Journal of Systems Engineering and Electronics, Vol.7, No.1, 1996, pp.35-42

An Approximate Algorithm of Generating Variates with Arbitrary Continuous Statistical Distributions

Zhang Zhengjun[1], *Yang Ziqiang*[2], *Zhang Chunming*[3] & *Feng Yuncheng*[1]

[1]School of Management, Beijing University of Aeronautics & Astronautics, Beijing 100083, China
[2] Institute of Computational Math. & Sci. Eng. Computing, Academia Sinica, Beijing 100080, China
[3] Dept. of Statistics, University of North Carolina, Chapel Hill, U.S.A.

(Received August 7, 1995)

Abstract: This paper discusses an approximate algorithm method which can be used to generate arbitrary non-uniform continuous variates. Percentile calculations of arbitrary continuous distributions are given. In addition, the idea of the algorithm is applied to probability computing.

Keywords: Variate generation, Monotonicity, Curve fitting, Probability computing.

1. INTRODUCTION

The Monte Carlo method, now known as simulation, describes a technique of solving problems through experimentation with random numbers. In many cases, real-world experimentation is difficult to implement due to complexity, expense, and lengthy execution time. Replacing real-world experimentation with computer simulation is a means of solving such problems. However, random numbers play a very important role in computer simulation and real random numbers are hard to come by. For this reason, real random numbers are commonly replaced by pseudo-random numbers which are produced by a numerical algorithm called random number generator.

With this in mind, various stochastic probability models exist in system simulation that require random variate generation during simulation processes. However, a general and exact method to satisfy such a requirement is difficult to find. An approximate method would most effectively remedy this situation.

Suppose, for example, that X is a continuous random variable with arbitrary density $f(x)$, and with a cumulative distribution function $F(x) = \int_{-\infty}^{x} f(t)dt$. The most rudimentary method for variate generation is the inverse distribution function $F^{-1}(x)$, which is also called percentile function. However, finding an explicit representation of the inverse cumulative distribution function (CDF) for most distribution densities is very difficult. When this problem occurs, an approximate method is always an efficient method.

In the 1980s, authors of [2], [8], [1], [5], [10], [12], [4], obtained a series of results. [3], [6], [9], [13], [15] summarized these works. For computer generations of arbitrary continuous distributions, some existing methods are limited only in special cases, for example, monotone decreasing or symmetric unimodal density functions.

[1], for example, proceeded to fit the inverse CDF with an eighth degree Lagrange interpolant over each of the subintervals and developed a fast interval searching table. Because of the oscillatory behavior exhibited by high degree Lagrange interpolant, this method is not

suitable for some distribution densities.

In addition, [12] constructed Runge-Kutta formula as the approximation of the inverse CDF, however, this formula does not preserve monotonicity of monotone data. The inverse CDF functions are monotone increasing functions in nature, so Runge-Kutta formula is not suitable for some distribution densities.

From the following information, a new algorithm can be constructed to effectively generate variates and calculate percentiles with arbitrarily given densities.

⋆ Table developed by Ahrens and Kohrt in [1],

⋆ Idea of solving differential equation used in [12],

⋆ Monotone piecewise curve fitting algorithm developed in [16].

A discussion of approximate computing of probability distribution functions is also included.

2. Monotone Piecewise Curve Fitting Algorithm

In order to preserve the monotonicity of data without modifying the assigned slopes, a piecewise cubic fitting algorithm is proposed. This algorithm has a satisfactory order of convergence. The cubic rational interpolation method is also discussed.

Related to simulation probability model, the assigned slopes are the value of given density functions at some knots.

Let $g(x) \in C^1[a,b]$ be a monotone increasing function, and $\pi : a = x_1 < x_2 < \cdots < x_n = b$ be a partition of the interval $I = [a,b]$.

Suppose that y_i and d_i are the approximate values of $g(x)$ and $g'(x)$ at the partition points x_i respectively.

Let $h_i = x_{i+1} - x_i,\ \Delta y_i = y_{i+1} - y_i,\ \Delta_i = \Delta y_i / h_i, i = 1, 2, \ldots, n.$

In particular, suppose an integer $q > 0$ exists such that

$$y_i = g(x_i) + O(h^q), \quad d_i = g'(x_i) + O(h^q), \ i = 1, 2, \ldots, n \tag{1}$$

where $h = \max\{h_i\}$.

Now, construct a piecewise cubic function $s(x) \in C^1[I]$ such that $s(x_i) = y_i, s'(x_i) = d_i, i = 1, 2, \ldots, n$.

A Hermite cubic interpolation polynomial $s(x)$ in each subinterval $I_i = [x_i, x_{i+1}]$, is defined as:

$$\begin{aligned} s_i(x) \ &= \ \frac{d_i + d_{i+1} - 2\Delta_i}{h_i^2}(x - x_i)^3 \\ &+ \frac{-2d_i - d_{i+1} + 3\Delta_i}{h_i}(x - x_i)^2 + d_i(x - x_i) + y_i \end{aligned} \tag{2}$$

A necessary condition for monotonicity is :

$$\mathrm{sgn}(d_i) = \mathrm{sgn}(d_{i+1}) = \mathrm{sgn}(\Delta_i) \tag{3}$$

Let $\alpha_i = d_i/\Delta_i, \beta_i = d_{i+1}/\Delta_i$, then the following lemmas are obtained as denoted in [7].

Lemma 1. If $\alpha_i + \beta_i - 2 \leq 0$ then $s(x)$ is monotone on I_i if and only if (3) is satisfied.

LEMMA 2. If $\alpha_i + \beta_i - 2 > 0$, and (3) is satisfied, then $s(x)$ is monotone on I_i if and only if one of the following conditions is satisfied.

(i) $2\alpha_i + \beta_i + 3 \leq 0$

(ii) $\alpha_i + 2\beta_i - 3 \leq 0$ or

(iii) $\phi(\alpha_i, \beta_i) \geq 0$

where $\phi(\alpha, \beta) = \alpha - (2\alpha + \beta - 3)^2/3(\alpha + \beta - 2)$.

If $s(x)$ is not monotone on I_i, we denote

$$\tilde{s}_i'(x) = \begin{cases} a_1(x - c_1)^2 + c, & x \in [x_i, c_1] \\ cg_1(x), & x \in [c_1, \bar{x}] \\ cg_2(x), & x \in [\bar{x}, c_2] \\ a_2(x - c_2)^2 + c, & x \in [c_2, x_{i+1}] \end{cases} \tag{4}$$

where $\bar{x}$ is the extreme point of $s'(x)$ on I_i, $g_1(x) = (\bar{x} - x)/(\bar{x} - c_1), g_2(x) = (\bar{x} - x)/(\bar{x} - c_2)$. The constants a_1, a_2 and c and the additional points c_1 and c_2 must be determined in such a way that $\tilde{s}'(x)$ satisfies all interpolation and monotonicity requirements. By integrating formula (4) on related interval, $s(x)$ is obtained with the following form:

$$\tilde{s}_i(x) = \begin{cases} \frac{1}{3}a_1(x - c_1)^3 + c(x - c_1), & x \in [x_i, c_1] \\ -\dfrac{(x - \bar{x})^2}{2(\bar{x} - c_1)} + \dfrac{\bar{x} - c_1}{2}, & x \in [c_1, \bar{x}] \\ \dfrac{(x - \bar{x})^2}{2(c_2 - \bar{x})} - \dfrac{c_2 - \bar{x}}{2}, & x \in [\bar{x}, c_2] \\ \frac{1}{3}a_2(x - c_2)^3 + c(x - c_2), & x \in [c_2, x_{i+1}] \end{cases}$$

For calculation of a_1, a_2, c_1, c_2, c, refer to [16], the results are given here. We denote μ, η, ω, as $\mu = \bar{x} - x_i$, $\eta = x_{i+1} - \bar{x}$, $\omega = s_i'(\bar{x})$. Let c be a free variable, $\theta = (d_i\mu + d_{i+1}\eta)/h_i, \rho(c) = 3(\Delta_i - \frac{c}{2})/(\theta + \frac{c}{2})$, then

$$c_1 = x_i + \rho(c)(\bar{x} - x_i), \quad c_2 = x_{i+1} - \rho(c)(x_{i+1} - \bar{x})$$

$$a_1 = \frac{d_i - c}{(c_1 - x_i)^2}, \quad a_2 = \frac{d_{i+1} - c}{(x_{i+1} - c_2)^2}$$

Let

$$m(x) = \begin{cases} s_i(x), & s_i(x) \text{ is monotone on } I_i \\ \tilde{s}_i(x), & \text{otherwise} \end{cases} \tag{5}$$

then the following convergence theorem is obtained.

THEOREM 2.1. Let $g(x) \in C^4[a, b]$ and $p = \min(4, q)$, if (1) holds and the variable c in $\tilde{x}_i(x)$ satisfies

$$0 \leq c < min(|\omega|, 2\Delta_i) \tag{6}$$

then for each interval I_i, we have $\| g(x) - m(x) \| = O(h^p)$.

This theorem shows that when $g(x)$ has sufficiently smooth properties $m(x)$ can sufficiently approximate $g(x)$. When c is determined, $m(x)$ is a monotone cubic function on all subintervals I_i, and can be used to approximate any other functions.

Another method, the cubic rational interpolation will be introduced using the above notations and the assumptions of (1) and (3). Furthermore, let parameter $r_i > -1$. Piecewise cubic rational interpolation $z(x)$ is then defined by

$$z(x) = \frac{p(\theta)}{q(\theta)} \quad x \in [x_i, x_{i+1}] \tag{7}$$

where

$$\theta = (x - x_i)/h_i$$

$$p(\theta) = y_{i+1}\theta^3 + (r_i y_{i+1} - h_i d_{i+1})\theta^2(1-\theta) + (r_i y_i + h_i d_i)\theta(1-\theta)^2 + y_i(1-\theta)^3$$

$$q(\theta) = 1 + (r_i - 3)\theta(1-\theta)$$

Interpolation conditions of the function and derivative values of given data are satisfied by $z(x)$. When $r_i \geq (d_i + d_{i+1})/\Delta_i$, $z(x)$ is monotone increasing. The following theorem is obtained.

THEOREM 2.2. Let $r_i - 3 = O(h_i^l)$. If (1) holds, then $\| g(x) - z(x) \| = O(h^p)$ where $p = \min(4, q, l+2)$.

This theorem shows that when $g(x)$ has sufficiently smooth properties $z(x)$ can sufficiently approximate $g(x)$. The proof of the theorem 2.1 and theorem 2.2 and the details about monotone cubic interpolation can be found in [16].

3. NONUNIFORM VARIATE GENERATION

Several reasons for using non-uniform variate generation exist. The main reason, however, is the requirement of random variables in order to run any simulation model. Some users need random variables with unusual densities for other purposes. For instance, some scientists need to solve a simulation problem with a very complex distribution density where variate generation is difficult by ordinary generators. The best process for generating these variates is to use an approximate algorithm.

Suppose that a distribution function $F_X(\cdot)$ for the random variable of interest is selected. It is desired to generate random variables $X_1, X_2, \cdots$ each of which follows this distribution function. If U is a uniform variable on (0,1), then the variable defined by $X = F^{-1}(U)$ has density $f(x)$. When $F(x_p) = p$ is satisfied for an arbitrary $p(0 < p < 1)$, x_p is called p percentile. In the case of variate generation, two equations are obtained:

$$X = F^{-1}(U) \text{ and } \frac{dX}{dU} = \frac{1}{f(x)}$$

From these two equations, interval [0,1] can be divided into

$$u_0 = 0 < u_1 < u_2 < \cdots < u_n = 1$$

and X_i calculated at each knot by using Runge-Kutta method. Let

$$y_i = X_i \text{ and } d_i = \frac{1}{f(X_i)}, \; i = 1, \cdots, n$$

Then equation (5) or (7) can be used to calculate percentiles with random number U. In other words,

$$X \approx m(U)$$

The same process for variate generation is used for percentile computing. However, the following algorithm is suggested.

Algorithm steps include:

STEP 1. *Initialization.*

a) Divide the interval [0,1] into n subintervals $(u_i, u_{i+1}), i = 1, 2, \ldots, n$.

b) Calculate x_i and x'_i at each knot $u_i, i = 1, 2, \ldots, n$.

c) Calculate parameters of interpolation formulas in each subinterval and the given monotone interpolation from section 2.

STEP 2. Generate $U \sim U(0, 1)$ or give a probability value p.

STEP 3. Use the formula on (u_i, u_{i+1}), when U or p is in the subinterval (u_i, u_{i+1}). Generate $X \sim F(x)$ or calculate $x_p = F^{-1}(p)$ through $m(x)$.

STEP 4. Repeat Step 2 to 3 until the number of requirement is achieved.

The initialization process can be done and stored before **step 2 through 4** is loaded. The partition table of interval [0,1] given in [1] is recommended.

4. Approximate Probability Computing

When a statistical analysis is made with stochastic data, the distribution function $F(x)$ does need to be calculated. There are several methods of calculating $F(x)$. One method is to create and save a table on computer diskettes. However, a large amount of data will need to be saved and may occupy a tremendous amount of computer memory space. A second method, the integrated method, is difficult and expensive. Programming is in computer code and computation time is costly. A third method, the approximate formula, is therefore the suggested alternative. The approximate formula uses a special distribution function constructed and stored on diskettes before probability computing begins. In this section, approximate probability computing is discussed and a general method for arbitrary continuous distribution functions is proposed.

Suppose that X is a continuous random variable with arbitrary density $f(x)$, and cumulative distribution function (CDF) $F(x) = \int_{-\infty}^{x} f(t)dt$. Then

$$F'(x) = f(x)$$

Let

$$\pi : -\infty < x_1 < x_2 < \cdots < x_n < \infty$$

be a partition of interval $(-\infty, \infty)$. Let

$$y_i = F(x_i) \text{ and } d_i = f(x_i) \; i = 1, 2, \cdots, n$$

Then Equation (5) or (7) can be used to calculate the probability value with distribution $F(x)$. i.e. $F(x) \approx m(x)$.

The approximate probability computing algorithm is

STEP 1. *Initialization.*

a) Let $\pi : -\infty < x_1 < x_2 < \cdots < x_n < \infty$ be a partition of interval $(-\infty, \infty)$.

b) Calculate $F(x_i)$ and $f(x_i)$ at each knot $x_i, i = 1, 2, \ldots, n$.

c) Calculate parameters of interpolation formulas in each subinterval using the monotone interpolation given in section 2.

STEP 2. Give an X, search the subinterval I_i to which x belongs.

STEP 3. Calculate $F(x) \approx m(x)$ or $F(x) \approx z(x)$

Table 1 gives a comparison of error of four distinguished distribution functions. From the table, the approximate algorithm can give a satisfactory precision for probability computing. ABSERR denotes absolute error, RELERR denotes relative error, MPCI denotes monotone piecewise cubic interpolant, MPCRI denotes monotone piecewise cubic rational interpolation in the table.

Table 1. Comparison of error for computing probability values

Distribution	MPCI		MPCRI	
	ABSERR	RELERR	ABSERR	RELERR
$N(0,1)$	2.0×10^{-6}	3.0×10^{-4}	9.8×10^{-6}	4.9×10^{-4}
$\chi^2(10)$	7.3×10^{-6}	7.8×10^{-5}	3.7×10^{-5}	3.6×10^{-4}
$F(10,10)$	3.2×10^{-5}	9.2×10^{-5}	3.7×10^{-5}	2.3×10^{-4}
$t(10)$	8.2×10^{-6}	7.4×10^{-4}	1.3×10^{-5}	6.5×10^{-4}

5. Numerical Results and Conclusions

In this section three tables show the efficiency of the approximation method. These tables are obtained by simulation on an IBM compatible computer 50 MHz 80486DX2, with FORTRAN code. A multiplicative congruential method with parameters (742938285, 2^{31}-1) is used to generate pseudorandom numbers.

Table 2 shows that the approximate algorithm can preserve monotonicity of Gamma (0.1) distribution functions, under requested accuracy of 1×10^{-8}, but Hermite cubic interpolant may not.

As shown in Table 3 that the algorithm has a quick initialization time. Time requirements can therefore be ignored because initialization process can be completed before the algorithm is applied to some practical applications. In addition, computing time of the algorithm is faster in comparison to other algorithms, as shown in Table 4. Other methods and programs are in [3].

Generally, monotone piecewise cubic interpolant is superior to other methods. The algorithm adapts to arbitrary continuous distributions. Tables 2 through 4 show, in turn, that the algorithm is an efficient approximation method.

Acknowledgment. The authors pay appreciation to Prof. Bernard J. Schroer for his helpful comments.

Table 2. Comparison of error for variate generation

Distribution	MPCI		MPCRI	
	ABSERR	RELERR	ABSERR	RELERR
Cauchy(0.5)	7.1×10^{1}	1.5×10^{-2}	2.5×10^{1}	1.1×10^{-3}
exponential $\lambda = 1$	3.6×10^{-3}	9.6×10^{-4}	1.1×10^{-3}	3.0×10^{-4}
$N(0,1)$	1.1×10^{-3}	5.9×10^{-4}	4.7×10^{-4}	2.3×10^{-4}
gamma(0.1)	3.1×10^{-3}	8.8×10^{-1}	9.1×10^{-4}	9.9×10^{-1}
gamma(0.1)*	3.1×10^{-3}	9.8×10^{0}		
gamma(2)	4.0×10^{-3}	7.2×10^{-4}	1.3×10^{-3}	3.3×10^{-4}
beta(5,5)	1.0×10^{-4}	4.9×10^{-4}	4.8×10^{-5}	2.2×10^{-4}
$\chi^2(10)$	9.9×10^{-3}	4.9×10^{-4}	3.5×10^{-3}	2.4×10^{-4}
$F(10,10)$	1.6×10^{-2}	1.5×10^{-3}	4.0×10^{-3}	3.7×10^{-4}
$t(10)$	2.8×10^{-3}	9.2×10^{-4}	8.5×10^{-4}	3.2×10^{-4}

gamma(0.1)* belongs to RK4 (Hermite cubic (2)) method of Ulrich and Watson.

Table 3. The initialization time of the algorithm (second)

Distribution	time
Cauchy(0.5)	0.44
exponential λ=1	0.39
N(0,1)	0.39
gamma(0.1)	0.88
gamma(2)	0.65
beta(5,5)	0.39
F(10,10)	0.99
T(10)	0.55
$\chi^2(10)$	1.10

Table 4. The average time per variate generation of our algorithm comparing with other methods (μsec)

Distribution	MPCI	MPCRI	other	methods
Cauchy(0.5)	78	90.2	80.2	Inverse CDF
exponential λ=1	82.4	90	61.4	Inverse CDF
$N(0,1)$	78	93.4	74.6	Box-Muller
gamma(0.1)	79	83.4	152.6	RGS
gamma(2)	82.4	91.2	156	RGKM3
beta(5,5)	77	91.2	173.6	*
$F(10,10)$	78	90.2	176.8	**
$T(10)$	78	90	286.8	***
$\chi^2(10)$	76.8	92.2	218.6	****

* Schmeiser and Babu's method (1980) (Source: [6]).

** Transformation of beta variate.

*** Best's method based on t3 generator (1978) (Source:[6]).

**** Transformation of five independent uniform variates.

REFERENCES

[1] Ahrens, J.H. and Kohrt K.D., "Computer methods for efficient sampling from largely arbitrary statistical distribution", Computing 26, 1981, pp.19–31.

[2] Archer, N.P., "The generation of piecewise linear approximations of probability distribution functions", J. Statist. Comput. Simul 11, 1980, pp.21–40.

[3] Bratley, P., Fox, B.L. and Schrage, L. E. A, Guide to Simulation, Springer-Verlag, Berlin, 1987.

[4] Chen, Q.Z., A new method of generating random numbers with unimodal densities, The Fourth Chinese-Japanese Statistics Symposium, Kunming, 1991, pp.33–34.

[5] Devroye L., "Random variate generation for unimodal and monotone densities", Computing, 32, 1984, pp.43–68.

[6] Devroye L., Non-uniform random variates generation, Springer-Verlag, Berlin, 1986.

[7] Fritsch, F. N. and Carlson, R. E., Monotone piecewise cubic interpolation, Siam. J. Numer. Anal., 17, 1980, pp.238–246.

[8] Hora S. C., "Estimation of the inverse function for random variate generation", Comm. Acm, 26, 1983, pp.590–594.

[9] Kennedy W. J. and Gentle, J. E., Statistical Computing, Marcel Dekker, New York, 1980.

[10] Marsaglia G. and Tsang W. W., "A fast, easily implemented method for sampling from decreasing or symmetric unimodal density functions", Siam J. Sci. Stat. Comput, 5, 1984, pp.349–359.

[11] Shampine, L. F. and Gordon, M. K., Computer solution of ordinary equation, the initial value problem, W. H. Freeman, San Francisco, 1975.

[12] Ulrich G. and Watson L. T., "A method for computer generation of variates from arbitrary continuous distributions", Siam J. Sci. Stat. Comput.8, 1987, pp.185–197.

[13] Xu Zhongji, Monte Carlo Method, Shanghai Science Technology Press, 1980. (in Chinese)

[14] Yang Ziqiang, Zhang Zhengjun and Zhang Chunming. "Statistical Computing and Simulation based on Piecewise Monotone Curve Fitting Algorithms", Proceedings of Asian Conference on Statistical Computing, Beijing, China, 1993, pp.296–299.

[15] Zhang Zhengjun, Studies of continuous random variate generation and preserved monotone interpolation methods. MA. Thesis, Computing Center, Academia Sinica, Beijing, 1993. (in Chinese)

[16] Zhang Zhengjun, Yang Ziqiang and Zhang Chunming, "Monotone Piecewise Cubic Curve Fitting Algorithms", Jour. of Comput. Math., Vol.12, No.2, 1994, pp.163–172.

Zhang Zhengjun, born in Sept. 1965, received the BS in Computational Mathematics and Applied Software from Yunnan University in 1986, MS in Computational Mathematics from the Computing Center, Academia Sinica in 1993, Ph.D in Managerial Engineering from Beijing University of Aeronautics and Astronautics in 1996. He also has four years research experience in system engineering from working at the Research Center of Systems Engineering at Yunnan University. His research interests are in random number generations, sampling techniques, variance reduction techniques, advanced simulation designs on discrete event systems and manufacturing systems.

YANG Ziqiang, born in 1937, is a research professor in the Institute of Computational Math. & Sci. Eng. Computing, Academia Sinica. His main interests are in probability and statistical computing, parallel computing, multivariate analysis, statistics software.

ZHANG Chunming, is a Ph.D student in Department of Statistics at the University of North Carolina, USA. She has the BS and MS both in Mathematical statistics from Nankai University and Computing Center, Academia Sinica, respectively. Her main interests are in probability and statistical computing, parallel computing, Monte Carlo method, quality control, multivariate analysis.

FENG Yuncheng, born in 1933, is a professor in School of Management at Beijing University of Aeronautics & Astronautics. He is chairman of the Chinese Society for Computer Simulation. His main interests are in discrete event system simulation, industrial engineering.