

Uniform Fractional Part Algorithm And Applications

Elham Shadkam

**Department of Industrial Engineering
K.N. Toosi University of Technology, Tehran, Iran**

Abdollah Aghaie

**Department of Industrial Engineering
K.N. Toosi University of Technology, Tehran, Iran**

Abstract

In today's rapid changing world, systems are more complex than those in the past and different techniques such as simulation must be used to analyze them. Due to random existent situations in most of systems, random aspect of simulation plays a significant role in this regard. Thus, randomization can be considered in simulation models by using random variates. This paper compares one of the algorithms of generating random variates named "Uniform Fractional Part" (UFP), with other algorithms in this area. The study shows that the performance of this algorithm is much better than others with respect to speed and accuracy.

Keywords

Random Variates generation, Uniform Fractional Part, Simulation, Algorithm.

1. Introduction

Non-uniform Random variates generation is a common area among computer science, statistics, operations research and mathematics. The knowledge of random variates began about the middle of the last century. During the World War II, when feasibility of Monte Carlo experiments had been studying, the random values generation from different distributions was considered as an important area of research [1]. The applications of this field are very multifarious, as an instance, for solving various problems in Monte Carlo methods such as random optimization, Monte Carlo integration, solving linear equations, etc.[2]. Random number generators are required for generating random values from different distributions, so it's assumed random number generators with desired characteristics are available. Random numbers are the nuts and bolts of simulation machine[3]. In simulation of a system or process which there is substantial random component, producing a model for random numbers, is needed.

2. Definition of uniform fractional part algorithm and benefits

A desired random value generation algorithm should satisfy different criteria that some of them include: speed of algorithm, accuracy, simplicity and comprehensibility, range of applications, the number of used random numbers, required memory, setup and implementation speed, setup time, length of compiled code, independent of the machine, robustness, etc. The studding Method in this paper, satisfies a large number of mentioned criteria. This algorithm is a sub-category of a new generation of algorithms that is presented by Mahlooji et al. [4]. It is an approximate algorithm that can be applied for continuous distributions.

Because of high setup and low CPU time it's suitable for models that the parameters of desired distribution during the simulation process not to be changed. Application of this algorithm in different distributions confirms the accuracy, speed and simplicity in compare with other algorithms. On the other hand it can be used for order statistics and correlated random variates. In fact, this method is a kind of approximate inverse transformation method, and has all its advantages, while; UFP is faster than it. Also the inverse transformation method cannot be used for all distributions, such as gamma, beta and normal distributions, while this algorithm can generate random values of them.

This algorithm is based on the theorem of Morgan that is explained in below [5]:

Theorem 1: fractional part of sum of two independent random variable with uniform distribution on $[0,1]$ is uniform distribution on $[0,1]$. In other words, if x and R are two independent random variable in $[0,1]$ and R is defined as

$R = R + x - [R + x]$ then R will be distributed in $u[0,1]$. The generalized theorem is expressed that if x has the desired continuous distribution, random variable R will be distributed on $u[0,1]$. Random values generation algorithm from continuous distribution of x can be summarized as follows:

- The generation Two independent random values and with distribution $u[0,1]$
- The generation one value from discrete distribution $[]$ (figure 1)
- If $\geq$ then calculate $= - + []$ and otherwise calculate $= - + [] + 1$
- Give the value of x

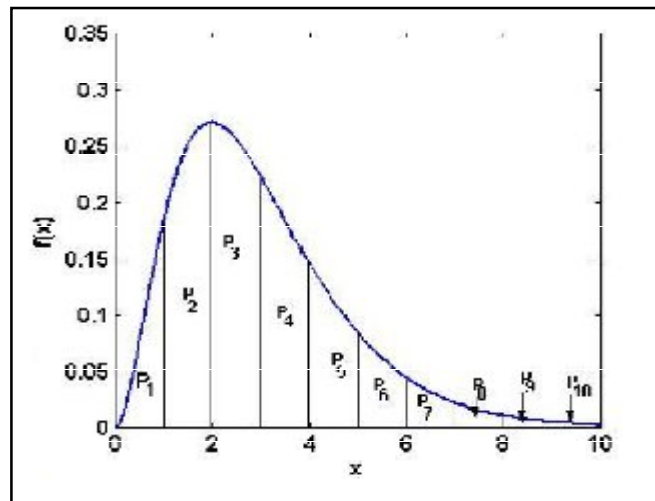


Figure 1. Generating random value from discrete distribution $[]$

This algorithm is known as Uniform Fractional Part (UFP). An essential condition of this algorithm is independent of random variables R and R , but they are dependent when one considers other distribution than uniform in $[0,1]$ for the random variable x . However specific settings on the parameters of distribution will satisfy this condition somehow. Also negative correlated random values for variation reduction techniques can be used antithetic complement of random number that is used to generating the value of $[x]$ in the second step[6].

3. Application of UFP algorithm to generate random values of gamma, beta and normal distribution

One of the most useful known distributions in simulation is gamma distribution. Shape and scale parameters of this distribution make it flexible. Gamma distribution can model a wide range of random phenomena. Random variate generation algorithms from gamma distribution often do not cover the whole range of values α . Often algorithms for generating gamma values are divided into two categories, one is for $0 \leq \alpha \leq 1$ and the other for $1 \leq \alpha$. However UFP covers the range of value α . Also the algorithm presented by Chang and Feast [7] covers the whole range of α in such a way that $\alpha \geq -\text{nez}$. UFP had been compared with other algorithms and has proved that although some algorithms are more accurate than UFP, but their accuracy changes with modifying of parameters then they aren't robust. Moreover, this algorithm is simple so that someone who is basically familiar with statistics and probability can use it. For generating random values from Gamma (α, β) distribution it is necessary to generate random value from Gamma $(\alpha, 1)$ distribution then the resulting value should be multiplied by β . The generating value from $[x]$ distribution is time consuming section of UFP algorithm and using faster algorithm can increase the speed of UFP. As mentioned before, independence condition of R and R is vital in this algorithm. But the studies on these two values show that increasing the amount of parameter β of gamma distribution can decrease the correlation between R and R and if $\alpha \geq 2$, almost there is no dependence between R and R . These results are verifiable from Monte Carlo simulation and statistical tests. Thus, the algorithm can change as follows:

- If $\alpha \geq 2$, the previous algorithm generates random values.
- If $\alpha < 2$ then algorithm is as follows:
 - Specify appropriate β
 - Generate random values from Gamma (α, β) distribution with the previous algorithm.

- Divide Resulted value to β .

The UFP had been compared with some other well known algorithms in this field, in terms of speed and accuracy[8]. Although the algorithm has highly setup time, generating random value by this algorithm is faster than other algorithms. The accuracy of algorithm measured with P-Value parameter and Kolmogorov-Smirnov test. Results from this test show that UFP is often more accurate than other algorithm.

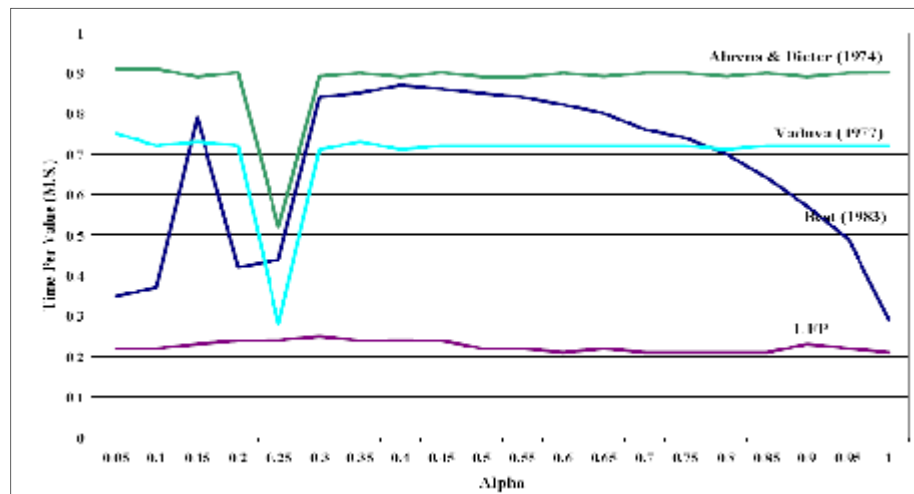


Figure 2. Required time for generating each random value (,) (< ≤)

Three of the most important distributions in the Monte Carlo simulation are beta, gamma, and normal distributions. Fortunately, UFP can be used for generating random values from all the mentioned distributions. This method is also used for normal and beta distributions and the results in compare with well known algorithms (Table 1). Based on information is demonstrated in the table, UFP is robust to the distribution parameters.

Table 1. Robustness of UFP method to change the parameters of gamma and beta distribution

Distribution	Parameters	P-value			Time (μs)
		k = 2 + 1	k = 2 + 1	k = 2 + 1	
Gamma	$\alpha = 0.1, \beta = 1$	0.32687	0.425883	0.529656	0.06
	$\alpha = 1, \beta = 1$	0.336946	0.524091	0.536118	0.05
	$\alpha = 5, \beta = 1$	0.241094	0.50367	0.536259	0.06
Beta	$\alpha = 1.5, \beta = 3$	0.373641	0.497531	0.510837	0.05
	$\alpha = 0.8, \beta = 2$	0.459897	0.498533	0.457957	0.05
	$\alpha = 0.2, \beta = 0.8$	0.164221	0.493317	0.459484	0.06

4. Application of UFP algorithm to generate correlated random values and copula

So far in this paper generation of only a single random variate from distributions at a time has been considered. By applying introduced algorithm repeatedly with independent sets of random numbers produced a sequence of IID¹ random variates from the desired distributions. In Some simulation models, someone may want to generate a random vector $X = (x_1, \dots, x_n)$ from a special joint distribution (or multivariate) which the individual components of the vector may not be independent (A indicates transpose of a vector or matrix A). In fact, sometimes correlated input random values are required for simulation process that disregarding this problem can reduce validation of simulation models. Many methods for generation correlated random values are presented which can be found in Meuwissen and Bedford in 1997 [9], Nelson and Yamnisky in 1998 [10], Kurowicka and Cooke in 2001 [11]. This

¹ Independent and Identically

issue is especially considered in Simulation of manufacturing processes and Portfolio risk in the stock market. For example Hormann and Leydold in 2010 used t-Copula model for efficient risk simulations in linear asset portfolios[12].

Another advantage of UFP method is independent random values generation. Hence one can generate two random variates with desired marginal and a pre-specified invariant measure of correlation. Generating joint function isn't necessary by this problem. This algorithm is based on the generating two uniform random values by specific correlation then transforming uniform generated vectors into arbitrary marginal distribution by inverse transformation method. If invariant correlations used, uniform values correlation is no change in transformed vector that would led to generate two random values with desired correlation and marginal distribution. Copula has been used for generating these types of values. In fact, Copula connects a multivariate distribution function with its marginal distribution. Copula was presented by Sklar in 1959 [13]. This function is defined on the square $[0,1]$ and marginal distributions are uniform distribution in $[0,1]$. Assuming H is bivariate Cumulative distribution function with marginal cumulative functions $F(x)$ and $G(x)$ and then C is available as a Copula, so that the following relationship is satisfied:

$$H(x,y) = C F(x), G(y) \quad (1)$$

Today, most copulas research is integrated risk management. Continues bivariate copula had been developed by using UFP method and then applied for generating random values of Beta $(2,\beta)$ distribution with $\beta > 1$. Algorithm for generating random values is as follow[14]:

- Determine the amount of β in such a way that satisfies the relation of Spearman correlation coefficient
- Generate X with Beta $(2, \beta)$ distribution
- Generate $U \sim [0,1]$
- $U = U + X - [U + X]$
- Return values $(F(U), G(U))$

5. Application of UFP algorithm in generating of order statistics

The order statistics from random sample $x_1, x_2, \dots, x_n$ of size n is single values sorted as ascending:

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)} \quad (2)$$

For generating random order statistics one may generate random samples and then sort them. Another method which is more efficient than previous method is using of $F(u_{(1)}), \dots, F(u_{(n)})$, where F is cumulative distribution function of x and $u_{(i)}$ is order statistic of uniform samples. UFP method is faster than the numerical inverse method or TDR methods for generating order statistics because this method doesn't need to indexed search as a part of the algorithm. As mentioned in the previous section UFP method can be used for generating beta random values as well as order statistics from beta distribution with parameters $\alpha = k$ and $\beta = n - k + 1$ and then use this value to generate $[x]$. Thus UFP method can be used for generating order statistics. Obtained values were compared from this algorithm with other well known methods and researches showed that this algorithm make out of other algorithms[15].

6. Application of UFP algorithm to generate random numbers (UFPG)

As mentioned before, random numbers are the nuts and bolts of simulation machine. Indeed, randomization creates random numbers in simulation models. UFP method can be used for generating random numbers. This method also borrows an idea from the alias method in order to form this generator. The resulted generator was named Uniform Fractional Part Generator (UFPG). UFPG has many advantages. For instance this algorithm is extremely fast, enjoys such properties as portability and reputability and the memory size requirement for this algorithm is minimal and its performance is quite well on the simple statistical test such as discrepancy, correlation and runs. Unlike linear and multiple recursive generators, which have finite period length, this generator leads to long cycle generator without any necessary condition that amounts to the imposition of a finite period length. UFPG initially takes two random numbers for seed, and then generates two independent random numbers in each iteration. The experiment shows the

length of period also may exceed 4 billion. Table 2 shows the comparison of UFGP with the MRG2 and LCG3 generators:

Table 2. Compare UFGP generator with other random number generators

Generator	Time (μ s)	Correlation test	Runs test	p-value (KS test)
UFGP	0.54	0.955	0.747	0.54
MRG	1.9	0.683	0.541	0.4
LCG	1.14	0.728	1.376	0.546
J.Carr	0.7	0.892	0.782	0.49

As it can be seen, UFGP is faster than other two generators and the average P-Value is better than the other rival generators in Kolmogorov-Smirnov test. UFGP has successfully passed randomness and run tests at mean levels 0.01 and 0.05 [16].

7. Improved uniform fractional part algorithm

There are different approaches for improving the algorithms such as: random number recycling, reduction number of random numbers, conversion approximate algorithms to exact algorithms, elimination explorer part of algorithm, etc. For example, random number recycling approach has been used for different algorithms such as numerical inverse, strip method and transformed density rejection. UFP takes three random numbers to deliver one random variate (one of these values is used for generating a value for $\text{Int}(X)$). Since random number generators aren't perfect and in fact generate pseudo-random numbers, using these values induce a substantial error of approximation. This error may increase with the number of pseudo-random numbers required for generating single variate. Therefore the number of used random numbers is a vital factor in the algorithm. This algorithm was improved and modified to an algorithm that uses just two random numbers [17].

- Generate uniform random values $u[0,1]$
- Produce a value for the $[X]$
- Calculated $X = [X] + u$

Not only this improved algorithm uses fewer number of random numbers but also it is easier and faster than others, because of its needlessness to compare U with U . Of course, this algorithm has improved again and the number of used random numbers has decreased to minimum number (just one random number) with recycling random number approach. Many reforms have been done on this algorithm that here only a brief description of these reforms was presented and the modified algorithms were discarded. For more information refer to the reference [17]. Because of the primary algorithm is approximate, it can be improved to exact algorithm by using the acceptance-rejection method. One of the other problems of this algorithm is column with integer values. Therefore the problem has been resolved in the improved algorithm and the width of the column can be decimal amount. Also, for generating $\text{Int}(X)$ and to accelerate the generation, known search methods can be used. Another approach to improve the accuracy of algorithms is to increase the number of intervals on the x-axis and it has been proved that this approach increases the accuracy because of reduced correlation between U and U .

The potential aspect of this algorithm can be further improved, of them, is the use of various acceptance-rejection methods in this regard. For example, by using the TDR algorithm, a squeeze function or normal hat function, similar to Aherence and Dieter methods [18] can be used. One of factors may cause inaccuracy of this algorithm is the infinite tails of distributions like gamma and normal distributions. By truncating these tails, algorithm can be improved. Therefore by using the ratio of uniform method, the accuracy of the algorithm may be increased. One of the sectors that can improve this algorithm is cut-off points. It can change the mode of cut-off points and then the other approaches can be used. So far this algorithm has been used for continuous distributions. Improving the algorithm probably make it applicable for discrete distributions.

² Multiple Recursive Generator

³ Linear Recursive Generator

8. Conclusion

Due to expanding the use of computer simulation, more attention has been done to different methods for generating random values; those are compatible with the computer methods. In this paper the comparison between UFP and other similar algorithms was presented. As it was appeared the performance of the UFP algorithm is much better than similar algorithms in respect to speed and accuracy. Although the algorithm has been improved, the potential advantages of this algorithm lead us for further improvement. Of them, is the use of various acceptance-rejection methods.

References

1. Hormann, W., Leydold, J., and Derflinger, G., 2004, Automatic nonuniform random variate generation, New York, Springer-Verlag.
2. Banks, J., Carson, J.S., Nelson, B.L., and Nicol, D.M., 2005, Discrete-event system simulation, Upper Saddle River, Pearson Prentice Hall.
3. Banks, J., 1998, Handbook of simulation principle, methodology, advances, applications, and practice, New York, John Wiley & Sons.
4. Izady, N., 2005, "The application of Uniform Fractional Part method in generating gamma values and correlated variables," MS. Dissertation, Sharif University of Technology.
5. Morgan, B.J.T., 1984, Elements of simulation, London, Chapman and Hall.
6. Mahlooji, H., and Izady, N., 2004, "A new method for generating gamma values," Proc. International Conference on Industrial Engineering, 294-305.
7. Cheng, R. C. H., and Feast, G. M., 1979, "Some simple gamma variate generators," appl statist, 28(3), 290-295.
8. Mahlooji, H., Jahromi, A.E., Mehrizi, H.A., and Izady, N., 2008, "Uniform Fractional Part: A simple fast method for generating continuous random variates," Scientia Iranica, 15(5), 613-622.
9. Meuwissen, A. H. M., and Bedford, T. J., 1997, "Minimal informative distributions with given rank correlation for use in uncertainty analysis," Journal of Statistical Computation and Simulation, 57, 143-157.
10. Nelson, B. L., and Yamnistky, M., 1998, In: Proceedings of the 1998 Winter Simulation Conference, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 105-112.
11. Kurowicka, D., and Cooke, R.M., 2001, In: Proceedings of the European Safety and Reliability Conference, Torino, Italy: ESREL, 1795- 1802.
12. Sak, H., Hormann, W., and Leydold, J., 2010, "Efficient risk simulations for linear asset portfolios in the t-copula model, European Journal of Operational Research," 202, 802-809.
13. Sklar, A., "Fonctions de répartition à n dimensions et leurs marges," Publ. Inst. Statist. Univ. Paris, 8, 229-231.
14. Mahlooji, H., and Izady, N., 2004, "Developing a wide class of bivariate copulas for modeling correlated input variables in stochastic simulation," Proc.35th International Conference on Computers and Industrial Engineering, 1349-1354.
15. Mahlooji, H., Mehrizi, H.A., Farzan, A., 2004, "A fast method for generating continuous order statistics based on uniform fractional part," Proc. 35th International Conference on Computers and Industrial Engineering, 1355-1360.
16. Mahlooji, H., Mehrizi, H., and Sedghi, N., An efficient, 2004, "fast and portable random number generator," Proc.35th International Conference on Computers and Industrial Engineering, 1361-1366.
17. Abuyi mehrizi, H., 2005, "The correction of Uniform Fractional Part algorithm and it's applied for generating order statistics," MS. Dissertation, Sharif University of Technology.
18. Ahrens, J.H., and Dieter U., 1982, "Generating gamma variates by a modified rejection technique," Communications of the ACM, 25, 47-54.