Scientia Iranica, Vol. 15, No. 5, pp 613–622

SCIENTIA
IRANICA

Uniform Fractional Part: A Simple Fast Method for Generating Continuous Random Variates

H. Mahlooji[1,*], A. Eshragh Jahromi[1], H. Abouee Mehrizi[1] and N. Izady[1]

A known theorem in probability is adopted and through a probabilistic approach, it is generalized to develop a method for generating random deviates from the distribution of any continuous random variable. This method, which may be considered as an approximate version of the Inverse Transform algorithm, takes two random numbers to generate a random deviate, while maintaining all the other advantages of the Inverse Transform method, such as the possibility of generating ordered as well as correlated deviates and being applicable to all density functions, regardless of their parameter values.

INTRODUCTION

Random variate generators are at the heart of any stochastic simulation. Generating samples from a diverse variety of distributions has become an established research area since World War II, when the feasibility of performing Monte Carlo experiments became a reality. The generation of non-uniform random deviates has come a long way, from methods dating back to a time prior to the era of the computer [1] to the latest novel methods, such as the Ziggurat and vertical strip [2]. While some methods are general in nature, some others are intended for a particular distribution.

Fishman [3] summarizes the milestones in the development of this field as follows: In 1951, von Neumann showed how "principles of conditional probability could be exploited for generating variates". In 1964, Marsaglia et al. demonstrated how "a synthesis of probabilistic and computer science considerations could lead to highly efficient algorithms" and, finally, in 1974, Ahrens and Dieter showed how "a bounded mean computing time could be realized for an arbitrary distribution".

Among the algorithms developed so far, some are widely used and/or are more efficient than others. For instance, while the Inverse Transform method is quite simple, if the desired cumulative distribution function cannot be expressed in a closed form, one has to resort to numerical methods, which significantly decrease the efficiency of the algorithm. If the distribution function can be stated as a convex combination of other distribution functions from which, perhaps, it is easier to generate values, the Composition method would be a competitive alternative. While these methods deal directly with the intended distribution itself, the Acceptance Rejection method targets a majorizing or hat function instead of the given density. The efficiency of this method is directly related to the chosen hat function, where identifying a 'perfect' hat function in each case has always posed as an elusive goal. There are many other algorithms in the literature, such as: Forsythe-von Neumann's Ratio of Uniforms and the like, which all can be found with in-depth analysis and examples in Devroye [4] or Fishman [3]. Hormann [5] proposed a method named the Transformed Density Rejection that can be applied to all distributions. This is a complex method and is usually time-consuming when it comes to find the hat and the squeeze functions. Some other universal random variate generators, such as the Strip method, have also been discussed thoroughly in Hormann et al. [5,6].

One of the most difficult problems in random variate generation is selecting an appropriate well-suited algorithm. Devroye [4] suggests speed, set-up time, length of compiled code, range of set of applications and simplicity as the factors for evaluating different methods. Law and Kelton [7] add exactness and robustness to this list. Exact algorithms generate variates according to the desired distribution, with the assumptions of availability of a perfect random number generator and the computer capability to store real numbers, while approximate methods need more assumptions. Robustness deals with the efficiency of

1. *Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran.*
*. *To whom correspondence should be addressed. E-mail: mahlooji@sharif.edu*

the algorithm over the entire range of distribution parameters.

In this article, a method is presented, which is applicable to the distributions of all continuous random variables. Although this algorithm belongs to the approximate category, its simplicity, speed, robustness and coverage make it a powerful competitor against exact methods, while its accuracy can be enhanced to almost any desired level. This method, which can be looked upon as a piecewise linear approximation generator, demands a rather high cost of set-up time, in the case of a rare event simulation. An added set-up time is expected also for cases when the density is somehow changed during the simulation run.

The structure of this paper is as follows: First, there will be a discussion on theoretical considerations when developing the probabilistic interpretation of the Uniform Fractional Part (UFP) method. Then, the dependence issue between the random numbers used as input to the algorithm will be elaborated on and the initial algorithm will be presented. After that, Simplifications to the initial algorithm are included and computational results are presented. Finally, the paper ends with concluding remarks and suggestions for further developments.

THEORETICAL BASIS OF THE UFP METHOD

The uniform fractional part method is based on the following theorem, which appears as an exercise on page 72 in Morgan [8]:

Theorem 1

Suppose X and U_1 to be two independent uniform $[0,1]$ random variables. Then, U_2, defined as:

$$U_2 = U_1 + X - \lfloor U_1 + X \rfloor , \tag{1}$$

is uniformly distributed over the interval $[0,1]$, where $\lfloor \bullet \rfloor$ stands for the "largest integer smaller than, or equal to $\bullet$".

There is an interesting extension to this problem (see [8] p. 72). If the continuous random variable, X, follows any distribution other than uniform $[0,1]$, U_2 is still distributed uniformly over the interval $[0,1]$.

It is tempting to infer that, based on Equation 1, one can generate values for the continuous random variable, X, with the use of two uniform random numbers, such as U_1 and U_2. Following this notion, one can isolate X in Equation 1 to get:

$$X = U_2 - U_1 + \lfloor U_1 + X \rfloor . \tag{2}$$

Since U_1 takes values between 0 and 1, then $\lfloor U_1 + X \rfloor$ is equal to either $\lfloor X \rfloor$ or $\lfloor X \rfloor + 1$. This means that one of the following two cases is relevant:

a) If $\lfloor U_1 + X \rfloor$ is equal to $\lfloor X \rfloor$, Equation 2 simplifies to:

$$X = U_2 - U_1 + \lfloor X \rfloor . \tag{3}$$

Since the fractional part of X (i.e., $X - \lfloor X \rfloor$) falls between 0 and 1, one can write:

$$0 \leq X - \lfloor X \rfloor = U_2 - U_1 \leq 1$$

$$\Rightarrow U_1 \leq U_2 \leq 1 + U_1 \Rightarrow U_2 \geq U_1. \tag{4}$$

b) If $\lfloor U_1 + X \rfloor$ is equal to $\lfloor X \rfloor + 1$, Equation 2 takes the following form:

$$X = U_2 - U_1 + \lfloor X \rfloor + 1. \tag{5}$$

Since the fractional part of X falls between 0 and 1, one can write:

$$0 \leq X - \lfloor X \rfloor = U_2 - U_1 + 1 \leq 1$$

$$\Rightarrow U_1 - 1 \leq U_2 \leq U_1 \Rightarrow U_2 \leq U_1. \tag{6}$$

Thus, Equation 2 can be rewritten as follows:

$$X = \begin{cases} U_2 - U_1 + \lfloor X \rfloor , & U_2 \geq U_1 \\ U_2 - U_1 + \lfloor X \rfloor + 1, & U_2 < U_1 \end{cases} \tag{7}$$

All this means that, in generating deviates for any continuous random variable X, three values are needed: U_1, U_2 and $\lfloor X \rfloor$. U_1 and U_2 are easily provided by random number generators, such as the one developed by Marsaglia and Tsang [9]. To generate values for $\lfloor X \rfloor$, one can use the Inverse Transform method. In fact, suppose $X \sim F_X$. If one assumes that X takes non-negative values only, then:

$$p_i = P(\lfloor X \rfloor = i) = P(i \leq X < i + 1)$$

$$= F_X(i+1) - F_X(i), \qquad i = 0, 1, 2, \cdots \tag{8}$$

Hence, $\lfloor X \rfloor$ has a discrete distribution and takes value i with probability p_i, as illustrated in Figure 1, for an arbitrary distribution. Nowadays, one can easily have access to sources that provide very accurate values of p_i for any density function once the mesh points, $\lfloor X \rfloor = i$, are specified. This obviously can also be done for the special case of $p_i = p$, such that $\sum_{\forall i} p_i = 1.0$ always holds. Now, the following algorithm can be proposed.

Algorithm 1

1. Generate U_1 and U_2 as two independent uniform $[0,1]$ random numbers,
2. Generate a value for $\lfloor X \rfloor$,

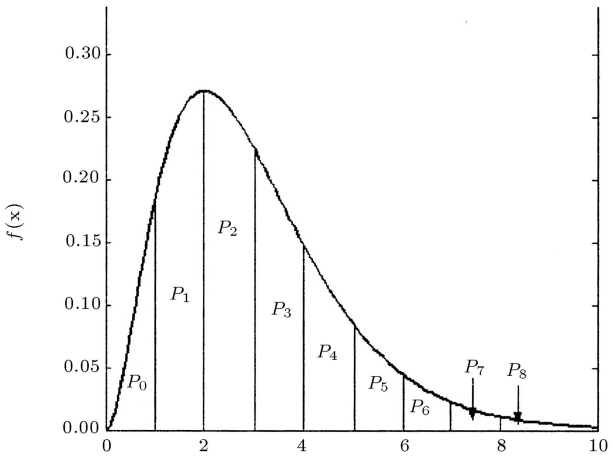


Figure 1. Probabilities associated with $\lfloor X \rfloor$.

3. Generate U_1 and U_2 as two independent uniform $[0, 1]$ random numbers,
4. Generate a value for $\lfloor X \rfloor$,
5. If $U_2 \geq U_1$, then, $X = U_2 - U_1 + \lfloor X \rfloor$; otherwise, $X = U_2 - U_1 + \lfloor X \rfloor + 1$.

This algorithm works exactly, provided that U_1 and U_2 in Equation 1 are independent. In fact, if $X \sim U[m, n]$, it can be easily shown (see the Appendix for the case of integer m and n) that U_1 and U_2 are independent. However, as will be investigated in detail, this is not true in general. Hence, in cases where U_1 and U_2 do not behave as independent random variables, this algorithm can be modified and used as an approximate method toward generating random deviates. This matter is discussed in following sections.

Up to this point, all this method does is randomly to select one of the integer values within the range of X(one of the columns in Figure 1) and then adds $U_2 - U_1$ to it, to make a random deviate. Obviously, it is quite restrictive if one is to select only integer values for X as the mesh points. So, Theorem 1 is now generalized in such a way that once needed any set of non-equidistant real values within the range of X can also be chosen and implemented in the algorithm. To achieve this purpose, first, the following definition is presented.

Definition 1

The operand, *Int*, is defined as:

$$Int\,(X + \bullet) = \begin{cases} a_i, & X + \bullet < a_{i+1} \\ a_{i+1}, & X + \bullet > a_{i+1} \end{cases} \tag{9}$$

where X is a continuous random variable taking values in the interval $[a_i, a_{i+1})$ and a_i's are distinct real values, such that $a_i < a_{i+1}, i = 1, 2, \cdots$.

Theorem 2

Suppose X is any continuous random variable and $(-\infty, a_1)$, $[a_1, a_2), \cdots, [a_k, \infty)$ is a finite partition of the range of X. Now, define A as the set of mesh points, i.e., $A = \{a_1, a_2, \cdots, a_k\}$. Also, define d_i as the distance between two successive members of A, as $d_i = a_{i+1} - a_i$, $i = 1, 2, \cdots, k-1$. Given that U_1 is a uniform $[0, d_i]$ random variable for any value of i, when X takes a value in an interval of the form $[a_i, a_{i+1})$, then, U_2 defined as:

$$U_2 = U_1 + X - Int\,(X + U_1), \tag{10}$$

is uniformly distributed over the interval $[0, d_i]$.

To prove Theorem 2, the following lemma is first presented.

Lemma 1

Given that $d_i \in \Re^+$, $x \in \Re$ and $y \in [0, d_i]$ are constant real numbers and $U \in [0, d_i)$ is a real variable for $i = 1, 2, \cdots, k-1$, then the following equation:

$$x + U - Int\,(x + U) = y, \tag{11}$$

has one, and only one, solution, such as $U = u$.

Proof

First, it is shown that Equation 11 will always have an answer. If U is isolated in Equation 11, one will arrive at:

$$U = y + Int\,(x + U) - x. \tag{12}$$

It is obvious that, for any particular i, $Int\,(x + U)$ is either equal to $Int\,(x)$ or $Int\,(x) + d_i$. So, Equation 12 can be presented as one of the following two cases:

- If $Int\,(x + U) = Int\,(x)$, then, $U = y + Int\,(x) - x \Rightarrow u = y + Int\,(x) - x$,
- If $Int\,(x + U) = Int\,(x) + d_i$, then, $U = y + Int\,(x) + d_i - x \Rightarrow u = y + Int\,(x) + d_i - x$.

Thus, one answer always exists. If there were two different solutions as:

$$u_1 = y + Int\,(x) - x,$$

and:

$$u_2 = y + Int\,(x) + d_i - x,$$

one would have:

$$0 \leq u_1 < d_i \Rightarrow 0 \leq y + Int\,(x) - x < d_i$$

$$\Rightarrow d_i \leq y + Int\,(x) + d_i - x < 2d_i$$

$$\Rightarrow d_i \leq u_2 < 2d_i, \tag{13}$$

which contradicts the assumption of $0 \leq u_2 < d_i$. A similar argument holds for u_1. Hence, Equation 11 will always have a unique solution, like $U = u$.

Now, based on Lemma 1, an effort will be made to prove Theorem 1. Bearing in mind that, for the case of any continuous random variable like X one can write:

$$P(X = x) = P\left(\lim_{h \downarrow 0} x \leq X < x + h\right)$$

$$= \lim_{h \downarrow 0} P(x \leq X < x + h) \cong \lim_{h \downarrow 0} h.f_X(x)$$

$$\Rightarrow f_x(x) \cong \frac{P(X = x)}{h}.$$

One can find the density of U_2 conditioned on the interval in which X falls as:

$$\Pr\{U_2 = u_2 | X \in [a_i, a_{i+1})\}$$

$$= \Pr\{X + U_1 - Int\,(X + U_1)$$

$$= u_2 | X \in [a_i, a_{i+1})\}$$

$$= \int_{a_i}^{a_{i+1}} \Pr\{X + U_1 - Int\,(X + U_1)$$

$$= u_2 | X = x,$$

$$X \in [a_i, a_{i+1})\} d(\Pr\{X \leq x | X \in [a_i, a_{i+1})\})$$

$$= \int_{a_i}^{a_{i+1}} \Pr\{x + U_1 - Int\,(x + U_1)$$

$$= u_2\} \frac{f_X(x)}{\Pr\{X \in [a_i, a_{i+1})\}} dx. \quad (14)$$

By considering Lemma 1, which states that Equation 11 always has the unique solution, $U = u$, Equation 14 simplifies to:

$$\Pr\{U_2 = u_2 | X \in [a_i, a_{i+1})\}$$

$$= \int_{a_i}^{a_{i+1}} P\{U_1 = u_1\} \frac{f_X(x)}{\Pr\{X \in [a_i, a_{i+1})\}} dx. \quad (15)$$

Now, suppose Δh is a small positive value, then:

$$\Pr\{U_2 = u_2 | X \in [a_i, a_{i+1})\}$$

$$\cong \frac{1}{\Pr\{X \in [a_i, a_{i+1})\}} \int_{a_i}^{a_{i+1}} (\Delta h f_{U_1}(u_1)) f_X(x) dx$$

$$= \frac{1}{d_j \Pr\{X \in [a_i, a_{i+1})\}} \int_{a_i}^{a_{i+1}} \Delta h f_X(x) dx$$

$$= \frac{\Delta h}{d_i}. \quad (16)$$

So, one has:

$$\frac{P(U_2 = u_2 | X \in [a_1, a_{i+1}))}{\Delta h} \cong \frac{1}{d_i}$$

$$\Rightarrow f_{U_2}(u_2 | X \in [a_i, a_{i+1})) = \frac{1}{d_i}, \quad (17)$$

when $\Delta h \to 0$. Therefore, U_2 is uniformly distributed over the interval $[0, d_i]$ and Theorem 2 is proved.■

Note that, if A consists of integer values, $Int\,(x)$ will result in the same values as $\lfloor X \rfloor$ and Theorem 1 becomes a special case of Theorem 2, with $d_i = 1$, $i = 1, 2, \cdots, k-1$. In general, Equation 8 takes the form:

$$p_i = P(Int\,(X) = a_i) = P(a_i \leq X < a_{i+1})$$

$$= F_X(a_{i+1}) - F_X(a_i), \qquad i = 1, \cdots, k-1. \quad (18)$$

In analogy to Algorithm 1, one will arrive at the following Algorithm.

Algorithm 2

1. Generate a value for $Int\,(X)$ and determine the value of i;
2. Generate U_1 and U_2 as two independent uniform $[0, d_i]$ random variates, where d_i denotes the length of the interval into which X falls;
3. If $U_2 \geq U_1$, then $X = U_2 - U_1 + Int\,(X)$, otherwise $X = U_2 - U_1 + Int\,(X) + d_i$.

Algorithm 2 relaxes the restriction of using only integer values for X. As will be discussed later on, this would help significantly when U_1 and U_2 show strong dependence.

INVESTIGATING THE DEPENDANCE BETWEEN U_1 AND U_2

Investigating the nature of the relation between the uniformly distributed random variables, U_1 and U_2, can be quite intriguing. To address this issue, first, the authors resort to a number of experiments in which they initially assume a uniform partition of the range of X and, hence, a constant mesh size $d_i = d, \forall i$. For the sake of experiments presented here, it is assumed that X is distributed according to Gamma (Figures 2, 3, 4 and 6) or Normal (Figure 5). Then, by Monte Carlo sampling on a computer, samples of arbitrary size are generated from distributions of the independent random variables, U_1 and X. Each time a pair of values (u_1, x) is generated, the value u_2 is computed according to Equation 9. In this way, a sample of size, say, n, is generated for (U_1, U_2). By plotting the values (u_1, u_2) as points inside the square, $[0, d]^2$, one

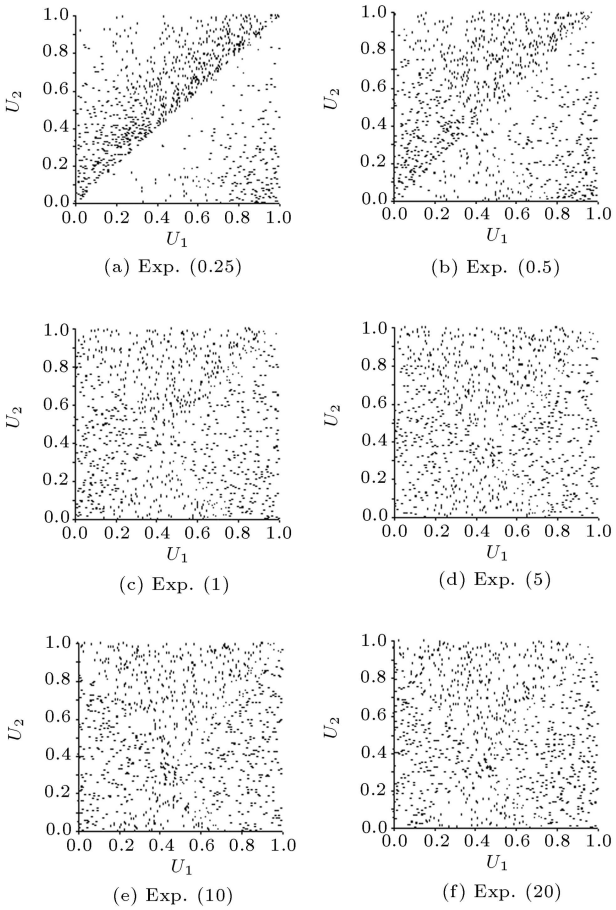


Figure 2. Scatter diagrams for (U_1, U_2), $d = 1$ and $X \sim$ exponential (β).

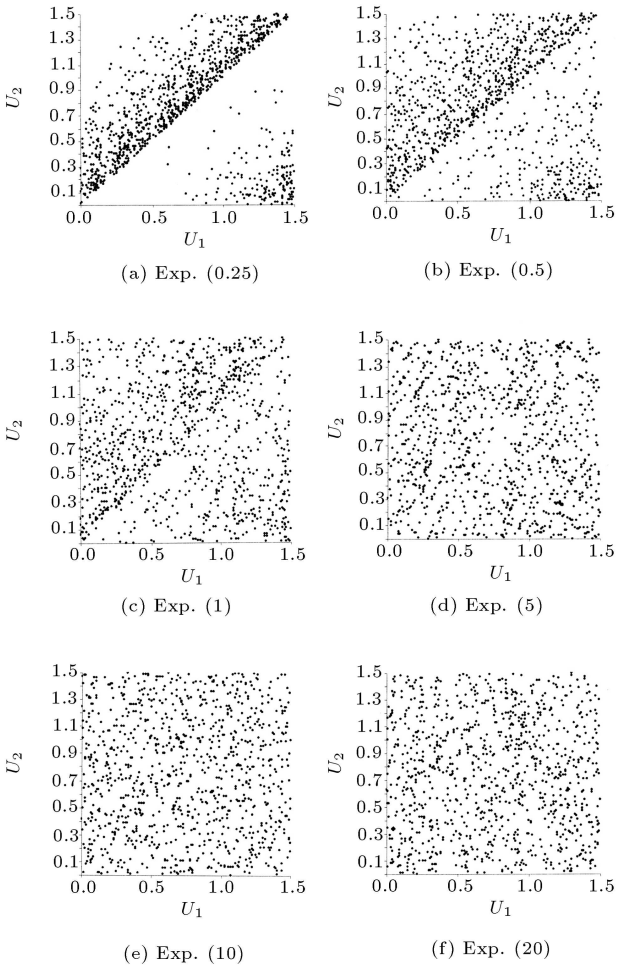


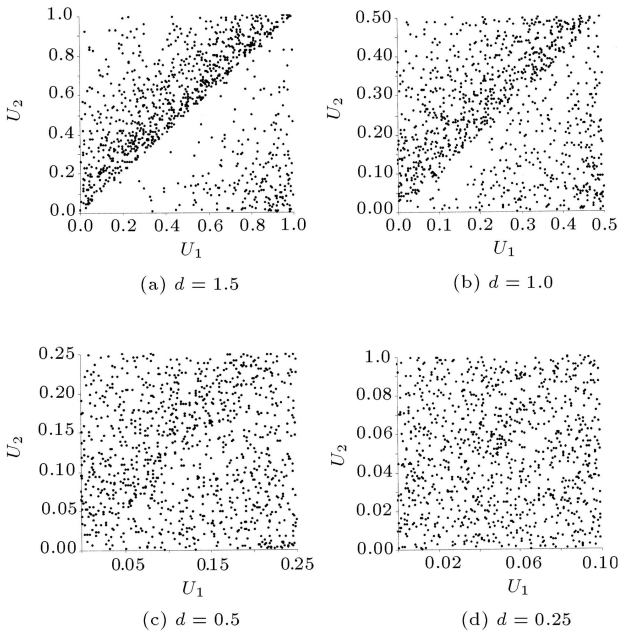
Figure 3. Scatter diagrams for (U_1, U_2), $d = 1.5$ and $X \sim$ exponential (β).

can make some interesting observations. As can be seen in Figure 2, for a uniform spacing with $d = 1$, when dispersion in the distribution of X is small, clear patterns between U_1 and U_2 can be observed in each plot, such that it can be concluded that U_1 and U_2 are not behaving independently. Figure 3 displays the same behavior for $d = 1.5$. Figure 4 shows that instead of increasing the dispersion one can define a more compact set of mesh points (i.e., decreasing the value of d), in order to make the patterns fade away. When one considers other distributions for the random variable X and manipulates the parameter(s) of each distribution with the aim of increasing the dispersion, similar observations are made (Figures 5 and 6). At this point, it is tempting to loosely interpret a 'no visible pattern' situation as indicative of the independence of U_1 and U_2 on $[0, d]^2$.

Even though such experiments shed light on the dependence structure of U_1 and U_2, it was decided that it would be more convincing if one were somehow able to measure the dependence between U_1 and U_2. There are several correlation notions in the statistics literature, such as the Pearson linear correlation, Spearman's ρ and Kendall's τ. All these measures have a substantial drawback for the purposes of this paper: Being zero does not necessarily imply independence between the random variables involved. Devroye ([4], page 575) introduces a good measure of dependence for continuous marginals, which, for the special case at hand, is defined as:

$$L = \frac{1}{2} \int |f(u_1, u_2) - f_1(u_1) f_2(u_2)| du_1 du_2, \qquad (19)$$

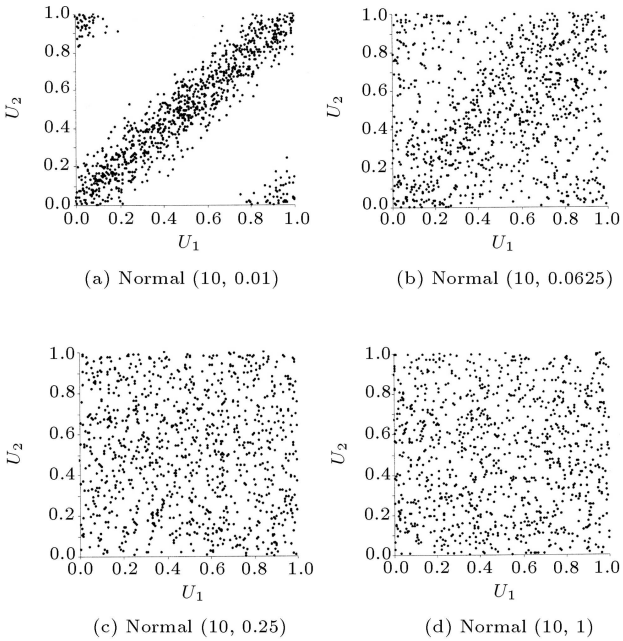
where f_1 and f_2 are marginal densities of U_1 and U_2, respectively, and $f(u_1, u_2)$ stands for their joint density function. The fact is that, if this measure of dependence between two random variables is zero, then the random variables are independent and, conversely, if the random variables are independent, then $L = 0$ holds true. Thus, U_1 and U_2 are independent if, and only if, $L = 0$. In cases where L cannot be found explicitly, one can try to calculate its unbiased estimator, based on a sample of size n composed of observations such as (u_{1i}, u_{2i}), through the following expression:



(a) $d = 1.5$ (b) $d = 1.0$

(c) $d = 0.5$ (d) $d = 0.25$

Figure 4. Scatter diagrams for (U_1, U_2), different values of d and $X \sim$ exponential (0.25).



(a) Normal (10, 0.01) (b) Normal (10, 0.0625)

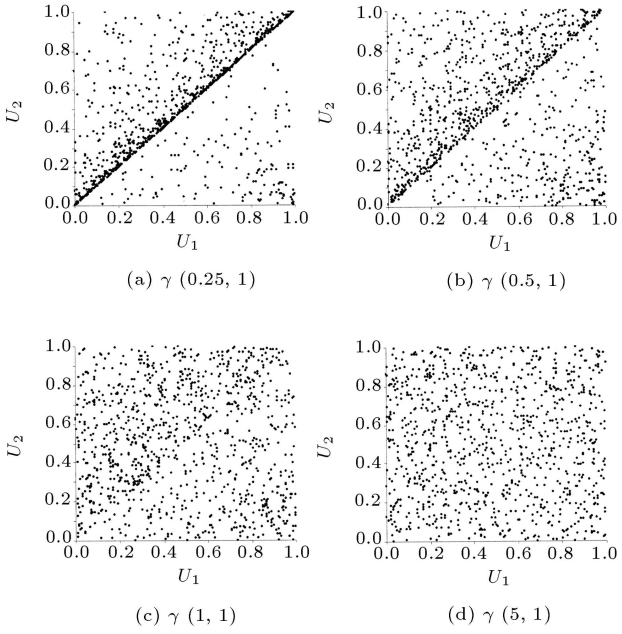
(c) Normal (10, 0.25) (d) Normal (10, 1)

Figure 5. Scatter diagrams for (U_1, U_2), $d = 1$ and $X \sim$ normal $(10, \sigma^2)$.

$$\hat{L} = \frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - \frac{f_1(u_{1i}) f_2(u_{2i})}{f(u_{1i}, u_{2i})}\right), \qquad (20)$$

where f_1, f_2 and $f(u_1, u_2)$ are estimated via methods such as Kernel Density Estimation [5].

For special cases, where X is exponentially distributed with mean β and for an arbitrary partition of the range of X, this measure of association has been derived and has been shown to be equal to:



(a) γ (0.25, 1) (b) γ (0.5, 1)

(c) γ (1, 1) (d) γ (5, 1)

Figure 6. Scatter diagrams for (U_1, U_2), $d = 1$ and $X \sim$ Gamma $(\alpha, 1)$.

$$L = \frac{(d_i p_i - 2\beta p_i(\ln(d_i) - \ln(\beta) + 1 - \ln(p_i)) + 2a_i p_i + d_i e^{-\frac{(a_i + d_i)}{\beta}} + e^{-\frac{a_i}{\beta}} d_i)}{2p_i d_i}, \qquad (21)$$

given that X falls in an interval of the form $[a_i, a_{i+1})$, p_i is defined as in Equation 18 and d_i is defined as in Theorem 2. The plot of L versus different values of β and d_i is presented in Figure 7. It is observed that L tends to zero as β goes to infinity or as d_i approaches zero.

Note that, because of the difficulty in deriving L, or even estimating it when X follows other common distributions, other means were relied upon to justify the conditions under which U_1 and U_2 could be considered as independent random variables. For

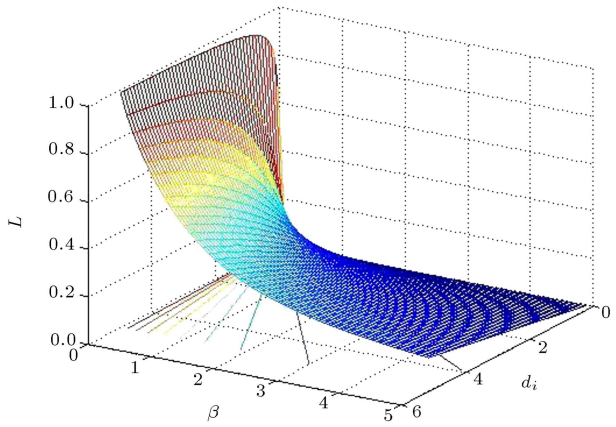


Figure 7. L vs. β and d_i.

instance, since, when the joint density of U_1 and U_2 is uniform on the unit square, then U_1 and U_2 will be independent (see, e.g. [4] p. 576). The following hypothesis has been tested by using the chi-square statistic for different densities and a wide range of their parameter values. For this purpose, when $d = 1$, the unit square was divided into 100 squared cells of equal area and, each time, samples of 1000 (u_1, u_2)'s were generated, as described above. The difference between the expected and observed number of points in each cell was calculated through the $\chi^2 = \sum_{i=1}^{10}\sum_{j=1}^{10} \frac{(y_{ij}-10)^2}{10}$ statistic, where y_{ij} stands for the number of observed points actually falling inside the (i, j) cell. In this way, one will have a chi-square statistic with 99 degrees of freedom.

$$\begin{cases} H_0 : f(u_1, u_2) = 1 \\ H_1 : \text{Otherwise} \end{cases} \tag{22}$$

Based on the values taken by the test statistic, the null hypothesis may not be accepted at first. But, by gradually increasing the dispersion of the density of X and by feeding the same streams of random numbers to generate values for U_1 and X, the values assumed by the test statistic improved to a point where H_0 could no longer be rejected. Increasing the dispersion even further, only amounts to accepting H_0 by ever-increasing p-values. These results were in line with the initial notion that 'to reach the needed independence of U_1 and U_2, one can increase the dispersion in the density of X either by increasing the variance of X or by expanding its range (like in the case of beta distribution)'. When the mesh size, d_i, assumes any fixed value other than 1, quite similar results are obtained from experimenting within the d_i square, $[0, d_i]^2$. In the case of varying d_i, still the same behavior can be observed, except that now, the patterns of (u_1, u_2) must be studied in, at most, $k - 1$ squares, $[0, d_i]^2$, $i = 1, \cdots, k - 1$.

Even though Figure 7 suggests that, for the case of X as a negative exponential distribution, one can treat U_1 and U_2 as independent random variables when the mean of X actually tends to infinity, from a practical point of view, however, one can work with very moderate values of $E(X) = \beta$ (values as low as 8 or 9) and yet expect to generate quite satisfactory results. In the following section, among others, the mechanism of defining values for a_i's (and hence d_i's) are discussed, which can almost 'assure' us of the independence of U_1 and U_2.

SETTING UP THE ALGORITHM

The most notable task in setting up the algorithm deals with defining the set of mesh points $a_1, \cdots, a_k$. The mesh points are obviously defined once an arbitrary partition of the range of X is formed. In general, the partition can be presented as $(-\infty, a_1]$, $[a_1, a_2), \cdots, [a_k, +\infty)$. In case the range of X is bounded, as $a_1 \leq X \leq a_k$, the partition simplifies to $[a_1, a_2), [a_2, a_3), \cdots, [a_{k-1}, a_k]$. From a practical point of view, in order to implement the UFP method, one has to cut off the range of X at one or both ends, if the range is open on one or both sides. In other words, in the case of an open range, one needs to substitute $X \leq a_k$, $X \geq a_1$ or $X \in \Re$ by $a_1 \leq X \leq a_k$. This amounts to discarding a segment in the partition that includes either $(-\infty, a_1]$ or $[a_k, +\infty)$ or both with corresponding area(s) p_0 and/or p_k. Depending on the probability associated with the discarded interval(s), one should expect a serious or mild deficiency in the generation of tail values by the algorithm. It is only logical that a_1 and/or a_k must be chosen in such a way that the risk of such deficiency becomes negligible. By resorting to experimentation, it has been noted that the cut off value(s) a_1 (and a_k) must be selected in such a way that p_0 (and p_k) does (do) not exceed 0.001. This threshold is suggested as a maximum only.

Once end points a_1 and a_k are given or decided upon, points $a_2, \cdots, a_{k-1}$ must also be defined. Among different available ways, only the 'uniform probability partition' scheme is discussed. This is the alternative that makes the areas of all segments equal (i.e., $p_1 = p_2 = \cdots = p_{k-1} = p$, where p is equal to $\frac{1}{k-1}$). Following this rule, if $k = 11$, for instance, the mesh points will be $a_1, a_2, \cdots, a_{11}$ and there will be ten segments each with an associated probability of 0.1, except, possibly, the very first and/or the very last segment(s), depending on the necessity of a cutoff operation on the left and/or the right extreme(s) of the range of X.

By choosing very small values for p, the accuracy of the UFP method will become quite acceptable. In fact, as $p \to 0$, the accuracy of the UFP method approaches the accuracy attainable by the Inverse Transform method when the latter is applicable. By adopting a uniform probability partition of the range of X and in light of Step 1 in Algorithm 2, one just needs to generate a random number, such as U', divide it by the chosen value of p and identify index $i = 1, \cdots, k - 1$ as the integer part of $\frac{U'}{p} + 1$, which, in turn, identifies $Int\,(X) = a_i$. In this way, the first step in Algorithm 2 is executed without any need for a formal search routine.

SIMPLIFYING THE ALGORITHM

The proposed algorithm takes three uniform random numbers to deliver one random variate (one of these random numbers is used for generating a value for $Int\,(X)$). Random number generators are never perfect

and, in fact, produce pseudorandom numbers. Replacing uniform deviates by pseudorandom numbers may induce a substantial error of approximation. According to Fishman [3], this error may increase with the number of pseudorandom numbers required for generating a single deviate. The need of our algorithm for pseudorandom numbers is more than that of other methods, which mostly need less than three on the average (for the inverse transform method this number is just 1). In this section, it is explained how and when this number can be reduced to only 2.

It begins with Theorem 2, with the specific assumption that random variable X is uniformly distributed over the interval $[0, d]$ and the mesh size is equal to d. Given that U_1 and U_2 can be treated as independent uniform $[0, 1]$ random variables, in line with Step 3 in Algorithm 2, one will arrive at:

$$X = \begin{cases} U_2 - U_1, & U_2 \geq U_1 \\ U_2 - U_1 + d, & U_2 < U_1 \end{cases} \tag{23}$$

This is because $Int\ (X)$ is equal to zero. Since it is assumed that $X \sim U[0, d]$, thus, the role of U_1 and U_2 in Algorithm 2 can be played by a single uniformly distributed random variable over interval $[0, d]$. This single random variable is designated by U. Note that, in this case, U_1 and U_2 are supposed to be independent before such a role reversal can take place, so, necessary measures should first be taken to make U_1 and U_2 behave almost independently. In the case of a non-uniform partition of the range of X, where the mesh size, $d_i = a_{i+1} - a_i$, is variable, a similar argument holds and just two random numbers are sufficient to generate a random deviate for X within each of the $k - 1$ intervals $[a_i, a_{i+1})$. Thus, Algorithm 2 can be modified as follows.

Algorithm 3

1. Input p, uniform probability mesh points $a_1, \cdots, a_k$ and $d_i = a_{i+1} - a_i$ for $i = 1, \cdots, k - 1$,
2. Generate a random number, U',
3. Determine the index i, as $\left\lfloor 1 + \frac{U'}{p} \right\rfloor$,
4. Based on the value of i, identify $Int\ (X) = a_i$ and $d_i = a_{i+1} - a_i$,
5. Generate a random number, U_1,
6. Deliver X, as $X = U_1 d_i + a_i$.

Algorithm 3 possesses almost all the advantages of the Inverse Transform method. In other words, UFP makes it possible to generate correlated random variates, as well as values from the densities of order statistics.

Using the antithetic of the random number consumed for generating a value from $Int\ (X)$ in Step 2 of Algorithm 2 or 3, leads to a random variate, which is negatively correlated to the one generated from that random number. This procedure allows one to generate pairs of negatively correlated random variates from any density function. Notice that the Inverse Transform method is not able to generate correlated variates from all distributions.

Order statistics can simply be generated by this algorithm as well, because it can be easily applied to any Beta density function. While the Inverse Transform method can match this property on a very limited basis, the method developed in this work can be applied to any underlying density function and any order statistic.

COMPUTATIONAL RESULTS

While it is obvious that UFP is not capable of working as fast as the inverse transformation method in the special case of $X \sim U[a, b]$, it can be considered as a fast method in many other cases.

In this section, the authors present partial results obtained in generating random deviates from two of the most popular distributions in the Monte Carlo literature, i.e. Gamma (α, β) and Beta (α, β). The performance of UFP is evaluated on the basis of speed and accuracy. The time (in micro seconds) required to generate one random deviate is the yardstick by which the speed is measured. To study the accuracy, the average p-value is adopted in testing the hypothesis, $H_0 : X \sim F_X(x)$. In fact, the Kolmogorov-Smirnov test statistic is used to decide whether samples generated from a distribution function actually demonstrate the needed conformity to that distribution or not. Because of the sensitivity of the results to the seeds and streams of random numbers, the conformity is chosen to be judged based on the average p-value in a series of 100 samples of 1000 random deviates from the intended distribution functions. The 100 seeds have been chosen randomly and the experiments have been run on an AMD Athlon 978 MHz processor using the Borland C^{++} 5.02 compiler under the win32 platform.

Since there are two schemes which can be implemented to make U_1 and U_2 behave (almost) independently, the numerical results for each scheme are presented separately. In one scheme, a more compact mesh is chosen to be defined without changing the dispersion, while, in the other scheme, it is endeavored to increase the dispersion of the intended density function. In order to measure speed, samples of 1000000 deviates were generated.

To take advantage of the capabilities of the C language to make the algorithm work faster, when using the first scheme, partitions of the set of positivity

of the random variable, X, were examined in terms of 2^n segments (n being a positive integer). By doing so, very fast times were recorded. Specifically, UFP performed twice as fast as the Marsaglia approach in generating deviates from Gamma distribution.

Table 1 shows the average p-values and speeds in generating deviates from different Gamma and Beta distributions. In this table, Scheme 1 is adopted and only 16, 32 and 64 segments are considered in the partition of the range of X. As can be seen, UFP performs at a speed which is almost constant for the parameter values shown, as well as the number of segments in the partition.

Tables 2 and 3 display the same performance measures for the second scheme, in which the dispersion of the densities is increased. In these tables, the mesh size is constant ($d = 1$) and positive integers are considered as the mesh points. The results indicate that the first scheme more often leads to better p-values.

In general, it can be said that the UFP algorithm is robust, with respect to speed, against the change of distribution, distribution parameters and also the number of segments in the partition (for each scheme). The notable difference in speed between the two schemes (0.05 vs. 0.16 μs) is due to using the power of 2 (16, 32 and 64) as the number of segments in the uniform probability partition scheme. This idea provides the possibility of working with bit operands in Step 3 of Algorithm 3, which operates very fast in C^{++} language.

Even though these experiments are far from being exhaustive, the comparisons with other popular methods for generating deviates from a number of distributions, such as Gamma, Normal and Beta, show that UFP is the fastest when the number of segments in the partition is chosen as an integer power of 2.

For the sake of computations, a C program was written to generate values from continuous distributions, when the set of positivity of the density consists of $[0, \infty]$. Obviously one can make simple modifications to prepare it for other cases. This code uses the uniform probability partition scheme, as discussed previously, and benefits from the C inline macros, as well as the extremely fast random number generator of Marsaglia and Tsang [9]. The k macro has been used to define the desired number of segments in the partition. One can

Table 1. Average p-values and speed under Scheme 1.

Distribution	**Parameters**	p-**Values**			**Speed**
		$k = 2^4 + 1$	$k = 2^5 + 1$	$k = 2^6 + 1$	(μs)
Gamma	$\alpha = 0.1,\ \beta = 1$	0.326870	0.425883	0.529656	0.06
	$\alpha = 1,\ \beta = 1$	0.336946	0.514091	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

Table 2. Average p-values and speed for Gamma distribution under Scheme 2.

Distribution	**Parameters**	p-**Values**			**Speed**
		$\beta = 10$	$\beta = 100$	$\beta = 1000$	(μs)
Gamma	$\alpha = 0.1,\ \beta = 1$	0	0	0	-
	$\alpha = 1,\ \beta = 1$	0.449565	0.466253	0.469371	0.17
	$\alpha = 5,\ \beta = 1$	0.462224	0.468812	0.468492	0.16

Table 3. Average p-values and speed for Beta distribution under Scheme 2.

Distribution	**Parameters**	p-**Values**			**Speed**
		Range = 10	**Range = 100**	**Range = 1000**	(μs)
Beta	$\alpha = 1.5,\ \beta = 3$	0.439006	0.457462	0.469494	0.16
	$\alpha = 0.8,\ \beta = 2$	0.244152	0.463282	0.470577	0.17
	$\alpha = 0.2,\ \beta = 0.8$	0	0	0	-

easily replace this number by any other value, which should be a suitable power of 2 (for example, 128).

Since this substitution decreases the speed, only powers of 2 are recommended as the number of segments in the partition.

CONCLUSIONS

Based on an all probabilistic reasoning, in this work, a method for generating continuous random deviates is developed. This method takes two random numbers to generate a deviate from any density function. The method in this paper is based on the uniform probability partition of the intended density function and is categorized among the approximate generating methods. While the proposed method enjoys very nice properties, in terms of conformity of the results and speed, it suffers very minor limitations in the case of the rare event simulation. Future research may aim at encouraging this method to behave in an exact manner. Reducing the number of required random numbers from two to one is another area to explore. Finally, one can concentrate on the evaluation of the merits of this method in generating deviates for order statistics and some of the important densities, like normal, gamma and beta, as compared to other methods.

REFERENCES

1. Teichroew, D. "A history of distribution sampling prior to the era of the computer and its relevance to simulation", *American Statistical Association*, pp 27-49 (1965).
2. Pang, W.K., Yang, Z.H., Han, H.S. and Leung, P.K. "Non-uniform random variate generation by the vertical strip method", *European Journal of Operational Research*, **43**, pp 595-604 (2002).
3. Fishman, G.S., *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, New York (1996).
4. Devroye, L., *Non-Uniform Random Variate Generation*, Springer-Verlag, New York (1986).
5. Hormann, W., Leydold, J. and Derflinger, G., *Automatic Non-Uniform Random Variate Generation*, Springer-Verlag, Berlin (2004).
6. Hormann, W. and Leydold, J. "Continuous random variate generation by fast numerical inversion", *ACM Transactions on Modeling and Computer Simulation*, **13**(4), pp 347-362 (2003).
7. Law, A.M. and Kelton, W.D., *Simulation Modeling and Analysis*, 3rd Ed., McGraw-Hill, New York (2000).
8. Morgan, B.J.T., *Elements of Simulation*, Chapman & Hall, London (1984).
9. Marsaglia, G. and Tsang, W.W. "A simple method for generating gamma variables", *ACM Transactions on Mathematical Software*, **26**(3), pp 363-372 (2000).

APPENDIX

Here, the independence of U_1 and U_2 is shown in Equation 1, in which $U_2 = U_1 + X - \lfloor U_1 + X \rfloor$ for special cases where X is uniformly distributed over the interval $[m, n)$ (m and n are integers). One starts with finding the distribution function of $Y = X - \lfloor X \rfloor$. By using the transformation technique, it can be easily shown that:

$$f_Y(y) = \sum_{i=m}^{n-1} f_X(y+i) = \frac{n-m}{n-m} = 1, \qquad \forall y \in [0,1).$$

Now, $X - \lfloor X \rfloor$ is replaced, in Equation 1, by Y to get:

$$U_2 = U_1 + Y - \begin{cases} 0, & Y < 1 - U_1 \\ 1, & Y > 1 - U_1 \end{cases}$$

To find the joint distribution function of U_1 and U_2, the following transformations are performed:

$$u_2 = u_1 + y \Rightarrow y = u_2 - u_1,$$

$$u_1 = u_1 \Rightarrow u_1 = u_1,$$

when $u_1 < u_2$ and:

$$u_2 = u_1 + y - 1 \Rightarrow y = u_2 - u_1 + 1,$$

$$u_1 = u_1 \Rightarrow u_1 = u_1,$$

when $u_1 > u_2$. Then, the joint distribution function would be:

$$f(u_1, u_2) = \begin{cases} f_y(u_2 - u_1) f_{U_1}(u_1) = 1 & u_1 < u_2 \\ f_y(u_2 - u_1 + 1) f_{U_1}(u_1) = 1 & u_1 > u_2 \end{cases}$$

So, $f(u_1, u_2) = 1$ over the square $[0,1]^2$ and that means that U_1 and U_2 are independent.