Monte Carlo methods for financial applications

Antonino Zanette
University of Udine
antonino.zanette@uniud.it

Uniform distribution

$X \sim Un(a,b)$ if the density function is given by :

$$f_X(x) = \frac{1}{b-a} \mathbb{1}_{(a,b)}(x)$$

The distribution function $F_X(x)$ of X is given by :

$$F_X(x) = \begin{cases} 0 & \text{if} \quad x < a \\ \frac{x-a}{b-a} & \text{if} \quad a \leq x < b \\ 1 & \text{if} \quad x \geq b \end{cases}$$

Remark

If $X \sim Un(a,b)$ then X has expectation $\mathbb{E}(X) = \frac{a+b}{2}$ and variance $Var(X) = \frac{(b-a)^2}{12}$.

Remark

If $U \sim Un(0,1)$ and $Z = a + (b-a)U$. Then $Z \sim Un(a,b)$.

Standard Normal distribution

$X \sim N(0,1)$ if

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \qquad x \in \mathbb{R}$$

The distribution function $F_X(x)$ is given by :

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt$$

If $X \sim N(0,1)$ then X has expectation $\mathbb{E}(X) = 0$ and variance $Var(X) = 1$.

The moment generating function of $X \sim N(0,1)$ is given by :

$$M_X(t) = \mathbb{E}(e^{tX}) = e^{\frac{1}{2}t^2}$$

The characteristic function is :

$$\phi_X(t) = \mathbb{E}(e^{itX}) = e^{-\frac{1}{2}t^2}$$

General normal distribution

Consider $Z \sim N(0,1)$, $\mu \in \mathbb{R}$, $\sigma \neq 0$ and X the transformed random variable

$$X = \mu + \sigma Z$$

Then $X \sim N(\mu, \sigma^2)$ with $\mathbb{E}(X) = \mu$ e $Var(X) = \sigma^2$ and density

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The moment generating function of $X \sim N(\mu, \sigma^2)$ is given by :

$$M_X(t) = \mathbb{E}(e^{tX}) = e^{\mu t + \frac{\sigma^2}{2} t^2}$$

The characteristic function is :

$$\phi_X(t) = \mathbb{E}(e^{itX}) = e^{i\mu t - \frac{\sigma^2}{2} t^2}$$

Log-Normal distribution

$Y \sim LN(\mu, \sigma)$ if $\log Y \sim N(\mu, \sigma^2)$

Then

$$Y = e^X \qquad with \quad X \sim N(\mu, \sigma^2)$$

The density of Y is given by:

$$f_Y(y) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}\, y} e^{-\left(\frac{(\log y - \mu)^2}{2\sigma^2}\right)} & \text{if} \quad y > 0 \\ 0 & \text{if} \quad y \leq 0 \end{cases}$$

The expectation of Y log-normal is given by

$$\mathbb{E}(Y) = e^{\mu + \frac{\sigma^2}{2}}$$

and the variance is given by

$$Var(Y) = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$$

Simulation methods of classical law

Suppose that we known how to build a sequence of numbers $(U_n)_{n\geq 1}$ that is the realization of a sequence of a indipendent, identically distribuited uniform random variables on $[0, 1]$.

Then we look for a function $(u_1, \ldots, u_p) \mapsto F(u_1, \cdots, u_p)$ such that $F(U_1, \cdots, U_p)$ follows the distribution of X.

The sequence of random variable $(X_n)_{n\geq 1}$ where $X_n = F(U_{(n-1)p+1}, \cdots, U_{np})$ is then a sequence of i.i.d. random variables following the distribution of X.

The sequence $(U_n)_{n\geq 1}$ is obtained in practice from successive calls to a pseudo-random number generator. Most languages available on computers provide such a random function.

Examples

- Discrete : Bernoulli and Binomial laws
- Continous : Normal, Log-Normal, Exponential laws
- General method : Inverse transform method

Simulation of a uniform distribution on $[0, 1]$

To generate a deterministic sequence which "looks like" indipendent r.v. uniformly distribuited on $[0, 1]$ the simplest methods are the congruential methods.

They are defined through the following integers chosen cautiosly in order to obtain satisfactory characteristics for the sequence:

- a the multiplicative term.
- b
- m the order of the congruence or the period lenght.
- x_0 the seed of the generator.

A pseudo random sequence $(x_n)_{n\geq 0}$ of integers between 0 et $m-1$ is generated as follows :

$$\begin{cases} x_0 = \text{intial value} \in \{0, 1, \cdots, m-1\} \\ x_{n+1} = T(x_n) = ax_n + b \text{ (modulo } m) \end{cases}$$

This method enables us to simulate pseudo-random integers between 0 and $m-1$;
to obtain a random real-valued number between 0 and 1, we can divide the random integer by m

$$u_{n+1} = \frac{x_{n+1}}{m}.$$

Observe that a pseudo-random number generator consists in a completely deterministic algorithm. Such an algorithm produces sequences which statistically behaves (almost) like sequences of i.i.d. uniformly distribuited random variables.

Example

$$\begin{cases} x_0 = \text{initial value } \in \{0, 1, \cdots, m-1\} \\ x_{n+1} = T(x_n) = ax_n + b \text{ (modulo } m) \end{cases}$$

with a, b, m integers.
Consider $a = 6, b = 0, m = 11, x_0 = 1$

$$x_1 = 6 \text{ (modulo } 11) = 6$$

$$x_2 = 36 \text{ (modulo } 11) = 3$$

Then we obtain the sequence

$$1, 6, 3, 7, 9, 10, 5, 8, 4, 2, 1, 6, ..$$

Since a computer can represent only a finite number of values, any recurrence relation of this form will return to a previous x_i and then repeat itself.

Remark

All the possible values are generated before the sequence repeats itself.
Generator of this type are said to have full period.

Example

$$\begin{cases} x_0 = \text{initial value} \in \{0, 1, \cdots, m-1\} \\ x_{n+1} = T(x_n) = ax_n + b \text{ (modulo } m) \end{cases}$$

Consider $a = 3, b = 0, m = 11, x_0 = 1$

Then we obtain the sequence

$$1, 3, 9, 5, 4, 1, 3, ..$$

Remark

This generator dont have full period. The multiplicative term $a = 3$ produce only 5 values before the sequence repeats itself.

Period lenght

- We would like to be able to generate a lot of distinct values before the sequence starts repeating itself.
- We would like to find a generator with full period with large m.
- But choosing large m is not sufficient to guarantee this property.
- So we have to choose a, b in order to have full period property with large m.

For example, Park-Miller propose a generator with $m = 2^{31}$:

$$\begin{cases} a & = & 16807 \\ b & = & 1 \\ m & = & 2147483647 \end{cases}$$

There is no theoretical criterion which ensures that a pseudo random generator is statistically accettable.
Such a property is established on the basis of empirical tests.

For example, one builds a sample from successive calls to the generator, and one then applies the Chi-square test or the Kolmogorov-Smirnov test in order to test wheter one can reasonably acccept the hypothesis that the sample results from i.i.d. uniformly distribuited random variables.

Simulation of Bernoulli law

$X \sim B(p)$ is

$$P(X = 1) = p$$

$$P(X = 0) = 1 - p$$

Let U be uniform $[0, 1]$, then

$$X = \mathbf{1}_{\mathrm{U} \leq \mathrm{p}}$$

have Bernoulli law.
In fact

$$P(X = 1) = P(0 < U < p) = F_U(p) - F_U(0) = p$$

$$P(X = 0) = P(p < U < 1) = F_U(1) - F_U(p) = 1 - p$$

because $F_U(u) = u$ with $u \in [0, 1]$

Simulate a trajectory in the CRR model

Dynamic hedging algorithm in the CRR model

```
function [S]=main()
  R=0.02;
  S0=40;
  T=10;
  K=40;
  N = 10;
  down =0.9;
  up =1.1;
   // Historical probability
  p=0.5;
  S=trajectory(N,up,down,R,S0,p);
endfunction
S=main();
```

```
//Simulation of Bernoulli law
function [ber]=bernoulli(p,down,up)
z=rand();
if z<=p then ber=up;
else ber=down;
end
endfunction

//Simulate a trajectory in the CRR model
//under Historical probabiltiy
function [S]=trajectory(N,up,down,R,S0,p)
  S=zeros(1,N+1);
  S(1)=S0;
  S(2) = S0*bernoulli(p,down,up);
  for n=3:N+1
     S(n)=S(n-1)*bernoulli(p,down,up);
  end;
endfunction
```

Simulation of Binomial Law

- $X \sim Bin(n, p)$

$$P(X = k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k}, k = 0, 1, ., n$$

- Let $Z_1, .., Z_n$ be Bernoulli r.v. with parameter $p \in [0, 1]$. Then

$$X = \sum_i^n Z_i \sim Bin(n, p)$$

- Simulation

$$X = \sum_i^n Z_i(U_i) \sim Bin(n, p)$$

Example

- Let us consider a discrete r.v X

$$P(X = x_i) = p_i, i = 1, .., n$$

- If $U \sim U(0, 1)$ then

$$X = x_i \mathbb{1}_{(p_1 + .. + p_{i-1} < U \leq p_1 + .. + p_i)}$$

Inverse distribution function method

Let X be a real random variable with continuous density function $f_X(x)$.
Let $F_X(x) = P(X \leq x)$ be its distribution function.

If $F_X(x)$ is continuous, increasing and non negative we can obtain $F_X^{-1}(u)$ such that

$$u = F_X(x) \Leftrightarrow x = F_X^{-1}(u)$$

Proposition

Let U be a uniformly distribuited in $[0, 1]$ random variable. Then

$$F_X^{-1}(U)$$

and X have the same law.

Remark

If F is piecewise costants function then we can define the pseudo inverse function

$$F_X^{-1}(u) = \inf\{x : F_X(x) \geq u\}$$

Example

- Exponential law λ with density function $f_X(x) = \lambda e^{-\lambda x} 1_{x \geq 0}$
- $F_X(x) = 1 - e^{-\lambda x}, x \geq 0$
- $F_X^{-1}(u) = -\frac{log(1-u)}{\lambda}$
- If $U \sim U(0,1)$ then

$$-\frac{log(1-U)}{\lambda}$$

has exponential law with parameter λ. We can simulate

$$-\frac{log(U)}{\lambda}$$

because U and $1 - U$ have the same law.

Simulation of a Gaussian distribution

- The most widely used simulation method of a Gaussian law is the Box-Muller method.

  Let (U_1, U_2) be two i.i.d. random variables following a uniform law on the interval $[0, 1]$, then :

  $$(\sqrt{-2\log(U_1)}\cos(2\pi U_2), \sqrt{-2\log(U_1)}\sin(2\pi U_2))$$

  are two indipendent standard Gaussian random variables.

- To simulate a Gaussian random variable with mean μ and variance σ^2, it suffices to set $X = \mu + \sigma g$, with $g \sim N(0, 1)$.

Algorithm $g \sim N(0,1)$

```
double gaussian()
{
  xs=random();
  ys=random();
  return sqrt(-2.0*log(xs))*cos(2.0*PI*ys);
}
```

Algorithm $g \sim N(m, \sigma^2)$

```
double gaussian_ms(double m,double sigma)
{
  return m+sigma*gaussian();
}
```

Proposition (Box-Muller)

Let r be an exponential random variable with parameter $\frac{1}{2}$ and θ Uniform on $(0, 2\pi)$ indipendent.

Then $X = \sqrt{r}\cos(\theta)$ and $Y = \sqrt{r}\sin(\theta)$ are standard indipendent normal random variables $N(0, 1)$.

Monte Carlo methods

Problem : Compute expectations of a random variable X with known density function $f_X(x)$

$$\mathbb{E}(X) = \int_{-\infty}^{+\infty} x f_X(x) dx$$

We need approximation in order to compute this integral.

The Monte Carlo method is on the possibility to generate a sequence of indipendent trials $X_1, \ldots, X_n, \ldots$ following the distribution of X.

Monte Carlo methods for the computation of $\mathbb{E}(X)$

- simulate a sequence i.i.d. $X_1, .., X_N$ with law X
- approssimate $\mathbb{E}(X)$ with the empirical mean :

$$\mathbb{E}(X) \approx \bar{E}_N = \frac{1}{N}(X_1 + \cdots + X_N).$$

The estimator is unbiased, that is $E[\bar{E}_N] = E(X)$.

- Why this method work? Strong law of Large Numbers
- How it works? Central Limit Theorem

Strong law of Large Numbers

Theorem

Let $(X_i, i \geq 1)$ be a sequence of i.i.d. random variables such that $\mathbb{E}(|X_1|) < \infty$. Then

$$\mathbb{P}\Big(\lim_{n \to +\infty} \frac{1}{n}(X_1 + \cdots + X_n) = \mathbb{E}(X_1)\Big) = 1$$

Remark

In most cases where we can use Monte Carlo metods, one can obtain estimates for the error :

$$\epsilon_n = \mathbb{E}(X) - \frac{1}{n}(X_1 + \cdots + X_n).$$

The Central Limit Theorem says us that the asymptotics behaviour of ϵ_n for large n

Example

Compute the expectation of the r.v. $Z = e^{\beta g}$ with $g \sim N(0,1)$:

$$E = \mathbb{E}\left(e^{\beta g}\right) = e^{\frac{\beta^2}{2}}.$$

Let us consider $f(g) = e^{\beta g}$.

$$E \approx \frac{f(X_1) + \cdots + f(X_n)}{n}$$

with $X_1, .., X_N$ i.id. having common law law $N(0,1)$.

We can use the following Monte Carlo algorithm :

Algorithm

```
main()
{
  double mean,g,sample,beta=5;

  mean_price= 0.0;
  for(i=1;i<=N;i++)
    {
      /*Simulation standard Gaussian*/
      g=gaussian();

      sample=exp(beta*g);

      mean=mean+sample;
    }

  /* Expectation of  Z */
  mean=mean/N;
}
```

Empirical variance

An estimator for the variance is given by :

$$\bar{S}_n^2 = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \bar{E}_n\right)^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \bar{E}_n^2.$$

$\bar{S}_n^2$ is often called the empirical variance of the sample. But this estimator is biased

$$\mathbb{E}\left(\bar{S}_n^2\right) = \frac{n-1}{n}\sigma^2.$$

Therefore a more appropriate estimator for the variance is given by :

$$\bar{\sigma}_n^2 = \frac{n}{n-1}\bar{S}_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{E}_n\right)^2 = \frac{n}{n-1}\left(\frac{1}{n}\sum_{i=1}^{n} X_i^2 - \bar{E}_n^2\right).$$

Then

$$\mathbb{E}\left(\bar{\sigma}_n^2\right) = \sigma^2.$$

$$
\begin{aligned}
\mathbb{E}(\bar{S_n}^2) &= \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} X_i^2\right) - \mathbb{E}_n^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left(X_i^2\right) - \mathbb{E}\left(\bar{E}_n^2\right) \\
&= \mathbb{E}\left(X_i^2\right) - \mathbb{E}\left(\bar{E}_n^2\right) \\
&= \left[\mathbb{E}\left(X_i^2\right) - \mathbb{E}^2\left(X_i\right)\right] + \mathbb{E}^2\left(X_i\right) - \left\{\left[\mathbb{E}\left(\bar{E}_n^2\right) - \mathbb{E}^2\left(\bar{E}_n\right)\right] + \mathbb{E}^2\left(\bar{E}_n\right)\right\} \\
&= \left[\sigma^2\right] + \mathbb{E}^2\left(X_i\right) - \left\{\left[Var(\bar{E}_n)\right] + \mathbb{E}^2\left(\bar{E}_n\right)\right\} \\
&= \left[\sigma^2\right] + \mathbb{E}^2\left(X_i\right) - \left[\frac{1}{n^2}\sum_{i=1}^{n} Var(X_i)\right] - \mathbb{E}^2\left(X_i\right) \\
&= \sigma^2 - \frac{\sigma^2}{n} \\
&= \frac{n-1}{n}\sigma^2
\end{aligned}
$$

Central Limit Theorem

Theorem

Let $(X_i, i \geq 1)$ be a sequence of i.i.d. random variables such that $\mathbb{E}(X_1^2) < \infty$. *Let* σ^2 *denote the variance* X_1. *Then :*

$$\frac{\sqrt{n}}{\sigma}\epsilon_n \text{ converge in distribution to } G,$$

where $G \sim N(0,1)$.

From this theorem it follows that for all $c_1 < c_2$:

$$\lim_{n \to +\infty} \mathbb{P}\left(\frac{\sigma}{\sqrt{n}}c_1 \leq \epsilon_n \leq \frac{\sigma}{\sqrt{n}}c_2\right) = \int_{c_1}^{c_2} e^{-\frac{x^2}{2}} \frac{dx}{\sqrt{2\pi}}.$$

In practice, one applies an approximate rule: for large n the distribution of ϵ_n is a Gaussian random variable with mean 0 e variance σ^2/n.

Confidence interval

$$\mathbb{P}\left(|G| \leq 1.96\right) \approx 0.95,$$

$$\mathbb{E}(X) \in \left[\bar{E}_n - 1.96\frac{\sigma}{\sqrt{n}}, \bar{E}_n + 1.96\frac{\sigma}{\sqrt{n}}\right].$$

We can use the empirical standard deviation.
This leads to an (approximate) confidence integral for $\mathbb{E}(X)$:

$$\left[\bar{E}_n - \frac{1.96\bar{\sigma}_n}{\sqrt{n}}, \bar{E}_n + \frac{1.96\bar{\sigma}_n}{\sqrt{n}}\right]$$

So, with very little additional computational cost we can give a reasonable estimate of the error done by approximating $\mathbb{E}(X)$ with E_n.

Compute expectation of Z with confidence interval

```
main()
{
  double mean,mean,g,sample,beta=5;
  double error,inf,sup;
  mean= 0.0;
  mean2= 0.0;
  for(i=1;i<=N;i++)
    {
      /* Standard Gaussian Simulation */
      g=gaussian();

      sample=exp(beta*g);

      mean= mean+sample;
      mean2= mean2+SQR(sample);
    }
  /* Expectation of  Z */
  mean=mean/N;
  error= sqrt(mean2/N - SQR(mean))/sqrt(N-1);
  inf= mean - 1.96*error;
  sup= mean + 1.96*error;
}
```

Results

```
Exact                          :                    268 337
100 000 trials  :                                   854 267
Confidence interval 95 % :                [-467 647,2 176 181] !
```

Financial example 1

Put option :

$$P = \mathbb{E}\left[(K - e^{\sigma g})_+\right].$$

with $g \sim N(0,1)$.

We approximate P with the empirical mean

$$P \approx \frac{f(X_1) + \cdots + f(X_n)}{n}$$

$(X_i, i \geq 1) \sim N(0,1)$.

Monte Carlo Algorithm
Financial example 1

```
main()
{
  double mean_price,mean2_price,g,price,price_sample,error_price,inf_price,sup_price;
  mean_price= 0.0;
  mean2_price= 0.0;
  for(i=1;i<=N;i++)
    {
      /*Standard Normal Simulation */
      g=gaussian();

      price_sample=MAX(0.0,K-exp(sigma*g));

      mean_price= mean_price+price_sample;
      mean2_price=mean2_price+ SQR(price_sample);
    }
  /* Price */
  price=mean_price/N;
  error_price= sqrt(mean2_price/N - SQR(price))/sqrt(N-1);
  inf_price= price - 1.96*(error_price);
  sup_price= price + 1.96*(error_price);
}
```

Results

European Put option with $\sigma = 1$ and $K = 1$.

```
Exact Price : 0.238422

N =    100,     Confidence interval 95% : [0.1795222,0.2963258]
                Simulated price : : 0.237924
N =  1 000,     Confidence interval 95% : [0.2321021,0.2700134]
               Simulated price : 0.2510578
N = 10 000,     Confidence interval 95% : [0.2356108,0.2473688]
                Simulated price : 0.2414898:
```

Integral computation

How to compute

$$I = \int_0^1 f(x)dx \quad ?$$

Classical methods

$$\sum_{i=0}^{n-1} \omega_i f(x_i), \quad \sum_{i=0}^{n-1} \omega_i = 1$$

- Riemann sums $\omega_i = \frac{1}{n}$, $x_i = \frac{i}{n}$
- Trapezoidal sums $\omega_i = \frac{1}{n}$, except $\omega_0 = \omega_n = \frac{2}{n}$, $x_i = \frac{i-1}{n}$
- Monte Carlo The integral I has a simple probabilistic interpretation.

$$\int_0^1 f(x)dx = \mathbb{E}\left[f(U) \right]$$

$$\int_0^1 f(x)dx \approx \frac{1}{N}(f(U_1) + \cdots + f(U_N))$$

with $U_1, .., U_N \sim Un(0, 1)$.
This method converge for any integrable function on $[0, 1]$.
(f is not supposed smooth, no regularity assumptions are necessary).

- Integral in a hypercube of dimension d

$$\int_{[0,1]^d} f(x_1, .., x_d) dx_1 .. dx_d = \mathbb{E}\left[f(U_1, .., U_d) \right]$$

The Monte Carlo method requires the simulation of indipendent random vectors $(U_1, .., U_d)$, whose coordinates are indipendent and follows a uniform law on $[0, 1]$.

Thus compared to the computation of the one dimensional integral, the number of trials is multiplied by d only.

Conversely, it is clear that the quadrature methods requires to many points.

Rate of convergence

For n large

$$\mathbb{P}\left(\frac{\sigma}{\sqrt{n}}c_1 \leq \epsilon_n \leq \frac{\sigma}{\sqrt{n}}c_2\right) = \int_{c_1}^{c_2} e^{-\frac{x^2}{2}} \frac{dx}{\sqrt{2\pi}}$$

The converge speed is given by

$$\frac{\sigma}{\sqrt{n}}$$

This is quite poor.

Nevertheless there exists situations where this slow method is the only one which can give an acceptable result (integrals computations in large dimension, parabolic equation in the same kind of dimension, pricing basket options).

Variance reduction techniques:

- Increase n
- Decrease σ.

Variance reduction techniques

Problem of the computation of

$$\mathbb{E}(X)$$

Basic Idea : find an alternative representation for the expectation to compute

$$\mathbb{E}(X) = \mathbb{E}(Y)$$

using a random variable Y with lower variance.

$$var(Y) << var(X)$$

Techniques:

- Control variate (Call-Put Parity Theorem)
- Importance sampling (Gaussian case).

Control variate

The basic idea of control variate is to write $\mathbb{E}\Big[f(X)\Big]$ as

$$\mathbb{E}\Big[f(X)\Big] = \mathbb{E}\Big[f(X) - h(X)\Big] + \mathbb{E}\Big[h(X)\Big]$$

where

$$\mathbb{E}\Big[h(X)\Big]$$

can be explicitely computed and such that

$$Var\Big[f(X) - h(X)\Big] << Var\Big[f(X)\Big]$$

Now we use a Monte Carlo method to estimate $\mathbb{E}\Big[f(X) - h(X)\Big]$ and we sum with the explicit computation of $\mathbb{E}\Big[h(X)\Big]$. We will give a first basic example, the Call-Put arbitrage formula.

Financial example 2

The price of a Call option is given by :

$$C = \mathbb{E}\left[(e^{\sigma g} - K)_+\right].$$

Consider $\sigma = 1$ $K = 1$.

```
exact value : 0.887143
N =    100,   confidence interval95% : [0.5420312,1.4090656]
              simulated price :  0.9755484
N =  1 000,   confidence interval95% : [0.7590768,0.9910191]
              simulated price :  0.8750480
N = 10 000,   confidence interval95% : [0.8511120,0.9302569]
              simulated price :   0.8906844
```

Using the Call Put parity theorem we have:

$$C - P = \mathbb{E}\left[e^{\sigma G} - K\right] = e^{\sigma^2/2} - K.$$

and therefore

$$C = P + \mathbb{E}\left(e^{\sigma G} - K\right) = P + e^{\sigma^2/2} - K.$$

In this case, one can obtain explicit formulas for the variance of the Put and the Call options.

On these formulas, one can check that the variance of the Put is smaller than the variance of the Call.

Importance sampling

Basic Idea : changing the sampling distribution.

Suppose we want to compute

$$\mathbb{E}\Big[f(X)\Big] = \int_{\mathbb{R}} f(x)p(x)dx$$

X being a real random variable following the density $p(x)$ on $\mathbb{R}$.

Let $\tilde{p}(x)$ be another density such that $\tilde{p}(x) > 0$. Clearly one can write

$$\mathbb{E}\Big[f(X)\Big] = \mathbb{E}\Big[\frac{f(Y)p(Y)}{\tilde{p}(Y)}\Big]$$

where Y has density $\tilde{p}(x)$ under $\mathbb{P}$.

We thus can approximate $\mathrm{E}\Big[\mathrm{f(X)}\Big]$ by an alternative Monte Carlo estimator

$$\frac{1}{N}\Big(\frac{f(Y_1)p(Y_1)}{\tilde{p}(Y_1)} + \cdots + \frac{f(Y_N)p(Y_N)}{\tilde{p}(Y_N)}\Big),$$

where $Y_1, ..., Y_N$ are indipendent copies of Y.

If $Var(\frac{f(Y)p(Y)}{\tilde{p}(Y)}) < Var(f(X))$ then it is better to use the new estimator.

The Gaussian case

Suppose that $X \sim N(0,1)$ and we compute to compute

$$\mathbb{E}\Big[f(X)\Big]$$

We have $\mu \in \mathbb{R}$

$$\mathbb{E}\Big[f(X)\Big] = \mathbb{E}\Big[f(X+\mu)e^{-\mu X - \frac{1}{2}\mu^2}\Big].$$

In fact, we choose the distribution of $Y = X + \mu$ as the new sampling distribution.

$$\mathbb{E}\Big[f(X)\Big] = \mathbb{E}\Big[\frac{f(Y)p(Y)}{\tilde{p}(Y)}\Big] = \mathbb{E}\Big[f(Y)e^{-\mu Y + \frac{1}{2}\mu^2}\Big].$$

μ is a real costant that has to be determined carefully in order to reduce the variance.

Call Deep Out-the-Money Options

$$\mathbb{E}\Big[f(X)\Big] = \mathbb{E}\Big[f(X+\mu)e^{-\mu X - \frac{1}{2}\mu^2}\Big]$$

- Let us consider

$$C = \mathbb{E}\left[(e^{\sigma g} - K)_+\right].$$

  with $K >> 1$.

- $\mathbb{P}\Big(e^{\sigma g} > K\Big)$ is very small and it is very unlikely that the option will be exercised.
- This fact can lead to a very large error in a standard MC method.
- Idea : to push the random variable towards K using μ
- How to chose μ?

$$C = \mathbb{E}\Big[(e^{\sigma(g+\mu)} - K)_+ e^{-\mu g - \frac{1}{2}\mu^2}\Big]$$

- Heuristic approach : μ such $e^{\sigma\mu} = K$.

$$\mathbb{P}\Big(e^{\sigma(g+\mu)} \geq K\Big) = \frac{1}{2}$$

- It will drastically improve the efficiency of the MC method when $K >> 1$.

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 f. Thus we can implement it very easily if we are able to generate the variable X according to his distribution.
  - The estimator is unbiased, that is $E[\bar{E}_N] = E(X)$.
  - 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.
  But to reduce computing time, we can use variance reduction techniques.