Chapter 1:

Random Number Generation, Simulation of Random Variates (v.A.1)

The main objective of this chapter is to generate/simulate sequences of *i.i.d.* random variables that come from discrete or continuous distributions. The sequences of generated numbers will not be truly random, they are "pseudorandom" numbers, but with sufficient care, these sequences can be made to have the most (and major) properties of random numbers. Most algorithms that will be used to generate sequences of "pseudorandom" numbers are of the form $X_{n+1} = g(X_n)$ for some function g. We will consider only linear functions g below.

Starting with a seed X_0, and constants *a, b*, and m, one can build up a sequence of "pseudorandom" numbers by the following rule:

$$X_{n+1} = (aX_n + b) \ \ modulo\ m.$$

That is, starting at X_0, the next number X_{n+1} in the sequence is constructed as the remainder of the division of a linear form $aX_n + b$ by the number m. Two most popular choices for *m* are:

(1) m is a prime number.
(2) m is a power of 2.

In general, these random number generators are referred to as *mixed or linear congruential generators*. When $b = 0$, these generators are referred to as *multiplicative congruential generators.*

One of the concerns about the above-described algorithm (to generate pseudorandom numbers) is the length of the sequence of pseudorandom numbers generated, before they repeat themselves. Thus, it is important to be able to generate a "long" sequence of numbers that also appear to be random. The following results help in searching for such algorithms.

Lemma 1 [B.J.T. Morgan]. Define the sequence of numbers $\{X_n\}_{n=0}^{\infty}$ as follows:
X_0 is given and define $X_{n+1} = (aX_n + b)\ modulo\ m$ for $n > 0$. Take $m = 2^k,\ a = 4c + 1$, and let b be odd number and *c* be a positive integer. Then, the sequence of pseudorandom numbers generated by this algorithm has a cycle length of $m = 2^k$.
This result allows one to construct sequences of numbers that have long-enough lengths, by selecting the constants.

Lemma 2. Assume m is a prime number. The *multiplicative congruential generator,* $X_{n+1} = aX_n\ modulo\ m$, where $a \neq 0$, has maximal period $(m - 1)$ if and only if, $a^k \neq 1\ modulo\ m$, for all $k = 1, 2, \ldots, m - 1$.
This result provides a necessary and sufficient condition for generating a sequence of pseudorandom numbers with the largest possible length.

Lemma 3. If $m = 2^k$ for $k \geq 3$, and if $a\ modulo\ 8 = \{3\ or\ 5\}$, and X_0 is an odd number, then the *multiplicative congruential generator* $X_{n+1} = aX_n\ modulo\ m$, where $a \neq 0$, has maximal period of 2^{k-2}.

Lemma 4. The *mixed congruential generator*, $X_{n+1} = (aX_n + b)\ modulo\ m,$ has full period m if and only if the following three conditions hold:

(i) m and b are relatively prime.
(ii) Each prime factor of m is also a factor of $(a - 1)$.
(iii) If m is a multiple of 4, then so is $(a - 1)$.

When m is a prime number, then condition (ii) along with the assumption that $a < m$ implies that m must be a multiple of $(a - 1)$, which implies that $a = 1$. Therefore, for prime number m, the only full-period generators correspond to the following value of a: $a = 1$. Prime numbers m are desirable for long periods in cases of multiplicative generators. However, in cases of mixed congruential generators, the $X_{n+1} = (X_n + b)\ modulo\ m$ generator achieves maximal period m if m is a prime number. This covers one of the popular cases: $m = 2^{31} - 1$ (which is a prime number).

For the generators $X_{n+1} = (aX_n + b)\ modulo\ m$, where $m = 2^k$ for $k \geq 2$, the condition to achieve full period 2^k requires that b be odd, and $a = 4i + 1$ for some integer i.

One of the most popular generators is the RANDU generator which is a particular case of the generator $X_{n+1} = (aX_n + b)\ \ modulo\ m$, where $a = 2^{16} + 3, b = 0, m = 2^{31}$.

Another popular example is the generator of Lewis, Goodman, and Miller (LGM). This one has been used by IBM and seems to do a satisfactory job in generation of random sequences.
The LGM algorithms is given by $X_{n+1} = 7^5 X_n\ \ modulo\ (2^{31} - 1)$. It has also been the basic random number generator for the IMSL statistical package.
What are the most important properties that sequences of pseudorandom numbers need to possess? The most important properties we would like these sequences of pseudorandom numbers to have are the following:

- *Be uniformly distributed over a certain set*,
- *Be mutually independent.*

The table below shows some popular random generators:

Table 1: ***Random Number Generators.***

m	**a**	**b**	**Name**
$2^{31} - 1$	7^5	0	(LGM) Lewis Goodman, Miller - IBM
2^{31}	$2^{16} + 3$	0	RANDU
2^{35}	5^{13}	0	APPLE
2^{32}	134,775,813	1	Turbo-Pascal
$2^{61} - 1$	$2^{19} - 1$	0	Wu-1997

We will assume that the pseudo-random sequences of numbers generated by the above algorithms satisfy the two properties above:

- are **uniformly distributed** over the set $X_n \in \{0,1, \dots, m-1\}$,
- are **mutually independent.**

See P. Glasserman, Monte Carlo Methods in Financial Engineering, for more details.

These algorithms allow users to apply a simple transformation to generate a sequence of random variates that come from $Uniform$ distribution on [0,1]. The algorithm is described below.

Generation of Uniform[0,1]:

Assume the sequence $X_n \in \{0,1,\dots,m-1\}, n = 1,2,\dots$ is an acceptable sequence of pseudorandom numbers. Define $U_n = \frac{X_n}{m}$, or $U_n = \frac{X_n+1/2}{m}$.

Then, $U_n \sim Uniform[0,1]$ and $\{U_n\}^N{}_{n=1} \sim i.i.d.\ U[0,1]$.

Notes.

1. Here, $U_n \sim Uniform[0,1]$ means that , U_n has Uniform distribution on [0,1].
2. Assume a random variable X has $Uniform[0,1]$ distribution. Then, its Cumulative Distribution Function (CDF) is given by:

$$F(x) = P(U \le x) = \begin{cases} 0, & if \quad x \le 0 \\ x, & if \quad 0 < x \le 1 \\ 1, & if \quad x > 1 \end{cases}$$

and, its probability density function (PDF) is given by:

$$f(x) = \begin{cases} 0, & if \quad x < 0 \; or \; x > 1 \\ 1, & if \quad 0 \le x \le 1 \end{cases}$$

From now on, we will assume that we are able to generate sequences of independent and identically distributed random variates from the $Uniform[0,1]$ distribution.

Next, we will try to generate sequences of random variates that come from various discretely- and continuously-distributed families of distributions.

1.1 Discrete Distributions

1.1.1 **Bernoulli** (p).

We will start with the most basic discrete distribution – the Bernoulli distribution.

The Bernoulli(p) distribution is defined as follows: Random variable X has Bernoulli(p) distribution if it takes two values:

$$X = \begin{cases} 1, & with\,prob. \quad p \\ 0, & with\,prob. \; 1-p \end{cases}$$

How to generate sequences of random variates that are $i.i.d.$ Bernoulli (p)?

The following steps will result in generation of random variates with Bernoulli(p) distribution:

STEP 1: Generate a random variable U that has $Uniform[0,1]$ distribution: $U \sim U[0,1]$

STEP 2: $if\ U < p$ then set $X = 1$, else, set $X = 0$.

Then, the random variable X will have Bernoulli (p) distribution:

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

To generate a sequence ${\{X_i\}^n}_{i=1}$ of $i.i.d.$ random variables with Bernoulli (p) distribution, the above steps are releated for a sequence of ${\{U_i\}^n}_{i=1}$:

For $i = 1$ *to n:*

{ STEP 1: Generate a random variable U_i that has $Uniform[0,1]$ distribution: $U_i \sim U[0,1]$

STEP 2: $if\ U_i < p$ then set $X_i = 1$, else, set $X_i = 0$.}

1.1.2 General Discrete Distributions

Using the same method as in the Bernoulli case above, we will generate a sequence of discretely distributed random variates with finitely many values:

$$X = \begin{cases} x_1, & with\,prob.\ p_1 \\ x_2, & with\,prob.\ p_2 \\ x_3, & with\,prob.\ p_3 \\ & \quad . \\ & \quad . \\ x_m, & with\,prob.\ p_m \end{cases}$$

where $\sum_{i=1}^{m} p_j = 1$.

The following steps will result in generation of a sequence ${X_i}^n{}_{i=1}$of $i.i.d.$ random variables with the distribution described above.

For the ith random variable in the sequence, we follow these steps:

STEP 1: Generate a random variable U_i that has $Uniform[0,1]$ distribution.

STEP 2: $if\ U_i < p_1\ then\ set\ X_i = x_1, else,\ if\ U_i < p_1 + p_2\ then\ set\ X_i = x_2,$ $else, \ldots\ ,\ if\ U_i < p_1 + p_2 + \cdots + p_{m-1}\ then\ set\ X_i = x_{m-1},\ else, set\ X_i = x_m.$

Then, X_i will have the above distribution because

$$P(X_i = x_j) = P(p_1 + \cdots + p_{j-1} \le U \le p_1 + \cdots + p_{j-1} + p_j) = p_j,$$
$$for\ any\ j = 1, \ldots, m, and\ p_0 = 0.$$

1.1.3 Binomial(n, p)

We say that the random variable X has Binomial(n, p) distribution if

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \text{ for any } k = 0, 1, \ldots, n$$

The following steps will result in generation of random variates with Binomial (n, p) distribution. We will provide two methods to generate sequences ${X_i}^n{}_{i=1}$of $i.i.d.$ random variables with the Binomial(n, p) distribution.

Method 1: We use the following recursive property: $P(X = k+1) = \frac{n-k}{k+1}\frac{p}{1-p}P(X = k)$.

Define $z = \frac{p}{1-p}\ and\ \ y = (1-p)^n$.

STEP 1: Generate a random variable U_i that has $Uniform[0,1]$ distribution.

STEP 2: Set $k = 0, x = y$.

STEP 3: if $if\ U_i < x\ then\ set\ X_i = k\ and\ exit, \quad else$

STEP 4: set $y = \frac{n-k}{k+1} zy, \quad x = x + y, \;\; k = k + 1$

STEP 5: Repeat from *STEP 3*.

The resulting sequence will have the desired distribution, which follows from its construction.

Method 2: We will use the Bernoulli-Decomposition of a Binomial(n, p) distribution provided by the following result.

Lemma: Assume the random variable X has $nomial(n, p)$distribution: $X \sim Binomial(n, p)$. Then, for any sequence $\{Y_i\}^{n}_{n=1}$ $of\ i.i.d.\ Bernoulli(p)$ random variables, X has the same distribution as $Y = Y_1 + Y_2 + \ ... + \ Y_n$.

The following steps will result in a generation of random variates with Binomial (n, p) distribution using Method 2:

STEP 1: Generate a sequence $\{U_i\}^{n}_{i=1} \sim i.i.d.\ U[0,1]$,

STEP 2: Using the numbers in STEP 1 (and the Bernoulli(p) –generation method) generate a sequence of $\{Y_i\}^{n}_{n=1} \sim i.i.d.\ Bernoulli(p)$,

STEP 3: Define $X = Y_1 + Y_2 + \ ... + \ Y_n$.

Then, X will have $Binomial(n, p)$ distribution by its construction.

Note: For the generation of one variate that is $\boldsymbol{Binomial(n, p)}$ **− distributed**, we needed to generate *n* independent random variates $\{\boldsymbol{Y_i}\}^{\boldsymbol{n}}_{\boldsymbol{n=1}} \sim \boldsymbol{i.i.d.\ Bernoulli(p)}$.

1.1.4 Poisson ($\boldsymbol{\gamma}$)

A random variable X is said to have Poisson(γ) distribution, if its distribution of X is given by $P(X = k) = \frac{e^{-\gamma}\gamma^k}{k!},\ for\ k = 0, 1, \dots, n, \dots$

The following steps will result in generation of random variates with Poisson(γ) distribution:

Method: By recursion.

Notice that, $P(X = k+1) = \frac{\gamma}{k+1} P(X = k)$. Define $z = e^{-\gamma}$.

STEP 1: Generate a random variable U that has $Uniform[0,1]$ distribution.

STEP 2: Set $k = 0, x = z$.

STEP 3: if $U < x$ then set $X = k$ and exit, else

STEP 4: Set $z = \frac{\gamma}{k+1} z,\ x = x + z, k = k + 1$

STEP 5: Repeat from STEP 3.

Then, X will have Poisson(γ) distribution, which follows from its construction.

1.2 Continuous Distributions

The generation of uniformly-distributed random variates was discussed earlier. To generate non-uniform continuous distributions, we will use with the following result.

Theorem. Let X be a continuously-distributed random variable with cumulative distribution function $F(\cdot)$ [that is, $F(x) = P(X \leq x)$]. Assume $F(\cdot)$ is an increasing and continuous function. Let U be a Uniform[0,1]-distributed random variable. Then the random variable

$Y = F^{-1}(U)$ has the same distribution as X.

Proof:

$$P(Y \leq y) = P(F^{-1}(U) \leq y) = P(F(F^{-1}(U)) \leq F(y)) = P(U \leq F(y)) = F(y) = P(X \leq y).$$

This proves the theorem.

Comment: If $F(\cdot)$ is a non-decreasing and (potentially) discontinuous function, then the inverse of the function F does not exist in the traditional sense. We will define the pseudo-inverse function of F as follows:

$$F^{-1}(y) = \min\{x: F(x) \geq y\}$$

Then, $Y = F^{-1}(U)$ has the same distribution as the random vatiable X.

1.2.1 Exponential(γ)

Definition: Random variable X has Exponential (γ) distribution, $X \sim Exp(\gamma)$, if the CDF of X is given by

$$F(y) = P(X \leq y) = 1 - e^{-\frac{y}{\gamma}} \text{ for } y \geq 0$$

Note: We have the pdf of X given by $f(x) = \frac{1}{\gamma} e^{-\frac{x}{\gamma}}$ for $x \geq 0$, and $E(X) = \gamma$.

The exponential distribution is common in many applications of probability theory. Due to its relationship to the Poisson distribution, it is used to model the *time between two events*, the occurrence of which follows a Poisson process. An example of such an event could be the default of a security in a pool.

Assume $X \sim Exp(\gamma)$ and its CDF is given by: $F(y) = P(X \leq y) = 1 - e^{-\frac{y}{\gamma}}$ for $y \geq 0$.

Using the inverse transformation method, we see that $Y = -\gamma \ln(1 - U)$also has Exp(γ) distribution. Therefore, the following two steps will allow us generate a sequence ${\{X_i\}^n}_{i=1}$of $i.i.d.$ exponentially-distributed random variates.

For the ith random variable in the sequence, we follow these steps:

> *STEP 1:* Generate a random variable U_i that has $Uniform[0,1]$ distribution.
>
> *STEP 2:* Define $X_i = -\gamma \ln(1 - U_i)$

Then, ${\{X_i\}^n}_{i=1}$ will be a sequence of $i.i.d.$ exponentially-distributed random variates.

Comment: Notice that $Y = -\gamma \ln(U) \sim Exp(\gamma)$ since U and $(1 - U)$ have the same $U[0,1]$ distribution.

1.2.2 Gamma(n, γ).

To generate a random variable X that has $\Gamma(n, \gamma)$ distribution, we will use the fact that X can be written as (has the same distribution as) a sum of *n* independent and Exponentially-distributed random variables:

$$\boldsymbol{X = Y_1 + Y_2 + \cdots + Y_n}$$

where $\{\boldsymbol{Y_i}\}^{\boldsymbol{n}}{}_{\boldsymbol{i=1}}$ are *i.i.d., and* $\boldsymbol{Y_i \sim Exp(\gamma)}$**.** Since we discussed the generation of exponentially-distributed random variables above, then the generation of $\boldsymbol{\Gamma(n,\gamma)}$ –distributed random variate follows easily as the next steps show.

STEP 1: Generate a sequence $\{U_i\}^n{}_{i=1} \sim i.i.d.\ U[0,1]$,

STEP 2: Using the numbers in STEP 1 (and the $Exp(\gamma)$–generation method) generate a sequence of $\{Y_i\}^n{}_{n=1} \sim i.i.d.\ Exp(\gamma)$.

STEP 3: Define $X = Y_1 + Y_2 + \ ... + \ Y_n$.

Then, X will have $\Gamma(n,\gamma)$ – distribution by its construction.

1.2.3 Logistic$(\boldsymbol{a},\boldsymbol{b})$

Definition: Random Variable X has Logistic distribution, $X \sim Logistic\ (a,b)$ if its SDF is given by: $P(X \le t) = \frac{1}{1+e^{-\frac{t-a}{b}}}$

We can use the inverse-transformation method to generate a variate with Logistic distribution:

$$Y = a + b\ln\left(\frac{U}{1-U}\right) \sim X \sim\ Logistic\ (a,b)\ .$$

1.2.4 Student's t

Definition: Random Variable X has X has t distribution with $\mathrm{n}-1$ degrees of freedom, $X \sim t_{n-1}$, if its density function is given by

$$f(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\frac{n}{2}\right)}\left(1+\frac{x^2}{n}\right)^{-(n+1)/2}$$

The following result will help in generation of a random variate with the student's t-distribution.

Lemma: Assume ${\{X_i\}^n}_{i=1}$ are $i.i.d.$ $N[0,1]$ random sample and $\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$ is the Sample Mean, and $S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2}$ is the Sample Standard Deviation. Then, $t = \frac{\bar{X}}{s/\sqrt{n}}$ has Student's t-distribution with $(n-1)-$ degrees of freedom.

Thus, if one can generate *i.i.d.* Standard Normally-distributed variates, then the above result will help to generate a random variate with the student's t-distribution

Problem: Suppose that ${\{U_i\}^n}_{i=1} \sim i.i.d.$ $U[0,1]$. Define $S_n = U_1 + U_2 + \ldots + U_n$. Define $N = \min\{k: S_k > 1\}$

(a) Show that $P(S_k \le x) = \frac{x^k}{k!}$ for $0 \le x \le 1$.

(b) Show that $E(N) = e$

(c) Use part (b) to estimate e by simulation (using the Law of Large Numbers).

(d) Generate (approximately) standard normally distributed random variates using S_n. (Use the Central Limit Theorem.)

The following random variables with given distributions can be generated by using the inverse transformation algorithm:

a. $Rayleigh(b)$ distribution with CDF $F(x) = 1 - e^{2x(x-b)}$ for $x \ge b$.

b. $Pereto(a,b)$ distribution with CDF $F(x) = 1 - \left(\frac{b}{x}\right)^a$ for $x \ge b > 0$.

c. $Arcsine$ distribution with CDF $F(x) = \frac{2}{\pi}\arcsin(\sqrt{x})$ for $0 \le x \le 1$.

In the next subsection we will discuss a few techniques that would allow one to generate random variates that have standard normal distribution.

1.2.5 Normal Distribution

To generate normally distributed random variates one cannot use the inverse transformation method as there is no closed-form expression for the inverse CDF function in that case. Below we consider two methods for generation of normally distributed random variables. In both cases, the outcome of the method is a pair of independent, standard normally-distributed random variables.

Method 1. Box-Muller Method.

The following result will help to generate a pair , Z_1 *and* Z_2, of $i.i.d.N(0,1)$ variates.

Theorem: Assume $\{U_i\}^2{}_{i=1} \sim i.i.d.\ U[0,1]$. Define

$$\begin{cases} Z_1 = \sqrt{-2Ln(U_1)}\cos(2\pi U_2) \\ Z_2 = \sqrt{-2Ln(U_1)}\, sin(2\pi U_2) \end{cases}.$$

Then, Z_1 *and* Z_2 are $i.i.d.N(0,1)$.

For the proof of the Theorem we need the following result.

Lemma:

(a) Let $X, Y \sim i.i.d.N(0,1)$. Define the polar coordinates of the couple (X, Y) as:

$\begin{cases} R^2 = X^2 + Y^2 \\ \theta = \arctan\frac{X}{Y} \end{cases}$. Then, $R^2 \sim Exp(2)$, and $\theta \sim U[0,2\pi]$ and R^2 is independent of θ.

(b) If $R^2 \sim Exp(2)$, and $\theta \sim U[0,2\pi]$ and R^2 is independent of θ, then, if we define

$X = R\sin\theta,\ Y = R\, cos\,\theta$, then $X, Y \sim i.i.d.N(0,1)$.

Proof of Lemma:

(a) Use the fact that $f_{R^2,\theta}(R^2,\theta) = |J| f_{X,Y}(x(R^2,\theta), y(R^2,\theta))$, where the Jacobian is

$|J| = \begin{vmatrix} \frac{1}{2\sqrt{R^2}}\cos\theta & -R\sin\theta \\ \frac{1}{2\sqrt{R^2}}\sin\theta & -R\cos\theta \end{vmatrix} = \frac{1}{2}$. The joint density of (R^2, θ) can be written as

$$f_{R^2,\theta}(d,\alpha) = \frac{1}{2}\frac{1}{2\pi}e^{-d/2} = \left(\frac{1}{2}e^{-\frac{d}{2}}\right)\left(\frac{1}{2\pi}\right) \; for\ 0 < d < \infty, and\ 0 < \alpha < 2\pi.$$

This implies that $R^2 \sim Exp(2), and\ \theta \sim U[0,2\pi]\ and\ that\ R^2\ is\ independent\ of\ \theta.$

(b) The proof is a replication of (a), in a reversed order.

Proof of Theorem: Let $U_1, U_2 \sim i.i.d.U[0,1]$. Define: $\begin{cases} R^2 = -2LnU_1 \\ \theta = 2\pi U_2 \end{cases}$. Then, it is easy to see that $R^2 \sim Exp(2)$, and $\theta \sim U[0,2\pi]$ and that R^2 is independent of θ.

From the Lemma above it follows that if we define $Z_1 = R\sin\theta, Z_2 = R\ cos\,\theta$, then

$\begin{cases} Z_1 = \sqrt{-2Ln(U_1)}\cos(2\pi U_2) \\ Z_2 = \sqrt{-2Ln(U_1)}\ sin(2\pi U_2) \end{cases}$ and $Z_1\ and\ Z_2$ are $i.i.d.\ N(0,1)$.

The Box-Muller method can be used to generate a pair of normalls as follows:

<u>*STEP 1:*</u> Generate a sequence $\{U_i\}^2_{i=1} \sim i.i.d.\ U[0,1]$,

<u>*STEP 2:*</u> Define $\begin{cases} Z_1 = \sqrt{-2Ln(U_1)}\cos(2\pi U_2) \\ Z_2 = \sqrt{-2Ln(U_1)}\ sin(2\pi U_2) \end{cases}$

Then, $Z_1\ and\ Z_2$ are $i.i.d.$ random variates that have $N(0,1)$ distribution.

Method 2. <u>Polar-Marsaglia Method</u>

The following result will help to generate a pair , $Z_1\ and\ Z_2$, of $i.i.d.N(0,1)$ variates.

Theorem: Assume $\{U_i\}^2_{i=1} \sim i.i.d.\ U[0,1]$. Define $V_1 = 2U_1 - 1, V_2 = 2U_2 - 1, and\ W = V_1^2 + V_2^2$. IF $W > 1$ then ignore the pair $\{U_i\}^2_{i=1}$. IF $W \le 1$, then define

$$\begin{cases} Z_1 = V_1 \sqrt{\dfrac{-2 \ln \mathrm{W}}{\mathrm{W}}} \\ Z_2 = V_2 \sqrt{\dfrac{-2 \ln \mathrm{W}}{\mathrm{W}}} \end{cases}$$

Then, Z_1 *and* Z_2 are $i.i.d. N(0,1)$.

Proof. If ${\{U_i\}^2}_{i=1} \sim i.i.d.\ U[0,1], then\ (V_1, V_2) \sim U[-1,1]x[-1,1]$. The algorithm suggests that we choose only those couples (U_1, U_2) that are in the unit circle: $W \leq 1$. Then, $(U_1, U_2) \sim U(in\ unit\ circle)$.

The polar coordinates of (V_1, V_2) are (R, θ), and $R \sim U[0,1]$, $\theta \sim U[0,2\pi]$, and R is independent of θ. Since

$$\begin{cases} sin\theta = \dfrac{V_2}{R} = \dfrac{V_2}{\sqrt{V_1^2 + V_2^2}} \\ cos\theta = \dfrac{V_1}{R} = \dfrac{V_1}{\sqrt{V_1^2 + V_2^2}} \end{cases}$$

Then, it follows from the results of the Box-Muller method that if we define Z_1 *and* Z_2 as follows,

$$\begin{cases} Z_1 = (-2\mathrm{Ln\ W})^{1/2}\ \dfrac{V_1}{\sqrt{V_1^2 + V_2^2}} \\ Z_2 = (-2\mathrm{Ln\ W})^{1/2}\ \dfrac{V_2}{\sqrt{V_1^2 + V_2^2}} \end{cases}$$

then, Z_1 *and* Z_2 are $i.i.d. N(0,1)$ because $(U_1, U_2) \sim U(in\ the\ unit\ circle)$.

Comment: In this method, not every pair (U_1, U_2) of uniforms is utilized in generation of standard normals. We only use those pairs that are inside the unit circle (with its center at the origin and a radius of 1). That is, the proportion of the pairs being utilized will be the probability of a random pair of uniforms $(V_1, V_2) \sim U[-1,1]x[-1,1]$ falling inside the unit circle: $V_1^2 + V_2^2 \leq 1$. This probability is $\pi/4$.

Below we provide a few important results from probability and Statistics that will help in justifying the uses of Monte Carlo Simulations to obtain estimates for unknown parameters.

Law of Large Numbers (LLN)

Suppose ${\{X_i\}^n}_{i=1} \sim i.i.d.$ sample and $E(X_i) = \mu,\ Var(X_i) = \sigma^2 < \infty$.

Define $S_n = \frac{X_1+X_2+\ldots+X_n}{n}$. Then, $\lim_{n\to\infty} E(S_n - \mu)^2 = 0$, that is $S_n \approx \mu$ for large enough n.

Central Limit Theorem (CLT)

Suppose ${\{X_i\}^n}_{i=1} \sim i.i.d.$ sample and $E(X_i) = \mu,\ Var(X_i) = \sigma^2$. Then,

$Z_n = \frac{S_n-\mu}{\sigma\sqrt{n}} \underset{n\to\infty}{\longrightarrow} Z \sim N(0,1)$. The convergence is in distribution.

1.3 Acceptance-Rejection Method

Suppose we want to generate random variates that come from a distribution with density function $f(x)$ and that it is difficult (or impossible) to invert the corresponding CDF. That is, the inverse transform method is not applicable.

Assume there exists a function $g(x)$ such that $g(x) \geq f(x)$ for any x for which $f(x) \neq 0$.

Assume $\int_{-\infty}^{\infty} g(x) < \infty$. Define $h(x) = \frac{g(x)}{\int_{-\infty}^{\infty} g(x)}$. Then, $h(\cdot)$ is a density function.

The idea behind the acceptance-rejection method is to select a function g in such a way that it is relatively easy to simulate a random variate that has a density function of h. Then, the following steps will result in generation of a random variate X that has a distribution with density function f:

STEP 1. Generatea random variate Y from the $h(\cdot)$ distribution,

STEP 2. Generate $U \sim U[0,1]$ that is independent of Y,

STEP 3. If $U \leq \frac{f(Y)}{g(Y)}$ then **ACCEPT** Y and set $X = Y$ and exit,

else, if $U > \frac{f(Y)}{g(Y)}$ **REJECT** Y and go to the next STEP,

STEP 4. Repeat Steps 1-3.

Comment: If the support of the function f is a finite interval, say $[a, b]$, then we can take

$$g(x) = \begin{cases} 0, & if \quad x < a, \ or \ x > b \\ \max\limits_{x \in [a,b]} f(x), if & \qquad a \leq x \leq b \end{cases} \qquad \text{and thus,}$$

$$h(x) = \begin{cases} 0, & if \ \ x < a, \ \ or \ x > b \\ \frac{1}{(b-a)}, & if \qquad a < x \leq b \end{cases}.$$

Proof of the Acceptance-Rejection algorithm for continuous random variables:

It is necessary to show that the conditional distribution of Y, given that $U \leq \frac{f(Y)}{g(Y)}$, is F.

Using the Bayes Theorem it follows that

$$P\left(Y \leq y \middle| U \leq \frac{f(Y)}{g(Y)}\right) = \frac{P\left(U \leq \frac{f(Y)}{g(Y)} \middle| Y \leq y\right) P(Y \leq y)}{P\left(U \leq \frac{f(Y)}{g(Y)}\right)}$$

Notice that

$$P\left(U \leq \frac{f(Y)}{g(Y)}\right) = \int_{-\infty}^{+\infty} P\left(U \leq \frac{f(Y)}{g(Y)} \middle| Y = y\right) h(y)dy \ = \int_{-\infty}^{+\infty} \frac{f(y)}{g(y)} h(y)dy \ = \frac{1}{\int_{-\infty}^{+\infty} g(x)dx} \ .$$

For notational convenience, define $c = \int_{-\infty}^{+\infty} g(x)dx$. Observe further that

$$P\left(U \le \frac{f(Y)}{g(Y)} \middle| Y \le y\right) = \frac{P\left(U \le \frac{f(Y)}{g(Y)}, Y \le y\right)}{P(Y \le y)}$$

$$= \frac{1}{H(y)} \int_{-\infty}^{y} P\left(U \le \frac{f(Y)}{g(Y)} \middle| Y = w\right) h(w)dw = \frac{1}{H(y)} \int_{-\infty}^{y} \frac{f(w)}{g(w)} h(w)dw = \frac{F(y)}{cH(y)}.$$

Therefore, it follows that

$$P\left(Y \le y \middle| U \le \frac{f(Y)}{g(Y)}\right) = \frac{\frac{F(y)}{cH(y)} H(y)}{\frac{1}{c}} = F(y).$$

This completes the proof.

Remarks: In the Acceptance-Rejection algorithm, we accept only a fraction of all generated variates, therefore one needs to evaluate the efficiency of the algorithm. The answers to the following questions will shed light on the efficiency of the algorithm:

(a) What is the probability of acceptance in this algorithm?

(b) Define X to be the time of the first "success", defined as accepting the generated number. What is E(X)?

(c) How to choose g(x) to minimize the computational cost of the algorithm?

Suggested Exercises:

1. Use the Acceptance-Rejection method to generate N(0, 1)-distributed random variates. Use Double-Exponential distribution for g: $g(x) = \frac{1}{2} e^{-|x|}$. Notice that here $\frac{f(x)}{g(x)} < c = 1.32$.

2. The answers to the following questions will shed further light on the efficiency of the Acceptance-Rejection algorithm. Define N to be the time of the first "success" – accepting the generated number.

(a) What is the probability distribution of N?

(b) Compute EN and show that E$N = c$.

(c) How to choose $g(x)$ to minimize the computational cost of the algorithm? Ideally, we would like to choose g 'close' to f to minimize the computational load. In such cases, c will be close to 1 and we will need to perform fewer iterations. However, there is a tradeoff: g being 'close' to f helps to have fewer iterations, but if it is difficult to generate variates from f, then it must be difficult to generate them from g as well.

Examples:

1. Generate $Beta(a,b)$ random variable using the Acceptance-Rejection Method.

   The density function of $Beta(a,b)$ is given by:

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} \quad for\ 0 \le x \le 1, where\ \Gamma(a) = \int_0^\infty x^{a-1}e^{-x}dx$$

Use a specific example for demonstration: $Beta(4,3)$. Then, the density function is

$f(x) = 60x^3(1-x)^2\ for\ 0 \le x \le 1$. It is easy to see that $f(x) \le 4$ for $0 \le x \le 1$ (verify this). We will take

$$g(x) = \begin{cases} 4, & for \quad 0 \le x \le 1 \\ 0, & else \end{cases}$$

and follow the algorithm.

Comments: 1. Notice that the function $g(x) = \begin{cases} 0, & else \\ 3, & for \quad 0 \le x \le 1 \end{cases}$ would also work in this case. So, the obvious question is: is there an "optimal" choice of g?

The answer is: the optimal g (among constants) is the following function:

$$g(x) = \begin{cases} max\frac{f(x)}{g(x)}, & for \quad 0 \le x \le 1 \\ 0, & else \end{cases}$$

The reasoning of this is that, the average number of trials in the algorithm until acceptance is the number that we choose in g (3 or 4 above), therefore $max f(x)$ would be the best choice to minimize that number.

1.4 Evaluating Integrals

Suppose we want to estimate the following integral which cannot be computed explicitly: $\int_0^1 f(x)dx$.

It is easy to see that the integral can be written in the following form $\int_0^1 f(x)dx = Ef(U), where\ U \sim U[0,1]$. The latter expectation can be estimated by using the Law of Large Numbers (LLN) as follows:

For a sequence of $\{U_i\}^n{}_{i=1} \sim i.i.d.\ U[0,1]$, $\boldsymbol{E}f(U) \approx \frac{f(U_1)+f(U_2)+\cdots+f(U_n)}{n}$ for large enough n. This is the main idea behind Monte-Carlo simulation. Thus, the following steps will allow one to estimate integrals:

STEP 1: Generate a sequence $\{U_i\}^2{}_{i=1} \sim i.i.d.\ U[0,1]$,

STEP 2: Estimate $\theta = \int_0^1 f(x)dx$ as $\tilde{\theta} \approx \frac{f(U_1)+f(U_2)+\cdots+f(U_n)}{n}$.

Comments:

(1) If the integral is not on the [0,1] interval, a simple rescaling would suffice to convert it to an integral on the [0,1] interval:

$$\int_a^b f(x)dx = (b-a)\int_0^1 f(a+(b-a)t)dt$$

$$\int_o^\infty f(x)dx = \int_0^1 \frac{f\left(\frac{1}{t}-1\right)}{t^2}dt$$

(2) Multi-dimensional integrals are computed similarly:

$$\int_0^1 \int_0^1 f(x,y)dx\,dy = \boldsymbol{E}f(U_1,U_2)$$

$where\ U_1, U_2 \sim U[0,1]\ and\ are\ independent.$ LLN implies that

$$Ef(U_1,U_2) \approx \frac{f(U_1^1,U_2^1) + f(U_1^2,U_2^2) + \cdots + f(U_1^n,U_2^n)}{n}$$

for large enough n.

(3) The estimator $\hat{\theta}$ is an unbiased estimator for θ because $\mathrm{E}\tilde{\theta} = \theta$.

(4) The estimator $\hat{\theta}$ is a consistent estimator for θ because $\tilde{\theta} \to \theta$ almost surely as $n \to \infty$.

1.5 The Kolmogorov-Smirnov Test

One simple test to determine whether a random number generator is yielding "close-to-random" realizations from a distribution with CDF $F(x)$ is the Kolmogorov-Smirnov test (KS test). Kolmogorov and Smirnov developed a goodness of fit test for continuous data to determine if a sample comes from a given distribution. Currently, the method continues to be one of the most widely used goodness-of-fit tests. This test is based on the empirical distribution function (EDF), which converges uniformly to the population's cumulative distribution function (CDF) with probability one (this result is based on the Glivenko-Cantelli Theorem). Even though many goodness of fit tests have been developed in recent years with higher statistical power than the KS test, the KS test remains popular because it is simple and intuitive, comparing the EDF to the CDF.

Assume we have a sample of size of n. Order the sample the following way: $x_{(1)} \le x_{(2)} \le \cdots x_{(n-1)} \le x_{(n)}$.

Define the sample distribution function S_n as follows

$$S_n(x) = \begin{cases} 0 & for \qquad x < x_{(1)} \\ \dfrac{r}{n} & for\ x_{(r)} \le x < x_{(r+1)} \\ 1 & for \qquad x_{(n)} \le x \end{cases}$$

Then, for large enough n ($n > 40$), it can be shown that IF x's are indeed from a distribution with CDF F, then, with probability $1 - \alpha$

$$|S_n(x) - F(x)| < \frac{K_\alpha}{\sqrt{n}}$$

The table below shows the critical values of the KS test.

Table 2: ***Approximate Critical Values of the Kolmogorov-Smirnov Test for Large n.***

α	0.20	0.10	0.05	0.02	0.01	0.005	0.001
K_α	1.07	1.22	1.36	1.52	1.63	1.73	1.95

To test for the uniformity and/or randomness of a sequence of generated random variates, one can use one of the following tests:

1. Runs Test – tests if a given sequence has *i.i.d.* property;
2. Serial Correlation Test – tests for independence;
3. Chi-Squared Test – tests for uniformity of a sequence;
4. Spectral Test – tests for randomness.

1.6 Exercises

1. Use the Random Number generators discussed in the class to do the following:
   (a) Generate 10,000 Uniformly distributed random numbers on [0,1] and plot the histogram of them using LGM method;
   (b) Now use built-in functions of whatever software you use to do the same thing as in (a).
   (c) Compare the histograms of the above random number sequences (in (a) and (b)) – use any method of comparison you like.
2. Use the numbers of part (a) of question 1 to do the following:
   (a) Generate 10,000 random numbers with the following distribution;

$$X = \begin{cases} -1 \; with\, prob.\, 0.3 \\ 0 \;\; with\, prob.\, 0.5 \\ 1 \;\; with\, prob.\, 0.2 \end{cases}$$

(b) Draw the histogram and the empirical distribution function by using the 10,000 numbers generated above in part (a).

3. Use the idea of part (a) of question 1 to do the following:

   (a) Generate 1,000 random numbers with Binomial distribution with $n = 44$ and $p = 0.64$. (*Hint*: A random variable with Binomial distribution (n, p) is a sum of n Bernoulli (p) distributed random variables, so you will need to generate 44,000 Uniformly distributed random numbers, to start with).

   (b) Draw the histogram. Compute the probability that X, with Binomial (44, 0.64) distribution, is at least 40.

   (c) Use any statistics textbook for the exact number for the above probability and compare them.

4. Use the numbers of part (a) of question 1 to do the following:

   (a) Generate 10,000 Exponentially distributed random numbers with parameter $\lambda = 1.5$.

   (b) Draw the histogram by using the 10,000 numbers of part (a).

5. Use the idea of part (a) of question 1 to do the following:

   (a) Generate 5,000 Uniformly distributed random numbers on [0,1].

   (b) Generate 5,000 Normally distributed random numbers with mean 0 and variance 1, by **Box-Muller** Method.

   (c) Draw the histogram by using the 5,000 numbers of part (b).

   (d) Now use the **Polar-Marsaglia** method to do the same as in (b). here you will not have the same number of RVs)

   (e) Draw the histogram by using the numbers you got in (d).

(f) Now compare the **times** it takes the computer to generate 5,000 normally distributed random numbers by the two methods. Which one is more efficient?

If you do not see a clear difference, you need to increase the number of generated realizations of random variables.

6. (a) Use the density formula for the standard normal distribution to construct the density curve. Plot it from –4 to 4 by 0.0005 step size.
   (b) Compare it to the ones obtained in question 5. Do you see any differences?

7. (a) Now use the built-in function (of the software you are using : Matlab, C/C++, VBA, etc.) to generate 5,000 Normally distributed random numbers with mean 0 and variance 1.
8. (b) Compare the histogram of these to the other three cases. Are there any biases? (You may compute the means of your sequence of numbers and compare them to 0. Ideally they must be equal to 0. Are they?)
9. Generate 10,000 of each of the following distributions using Excel and draw their histograms: Uniform[0,1], Bernoulli (0.65), Normal(0,1), Exponential(2).
10. How many uniform random variables do you have to generate on average to get 1000 normally distributed random variables via the Polar-Marsaglia method?
11. Estimate the following by using random number generation techniques:
    a. $P(U^2 + V^2 \leq 0.8)$ for U, V being iid $U[-1/2, 3/2]$
    b. $E(U^2V^2|U \leq 0.8)$ for U, V being iid $U[-1/2, 3/2]$
12. Which algorithm is faster: Box Muller or Polar Marsaglia? Discuss.
13. Let Z be a standard normal random variable. We want to generate random variables which are distributed as the absolute value of Z using the Acceptance-Rejection algorithm.
    a. Find the probability density function f of $|Z|$.
    b. Find a function $g(x)$ such that $g(x) \geq f(x)$ for any x for which $f(x) \neq 0$. (Hint: use exponential density).
    c. Formulate the Acceptance-Rejection algorithm to generate random variables distributed as $|Z|$.
    d. How can you extend this algorithm to generate standard normal random variables?
14. Suppose you have generated n independent $U[0,1]$ random variables $U1, U2, ...$ Find an algorithm for generating a random variable X, where:

a. X has the following probability density function:

$$f(x) = \frac{1}{\sqrt{2\pi}2x}\exp(-\frac{(\ln(x)-2)^2}{8}$$

b. X has the following probability density function:

$$f(x) = \frac{1}{96}x^3\exp(-\frac{x}{2}) \text{ for } x \geq 0.$$

15. Evaluate the following integral by simulation:

a. $\int_0^1 (1-x^2)^{\frac{3}{2}}dx$

b. $\int_0^1 \int_0^1 (1-x^2)^{3/2}\, e^{(x+y)^2/2}dx\, dy$