Chapter 3

Generating Uniform Random Variables

Generating Uniform Random Variables

In any kind of simulation, we need data, or we have to produce them. Especially in Monte Marco simulation we have to use random number that fits to real data as one consider the probability density function (pdf).

Fro example, I'll give you and example that we need to used a random number.

The randomized response techinque

In conducting surveys of individuals regarding an embarrassing question such as driving, sex, tax, and drugs, we make the following procedure:

- E : embarrassing question (driving offence)
- N : Non-embarrassing question that we know positive (do you like response with probability p among this population)

- 0 : Random digit simulator with probability : to answer N
- 1 : Random digit simulator with probability : to answer E

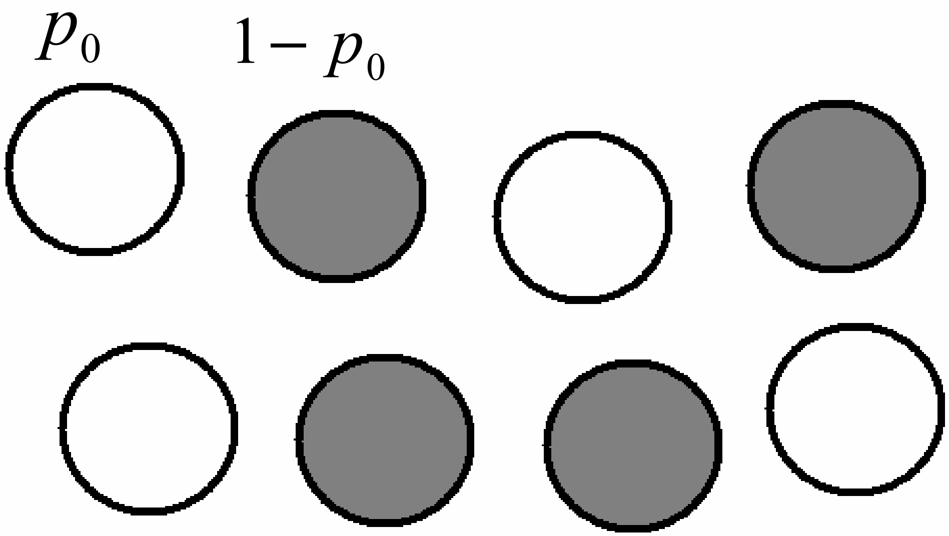
$$\Pr(res = yes) = \Pr(res = yes \mid N)\, p_0 + \Pr(res = yes \mid E)(1 - p_0)$$

- Of course the interviewer does not see the random digit . So if answer to question is yes it is not embassing for interviewee.

$$p = \Pr(res = yes \mid N) = 0.9$$

$$\Pr(res = yes) = 0.8$$

$$p_0 = 0.5$$



$$0.8 = p \times p_0 + \Pr[res = yes \mid E](1 - p_0)$$

$$0.8 = 0.9 \times 0.5 \quad + \Pr[res = yes \mid E] \times 0.5$$

$$\Pr[res = yes \mid E] = \frac{0.8 - 0.45}{0.5} = \frac{0.35}{0.5} = 0.7$$

$$p = 0.9$$

$$\Pr[res = yes] = 0.45 \quad from \quad survey$$

$$0.45 = 0.9 \times 0.5 + \Pr[res = yes \mid E] \times 0.5$$

$$\Pr[res = yes \mid E] = 0$$

- Even 45 % of survey said yes, but , we can see that no positive response is for embarrassing question .

Form this equation one can estimate when we know and and we can find form the survey.

- So, here we see how a random number is useful to conduct this problem.

In the above example, we can use a random number with any distribution. However, if we have a supply of U(0,1) (uniform random number), we can generate any distribution random number.

Dice and Machines

The simplest random number generators are coins, dice and bags of colored balls.

Thus in the RRT example above, the interviewee could be given a well-shaken bag of balls.

- : white balls with probability to answer N question.
- : Black balls with probability to answer E question.
- Other machines are used in games, such as roulette and lotteries.
- A fair coin is tossed four times. If we record a head as “0” and a tail as “1”, then the result of the experiment is four digits, abcd, for example 0110, so we can generate a random number: 6

$$R = a \times 2^3 + b \times 2^2 + c \times 2 + d$$

In our example :

0110 We can reject a number when it is larger than 9.

This is a simple method to generate a random number with uniform distribution.

Using a physical device such as dice or coin to generate random digits, one has to test the generated random number to ensure its randomness.

If dice or coin are biased, then the number may not be a true random number.

- By reading the last three digits of successive telephone numbers directory, is another generator of random numbers.
- Large scale simulations are conducted by using computers.
- Isida in 1982 presented a compact physical random number generator based on the noise of a Zener diode. In modern had- calculator now we have RND button for generating U(0,1).

Pseudo-Random Numbers :

By using a recursion formula such as :

$U_{n+1} = fractional \quad part \quad of\,(\pi + u_n)^5$ for $n \geq 0$

We can generate a random-like numbers which are generated one after each.

We say random-like because they were generated by a certain formula. We call them a Pseudo-random number. They are pseudo, because by selecting a “seed” $0 < u_0 < 1$, all numbers of sequence are determined.

They are most suitable for use on computers and calculators. Their properties could be investigated mathematically. If the result of tests are satisfied .

Then additional test is not necessary.

By applying the same seed, se can generate the same sequence, which is useful for some application.

This ‘re-run’ facility is not of course possible with physical generators such as coin or dice.

Congruential Pseudo-Random Number Generators :

An alternative mathematical representation of above formula is :

$U_{n+1} = (\pi + U_n)^5 \pmod 1$ for $n \geq 0$

In general, the recursion formula is :

$x_{n+1} = ax_n + b \pmod m$ for $n \geq 0$

x_0 : is an integer called seed.

a, b, m are integer constants.

When $b = 0$: called “multiplicative”

When $b \neq 0$: called “mixed”

- By setting $u_i = \frac{x_i}{m}$, we can generate Formula (1) may not give a true random number, however (2) is more reliable.
- Example:
- $x_0 = 89,$ a=1573, b=19, m=1000

  $$x_1 = 140016 (\bmod 10^3) = 16$$

  $$x_2 = 25187 (\bmod 10^3) = 187$$
- etc.
- for computer arithmetic, is commonly used.
- “m” should be large number. Because the formula (2) can produce no more than m different number before the cycle repeats itself.

- The maximum possible cycle length m is obtained if and only if the following three conditions are hold:

1. b and m have no common factors other than 1.
2. $(a-1)$ is a multiple of every prime number that divides m.
3. $(a-1)$ is a multiple of 4 if m is a multiple of 4.

If $m = 2^K$, then a=4c+1, b= any odd positive number.

If $m = 2^K$, for multiplicative generator the cycle length of may be obtained.

Wichmann man Hill in 1982 present a generator with a cycle length greater than. If one uses 1000 numbers/second, it would take more than 800 years for the sequence to repeat.

The choice of parameters a, b, m should be such that the generated successive numbers have small correlation.

Greenbeger has shown that an approximation to the correlation between and is given by:

$$p \approx \frac{1}{a} - \frac{6b}{am}(1 - \frac{b}{m}) \pm \frac{a}{m}$$

Two examples are:

a	b	m	p
$2^{34}+1$	1	2^{35}	0.25
$2^{18}+1$	1	2^{35}	$\prec\prec 2^{-18}$

But by choosing a, b, m to ensure small p can result in a poor generator cycle.Any way, it is worthwhile to choose constants a, b, m in order to have good generator, choosing both, full cycle and randomness test.

- Also, in generating pseudo random number, by computer, one can has to consider the arithmetic involved in operating the formula, with round off error.
- The correlation between and can be large more depending as we can see in the following example: (multiplicative)
- EX: $x_{i+1} = 5x_i \, (\text{mod} \quad m)$

This formula can be written as : $x_{i+1} = 5x_i - h_i m$

- So, there are fine lines for pair (x_{i+1} , x_i),
- Because $h_i = 0, 1, 2, 3, 4.$
- The larger, the m, the sequence will remain on one line more, before moving to another line.

If $x_0 = 1, \quad m = 11$

$$x_1 = 5\times 1 (\bmod 11) = 5 = 5x_0 - 0\times 11$$

$$x_2 = 5\times 5 (\bmod 11) = 3 = 5x_1 - 2\times 11$$

$$x_3 = 5\times 3 (\bmod 11) = 4 = 5x_2 - 1\times 11$$

$$x_4 = 5\times 4 (\bmod 11) = 9 = 5x_3 - 1\times 11$$

$$x_5 = 5\times 9 (\bmod 11) = 1 = 5x_4 - 4\times 11$$

So, each time the generated number is on a different line (so we have less dependency).

- Now, if $x_0 = 1$ but m=1000 (which is a big value) Then

$$x_1 = 5\times 1(\mathrm{mod}\,1000) = 5 = 5x_0 - 0\times 1000$$

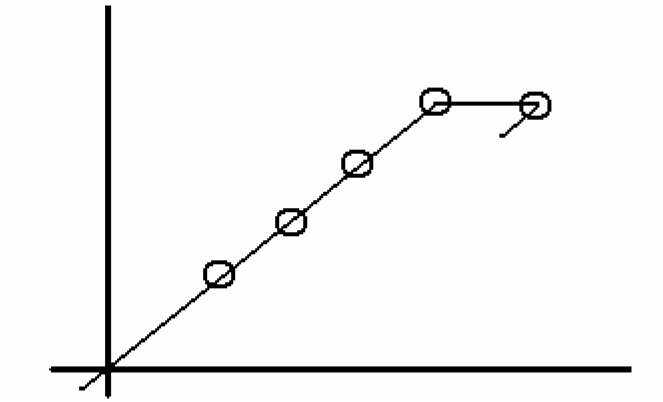
$$x_2 = 5\times 5(\mathrm{mod}\,1000) = 25 = 5x_1 - 0\times 1000$$

$$x_3 = 5\times 25(\mathrm{mod}\,1000) = 125 = 5x_2 - 0\times 1000$$

$$x_4 = 5\times 125(\mathrm{mod}\,1000) = 625 = 5x_3 - 0\times 1000$$

$$x_5 = 5\times 625(\mathrm{mod}\,1000) = 125 = 5x_4 - 3\times 1000$$

So, always are located on line And after the sequence is degenerated into a simple alternation.



In mixed congruential generator of the following example:

$$x_{n+1} = 781x_n + 387(\mathrm{mod}\,1000)$$

We have full cycle of length 1000. (Fig 3.2)

Since in this generator, we can see a kind of pattern, that may show some dependency, we prefer to shuffle them before use.

After shuffling, the result is shown in Fig. 3-2 that shows randomness.

Other method for pseudo-random number generator is given by Miller and Prentice (1968), for instance, use the third-order recurrence

$$x_j = x_{j-1} + x_{j-2} (\mathrm{mod}\ p)$$

p is a prime number

- Computer word length dependency for determining modulus m is not a desirable feature, as difficult for reproducing.
- An alternative approach is given by Wichmann and Hill (1982) who combine three multiplicative of the individual cycle-length.

$$x_{i+1} = a_1 x_i (\mathrm{mod}\, m_1)$$

$$x_i = a_2 x_{i-1} (\mathrm{mod}\, m_2)$$

$$x_{i-1} = a_3 x_{i-2} (\mathrm{mod}\, m_3)$$

A portable generator, with cycle-length is obtained. As well as in FORTRAN and HP-67 hand calculator.

In all kind of simulation, it’s important to specify the algorithm for random number. The result of simulation should be verified using different generator of random number.

- In minimal BASIC, we have two statements:

```
10  RANDOMIZE
20  U=RND
```

The first statement select a seed in a random fashion. Without that, the sequence is always the same.

Chapter 4

Particular Methods for Non-Uniform Random Variables

Here we want to use some trnasformation in order to convert uniform random variables into other distribution.

- By using control limit theorem, we can convert the uniform distribution to normal distribution.

If we simulate n independent U(0,1) random numbers , $U_1, U_2, \cdots, U_n$ then

$$N = \sum_{i=1}^{n} U_i$$

Then as $n \to \infty$ the distribution of N tends to be a normal distribution.

If $n = 2$, N has a triangular distribution.

If $n = 3$, N has a nice bell-shaped distribution.

With n=12, since $E[U_i] = \frac{1}{2}$ and $Var[U_i] = \frac{1}{12}$

Then $N = \sum_{i=1}^{12} U_i - 6$ is approximately normal random variable with $E(N) = 0, \quad Var[N] = 1$

A BASIC program is given in Fig.4.1 which is very simple.

The Box-Muller Method:

If U_1 and U_2 are two independent,
Then in 1958 (Box abd Muller) showed that

$$N_1 = (-2\log_e U_1)^{\frac{1}{2}} \cos(2\pi U_2)$$

$$N_2 = (-2\log_e U_1)^{\frac{1}{2}} \sin(2\pi U_2)$$

are two independent random variable (exactly).

- If we have two N_1 , N_2 with $N(0,1)$ and define point (N_1 , N_2) in Cartesian plane (coordinate), then in the polar coordinate we have:

$$N_1 = R\cos\theta$$

$$N_2 = R\sin\theta$$

  R, θ are two independent variables with θ having $U(0,2\pi)$ distribution and with $R^2 = N_1^2 + N_2^2$ exponential distribution of mean 2.

Then to simulate θ we need take $2\pi U_2$ and to simulate R we take $(-2\log_e U_1)^{\frac{1}{2}}$.

So, from (R,θ) we go to generate (N_1, N_2) .

$[2\pi U_2,(-2\log_e U_1)^{\frac{1}{2}}] \rightarrow [N_{1,} N_2]$

The BASIC program is given in Fig.4.1

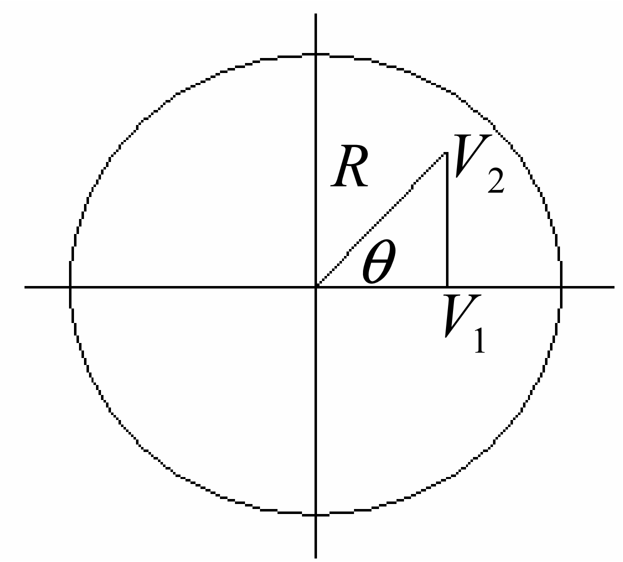
The Polar Marsaglia Method:

If $U(0,1)$ then: 2U is U(0,2) and V=2U-1 is U(-1,1)

Then specify a point at random in the sequence with :

$$R^2 = V_1^2 + V_2^2$$

$$\tan\theta = \frac{V_2}{V_1}$$



By repeating of selection of such point and rejection of points outside unit circle, the polar coordinates (R, θ) are independent with θ has $U(0,2\pi)$ and R^2 has.

So this pair is the same as required in Box-Muller method.

$$\begin{cases} \sin\theta = \dfrac{V_2}{R} = V_2(V_1^2 + V_2^2)^{-\frac{1}{2}} \\ \cos\theta = V_1(V_1^2 + V_2^2)^{-\frac{1}{2}} \end{cases}$$

So a pair of N_1, N_2 are given by,

$$\begin{cases} N_1 = (-2\log_e R^2)^{\frac{1}{2}} V_2(V_1^2 + V_2^2)^{-\frac{1}{2}} \\ N_2 = (-2\log_e R^2)^{\frac{1}{2}} V_1(V_1^2 + V_2^2)^{-\frac{1}{2}} \end{cases}$$

$$\begin{cases} N_1 = (-2\log_e(V_1^2 + V_2^2))^{\frac{1}{2}} V_2 (V_1^2 + V_2^2)^{-\frac{1}{2}} \\ N_2 = (-2\log_e(V_1^2 + V_2^2))^{\frac{1}{2}} V_1 (V_1^2 + V_2^2)^{-\frac{1}{2}} \end{cases}$$

If : $W = (V_1^2 + V_2^2)$

$$N_1 = V_2 (\frac{-2\log W}{W})^{\frac{1}{2}}$$

$$N_2 = V_1 (\frac{-2\log W}{W})^{\frac{1}{2}}$$

The rejection proportion is $1 - \frac{\pi}{4}$

A BASIC program for polar Marsaglia (Marsaglia and Bray 1964) is given in Fig.4.3. and is used in IMSL routine GGNPM .

Exponential variables

Random variables with exponential and gamma distributions are frequently used to model waiting times in queues of various kind.

The simplest way to produce an exponential PDF of e^{-x} for $x \geq 0$ is to set : $X = -\log_e U$

Where: U(0,1)

If : $Y = \frac{X}{\lambda}$

Then Y has exponential p.d.f of $\lambda e^{-\lambda Y}$

Gamma variates

To produce Gamma distributed random variables, first we choose $Y_1, Y_1, \cdots, Y_n$ with p.d.f $\lambda e^{-\lambda Y}$ for $x \geq 0.$

Then : $G = \sum_{i=1}^{n} Y_i$

has a Gamma $\Gamma(n, \lambda)$ distribution

Then, to generate Gamma from uniform distribution, we set: $$G = -\frac{1}{\lambda} \sum_{i=1}^{n} \log_e U_i$$

Where $U_1, U_1, \cdots, U_n$ are independent U(0,1).

We can also write :

$$G = -\frac{1}{\lambda} \log_e \prod_{i=1}^{n} U_i$$

Chi-square variables

For Chi-square ${X_m}^2$ distribution, we simply set $\Gamma(\frac{m}{2},\frac{1}{2})$

For m, even, we use the above approach.

For m , odd, we can obtain ${X_{m-1}}^2$, at frist,

By : $\Gamma(\frac{m-1}{2},\frac{1}{2})$

Then adding to it , where N is an independent N(0,1) normal random variable.

Both NAG and IMSL computer package use convolutions of exponential p.d.f in their routine to simulate of Gamma and Chi-square random variables.

Binomial Variables

A binomial $B(n, p)$ random variable X can be written as:

$$X = \sum_{i=1}^{n} B_i$$

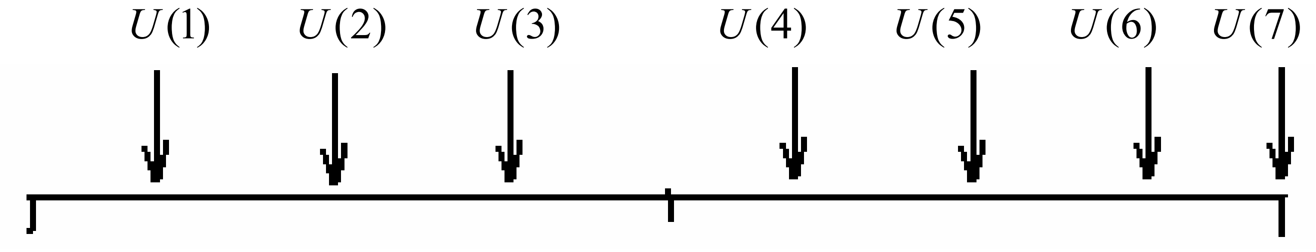
Where, the B_i are independent Bernoulli random variables, each taking the values $B_i = 1$ with probability p , or , with probability (1-p).

Thus to simulate an X, we need just to simulate n independent U(0,1) random variables, $U_1, U_1, \cdots, U_n$, and set $B_i = 1$ if $U_i \leq p$ and set $B_i = 0$ if $U_i \geq p$.

In IMSL routine, and GGBN for n<35, re-use of uniform random variable is employed

Belles (1972) simulate B(n,p) by simply count how many of U_i are less than p, (for n>35).

- If n is large, we can first ordering the $\{U_i\}$ and them observe the location of p within the ordered samples.
- For n=7 and p=0.5 we denote the ordered samples $\{U_i\}$



In this example X=3 as a realization of B(7, $\frac{1}{2}$).

If we write U in binary form to n places, the probability of 0,1 is $\frac{1}{2}$. Then we add number of 1 to generate of B(n,$\frac{1}{2}$)=9.

EX: U_1=0.10101011100101100

To generate B(n,$\frac{1}{4}$), we generate another independent

U(0,1): U_2= 0.10101101100110101

Then take place-by-place multiplication of the binary digits

0.10101001100100100

X=7 with B(17,$\frac{1}{4}$)