Computer Generation of Random Variables Using the Ratio of Uniform Deviates

A. J. KINDERMAN
California State University at Northridge
and
J. F. MONAHAN
Brookhaven National Laboratory

The ratio-of-uniforms method for generating random variables having continuous nonuniform distributions is presented. In this method a point is generated uniformly over a particular region of the plane. The ratio of the coordinate values of this point yields a deviate with the desired distribution. Algorithms which utilize this technique are generally short and often as fast as longer algorithms.

Key Words and Phrases: random number generation, uniform distribution
CR Categories: 5.5, 5 25

1. INTRODUCTION

In writing an algorithm for generating random deviates from a specific (nonuniform) distribution or family of distributions, a researcher has many tools available to help construct an efficient algorithm. Direct and inverse-distribution methods are well known but often have limited applicability. Acceptance-rejection methods are often powerful, especially when close and easy-to-compute inner and outer bounds on the density are known. Probability mixing allows the researcher to piece together efficient algorithms whose composite output has the desired distribution. The method presented here uses the ratio of two uniform deviates to obtain the desired distribution. Used alone or combined with other familiar tools, this method often leads to algorithms superior to those in present use.

The basic ratio-of-uniforms method and simple improvements are discussed in Section 2. Section 3 compares algorithms evolving from the basic method with the best available algorithms with comparable storage requirements.

2. METHOD AND DISCUSSION

In the ratio-of-uniforms method a point is generated uniformly over a certain region in the plane. The deviate with the desired distribution is then obtained by

This research was supported in part by the National Science Foundation under Grants SOC73-09243 and GS38609.

Authors' addresses: A.J. Kinderman, Department of Management Science, California State University at Northridge, Northridge, CA 91330; J.F. Monahan, Applied Mathematics, Brookhaven National Laboratory, Upton, NY 11973.

taking the ratio of the coordinate values of the point. To generate a random variable X with density $f(x)$, let $C_f = \{(u, v): 0 \le u \le f^{1/2}(v/u)\}$. Then if (U, V) is a pair of random variables distributed uniformly over C_f, $X = V/U$ has the desired density f. To see this, consider the transformation T: $(u, v) \to (v/u, u)$ whose inverse has Jacobian $J(u, v/x, y) = -y$. The joint density of $(X, Y) = T(U, V)$ is

$$g(x, y) = \begin{cases} 2y, & 0 \le y \le f^{1/2}(x)\,; \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

and hence the marginal density of X is

$$\int_0^{f^{1/2}} 2y\, dy = f(x). \tag{2.2}$$

Note that the constant 2 must appear in eq. (2.1) so that the marginal density of eq. (2.2) will integrate to 1. This implies that the area of C_f is always $\frac{1}{2}$.

Remark 1. This process can be reversed to generate (U, V) uniformly over C_f. Let X with density $f(x)$ be generated by some other means. Conditional on $X = x$, let Y equal $f^{1/2}(x)$ times the maximum of two independent uniform $(0, 1)$ deviates to get the joint density of (x, y) as in eq. (2.1). By applying the transformation $T^{-1}(x, y) = (y, xy)$, we get $(U, V) = T^{-1}(X, Y)$ distributed uniformly over C_f.

Remark 2. For any non-negative function $h(x)$ with finite integral K, let $C_h = \{(u, v): 0 \le u \le h^{1/2}(v/u)\}$. By an argument parallel to the one above, if (U, V) is distributed uniformly on C_h, then $X = V/U$ has the density $f(x) = K^{-1}h(x)$ and the area of C_h is $K/2$.

In many cases the equation $u = h^{1/2}(v/u)$ can be solved (typically for v in terms of u) and the boundary of C_h can be found explicitly. The following examples illustrate this point, as well as the basic method.

Example 1. (*Uniform*). Let $f(x) = (b-a)^{-1}$, if $x \in (a, b)$, and zero elsewhere. Then

$$C_f = \{0 \le u \le (b-a)^{-1/2}, a \le v/u \le b\},$$

which is a triangle with vertices $(0, 0)$, $(\lambda, \lambda a)$, $(\lambda, \lambda b)$ for $\lambda = (b-a)^{-1/2}$.

Example 2. (*Reciprocal Uniform*). Let $f(x) = x^{-2}(b-a)^{-1}$, if $x \in (b^{-1}, a^{-1})$, $0 \le a < b$, and zero elsewhere. Then

$$C_f = \{0 \le u \le v^{-1}u(b-a)^{-1/2}, b^{-1} \le v/u \le a^{-1}\},$$

which is a triangle with vertices $(0, 0)$, $(\lambda a, \lambda)$, $(\lambda b, \lambda)$ for $\lambda = (b-a)^{-1/2}$.

Example 3 (*Normal*). Let $h(x) = \exp(-x^2/2)$. Then

$$C_h = \{0 \le u \le \exp(-v^2/(4u^2))\} = \{v^2 \le -4u^2 \ln u\}$$

whose boundary is the curve $v = \pm 2u(-\ln u)^{1/2}, 0 \le u \le 1$.

Example 4. (*Cauchy*). Let $h(x) = (1 + x^2)^{-1}$. Then

$$C_h = \{0 \le u \le (1 + (v/u)^2)^{-1/2}\} = \{u^2 + v^2 \le 1, u > 0\},$$

or a half circle centered at the origin with radius 1.

Example 5 (*Exponential*). Let $f(x) = \exp(-x)$ for $x \ge 0$. Then C_f is bounded below by the u axis and above by $u=\exp(-v/2u)$ or $v = -2u \ln u$, $0 \le u \le 1$.

In most cases one must use acceptance-rejection techniques to obtain (U, V) uniform over C_h. Note that C_h is bounded in the u direction if $h(x)$ is bounded, and C_h is bounded in the v direction if $x^2h(x)$ is bounded. A pair (U, V) can be generated uniformly over a bounded region C_h by generating a point uniformly on the rectangle enclosing C_h and rejecting this point if it is not inside C_h. For example, if C_h is scaled to fit within the rectangle with vertices $(0, \pm b)$ and $(1, \pm b)$, the algorithm consists of two steps:

1. Generate u and v, independent uniform (0, 1) deviates, and let $x = 2b(v-0.5)/u$.
2. If $u^2 \le h(x)$, x is the desired deviate; otherwise, go to 1.

Sometimes the inequality to be checked in step 2 can be rewritten so that easily computed bounds can be used to replace more complicated computations. For the normal distribution (Example 3), the inequality can be written as $x^2 \le -4 \ln u$, and one of the family of bounds,

$$4(1 + \ln c) - 4cu \le -4 \ln u \le 4/(cu) - 4(1 - \ln c), \quad c > 0,$$

can be used to avoid the computation of the logarithm in most of the comparisons.

If the rectangle covering a bounded C_h is loosely fitting, indicating that many points will be rejected, we can improve the algorithm by partitioning C_h into disjoint subsets $C^1, \ldots, C^k$. Fewer points will be rejected if each subset C^i can be tightly enclosed in its own rectangle. By using a partition, we have represented X as a mixture of random variables formed by taking the ratio V/U separately on each subset C^i. The probability that $X = V/U$ is taken from C^i must be equal to the ratio of the area of C^i to the area of C_h. Another advantage of partitioning C_h is that often we can replace the ratio-of-uniforms method by more efficient techniques on some of the subsets. In particular, a single uniform deviate can be used when the subset is a triangle corresponding to either a uniform or reciprocal uniform distribution (Examples 1 and 2).

3. COMPARISON OF ALGORITHMS

To illustrate the potential of the ratio-of-uniforms method, we have implemented algorithms for the Cauchy, chi-square (3 degrees of freedom), Student's t (3 degrees of freedom), normal, and normal tail distributions. Function subprograms were written in Fortran and timed on the IBM 360/67 using the multiplicative congruential uniform generator of Lewis et al. [6]. The results are presented in Table I.

No algorithm in the table is both shorter and faster than its ratio-of-uniforms competitor. The basic ratio-of-uniforms algorithms are almost always short, owing to their simplicity. In the normal and normal tail cases, the use of bounds as discussed in Section 2 leads to even faster algorithms that still require very little storage. The Cauchy example of a composite algorithm that uses a partition of C_h and mixes the uniform and reciprocal uniform distributions with the ratio-of uniforms method was 20 per cent faster than the basic ratio-of-uniforms method at a cost of more than twice the storage. However, a fast composite algorithm fcr the normal is still smaller than its competitor.

The ratio-of-uniforms method produces algorithms that are short and often as

Table I

Distribution	Algorithm	Time*	Space†	Reference
Cauchy	$X = \tan(\pi(u - 0.5))$	424	49	
	R of U	320	63	
	composite R of U	249	149	[4]
Student's t_3	TMXS	670	374	[5]
	R of U	310	56	
Chi-square 3	GT	827	141	[2]
	R of U	747	68	
	$(\text{normal})^2$ + exponential	598	93	
Normal	PO	388	93	[1, 8]
	R of U (bounds)	363	83	
	GRAND	335	177	[3]
	composite R of U	330	168	
Normal tail ($x > 2.0$)	TL	654	70‡, 91§	[1, 7]
	R of U	584	66‡, 121§	[4]
	R of U (bound)	463	78‡, 137§	[4]

* Time is measured in microseconds per deviate.

† Space is measured in words for a function subprogram in Fortran and does not include 28 words for conversion constants and register-save area.

‡ Specific algorithm for $x > 2.0$.

§ General algorithm for $x > a$.

fast as longer algorithms. Of course the use of other machines, other languages, and other uniform generators would lead to different results from those given in Table I, but the ratio-of-uniforms method should still be useful to researchers wishing to construct short algorithms for other distributions.

REFERENCES

1. AHRENS, J.H., AND DIETER, U Computer methods for sampling from the exponential and normal distributions *Comm. ACM 15*, 10 (Oct 1972), 873–882.
2. AHRENS, J.H., AND DIETER, U. Computer methods for sampling from gamma, beta, Poisson and binomial distributions. *Computing 12* (1974), 223–246.
3. BRENT, R.P. Algorithm 488: A Gaussian pseudo-random number generator *Comm. ACM 17*, 12 (Dec. 1974), 704–706.
4. KINDERMAN, A J , AND MONAHAN, J.F. Generating random variables from the ratio of two uniform deviates. Computer Science and Statistics: Proc. Ninth Annual Symp. on the Interface, 1976.
5. KINDERMAN, A.J , MONAHAN, J.F., AND RAMAGE, J.G. Computer generation of random variables with normal and student's t distribution. Proc. Statist. Comptng. Sect., Amer. Statist. Assoc., Washington, D.C., 1975, pp. 128–131.
6. LEWIS, P.A W., GOODMAN, A.S., AND MILLER, J M. A pseudo-random number generator for the System/360 *IBM Syst. J 8* (1969), 136–146.
7. MARSAGLIA, G Generating a variable from the tail of the normal distribution. *Technometrics 6* (1964), 101–102.
8. MARSAGLIA, G., AND BRAY, T.A. A convenient method for generating normal variables. *SIAM Rev. 6* (1964), 260–264.

Received September 1976; revised December 1976