\sqrt{t}g$

• **Simulation (discretization) of a Brownian trajectory**, $0 \leq t \leq T$:
We now detailed two approaches for simulationg a Brownian path: the Forward one and the Backward one. Typically, for path-dependent options we have to simulate B over $\mathcal{T} = \{t_k; k = 0, \ldots, M, t_0 = 0, t_M = T\}$.

- Forward Simulation of B_t over $\mathcal{T}$ is given by:

$$\begin{aligned} &B(0) = 0, \\ &B(t_{k+1}) = B(t_k) + \sqrt{t_{k+1} - t_k} g_k, \end{aligned}$$

where $(g_1, \ldots, g_M)$ are independent gaussian standard variables.
If we use a discretization with evenly spaced intervals of size $h = \frac{T}{M}$, we have:

$$\begin{aligned} &B(0) = 0, \\ &B(t_{k+1}) = B(t_k) + \sqrt{h} g_k, \end{aligned}$$

- Backward simulation with Brownian Bridge
This other method is based on the following property for Brownian bridge:

$$\mathcal{L}(B_u, s < u < t | B_s = x, B_t = y) = \mathcal{N}\left(\frac{t-u}{t-s}x + \frac{u-s}{t-s}y, \frac{(t-u)(u-s)}{t-s}\right)$$

and particularly

$$\mathcal{L}(B_{\frac{t+s}{2}} | B_s = x, B_t = y) = \mathcal{N}\left(\frac{x+y}{2}, \frac{(t-s)}{4}\right)$$

This scheme consists in simulating B in a different order than with increasing times t_k.

$$\begin{aligned} &B(0) = 0, \\ &B(T) = \sqrt{T} g_1, \\ &B(\tfrac{T}{2}) = \tfrac{B(0)+B(T)}{2} + \sqrt{\tfrac{T}{4}} g_2, \\ &B(\tfrac{T}{4}) = \tfrac{B(0)+B(\frac{T}{2})}{2} + \sqrt{\tfrac{T}{8}} g_3, \\ &B(\tfrac{3T}{4}) = \tfrac{B(\frac{T}{2})+B(T)}{2} + \sqrt{\tfrac{T}{8}} g_4, \\ &\vdots \end{aligned}$$

where $(g_1, \ldots, g_M)$ are independent gaussian standard variables.

For this algorithm, we have to choose M as a power of 2. The first step

is directly for 0 to T. Intermediates steps are filled by taking successive subdivisions of the time intervals into halves.
It can be adapted with subdivisions of different length by considering the conditional law of brownian bridge between s and t.

- Remarks on this two schemes for MC and QMC simulations:
We need a vector of size M of independent gaussian variables.
For MC, these M variables can be simulated from the same pseudo random numbers generator. However, for a QMC simulation we need to use a M-dimensional low-discrepancy sequence to keep independence property.

4.2 Simulation of diffusion in Black and Scholes model

In the Black and Scholes model (reference), underlying asset price S_t follows the diffusion:

$$dS_t = rS_t dt + \sigma S_t dB_t$$

and then the price is a geometric Brownian process:

$$S_t = S_0 \exp\left(\left(r - \frac{\sigma^2}{2}\right)t + \sigma B_t\right)$$

In this particular case for which we have an explicit solution of the diffusion process, simulation of price paths is based on simulation of Brownian motion described in the last section.
As for Brownian path simulation, we present the forward and backward approaches.

• **Forward simulation:**
With scheme $\mathcal{T}$ for the discretization, we have:

$$S_{t_{k+1}} = S_{t_k} \exp\left((r - \frac{\sigma^2}{2})(t_{k+1} - t_k) + \sigma B_{t_{k+1}-t_k}\right)$$

and for a discretization with evenly spaced intervals of size h, we simply have:

$$S_{t_{k+1}} = S_{t_k} \exp\left((r - \frac{\sigma^2}{2})h + \sigma\sqrt{h}g_k\right)$$

• **Backward simulation:**
To construct this scheme, we use Backward simulation for Brownian path

described int the previous point. To expressed this scheme, we note $y_t = \log S_t$, that is $y_t = y_0 + \left(r - \frac{\sigma^2}{2}\right) t + \sigma B_t$.

$$\begin{aligned}
y_T &= y_0 + (r - \tfrac{\sigma^2}{2})T + \sigma\sqrt{T} g_1, \\
y_{\frac{T}{2}} &= \tfrac{y_0 + y_T}{2} + \sigma\sqrt{\tfrac{T}{4}} g_2, \\
y_{\frac{T}{4}} &= \tfrac{y_0 + y_{\frac{T}{2}}}{2} + \sigma\sqrt{\tfrac{T}{8}} g_3, \\
y_{\frac{3T}{4}} &= \tfrac{y_{\frac{T}{2}} + y_T}{2} + \sigma\sqrt{\tfrac{T}{8}} g_4, \\
&\vdots
\end{aligned}$$

Without problem, we endly take $S_{t_k} = \exp y_{t_k}$

4.3 Simulation of diffusion: Euler and Milshtein scheme

We consider the general diffusion process:

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t$$

If we don't have any explicit solution for X_t (like for Black and Scholes model), we have to use approximation schemes with a discretization of the process. The both most known schemes are Euler and Milshtein. They both take into account a discretization $\mathcal{T}$ of length h.

• The **Euler approximation scheme** for this diffusion is expressed as:

$$X_{t_{k+1}} = X_{t_k} + b(X_{t_k})h + \sigma(X_{t_k})(B_{t_{k+1}} - B_{t_k})$$

Simulation is obtained with a forward algorithm by:

$$X_{t_{k+1}} = X_{t_k} + b(X_{t_k})h + \sigma(X_{t_k})\sqrt{h} g_k$$

for $k = 0, \ldots, M-1$.

• The Milshtein approximation scheme for this diffusion is given by:

$$\begin{aligned}
X_{t_{k+1}} = \; & X_{t_k} + \left(b(X_{t_k}) - \tfrac{1}{2}\sigma'(X_{t_k})\sigma(X_{t_k})\right) h + \sigma(X_{t_k})(B_{t_{k+1}} - B_{t_k}) \\
& + \tfrac{1}{2}\sigma'(X_{t_k})\sigma(X_{t_k})(B_{t_{k+1}} - B_{t_k})^2
\end{aligned}$$

Simulation is obtained with a forward algorithm by:

$$X_{t_{k+1}} = X_{t_k} + \left(b(X_{t_k}) - \frac{1}{2}\sigma'(X_{t_k})\sigma(X_{t_k})\right) h + \sigma(X_{t_k})\sqrt{h} g_k + \frac{1}{2}\sigma'(X_{t_k})\sigma(X_{t_k}) h g_k^2$$

for $k = 0, \ldots, M-1$.

References

[1] E. PARDOUX B. LAPEYRE and R. SENTIS. *Methodes de Monte Carlo pour les equations de transport et de diffusion*. Number 29. Mathematiques Applications, 1998. 2

[2] COCHRAN. *Sampling Techniques*. Wiley Series in Probabilities and Mathematical Statistics, 1977. 2

[3] H.NIEDERREITER. Points sets ans sequences with small discrepancy. *Monatsh.Math*, 104:273–337, 1987. 11

[4] INRIA. *Probabilites numeriques. Chap 1: suites a discrepance faible et integration numerique*. 11

[5] HAMMERSLEY J. and HANDSCOMB D. *Monte Carlo Methods*. Chapman and Hall, London, 1979. 2

[6] W.J. MOROKOFF and R.E. CAFLISH. Quasi-random sequences and their discrepancies. *SIAM, Journal of Scientific Computing*, 15(6):1251–1279, nov 1994. 11

[7] N.BOULEAU. *Probabilits de l'ingnieur, variables alatoires et simulation*. Ed Hermann, 1986. 2

[8] H. NIEDERREITER. Random number generation and quasi-monte carlo methods. *SIAM*, 1992. 11

[9] R.Y. RUBINSTEIN. *Simulation and the Monte Carlo Method*. Wiley Series in Probabilities and Mathematical Statistics, 1981. 2, 5, 14

[10] H. NIEDERREITER P.J.S. SHIUE. Monte carlo and quasi-monte carlo methods in scientific computing. *Lecture Notes in Statistics,Ed Springer*, 106, 1995. 11