Monte Carlo Integration

Math 348: Class notes

March 14, 2008

1 Introduction: Probability distributions and random variables

1.1 Definition and examples

A random variable X is by definition a mapping that takes its values in a discrete set or a continuous interval of real numbers, according to a given probability distribution, P.

A random variable with values in a discrete set of real numbers is called *a discrete random variable* and a random variable that takes value in an interval is called *a continuous random variable.*

Examples of discrete random variables
a) Perhaps the simplest example of a discrete random variable is that of tossing a fair coin, which results in a head or tail with a 50/50 % chance. We can thus associate a random variable X that takes the value $X = 0$ if the outcome is head and $X = 1$ if it is tail with the probability distribution

$$P(\{X = 0\}) = \frac{1}{2},\ P(\{X = 1\}) = \frac{1}{2}.$$

b) A similar example is that of throwing a dice. The associated random variable takes its values in the discrete set $\{1, 2, 3, 4, 5, 6\}$ with the probabilities

$$P(\{X = 1\}) = p_1, P(\{X = 2\}) = p_2, \cdots, P(\{X = 6\}) = p_6$$

where $0 \leq p_1, p_2, \cdots, p_6 \leq 1$ and $p_1 + p_2 + \cdots + p_6 = 1$. If the dice is unloaded, then $p_1 = p_2 = \cdots = p_6 = 1/6$.

Examples of continuous random variables
Concrete examples of continuous random variables are ubiquitous in nature and human life. Often a continuous random variable represents an idealized approximation of a discrete random variable taking values in a large discrete set. Common examples include that of the price of an asset in the stock market or the exact dimensions of a manufactured object–there is always some deviations from the aimed dimensions due to imperfections in the devices used for making the product, etc.

The probability distribution for a continuous random variable is given in terms of the probability that X lies in a given interval $[a, b]$: $P(\{a \leq X \leq b\})$. For all practical purposes, the point-wise probability for a continuous random variable, $P(\{X = x\})$, (which is zero in theory, $P(\{X = x\}) = 0$) is meaningless.

a) Uniform random variable on $[0, 1]$:
If $[a, b] \subset [0, 1]$, then $P(\{a \leq X \leq b\}) = b - a$, the length of the interval $[a, b]$.
If $(a, b) \cap (0, 1) = \varnothing$, then $P(\{a \leq X \leq b\}) = 0$.

Note that accordingly, we have in general $P(\{a \leq X \leq b\}) = |(a, b) \cap (0, 1)|$ and $P(\{0 \leq X \leq 1\}) = 1$.

Very often, the probability distribution for a continuous random variable is given on the form of an integral

$$P(\{a \leq X \leq b\}) = \int_a^b f(x)dx$$

where $f(x)$ is a non-negative real valued function with the important property

$$\int_{-\infty}^{+\infty} f(x)dx = 1.$$

$f(x)$ is know as the *probability density function* (pdf for short). For the example of a uniform random variable on $[0, 1]$ we have

$$f(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Exercise: Verify that indeed $P(\{a \leq X \leq b\}) = \int_a^b f(x)dx$, for the example of a uniform random variable on $[0, 1]$.

The function

$$F(x) = \int_{-\infty}^{x} f(x)dx$$

which is the probability that $-\infty \leq X \leq x$ is known as the *cumulative distribution function* (CDF).

b) Gaussian or normal distributed random variable:
A Gaussian random variable takes its values in $(-\infty, +\infty)$ according to the Gaussian probability density function

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

where $\mu, \sigma > 0$ are two real parameters known as the mean and standard deviation of the Gaussian distribution. Note that

$$P(\{a \leq X \leq b\}) = \frac{1}{\sqrt{2\pi}\sigma} \int_a^b \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx$$

is not known in closed form but it is easily evaluated using quadrature technique based algorithms that are already implemented in many of the available softwares. For example in matlab one can use the function *normcdf* which serves to evaluate the cumulative distribution function

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx.$$

The Gaussian distribution is widely known as the normal distribution. The reason for this is because it is ubiquitous in nature. Many naturally generated random variables are Gaussian (as this is reflected by the central limit theorem given below).

To verify that the Gaussian pdf satisfies $\int_{-\infty}^{+\infty} f(x)dx = 1$ the following important result from multivariable calculus is used (combined with a clever change of variables).

$$\begin{aligned}\left[\int_{-\infty}^{+\infty} e^{-x^2} dx\right]^2 &= \int_{-\infty}^{+\infty} e^{-x^2} dy \times \int_{-\infty}^{+\infty} e^{-y^2} dxdy \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-(x^2+y^2)} dx = \int_{0}^{2\pi} \int_{0}^{+\infty} re^{r} drd\theta = 2\pi.\end{aligned}$$

c) Exponentially distributed random variable:

The pdf of an exponential random variable is given by

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

where $\lambda > 0$ is a real parameter.

Exercise: Show that $\int_{-\infty}^{+\infty} f(x)dx = 1$.

d) Log normal distribution:

A positive random variable $Y > 0$ is said log-normal if $X = \ln Y$ is a normal random variable.

Other common and important examples of random variables or probability distributions include, the binomial and Poisson distributions, which are discrete, the Ξ (Chi) distribution, the Γ (Gamma) distribution, etc.

Notation:

In many textbooks and research papers the uniform distribution on an interval $[\alpha, \beta]$ is denoted by $\mathcal{U}(\alpha, \beta)$. $\mathcal{U}(0, 1)$ denotes the standard uniform distribution on $(0, 1)$. While the standard normal or Gaussian distribution is denoted by $N(\mu, \sigma)$. $N(0, 1)$ denotes the Gaussian distribution with mean zero and standard deviation 1.

The PDFs and CDFs of the standard uniform $\mathcal{U}(0, 1)$, standard normal $\mathcal{N}(0, 1)$, and standard exponential ($\lambda = 1$) distributions are plotted in Figure 1.

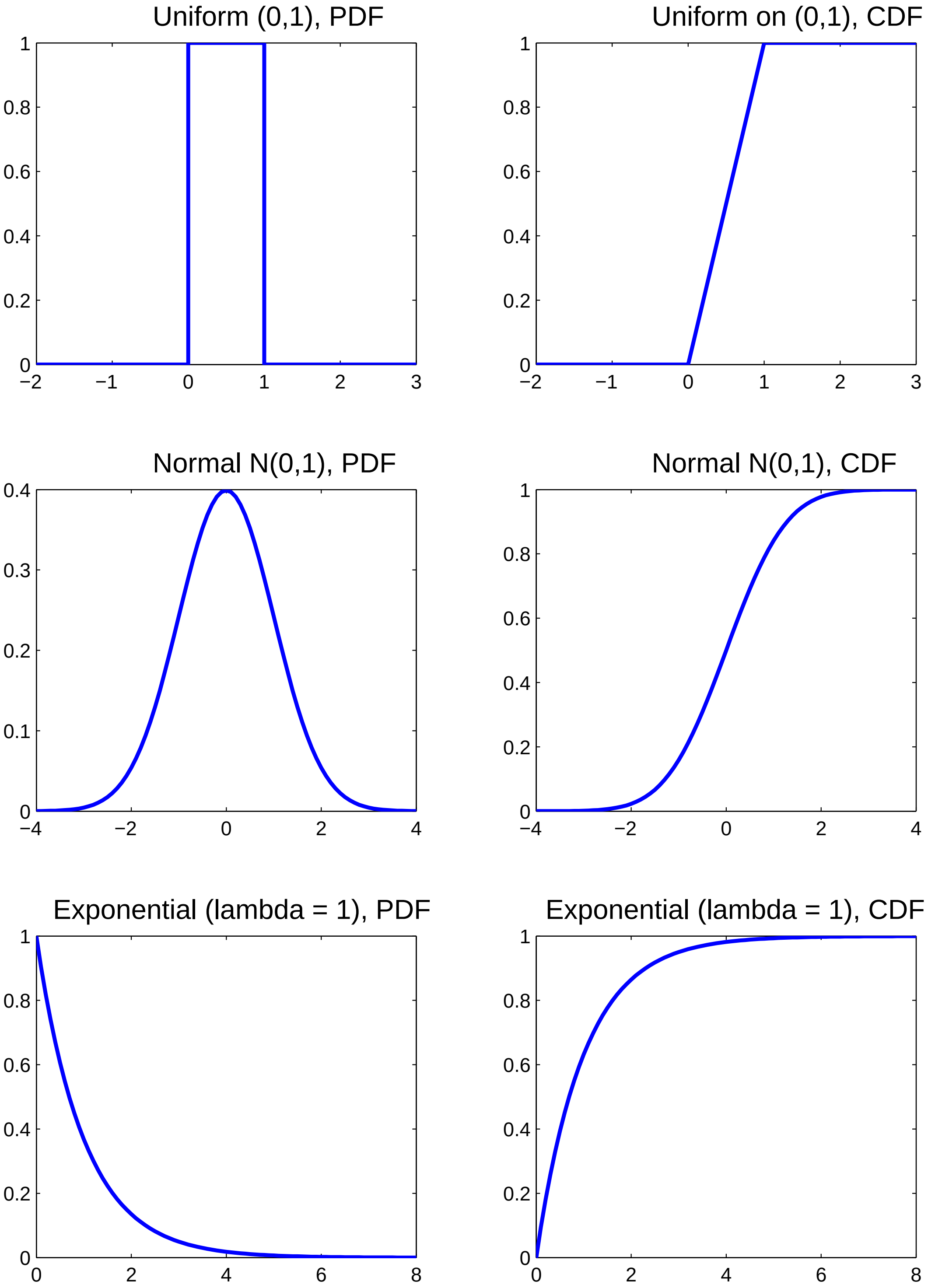

Figure 1: The PDFs and CDFs of the standard uniform, standard normal, and standard exponential distributions.

1.2 Mean, variance, and expectation

Let X denote a continuous random variable with a pdf $f(x)$. The *mean or expectation* of X is given by

$$E[X] = \int_{-\infty}^{+\infty} xf(x)dx.$$

Often the mean or expectation is equivalently denoted by an overbar or angle braces

$$E[X] \equiv \bar{X} \equiv \langle X \rangle.$$

For any given real valued function g we can define the expectation of $g(X)$ as

$$E[g(X)] = \int_{-\infty}^{+\infty} g(x)f(x)dx.$$

In a call option for example, if $X = S(T)$ is the market (selling) price of the underlying asset at the strike time T, distributed according to its pdf $f(x)$, and K is the strike price, then the profit is given by

$$\pi = \max(X - K, 0),$$

and the expected profit is

$$E[\pi] = \int_{-\infty}^{+\infty} \max(x - K, 0)f(x)dx = \int_{K}^{+\infty} xf(x)dx - KP(\{X \geq K\}).$$

The *variance* of X is given by

$$Var[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

and the standard deviation is given by

$$\sigma = \sqrt{Var[X]}.$$

Using simple integration rules, we can easily show that for

- $\mathcal{U}(0,1)$, uniform distribution on $[0,1]$, we have

$$E[X] = \int_0^1 xdx = \frac{1}{2} \text{ and } Var[X] = \int_0^1 x^2dx - \frac{1}{4} = \frac{1}{12}.$$

- $\mathcal{N}(\mu,\sigma)$, Gaussian distribution with parameters μ, σ, we have

$$E[X] = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} x \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = \mu$$
$$\text{and } Var[X] = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} (x-\mu)^2 \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx - \mu^2 = \sigma^2.$$

- exponential distribution with a parameter $\lambda > 0$, we have

$$E[X] = \lambda \int_0^{+\infty} xe^{-\lambda x}dx = \frac{1}{\lambda} \text{ and } Var[X] = \lambda \int_0^{+\infty} x^2 \exp\left(-\lambda x\right) dx - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}.$$

1.3 Conditional probability and notion of dependent and independent random variables

Let X, Y be two random variables with their respective probability density functions, f_X, f_Y. The probability $P(\{a \leq X \leq b, c \leq Y \leq d\})$ that both $a \leq X \leq b$ and $c \leq Y \leq d$ at the same time, is know as the *joint probability distribution.* The joint cumulative distribution function is the two variable function

$$F(x, y) = P(\{X \leq x, Y \leq y\}).$$

The joint probability density function of X, Y is given by

$$f(x, y) = \frac{\partial^2 F(x, y)}{\partial x \partial y}$$

so that

$$F(x, y) = \int_{-\infty}^{y} \int_{-\infty}^{x} f(t, s) dt ds.$$

We have the following compatibility conditions.

$$\begin{aligned} F_X(x) \equiv \int_{-\infty}^{x} f_X(t) dt &= \int_{-\infty}^{+\infty} \int_{-\infty}^{x} f(t, s) dt ds \qquad (1) \\ \text{and } F_Y(y) \equiv \int_{-\infty}^{y} f_Y(s) ds &= \int_{-\infty}^{+\infty} \int_{-\infty}^{y} f(t, s) ds dt. \end{aligned}$$

The two random variables X, Y are said *independent* if

$$P(\{a \leq X \leq b, c \leq Y \leq d\}) = P(\{a \leq X \leq b\}) \times P(\{c \leq Y \leq d\}).$$

If X, Y are independent then their joint probability density function satisfies

$$f(x, y) = f_X(x) f_Y(y).$$

The probability that $a \leq X \leq b$ given that $c \leq Y \leq d$ denoted by $P(\{a \leq X \leq b/c \leq Y \leq d\}$ is known as the conditional probability of the random variable X given that Y was realized. The following relationship holds.

$$P(\{a \leq X \leq b, c \leq Y \leq d\}) = P(\{a \leq X \leq b/c \leq Y \leq d\}) \times P(\{c \leq Y \leq d\}).$$

If X, Y are independent then the conditional probability satisfies

$$P(\{a \leq X \leq b/c \leq Y \leq d\}) = P(\{a \leq X \leq b).$$

The covariance between the two random variables X, Y is defined as

$$Cov(X, Y) = E[XY] - E[X]E[Y]$$

where

$$E[XY] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xyf(x,y)dxdy, \quad E[X] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} xf(x,y)dxdy = \int_{-\infty}^{+\infty} xf_X(x)dx,$$

$$\text{and } E[Y] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} yf(x,y)dxdy = \int_{-\infty}^{+\infty} yf_Y(y)dy,$$

where we used the compatibility condition (1). It is easy to verify that

$$Cov(X,X) = Var[X], \quad Cov(Y,Y) = Var[Y]$$

and if X, Y are independent then

$$Cov(X,Y) = 0.$$

Note that $Cov(X,Y) = 0$ doesn't necessarily mean that X, Y are independent. If $Cov(X,Y) > 0$, then X, Y are said positively correlated and if $Cov(X,Y) < 0$, they are said negatively correlated.

We have

$$E[X+Y] = E[X] + E[Y], \quad E[cX] = cE[X]$$

and

$$Var[X+Y] = Var[X] + Var[Y] + 2Cov(X,Y), \quad Var[cX] = c^2 Var[X]$$

Law of large numbers

Let $X_1, X_2, \cdots, X_n, n \geq 2$ be a sequence of independent and identically distributed (i.i.d) random variables, each having a mean μ and a standard deviation σ; $E[X_j] = \mu, Var[X_j] = \sigma^2, \forall j$. Define the 'average' random variable

$$X = \frac{1}{n}(X_1 + X_2 + \cdots X_n) = \frac{1}{n}\sum_{j=1}^{n} X_j.$$

Then the sample mean equals the population mean

$$E[X] = \frac{1}{n}\sum_{j=1}^{n} E[X_j] = \frac{1}{n}n\mu = \mu$$

and $Cov(X_j, X_k) = 0, j \neq k$ (because the r.v.'s are independent), implies

$$Var[X] = \frac{1}{n^2}\sum_{j=1}^{n} Var[X_j] = \frac{\sigma^2}{n}.$$

As a result the sample mean $\bar{X} = \lim_{n \longrightarrow +\infty} \frac{1}{n}\sum_{j=1}^{n} X_j$ converges (is said to) to the population mean μ in the probability or weak sense, i.e,

$$\lim_{n \longrightarrow +\infty} \frac{1}{n}\sum_{j=1}^{n} X_j = \mu \text{ with probability one}$$

or more precisely we have (Chebechev's inequality[1])

$$\forall \epsilon > 0, P(\{|X-\mu| \geq \epsilon\}) \leq \frac{Var[X]}{\epsilon^2} = \frac{\sigma^2}{n\epsilon^2}.$$

Therefore, in principle, the expectation $E[X] = \mu$ can be estimated by the sample average $\frac{1}{n}\sum_{j=1}^{n} X_j$ if n is large enough.

Central limit theorem

Let $X_1, X_2, \cdots, X_n, n \geq 2$ be a sequence of i.i.d random variables with mean μ and standard deviation σ. The sequence of random variables $\frac{X_1+X_2+\cdots+X_n-n\mu}{\sigma\sqrt{n}}$ converges in probability to a normally distributed random variable with mean zero and variance one. i.e, the limit in the probability sense,

$$Y = \lim_{n\longrightarrow+\infty} \frac{X_1+X_2+\cdots+X_n-n\mu}{\sigma\sqrt{n}}$$

has normal distribution with mean zero and standard deviation one: $Y \sim \mathcal{N}(0,1)$.

2 Crude Monte Carlo integration

Consider the integral

$$I = \int_0^1 g(x)dx.$$

This integral can be thought of as the expectation $E[g(U)]$ where U is a random variable uniformly distributed on $[0,1]$ ($U \sim \mathcal{U}(0,1)$). According to the law of large numbers above the expected value $I = E[g(U)]$ can be estimated by a sample mean. Let $U_1, U_2, \cdots, U_n$ be a sequence of random numbers *generated* according to the uniform distribution $\mathcal{U}(0,1)$. Then

$$I \approx \hat{I}_n = \frac{1}{n}\sum_{j=1}^{n} g(U_j).$$

The law of large numbers guarantees that with probability one,

$$\lim_{n\longrightarrow\infty} \hat{I}_n = I.$$

This is the basis for Monte-Carlo integration. However, the sampling of genuinely random numbers is not possible on a digital computer. Instead, most programming languages and computing environments such as matlab have one or more built in functions that can generate sequences of *pseudo-random* numbers. The function *rand()* of matlab can be used to generate sequences of pseudo-random numbers that are $\mathcal{U}(0,1)$ while *randn* samples the standard normal distribution $\mathcal{N}(0,1)$ (Gaussian distribution of mean zero and variance one).

[1] For the proof see the appendix at the end of these notes.

Example 1: Consider

$$I = \int_0^1 e^x dx = e - 1 \approx 1.7183.$$

To use Monte-Carlo integration, we view this integral as the expectation

$$E[e^U] \approx \frac{1}{n}\sum_{j=1}^{n} e^{U_j}, \text{ with } \mathcal{U}(0,1)$$

and use the function *rand()* of matlab to generate a sequence of pseudo-random numbers $\mathcal{U}(0,1)$. Recall that *rand(N,M)* returns an $N \times M$ matrix of random numbers, all uniformly distributed in $[0,1]$.

```
 >>rand('state',0) %this (re)initializes the function rand
 >>I = mean(exp(rand(1,10))
 ans=
      1.8318
 >>I = mean(exp(rand(1,10))
 ans=
      2.0358
>>I = mean(exp(rand(1,1000000))
 ans=
      1.7189
>>I = mean(exp(rand(1,1000000))
 ans=
      1.7178
```

Increasing the number of samples clearly improves the estimated integral.

Example 2:

Here we show how the Monte Carlo method can be used to compute an improper integral. As an example, we consider

$$I = \int_{-\infty}^{+\infty} \cos(x) e^{-x^2/2} dx.$$

To use Monte Carlo method we first rewrite the integral as

$$I = \int_{-\infty}^{+\infty} \sqrt{2\pi}\cos(x)\frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx.$$

Recognizing the normal probability density function $f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$, we view the given integral as the expectation $E[\cos(X)]$, where X is an $\mathcal{N}(0,1)$ random variable. Therefore

$$I \approx \frac{\sqrt{2\pi}}{n}\sum_{j=1}^{n} \cos(X_j)$$

where the X_j's are random numbers sampled (drawn) from the standard normal distribution. This can be easily accomplished in matlab by using the *randn()* function. The matlab lines below show how this is implemented. For the sake of comparison, we use the deterministic *quad* function (i.e. the composite Simpson rule) on a finite interval $[-A, A]$ with $A = 10$ and $A = 100$.

```
 >> quad('cos(x).*exp(-x.^2/2)',-10,+10)
ans =
    1.5203
 >> quad('cos(x).*exp(-x.^2/2)',-100,+100)
ans =
    1.5203
 >> sqrt(2*pi)*mean(cos(randn(1,100)))
ans =
    1.5682
>> sqrt(2*pi)*mean(cos(randn(1,100)))
ans =
    1.4070
>> sqrt(2*pi)*mean(cos(randn(1,1000000)))
ans =
    1.5212

>> sqrt(2*pi)*mean(cos(randn(1,1000000)))
ans =
    1.5225
```

Conclusion

The two examples above show that in fact the crude Monte Carlo method as utilized here to estimate integrals is not at all competitive with the various deterministic quadrature rules such as the trapezoidal, Simpson, Gauss, etc. rules. In fact the convergence is very slow and it requires a large number of samples. However the one major advantage of Monte Carlo integration lies in multi-dimensional integrals, because the number of integration points (i.e samples) required to achieve a given accuracy is independent of the dimension of the integral unlike the deterministic quadrature rules for which the number of integration points typically increases exponentially with the dimension: if n points are needed to achieve an accuracy ϵ in one dimension, then n^d points are needed in $\mathbb{R}^d$ (see the book, on page 223 for more discussion).

We finish this section by a finance example (taken from the book) dealing with the pricing of a vanilla European option. The discounted payoff is given through the expected value of the payoff at maturity.

$$f = e^{-rT} E[f_T],$$

where T is the maturity date, r is the risk free (i.e, the bank) interest rate, and f_T is the payoff at maturity. For a call option, we have, according to the Black-Scholes formula,

$$f_T = \max\left(0, S(0)e^{\left((r-\sigma^2/2)T+\sigma\sqrt{T}\ \xi\right)} - K\right)$$

where $S(0)$ is the initial price of the underlying asset, σ is the volatility, and ξ is a standard normal random variable $\xi \sim \mathcal{N}(0,1)$. The following matlab code uses Monte Carlo to compute the discounted payoff f (Book, page 224).

```
%M-file: BlsMC1.m
function Price = BlsMC1(S0,K,r,T,sigma,NRel)
nuT = (r - 0.5*sigma^2)*T;
siT = sigma * sqrt(T);
DiscPayoff = exp(-r*T)*max(0, S0*exp(nuT+siT*randn(NRepl,1))-K);
Price = mean(DiscPayoff);
%%%%%end of M-file

>>S0 = 50 ;
>>K =60;
>>r = 0.05;
>>T = 1;
>>sigma = 0.2;
>> randn('state',0)
>> BlsMC1(S0,K,r,T,sigma,1000)
ans =
1.2562
>> BlsMC1(S0,K,r,T,sigma,1000)
ans =
1.2562
>> BlsMC1(S0,K,r,T,sigma,1000)
ans =
1.8783
>> BlsMC1(S0,K,r,T,sigma,1000)
ans =
1.8764
>> BlsMC1(S0,K,r,T,sigma,1000000)
ans =
1.6295
>> BlsMC1(S0,K,r,T,sigma,1000000)
ans =
1.6164
>> BlsMC1(S0,K,r,T,sigma,1000)
ans =
1.6141
```

3 Random number generators: pseudo-random numbers

Generating pseudo-random variates from a given probability distribution starts by generating pseudo-random numbers from the uniform distribution $\mathcal{U}(0,1)$. The uniformly distributed pseudo-random numbers are then converted for the desired distribution according to various transformation methods. Here we discuss two main such methods, the inverse transform and the acceptance rejection method. Also pseudo-variates for distributions with different parameters can be generated from the standard distributions by simple change of variables.

- Pseudo-random numbers uniformly distributed on an arbitrary interval (α,β) can be generated from uniform variates on $(0,1)$.
$$U \sim \mathcal{U}(0,1) \implies X = (\beta-\alpha)U + \alpha \sim \mathcal{U}(\alpha,\beta)$$
- Pseudo normal variates with mean μ and variance σ generated from the standard normal distribution according to
$$X \sim \mathcal{N}(0,1) \implies Y = \mu + \sigma X \sim \mathcal{N}(\mu,\sigma)$$
- Pseudo-random exponential variates with a parameter $\lambda > 0$ are generated from the standard exponential variates with $\lambda = 1$
$$X \sim \mathcal{EXP}(1) \implies Y = \frac{X}{\lambda} \sim \mathcal{EXP}(\lambda)$$
- Log-normal variates are obtained from normal variates. How?

3.1 Linear congruential generators (LCG)

The most popular and simplest textbook examples of pseudo-random number generators is the family of linear congruential generators (LCG). They consist of generating a sequence of integers of the form
$$Z_{i+1} = (aZ_i + c) \mod m$$
where a, c, m are judiciously chosen integer parameters. a is called the multiplier, c the shift, and m is the modulus. Recall that the operation $r = x \mod m$ returns the integer remainder $0 \le r \le m-1$ from the division x/m. A sequence of pseudo-random variates uniformly distributed on $(0,1)$ is given by
$$U_i = Z_i/m, i = 1, 2, \cdots$$
To generate the sequence Z_i, an initial value Z_0 should be provided. Z_0 is know as the seed. In fact, for a given seed the whole sequence $Z_i, i \ge 0$ can be predicted i.e, the sequence of $U_i, i \ge 0$ are far from being independent random numbers. This is why they are called pseudo-random numbers, instead. When called with the same seed, the LCG algorithm will always generate the same sequence. To simulate a better resemblance to randomness, some programmers use a seed that changes with the machine clock time, eg. *seed = 100*clock* but this can reveal to be a very

bad choice, especially if we want to obtain the same result twice, for comparing or validating the numerical results.

The matlab code below (taken from the book) permits to generates N $\mathcal{U}(0,1)$-pseudo-random variates using an LCG generator with given parameters a, c, m and a given seed.

```
%M-file: LCG.m
function [Useq,Zseq] = LCG(a,c,m,seed,N)
Zseq = zeros(N,1);
Useq = zeros(N,1);
for i=1:N
 seed = mod(a*seed + c ,m) ;
Zseq(i) = seed;
Useq(i) = seed/m;
end
```

Assume for example we set $a = 5, c = 3, m = 16, seed = 7, N = 20$. Then, the matlab code above generates the two sequences

$$\{Z_i, i = 1, \cdots, 20\} = 6, 1, 8, 11, 10, 5, 12, 15, 14, 9, 0, 3, 2, 13, 4, 7, 6, 1, 8, 11$$

and

$$\begin{aligned}\{U_i, i = 1, \cdots, 20\} =& 0.3750, 0.0625, 0.5000, 0.6875, 0.6250, 0.3125, 0.7500, 0.9375, 0.8750, 0.5625, 0, \\ & 0.1875, 0.1250, 0.8125, , 0.2500, 0.4375, 0.3750, 0.0625, 0.5000, 0.6875.\end{aligned}$$

Two main conclusions can be drawn from these two sequences. The U sequence seems to jump back and forth within the interval $[0, 1)$ in an almost random fashion, uniformly covering the interval $[0, 1]$. The sequence Z however starts from and passes through all the 20 numbers $0, 1, \cdots, 19$ and comes back to number 6 after exactly 20 iterations and then repeats itself again. Thus, the sequence Z is periodic with period 20.

In fact, all LCG generators are periodic, because they operate within a finite set of numbers, the sequence will repeat itself as soon as it hits twice the same number. The period of the sequence Z satisfies $T \leq m$. However, not all LCG sequences always have a maximal period $T = m$ as in the previous example. In fact repeating the previous experiment with $a = 11, c = 5, m = 16$ and a seed $Z_0 = 3$ yields the sequence

$$Z = 6, 7, 2, 11, 14, 15, 10, 3, 6, 7, 2, 11, 14, 15, 10, 3, 6, 7, 2, 11$$

that displays a period $T = 8$, which is half the maximum possible period.

LCG's used in practice have a very large modulus m and the parameters a and c are chosen to achieve a maximum period. This ensures that the uniform variates $U_i = Z_i/m$ 'cover' uniformly the interval $(0, 1)$ with an apparent random behaviour, having samples that look independent. This is not an easy task however. An easy way to obtain a sequence with a maximal period, for any given m, is to set $a = c = 1$ and use a seed $Z_0 = 0$. This yields the sequence

$$U_i = \frac{i}{m}, i = 1, 2, \cdots, m-1.$$

In fact, this is a very bad choice. The generated variates do not look independent. Luckily, the random number generators found in commercial softwares were already tested to comply some minimal standards of having the required statistical properties to mimic well the desired randomness etc. and you don't have to worry about designing a random generator for yourself each time you want to use Monte Carlo to solve a numerical problem. The function *rand* installed in the latest matlab version (7.1.x) is in fact based on an algorithm that is far more sophisticated than LCGs and its details are beyond the scope of these notes.

Exercise: Read Example 4.5 of the book (page 228) and discuss the lattice structure of the pseudo-uniform numbers generated by the LCG method.

3.2 Inverse transform method

The inverse transform method is based on the following basic and general statement.

Given a random variable X and its probability density f_X and cumulative distribution function, $F_X(x)$, then the random variable

$$U = F_X(X) = \int_{-\infty}^{X} f_X(x)dx$$

is uniformly distributed on $(0, 1)$. To prove this it suffices to establish that

i) $P(\{U \leq y\}) = 1$ if $y \geq 1$

ii) $P(\{U \leq y\}) = 0$ if $y \leq 0$

iii) $P(\{a \leq U \leq b\}) = b - a$ if $(a, b) \subset (0, 1)$

The first two statements (i), (ii) are trivial given the fact $0 \leq \int_{-\infty}^{X} f_X(x)dx \leq 1$. For (iii) , we have

$$\begin{aligned} P(\{a \leq U \leq b\}) &= P(\{a \leq F_X(X) \leq b\}) = P(\{F_X^{-1}(a) \leq X \leq F_X^{-1}(b)\}) \\ &= \int_{-\infty}^{F_X^{-1}(b)} f_X(x)dx - \int_{-\infty}^{F_X^{-1}(a)} f_X(x)dx = F(F_X^{-1}(b)) - F(F_X^{-1}(a)) = b - a. \end{aligned}$$

Conversely, if $U \sim \mathcal{U}(0, 1)$, then we have

$$P(\{X \leq x\}) = P(\{F^{-1}(U) \leq x\}) = P(\{U \leq F(x)\}) = F(x)$$

This is represented schematically in Figure 2. If we are able to invert F easily[2], then the inverse transform method can be implemented to generate pseudo-random variates drawn from the distribution f_X as follows.

[2] which is of course guaranteed because $F(X)$ is an absolutely continuous and monotonically increasing function on the support of f_X

1. Draw a uniform pseudo-random variate $U \sim \mathcal{U}(0,1)$
2. Set $X = F^{-1}(U)$.

Example: As a typical easy example, we consider the standard exponential distribution $X \sim \exp(\lambda)$. We have

$$F(x) = 1 - e^{-\lambda x}.$$

For a given uniform variate, the inverse transform method yields

$$X = F^{-1}(U) = -\frac{1}{\lambda}\ln(1-U).$$

Given that the probability for drawing U and $1-U$ from the uniform distribution is the same, in practice exponential variates are usually generated by drawing uniform variates $U \in (0,1)$, then return $X = -\ln(U)/\lambda$. This is what it is implemented in the Statistics toolbox of matlab. As an example we use the above algorithm to compute the integral

$$I = \int_0^{+\infty} 2xe^{-2x}dx = \frac{1}{2},$$

which is the expectation of the exponential random variable $X \sim \exp(2)$.

```
 >> u = rand(1,1000000);
>> x = - log(u)/2;
>> mean(x)
ans =
    0.4993
>> u = rand(1,1000000);
>>x = - log(u)/2;
>>mean(x)
ans =
    0.5001
>> u = rand(1,1000000);
>>x = - log(u)/2;
>>mean(x)
ans =
    0.5004
```

3.3 Acceptance-rejection method

As mentioned above, the inverse transform method is only feasible when the cdf $F(x)$ can be easily inverted. Although, we can always resort to numerical root-finding techniques such as Newton's method to invert $F(x)$, it is not always a good idea because this can be very costly especially when we need to generate a large number of variates–to perform a Monte Carlo integration for example.

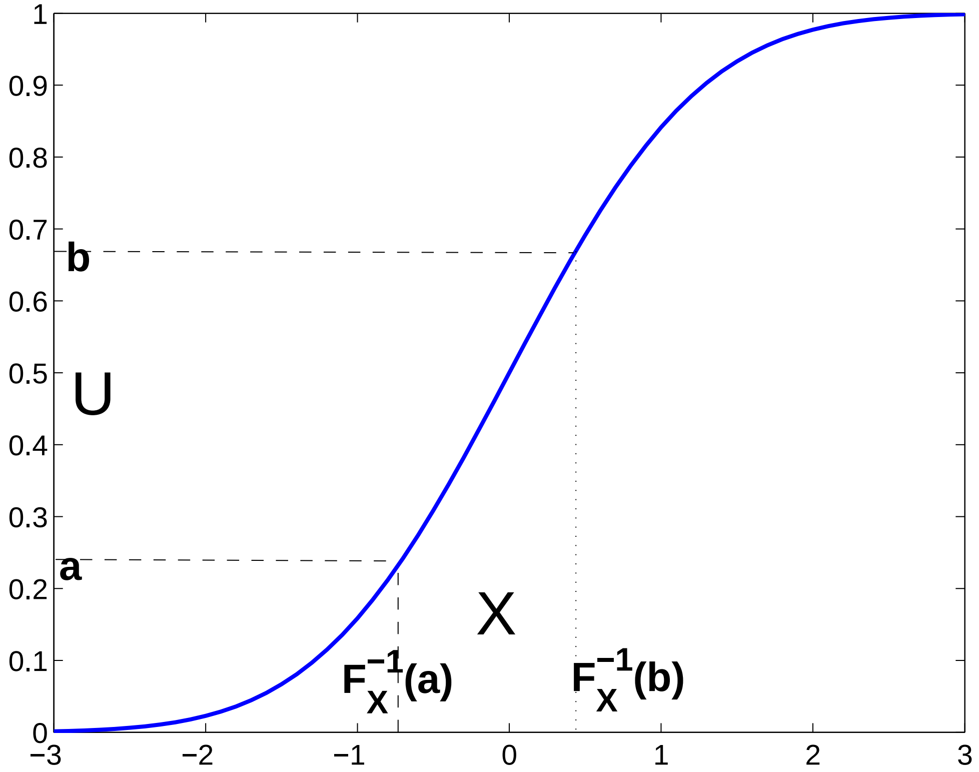

Figure 2: Schematic of the inverse method: $P(\{a \leq U \leq b\}) = P(\{F_X^{-1}(a) \leq X \leq F_X^{-1}(b)\})$

A better approach for the case when the inverse of $F(x)$ is not known in closed form or is expensive to evaluate is the *acceptance-rejection method* discussed here.

To simulate a random variable X obeying a given probability distribution with a pdf $f(x)$, the acceptance-rejection method starts by finding a function $t(x)$ such that

$$f(x) \leq t(x), \forall x \in \mathbb{R}$$

and

$$K = \int_{-\infty}^{+\infty} t(x)dx < +\infty$$

that is easy to invert. Once such a function $t(x)$ is found, the acceptance-rejection method consists of the three main steps listed below. Noting that $g(y) = t(y)/K$ (so that $\int_{-\infty}^{+\infty} g(y)dy = 1$) is a probability density function.

1. Use the inverse transform method to generate a pseudo random number Y corresponding to $g(y)$.
2. Draw a uniform variate U form $\mathcal{U}(0,1)$, independent of Y.
3. If $U \leq f(Y)/t(Y)$, then return $X = Y$ (accept), otherwise go to step 1.

Recall that for a given (fixed) Y, the probability for a uniformly distributed random number U to satisfy $U \leq f(Y)/t(Y)$ is $P(\{U \leq f(Y)/t(Y)\}) = f(Y)/t(Y)$. Therefore, the more this ratio is close to one the better are chances for the random number Y to be accepted and the above procedure to be terminated. The points Y where this ratio is close to 1 are likely to be accepted while those with a small $f(Y)/t(Y)$ are very unlikely to be accepted. To gain efficiency, it is thus important to choose a function $t(x)$ which as close as possible to $f(x)$. Also, it can be shown that the average number of iterations (acceptance and rejection trails) to terminate the procedure with an accepted value X is $K = \int_{-\infty}^{+\infty} t(x)dx$.

If the support of $f(x)$ is bounded, i.e, $f(x) = 0$ outside a bounded interval $[\alpha, \beta]$, then a natural choice for $g(x)$ is simply the uniform distribution on $[\alpha, \beta]$ and choose $t(x) =$ constant $= \max_{[\alpha,\beta]} f(x)$ for $\alpha \leq x \leq \beta$ and $t(x) = 0$ otherwise.

As an example we consider below the beta-distribution on $[0, 1]$.

$$f(x) = \frac{x^{\alpha_1 - 1}(1-x)^{\alpha_2 - 1}}{B(\alpha_1, \alpha_2)}, \qquad x \in [0, 1]$$

where

$$B(\alpha_1, \alpha_2) = \int_0^1 x^{\alpha_1 - 1}(1-x)^{\alpha_2 - 1}dx$$

For $\alpha_1 = \alpha_2 = 3$, we have

$$f(x) = 30(x^2 - 2x^3 + x^4), \qquad x \in [0, 1].$$

Note that this is a fourth order polynomial that is not easy to invert and for which the inverse transform method would be hard to apply. We thus use the acceptance-rejection method. Let $t(x) = \max_{[0,1]} f(x) = 30/16$. Using the uniform distribution as the reference density $g(x)$, we have the following algorithm

1. Draw two independent uniform random variates U_1, U_2 form $\mathcal{U}(0,1)$
2. If $U_2 \leq 16(U_1^2 - 2U_1^3 + U_1^4)$ accept $X = U_1$; otherwise, reject and go back to step 1.

Exercise:
As an exercise, you can try to verify statistically that the average number of iterations to generate one random number using this algorithm is $30/16 \approx 1.875$. Also, you can estimate the expectation $E[X] = \int_0^1 xf(x)dx$.

3.4 Polar approach for generating normal variates

The inverse and acceptance-rejection methods are universal and in theory they can be used for an arbitrary distribution. However, they both have their own limitations. As it is already mentioned above, it is not always easy for find the inverse of the CDF for the inverse method and for the acceptance-rejection method, except for the case of a bounded support, finding the easy function $t(x)$ is not always trivial. In some situations it is beneficial to design a sampling method for a specific distribution. The designed method may or may not be applicable for other distributions. Such methods are called ad hoc methods. Here we describe such a method for the normal distribution, called the polar method because it is based on the polar coordinates.

Consider the probability density function for the standard normal distribution in two dimensions.

$$f(x,y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2}.$$

This can be thought of as the joint pdf of two standard Gaussian distributions X, Y with their respective pdf's

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad f_Y(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}.$$

Let $D = R^2 = X^2 + Y^2$ and $\Theta = \arctan(Y/X)$ be two functions of the random variables X and Y. We view D, Θ as two new random variables in polar coordinates ($r = \sqrt{x^2+y^2}, \theta = \arctan(y/x)$). Let's compute the joint-cumulative probability function for D, Θ. First note that $D \geq 0$ and $\Theta \in [0, 2\pi]$. Therefore if suffices to find the probability $P(\{0 \leq D \leq d, 0 \leq \Theta \leq \theta\})$ for all $(d,\theta) \in [0,+\infty) \times [0, 2\pi]$.

$$P(\{0 \leq D \leq d, 0 \leq \Theta \leq \theta\}) = \int\int_{\{X^2+Y^2 \leq d, \quad 0 \leq \arctan(Y/X) \leq \theta\}} \frac{1}{2\pi} e^{-(x^2+y^2)/2} dxdy.$$

We express this integral in polar coordinates

$$P(\{0 \leq D \leq d, 0 \leq \Theta \leq \theta\}) = \int_0^{\sqrt{d}} \int_0^{\theta} e^{-r^2/2} r d\alpha dr = \frac{1}{2\pi} \int_0^{\theta} d\alpha \int_0^{\sqrt{d}} e^{-r^2/2} r dr.$$

The θ-integral can readily be viewed as the CDF of the uniform distribution for Θ on the interval $[0, 2\pi]$

$$P(\{0 \leq \Theta \leq \theta\}) = \frac{1}{2\pi} \int_0^{\theta} d\alpha = \frac{\theta}{2\pi}$$

while the remaining r-integral is further transformed by the change of variables $s = r^2$, yielding

$$P(\{0 \leq D \leq d\}) = \int_0^d \frac{1}{2} e^{-s/2} ds = 1 - e^{d/2},$$

which is simply the exponential distribution for D. Thus, the two random variables D, Θ are independent

$$P(\{0 \leq D \leq d, 0 \leq \Theta \leq \theta\}) = P(\{0 \leq \Theta \leq \theta\}) P(\{0 \leq D \leq d\})$$

and have much simpler distributions than X, Y: (D, Θ) can be easily sampled by the two methods described above; the inverse transform for D and (stretched) uniform variates for Θ. X, Y are then obtained by converting back to Cartesian coordinates. We obtain the following Box-Muller algorithm.

- Generate two independent uniform variates $U_1, U_2 \sim \mathcal{U}(0, 1)$
- Set $\Theta = 2\pi U_1$ and $D = -2 \ln(U_2)$
- Return $X = \sqrt{D} \cos \Theta$, and $Y = \sqrt{D} \sin \Theta$

Exercise: read example 4.8, page 237, from the book.

4 Controlling the sampling error and variance reduction techniques

Assume we use Monte Carlo integration to estimate the mean μ for a given probability distribution. We do this by building the sample mean

$$\mu \approx \bar{X}(n) = \frac{1}{n} \sum_{j=1}^{n} X_j,$$

where X_j are independent samples drawn from the given distribution. Statistically speaking, as mentioned earlier, $\bar{X}(n)$, when regarded as a random variable, is an unbiased estimate of μ since for all j we have

$$E[\bar{X}(n)] = E[X_j] = \mu.$$

However, we may wonder how good this estimate as an actual approximation for μ. We can quantify this by computing the expectation of the squared error $E[(\bar{X}(n) - \mu)^2]$, which in some sense, is a measure of the total error.

$$E[(X(n) - \mu)^2] = Var[X(n)] = \frac{\sigma^2}{n}$$

if σ is the standard deviation of our distribution. We see clearly from this inequality that as the number n of replications (or samples) increases the estimate is improved, since its variance decreases. In practice however, both μ and σ are unknown, therefore we may also rely on Monte Carlo to estimate σ

$$\sigma^2 \approx S^2(n) = \frac{1}{n}\sum_{j=1}^{n}(X_j - \bar{X}(n))^2$$

which has a certain unknown error $\sigma - S(n)$. If this error is small, then we can use the approximation $\sigma \approx S(n)$ to estimate the number n of samples required to get a good estimate of μ using the sample mean $\bar{X}(n)$.

Confidence Interval:

To control the sampling error, in practice, we rely on what is called the *confidence interval*, where a real number $\delta > 0$ is specified such that

$$\bar{X}(n) \in [\mu - \delta, \mu + \delta]$$

with a probability $(1 - \alpha)$. According the law of large numbers (combined with the central limit theorem), the confidence interval is such that

$$\delta = z_\alpha \sqrt{S^2(n)/n}$$

where z_α, called the $(1 - \alpha)$ quantile of the standard normal distribution satisfies

$$(1 - \alpha) = \int_{-\infty}^{z_\alpha} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx.$$

i.e, z_α can be computed by inverting the cumulative probability function of the standard normal distribution.

The relation $\delta = z_\alpha \sqrt{S^2(n)/n}$ suggests that the absolute error $|\bar{X}(n) - \mu|$ converges to zero on the order $1/\sqrt{n}$, when $n \longrightarrow +\infty$. This is in fact confirmed by the numerical example of an European vanilla call option, provided below.

Assume we want to control the absolute error in such a way that

$$|\bar{X}(n) - \mu| \leq \epsilon$$

with a given tolerance $\epsilon > 0$. The confidence interval states that

$$P(\{\bar{X}(n) - \delta < \mu < \bar{X}(n) + \delta\}) \approx (1 - \alpha), \quad \delta = z_\alpha \sqrt{S^2(n)/n}$$

(note that the $\approx$ here is because $S^2(n)$ is only an approximation for σ^2 and also standard normal distribution for z_α is just an approximation based on the central limit theorem).

Therefore, to obtain an estimate for μ within a tolerance ϵ, with a probability $(1 - \alpha)$, $\alpha > 0$ is small, all we need to do is sample until $\delta = z_\alpha \sqrt{S^2(n)/n} < \epsilon$ so that

$$|\bar{X}(n) - \mu| < \delta \implies |\bar{X}(n) - \mu| < \epsilon.$$

Example:
As an example we consider the pricing of an European vanilla call option. The matlab code below extends the Monte Carlo integration code used before to compute the expected pay off of an European vanilla call option to include the confidence interval (CI). It uses the matlab function *normfit* which for a given normally distributed vector of data X, it returns estimates for the mean $\hat{\mu}$, the standard deviation $\hat{\sigma}$, and the limits $CI = (CI_1, CI_2)$ of the 95% confidence interval, i.e, with a probality $1 - \alpha = 0.95$ the true mean μ satisfies

$$\hat{\mu} - CI_1 \leq \mu \leq \hat{\mu} + CI_2.$$

```
function [Price, CI] = BlsMC2(S0,K,r,T,sigma,NRepl)
% Returns estimated mean (Price) and confidence interval (CI)
nuT = ( r - 0.5*sigma^2)*T;
siT = sigma * sqrt(T) ;
DiscPayoff = exp(-r*T)*max(0,S0*exp(nuT+siT*randn(NRepl,1))-K);
[Price, VarPrice, CI] = normfit(DiscPayoff);
```

Note that *normfit* returns three fields. The expected pay off (Price), its variance (VarPrice), and the limits of the confidence interval CI, which is a 2-by-1 vector. Next the BlsMC2 function is called a few times with two different numbers of replications, NRepl=100,000 and NRepl=1,000,000. The CI vector is then used to compute some sort of a relative error; the relative length of the confidence interval $(CI_2 - CI_1)/\hat{\mu}$.

```
>>randn(’state’,0)%initializes the randn function,
                  %which generates standard normal variates
>>S0 = 50; K=55; r=0.05; T=5/12;sigma = 0.2;
>> [CallMC,CI] = BlsMC2(S0,K,r,T,sigma,100000)
CallMC =
    1.1840
CI =
    1.1666
    1.2014
>> (CI(2)-CI(1))/CallMC
ans =
    0.0294
>> [CallMC,CI] = BlsMC2(S0,K,r,T,sigma,100000)
CallMC =
    1.1838
CI =
    1.1663
    1.2013
>> (CI(2)-CI(1))/CallMC
ans =
    0.0295
>> [CallMC,CI] = BlsMC2(S0,K,r,T,sigma,100000)
CallMC =
    1.1752
CI =
    1.1578
    1.1926
>> (CI(2)-CI(1))/CallMC
ans =
    0.0297
>> [CallMC,CI] = BlsMC2(S0,K,r,T,sigma,1000000)
CallMC =
    1.1719
CI =
    1.1664
    1.1774
>> (CI(2)-CI(1))/CallMC
ans =
    0.0094
>> [CallMC,CI] = BlsMC2(S0,K,r,T,sigma,1000000)
CallMC =
    1.1650
CI =
    1.1595
    1.1704
>> (CI(2)-CI(1))/CallMC
ans =
    0.0094
>> [CallMC,CI] = BlsMC2(S0,K,r,T,sigma,1000000)
CallMC =
    1.1669
CI =
    1.1614
    1.1724
>> (CI(2)-CI(1))/CallMC
ans =
    0.0094
```

The important thing to learn from this example is that the length of the confidence interval depends on the number of replications. When the number of replications is increased from 100,000 to 1,000,000, the relative length of the confidence interval is dropped from about $\delta_1 = 0.0295$ to $\delta_2 = 0.0094$.

Note that the ratio $\delta_1/\delta_2 = 0.3241 \approx 1/\sqrt{10} = \sqrt{100,000/1,000,000}$ confirms the theoretical prediction that crude Monte Carlo has a rate of convergence on the order $O(1/\sqrt{N})$, where N is the number of replications. In fact this is rather slow. Doubling the number of replications will reduce the error by only $1/\sqrt{2}$

One way for improving the error, without taking astronomically large numbers of samples, is using what is called a variance reduction technique, which permits to reduce the sample variance (represented by $S^2(n)$ above) and thus to reduce the absolute-error bound δ. Many such techniques were developed and used by statisticians. A few of them are listed in the book. They all have their own strengths and own weaknesses but conceptually they are all similar. The idea is that instead of computing $\mu = E[X]$, using only one set of independent samples X_j, we use at least two different sets of samples, Y_j, Z_j that are associated with two **dependent** random variables Y, Z that are negatively correlated, $Cov(Y, Z) < 0$, such that

$$\mu = E[X] = E[Y + Z]$$

but

$$Var[Y + Z] = Var[Y] + Var[Z] + 2Cov(Y, Z) < Var[X].$$

Thus clearly sampling μ from the sum $Y+Z$ will yield a better approximation, with smaller absolute error, since $Y + Z$ has a smaller variance than the original random variable X. As an example, we discuss next one of the simplest variance reduction techniques, namely the *antithetic sampling* method.

Antithetic sampling

The antithetic sampling consists in generating two sets of samples $X_j^1, X_j^2, j = 1, 2, \cdots, n$ such that $\mu = E[X_j^1] = E[X_j^2], \sigma^2 = Var[X_1^j] = Var[X_j^2]$ and that $X_j^{i_1}, X_k^{i_2}$ are independent if $j \neq k$ or $i_1 \neq i_2$ $(i_1, i_2 \in \{1, 2\})$ but X_j^1, X_j^2 are dependent and satisfy

$$Cov(X_j^1, X_j^2) < 0.$$

The idea then is to set $Y_j = X_j^1/2$ and $Z_j = X_j^2/2$. Note that the sequence $Y_j + Z_j = (X_j^1 + X_j^2)/2, j = 1, 2, \cdots, n$ are i.i.d. random variables. Let

$$\bar{X}(n) = \frac{1}{n}\sum_{j=1}^{n}\frac{X_j^1 + X_j^2}{2}$$

we have

$$E[\bar{X}(n)] = E[(X_j^1 + X_j^2)/2] = (E[X_j^1] + E[X_j^2])/2 = \mu$$

i.e, $\bar{X}(n)$ provides an unbiased estimate for μ. The sample variance is given by

$$\begin{aligned} Var[\bar{X}(n)] =& \frac{Var[(X_j^1 + X_j^2)/2]}{n} = \frac{1}{n}\left(Var[X_j^1/2] + Var[X_j^2/2] + 2Cov(X_j^1/2, X_j^2/2)\right) \\ &= \frac{1}{4n}(Var[X_j^1] + Var[X_j^2] + 2Cov(X_j^1, X_j^2)) = \frac{\sigma^2}{2n}(1 + \rho(X_j^1, X_j^2)) \qquad (2) \end{aligned}$$

where $\rho(X_j^1, X_j^2) = Cov(X_j^1, X_j^2)/\sqrt{Var[X_j^1]Var[X_j^2]}$ is the normalized correlation. Thus, we get a reduced sample variance provided $-1 \leq \rho(X_j^1, X_j^2) < 0$. Note that by using the rectangular inequality $2|ab| < a^2 + b^2$ it is easy to show that $-1 \leq \rho(X_j^1, X_j^2) \leq 1$.

For a given X_j^1, the sequence X_j^2 can be chosen by exploiting possible symmetries of the given distribution. The cases of the uniform and normal distributions are discussed next.

- For the satandard uniform distribution for example we can set $X_j^1 = U_j \sim \mathcal{U}(0,1)$ and $X_j^2 = 1 - U_j$. It is easy to see in this case that X_j^1 and X_j^2 are negatively correlated.

  $$Cov(X_j^1, X_j^2) = E[X_j^1 X_j^2] - E[X_j^1]E[X_j^2] = E[U(1-U)] - E[U]E[(1-U)]$$

  since U and $1-U$ have the same (unifrom) distribution we have $E[U] = E[(1-U)] = 1/2$ and

  $$E[U(1-U)] = \int_0^1 u(1-u)du = \frac{1}{2} - \frac{1}{3} = \frac{1}{6}$$

  Thus

  $$Cov(X_j^1, X_j^2) = \frac{1}{6} - \frac{1}{4} = -\frac{1}{12} < 0$$

- For the standard normal distribution $Y \sim \mathcal{N}(0,1)$, we can set $X_1^j = Y_j$ and $X_2^j = -Y_j$. Clearly X_1^j and X_2^j are identically distributed since by symmetry we have

  $$\int_{-\infty}^{y} e^{-t^2/2} dt = \int_{-y}^{+\infty} e^{-t^2/2} dt$$

  i.e $P(\{Y \leq y\}) = P(\{-Y \leq y\}) = P(\{Y \geq -y\})$. But

  $$Cov(X_j^1, X_j^2) = E[Y(-Y)] - E[Y]E[-Y] = -E[Y^2] + E[Y]^2 = -Var[Y] = -1 < 0$$

Example

We illustrate this by the simple example of evaluating

$$I = \int_0^1 e^x dx = e - 1 \approx 1.7183.$$

Using only 100 samples, we have

```
>> rand(’state’,0) %
>> X = exp(rand(100,1));
>> [I, dummy, CI] = normfit(X);
>> I
I =
    1.7631
>> (CI(2)-CI(1))/I
ans =
    0.1089
%%Using antithetic sampling
>> U = rand(100,1);
>> X = [exp(U) + exp(1-U)]/2;
>> [I, dummy, CI] = normfit(X);
>> I
I =
    1.7104
>> (CI(2)-CI(1))/I
ans =
    0.0145
```

From this example we see clearly that antithetic sampling improves the confidence interval by a factor of about 10. However, the antithetic sampling also has its own limitations. As it is illustrated by Example 4.12, page 245, in the book, the antithetic sampling when applyed for estimating an integral

$$I = \int_a^b g(x)dx$$

as the expectation

$$I = E_f[h(X)] = \int_a^b h(x)f(x)dx$$

where f is a pdf and h is such that $h(x) = g(x)/f(x)$, one requirement is that the function h should be monotonic, in order to preserve the negative correlation between Y and Z, so that $Cov[H(Y)h(Z)]$ remains negative.

Appendix: Proof of Chebechev's Inequality

Chebechev's inequality states that, for a random variable of mean μ and variance σ^2, we have

$$\forall \epsilon > 0, P\{|(X - \mu)| > \epsilon\} = \frac{\sigma^2}{\epsilon^2}.$$

To prove this inequality we must use the more general Markov's inequality stating that if $Y \neq 0$, is a non-negative random variable then for all $d > 0$ we have

$$P\{Y \geq d\} = \frac{1}{d}E[Y].$$

Proof of Markov's inequality
Let

$$Z = \begin{cases} d & \text{if } Y \geq d \\ 0 & \text{otherwise} \end{cases}$$

We have

$$0 \leq Z \leq Y \implies E[Y] \geq E[Z] = dP\{Y \geq d\}.$$

Proof of Chebechev's inequality
Let $Y = (X - \mu)^2, d = \epsilon^2$. Using Chebechev's inequality yields

$$P\{(X - \mu)^2 > \epsilon^2\} = \frac{\sigma^2}{\epsilon^2}.$$