Journal of Al-Nahrain University Vol. 9(1), June, 2006, pp.85-90 Science

Various Techniques of Sampling Exponential Varieties by Monte Carlo Simulation

By
Dr. Akram M. AL-Abood[1], Miss Rana G. AL-Abdula[2]
[1]Dept. of Mathematics and Computer Applications, College of Science, Al-Nahrain University ,Baghdad Iraq
[2] Skeik Mohammed AL-Kasnazan University College Baghdad, Iraq

Key words: inverse transform method, order sample method, forsythe method, sequential method.

Abstract

In this paper, we introduce four methods of sampling random variaties from exponential distribution. These methods are investigated theoretically and assessed practically. Efficiency comparison is made among the methods by using a large scale Monte Carlo simulation. The advantages and disadvantages of these methods are also illustrated.

Introduction

Simulation in a " wide sense " [5] is defined as a numerical technique for conducting experiments on a digital computer, which involve certain types of mathematical or logical models that describe the system behavior, where as simulation in a " narrow sense " (also called stochastic simulation) is defined as experimenting with the model overtime, it includes sampling stochastic variaties from probability distribution. Simulation is often viewed as a " Method of Last Resort " to be used when everything else has failed, software building and technical developments have made simulation one of the most widely used and accepted tools for designer in system analysis and operational research.

Many methods and procedures are proposed for generating random varieties from different distributions [1],[2],[3],[6],[7],[8]. Four techniques are considered for generating random varieties from exponential distribution by Monte Carlo simulation namely :-

1- Inverse transform method.
2- Ordered sample method.
3- Forsythe method.
4- Sequential method.

We note that a random variable (r.v.) X is said to have an exponential distribution with parameter λ, denoted by $X \sim Exp(\lambda)$, if X has probability density function (p.d.f.) [1]

$$f(x) = \begin{cases} \frac{1}{\lambda} e^{-\frac{x}{\lambda}} , & 0 < x < \infty \\ 0 , & e.w. \end{cases} \quad (1)$$

and cumulative distribution function (c.d.f.)

$$F(x) = \Pr(X \le x) = \begin{cases} 0 , & x \le 0 \\ 1 - e^{-\frac{x}{\lambda}} , & 0 < x < \infty \\ 1 , & x = \infty \end{cases} \quad (2)$$

Inverse Transform Method, [4]

This method is the most common use for generating exponential variaties when the c.d.f. of the distribution exists and the inverse transform can be found analytically. This method is based on the following theorem :-

Theorem (1), [4]

The random variable $U=F(X) \sim U(0,1)$ if and only if , the random variable $X = F^{-1}(U)$ has c.d.f. $\Pr(X \le x) = F(x)$.

The E1-Algorithm describes the necessary steps for generating random variable by using the inverse transform method.

E1-Algorithm

1- Read λ .
2- Generate U from U(0,1).
3- Set $X = -\lambda \ln U$.
4- Deliver X as a random variaties generated from $Exp(\lambda)$.

The advantage of this method that is required n variaties from U(0,1) to generate a sample of size n from exponential distribution and the disadvanrage that it required n computation for natural logarithm of U and that include a power series expansion for each uniform variate generated which is also have time consuming.

Order Sample Method

This method is based on the proposition due to, [7]. The proposition depend on a mapping which transform the ordered sample from uniform U(0,1)- distribution to the unit exponential Exp(1).

 ISSN: 1814-5922

The theory behind the method is given by the following theorem :-

Theorem (2), [7]

Let $U_1, U_2, \cdots, U_n, U_{n+1}, \cdots, U_{2n-1}$ be a random sample of size 2n-1 from U(0,1), and let $W_1, W_2, \cdots, W_{n-1}$ be the order statistics corresponding to the random variable $U_{n+1}, U_{n+2}, \cdots, U_{2n-1}$. Assume $W_0 = 0$ and $W_n = 1$, then random variable

$$Y_k = (W_{k-1} - W_k)\ln\prod_{i=1}^{n} U_i = -(W_k - W_{k-1})\ln\prod_{i=1}^{n} U_i \quad ,k=1,2,\cdots,n$$

are independent and having Exp(1) distribution.

The E2-Algorithm describes the necessary steps for generating random variable by using the order sample method.

E2-Algorithm

1- Generate $U_1, U_2, \cdots, U_n, U_{n+1}, \cdots, U_{2n-1}$ from U(0,1).

2- Arrange $U_{n+1}, U_{n+2}, \cdots, U_{2n-1}$ in ascending order of magnitude which are represented by the order statistics $W_1, W_2, \cdots, W_{n-1}$.

3- Set $W_0 = 0, W_1 = \min(U_{n+1}, \cdots, U_{2n-1}), W_2 = 2^{nd}\min(U_{n+1}, \cdots, U_{2n-1})$ $, \cdots, W_{n-1} = \max(U_{n+1}, \cdots, U_{2n-1}), W_n = 1$.

4- Put $Y_k = -(W_k - W_{k-1})\ln\prod_{i=1}^{n} U_i \quad ,k = 1,2,\cdots,n$

5- Deliver $Y_k, k = 1,2,\cdots,n$ as a random sample of size n generated from Exp(1).

The advantage of this method that it requires one computation of the natural logarithm of the uniform variaties product for generating n exponential variaties while its disadvantage that it require the arrangement of the uniform variaties in ascending order of magnitude and then calculating the difference between two successive ordered variaties $(U_{(k-1)} - U_{(k)})$, which is also time consuming. Where it has been found practically that E2-Algorithm is faster than E1-Algorithm for n=3 until 6.

Forsythe Method

Forsythe's method is a rejection technique for sampling from a continuous distribution, which was originated to (1949) by J.N.Neumann. He presented on ingenious method for generating a sample from an exponential distribution, based on comparisons of uniform deviates alone, and suggests that his method could be modified to yield a distribution satisfying any 1st –order differential equation. Forsythe (1972) presented a general method on how the Von-Neumann idea be extended and the method efficiency is analyzed and applied on sampling from normal and other distributions. The method illustrated as follows:-

Suppose we wish to generate a random variable X having p.d.f. of the form

$$f(x) = ce^{-h(x)}, \quad x \geq 0 \quad \ldots\ldots\ldots(3)$$

where

$$c = \left[\int_0^\infty e^{-h(x)}dx\right]^{-1}$$

and h(x) is an increasing function $\forall x \in [0,\infty)$. In the first stage of the method an interval is selected for x, and in the second stage the value of x is determined within the interval by the acceptance-rejection method.

For each k=1,2,...,K (K is defined below) where we choose g_k as large as possible subject to the constraints

$$h(g_k) - h(g_{k-1}) \leq 1, \quad g_0 = 0$$

Next compute

$$r_k = \int_0^{g_k} f_X(x)dx, \quad k = 1,2,\cdots,K$$

Here the number of intervals, k, is chosen as the least index such that r_k exceeds

The largest number less than 1. (k may be chosen smaller if we set r_k), and if we are willing to truncate the generated variable by reducing any value above g_k to the interval $[g_{k-1}, g_k)$. Finally, compute

$$d_k = g_k - g_{k-1}, \quad k = 1,2,\cdots,K$$

and the function

$$G_k(x) = h(g_{k-1} + x) - h(g_{k-1}) \leq h(g_k) - h(g_{k-1}) \leq 1$$

The E3-Algorithm describes the necessary steps for generating random variable from equation (3). Steps 1 to 4 determine which interval of $[g_{k-1}, g_k)$, where the variable x will belongs to, $k = 1,2,\cdots,K$. Steps 5 to 9 determine the value of x with that interval.

E3-Algorithm

1- Set k=1.

2- Generate U from U(0,1).

3- If $U \leq r_k$, goto step 5 (the kth interval is selected).

4- If $U > r_k$, set k=k+1 and go back to step 2.

5- Generate another uniform deviate U and set $X = Ud_k$.

6- Set $t = G_k(X)$.

7- Generate $U_1, U_2, \cdots, U_N$ where N is such that

$t > U_1, t > U_2, \cdots, t > U_{N-1}$ but $t \le U_N$

($N=1$ if $t \le U_1$).

8- If N is even, reject X and return to step 2.

9- If N is odd, accept X.

we investigate Forsythe method by simulating one random variable from Exp(1) with ten independent repetitions of size 1000 and the efficiency of this method is presented in table (1).

Table (1)
Efficiencies of Forsythe Method

Trial	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
1	0.419	0.324	0.155	0.039	0.025	0.007	0.001	0	0	0
2	0.446	0.334	0.149	0.053	0.016	0.003	0.002	0	0	0
3	0.405	0.362	0.155	0.053	0.019	0.004	0.001	0.001	0	0
4	0.421	0.338	0.147	0.058	0.030	0.004	0.002	0	0	0
5	0.451	0.320	0.144	0.054	0.014	0.005	0.002	0	0	0
6	0.472	0.313	0.146	0.043	0.017	0.006	0.003	0	0	0
7	0.431	0.358	0.134	0.055	0.016	0.004	0.002	0	0	0
8	0.446	0.322	0.151	0.059	0.016	0.005	0	0	0.001	0
9	0.401	0.381	0.138	0.046	0.018	0.009	0.003	0.001	0	0
10	0.426	0.355	0.138	0.055	0.017	0.006	0.001	0.002	0	0
Mean	0.4351	0.3417	0.1457	0.0515	0.0188	0.0053	0.0017	0.0004	0.0001	0
Standard Deviation	$\pm 21646 \times 10^{-6}$	$\pm 21432 \times 10^{-6}$	$\pm 7243 \times 10^{-6}$	$\pm 6604 \times 10^{-6}$	$\pm 4917 \times 10^{-6}$	$\pm 1767 \times 10^{-6}$	$\pm 949 \times 10^{-6}$	$\pm 699 \times 10^{-6}$	$\pm 316 \times 10^{-6}$	0

The important of these efficiencies appear when k=1. From the above table, we see the method is stable on average of 0.4351 with standard deviation of $\pm 21646 \times 10^{-6}$.

The disadvantage of this method that its requires tables of the constants r_k, d_k and, g_k where it is found in the practice that the time consuming in this method is higher than the 1st and 2nd methods. And we discovered an advantage for this method that the percent of the exponential variaties generated lie in the interval $[g_{k-1}, g_k)$ decreases as the number of intervals increases, where 44% of the exponential variaties in $[g_0, g_1)$. This percent can be increased if one initially selects an optimum interval where h(x) increases but this optimality is difficult in practice.

Sequential Method

This method is based on the acceptance-rejection method for generating random variable from Exp(1) by using special type of standardized triangular distribution.

Mathematically speaking

Let $\{X_i, i = 0,1,\cdots\}$ be a sequence of identically independently distribution random variables from the standard triangular distribution. Where the distribution p.d.f.

$$f_x(x) = \begin{cases} \dfrac{2x}{\theta}, & 0 \le x \le \theta \\ \dfrac{2(1-x)}{\theta}, & 0 \le x \le 1 \end{cases} \qquad \text{..................(4)}$$

Define a random variable N, taking positive integer values through $\{X_i\}$ by the inequalities

$$X_1 < X_0,\ \sum_1^2 X_i < X_0, \cdots, \sum_1^{n-1} X_i < X_0,\ \sum_1^{n} X_i > X_0$$

we accept the sequence $\{X_i\}$ if N is odd, otherwise we reject it and repeat the process until N turns out odd. Let T be the number of sequences rejected before an odd N appears (

T=0,1,...)and let X_0 be the value of the first variable in the accepted sequence, then $Y = T + X_0$ is from Exp(1). More details given in [4].

To the best of our knowledge, the following E4-Algorithm seems to be new, where this algorithm describe the necessary steps for generating random variable from the Exp(1) by using the standardized triangular distribution.

E4-Algorithm

1- Read θ, $0<\theta<1$.

2- Set T =0.

3- Set K =0.

4- Generate U_k from U(0,1), goto step 5.

5- If $U_k < \theta$, set $Y_k = \sqrt{\theta U_k}$, if k=0, set k=k+1, goto step 4, else goto step 7.

6- Set $Y_k = 1 - \sqrt{(1-\theta)U_k}$, if k=0, set k=k+1, goto step 4, else goto step 7.

7- If $\sum_{i=1}^{k} Y_i > Y_0$, N=k, goto step 9.

8- If $\sum_{i=1}^{k} Y_i \le Y_0$, k=k+1, goto step 4.

9- Set $X = T + Y_0$.

10- If N is odd, accept X.

11- If N is even, reject X, T=T+1, goto step 3.

We investigate sequential method by simulating one random variable from Exp(1) with ten independent repetition of size 1000 with $\theta = 0.1, 0.2, 0.5$ is considered and the efficiency of the method is presented in the following tables (2), (3), and (4).

Table (2)

Efficiencies of Sequential Method ($\theta = 0.1$)

Trial	k=1	k=3	k=5	k=7	k=9	k=11	k=13	k=15	k=17	k=19
1	0.791	0.195	0.012	0.002	0	0	0	0	0	0
2	0.782	0.196	0.012	0.001	0	0	0	0	0	0
3	0.770	0.211	0.019	0	0	0	0	0	0	0
4	0.762	0.224	0.013	0.001	0	0	0	0	0	0
5	0.764	0.217	0.019	0	0	0	0	0	0	0
6	0.780	0.206	0.014	0	0	0	0	0	0	0
7	0.774	0.204	0.021	0.001	0	0	0	0	0	0
8	0.801	0.191	0.008	0	0	0	0	0	0	0
9	0.769	0.211	0.020	0	0	0	0	0	0	0
10	0.806	0.182	0.012	0	0	0	0	0	0	0
Mean	0.7799	0.2037	0.0150	0.0005	0	0	0	0	0	0
Standard Deviation	$\pm 15198 \times 10^{-5}$	$\pm 12772 \times 10^{-4}$	$\pm 4397 \times 10^{-6}$	$\pm 707 \times 10^{-6}$	0	0	0	0	0	0

The important of these efficiencies appear when k=1, we see the method is stable on average of 0.7799 with standard deviation of $\pm 15198 \times 10^{-6}$.

Table (3)
Efficiencies of Sequential Method (θ = 0.3)

Trial	k=1	k=3	k=5	k=7	k=9	k=11	k=13	k=15	k=17	k=19
1	0.830	0.170	0	0	0	0	0	0	0	0
2	0.791	0.207	0.002	0	0	0	0	0	0	0
3	0.817	0.177	0.006	0	0	0	0	0	0	0
4	0.824	0.176	0	0	0	0	0	0	0	0
5	0.826	0.172	0.002	0	0	0	0	0	0	0
6	0.812	0.185	0.003	0	0	0	0	0	0	0
7	0.816	0.181	0.003	0	0	0	0	0	0	0
8	0.817	0.182	0.001	0	0	0	0	0	0	0
9	0.811	0.185	0.004	0	0	0	0	0	0	0
10	0.806	0.190	0.004	0	0	0	0	0	0	0
Mean	0.815	0.1825	0.0025	0	0	0	0	0	0	0
Standard Deviation	$\pm$ 11146 $\times 10^{-6}$	$\pm$ 10596 $\times 10^{-6}$	$\pm$ 1900 $\times 10^{-6}$	0	0	0	0	0	0	0

The important of these efficiencies appear when k=1. From the above table, we see the method is stable on average of 0.815 with standard deviation of $\pm 11146 \times 10^{-5}$.

Table (4)
Efficiencies of Sequential Method (θ = 0.5)

Trial	k=1	k=3	k=5	k=7	k=9	k=11	k=13	k=15	k=17	k=19
1	0.959	0.041	0	0	0	0	0	0	0	0
2	0.953	0.047	0	0	0	0	0	0	0	0
3	0.952	0.048	0	0	0	0	0	0	0	0
4	0.951	0.049	0	0	0	0	0	0	0	0
5	0.956	0.044	0	0	0	0	0	0	0	0
6	0.952	0.048	0	0	0	0	0	0	0	0
7	0.954	0.046	0	0	0	0	0	0	0	0
8	0.957	0.043	0	0	0	0	0	0	0	0
9	0.965	0.035	0	0	0	0	0	0	0	0
10	0.953	0.047	0	0	0	0	0	0	0	0
Mean	0.9552	0.0448	0	0	0	0	0	0	0	0
Standard Deviation	$\pm$ 42635 $\times 10^{-7}$	$\pm$ 42635 $\times 10^{-7}$	0	0	0	0	0	0	0	0

The important of these efficiencies appear when k=1. From the above table, we see the method is stable on average of 0.9552 with standard deviation of $\pm 42635 \times 10^{-7}$. Where in practice we reach to a conclusion that there is no difference in comparison with E3 Algorithm but this method is faster than E3 Algorithm as θ increases and need only one interval for generating one random variate from unit exponential which can be seen clearly in (2), (3), and (4) tables.

Reference

[1]- Ahrens, J.H., and Dieter, U. (1972), " Computer methods for sampling from the Exponential and Normal distributions", Comm.Assoc.Comp.Mach.,15,873-882.

[2]- Ahrens, J.H., and Dieter, U. (1973), "Extensions of Forsyth's Method for Random Sampling from the Normal Distribution", Math.Comp., 27, 927-937.

[3]- Ahrens, J.H., and Dieter, U. (1974), "Computer methods for sampling from the Gamma, Beta, Poisson and Binomial distributions", Computing, 12, 223-246.

[4]- Law, A.M. and Kelton, W.D. (2000), "Simulation Modeling and Analysis", Third Edition, Mc Graw. Company.

[5]- Pllana, S. (1997), "History of Monte Carlo Method", Internet Online Search (http://www.geocities.com/College Park/Quad/2435/history.html).

[6]- Marsaglia, G. (1961), "Generating Exponential Random Variables", Ann. Math. Stat., 32, 899-900.

[7]- Rubinsten R.Y. (1981), "Simulation and the Monte Carlo Method", John Wiley and Sons. New York.

[8]- Woller, T.J. (1996), "The Basics of Monte Carlo Simulations", Internet Online Search (http://wwwitch.unl.edu/zeng/joy/mclab/mcintro.html).

المستخلص

في هذا البحث قدمنا أربعة طرق لتوليد المتغيرات العشوائية من التوزيع الاسي. هذه الطرق بحثت نظريا وطبقت عمليا. تمت مقارنة الكفاءة لهذه الطرق بأستخدام قياسات محاكاة مونت كارلو. قدمنا أيضا أستعراض لمنافع ومساوئ هذه الطرق.