Proceedings of the 1980 Winter Simulation Conference
T.I. Ören, C.M. Shub, P.F. Roth (eds.)

CHAPTER G OF THE IMSL LIBRARY - GENERATION AND TESTING OF RANDOM DEVIATES: SIMULATION

Peter A. W. Lewis*
Department of Operations Research
Naval Postgraduate School
Monterey, California
93940

*Member of the Advisory Board of International Mathematics and Statistics Library (IMSL), Inc., Houston, Texas.

ABSTRACT: This tutorial paper describes the capabilities and design philosophies of Chapter G of the IMSL (International Mathematics and Statistics Libraries) subroutine package. Several specific subroutines are described and further enhancements for later packages are alluded to.

In this tutorial we describe the capabilities and design philosophies of Chapter G of the IMSL (International Mathematics and Statistics Libraries) subroutine package. Chapter G is entitled "Generation and Testing of Random Numbers: Simulation" and contains subroutines which aid in testing pseudo-random deviates for randomness and which produce random deviates and stochastic processes. We believe that it is the most comprehensive and up-to-date collection of subroutines of this nature and should be of wide use to mathematicians, statisticians and operations analysts who require reliable and efficient building blocks for their simulations. Most of the comments in this exposition refer to Edition 8 of the IMSL Subroutine Library in which Chapter G has been considerably improved and extended from previous Editions. However, several other possibilities for later versions will be mentioned.

In addition to the testing routines, which will not be discussed here, the main subdivisions of Chapter G are:

(1) univariate deviates from the uniform distribution--that is, pseudo-randum numbers;

(2) discrete-valued deviates from non-uniform distributions;

(3) continuous valued deviates from non-uniform distributions;

(4) multivariate random variables;

(5) for simulations, various kinds of stochastic processes and point processes (series of events).

1. UNIVARIATE DEVIATES FROM THE UNIFORM DISTRIBUTION-PSEUDO-RANDOM NUMBERS

The design philosophy for the cornerstone of the Chapter--the psuedo-random numbers--has been that the algorithms should be simple enough to allow for portable implementations and should produce sequences with well tested and documented statistical properties, i.e. which will simulate sequences of independent random variables with a uniform (0,1) distribution. To this end, the prime modulus congruential generator developed by Lewis in 1969 and tested in Lewis and Learmonth (1973) has been used in subroutine GGUBS, with an option for self-shuffling given in subroutine GGUW. The shuffled version has considerably better statistical properties than the basic generator and should be used for extensive work. The shuffling in GGUBS is accomplished by using the low-order bits of the currently generated word to

80CH1617-0/80/0000-0357$00.75

exchange that word with one of 128 previously generated words which have been stored in a table. This obviates the need for another random number generator to do the shuffling and is thus very fast.

Because of recent test results (Learmonth, 1978, Fishman and Moore, 1978) another prime modulus congruential generator with a different multiplier is supplied in Edition 8 in subroutine GGUBT. The multiplier here is 397201094 while that used in GGUBS is 16807. An additional impetus to supplying both GGUBS and GGUBT is that in many cases simulators need two independent pseudo-random number generators. There is also a FUNCTION form of GGUBS called GGUBFS. We note, too, that both GGUBT and GGUW produce sequences of pseudo-random numbers which will be identical in the first 23 bits of the mantissa in all computer/compiler environments on which the IMSL library is supported.

This part of the chapter also supplies routines for generating pseudo-randum discrete deviates on (1,K) in subroutine GGUD. In addition order statistics from a continuous uniform distribution are produced in GGUO. This does not have to be all of the order statistics; it is possible, for instance, to produce only the first k of the order statistics from a sample of size n.

2. DISCRETE-VALUED DEVIATES FROM NON-UNIFORM DISTRIBUTIONS

For non-uniformly distributed univariate deviates the choice of algorithms, particular or general, is quite wide, and it is necessary to take into consideration

a. situations in which one random deviate is required at a time,

b. one number is required at a time but is repeatedly called for with the same parameters, and

c. situations in which large arrays of deviates will be called for at a time.

No algorithm for a particular type of random deviate will give the greatest possible speed, simplicity of programming and greatest precision for all of these cases and a judicious choice must be made. For several commonly used distributions, e.g. Poisson, it has been found necessary to give two algorithms to cover adequately all of the above situations.

For discrete-valued deviates from non-uniform distributions the inverse cumulative probability method gives the best results in most cases and a subroutine (GGDT) is provided to do this in Chapter G. It requires as input only a subroutine to compute the probabilities associated with the random deviate which is to be simulated. The discrete deviates are then obtained by generating uniform deviates and then searching a table of the cumulative probability function. The search employs bisection in the earlier stages, and then reverts to a linear search.

The only new general method to appear in the literature for generating discrete-valued deviates is the alias method due to Walker. This algorithm has been described recently by Kronmal and Peterson (1979). Essentially, the method (GGDA) represents the desired discrete N-valued deviate as an equi-probable mixture of N deviates which take on only two values. Each of the N two-point distributions, say f_i, takes on values i and L_i, which is called the alias and may be any value from 1 to N except i. Thus it is only necessary to store the set of N mixing probabilities π_i and the set of N aliases L_i. Once the initial computations have been done to find the appropriate mixture and the aliases, the method is extremely efficient.

For several commonly used random variables, e.g. Poisson, geometric and negative binomial, special algorithms have been programmed up as Chapter G subroutines. The methods used here are the best available which are consistent with the Chapter G design philosophy. If many replications are required from a specific case of these variables, greater efficiency will be obtained by using the inverse technique (GGDT) or the alias method (GGDA). An example is the Poisson case. Since it is so frequently used, GGPON is an implementation of a simple method using multiplication of pseudo-random deviates which requires little set-up.

Thus let λ be the Poisson parameter and let N be the number such that when i.i.d. uniform deviates $U_1, U_2, \ldots$ are successively multiplied together the product $U_1 \times \ldots \times U_N > e^{-\lambda}$ but $U_1 \times \ldots \times U_{N+1} \leq e^{-\lambda}$. Then N, which takes on values $0,1,2,\ldots$, is a Poisson deviate with parameter λ. The problems with this method are firstly that the expected time to generate a Poisson deviate N increases proportionally to λ, and secondly that for λ greater than, say, 20 numerical accuracy becomes a problem.

Despite the simplicity of this method the inverse cumulative probability method is competitive even for λ as small as 2. Thus for efficiency the Poisson generator supplied in GGPOS is the inverse method with the computations of the probabilities built into the subroutine. It is the preferred routine for many replications and when λ is, say, greater than 5.

3. CONTINUOUS VALUED DEVIATES FROM NON-UNIFORM DISTRIBUTIONS

For continuous-valued deviates from non-uniform distributions the situation is very different from the

discrete-valued case. A subroutine to implement the inverse probability integral transform is provided but requires a subroutine to compute the inverse probability function. This computation is generally difficult and consequently a considerable number of special algorithms for commonly used distributions have been implemented as subroutines. Even in cases when the inverse cumulative probability distribution is simple to compute, e.g. for the exponential where the inverse cumulative probability distribution is a simple logarithm, more elaborate methods will sometimes give faster algorithms. The price, of course, is memory and programming complexity.

An interesting case is that of the Gamma distribution (subroutine GGAMR). A considerable amount of work and ingenuity has gone into this case over the last ten years. It is only recently that simple, efficient methods have been derived to cover the whole range of values of the Gamma shape parameter, denoted by A.

For values of the shape parameter $A < 1$ the algorithm due to Ahrens used in Edition 7 is retained. For $A > 1$ the complicated three-part algorithm of Lewis and Robinson has been replaced by an extremely efficient algorithm due to Schmeiser and Lal (1979). This algorithm uses a combination of decomposition of the density into various parts and, within these parts, rejection-acceptance techniques enhanced by squeeze methods.

Several competing algorithms have appeared recently but none is markedly superior in all environments to the Schmeiser-Lal algorithm and this was consequently retained for Edition 8 because the implementation of the algorithm has been extensively tested. Reliability of the algorithm and the coding is always an important consideration, more important, say, than a 5% increase in speed.

Routines for continuous random deviates which are new to Edition 8 are those for stable random deviates and for mixtures of two exponential variates. The method for generating stable random deviates was published recently by Chambers, Mallows and Stuck (1976). This is used in subroutine GGSTA. Convex and non-convex mixtures of exponentials can be generated in GGEXT. The probability density function is

$$f(x) = \frac{p}{m_1} e^{-x/m_1} + \frac{(1-p)}{m_2} e^{-x/m_2} \qquad x \geq 0, \quad m_1 > m_2 \ .$$

When $0 < p < 1$ this is a convex mixture of two exponentials and, therefore, an exponential deviate with mean m_1 is generated with probability p or, with probability $1 - p$, an exponential deviate with mean m_2.

However, $f(x)$ is still a proper probability density function if $1 < p < m_1/(m_1-m_2)$. Generation of the deviate in this case uses a trick. This derives from the fact that the non-convex mixed exponential can be obtained as a mixture of an exponential deviate with mean m_1 (Probability $q = p - (p-1)m_1/m_2$) or the sum of two exponentials with means m_1 and m_2 respectively (probability $1 - q$).

4. MULTIVARIATE RANDOM DEVIATES

For multivariate random deviates the main continuous case of interest is that of multivariate normal random deviates with given covariance matrix (GGNSM). This is done by factoring the covariance matrix into the form LL^t, where L is triangular and is used to generate the desired N-vector as N linear combinations of N independent normal random variables. Since the matrix decomposition is time consuming, an option is provided to by-pass the decomposition during repeated calls to GGNSM with the same covariance matrix.

In the discrete case the main interest lies in multinomial random variables. This subroutine has been rewritten for Edition 8 as the previous version was extremely inefficient. For later editions of Chapter 8 it is hoped to supply subroutines to generate multivariate uniform, exponential and Gamma variables. The situation here is complex because there are so many variates with, say, uniform marginals. However, recent work has supplied uniform and exponential variates which are simple to simulate and have a broad range of dependencies.

5. STOCHASTIC PROCESSES AND POINT PROCESSES

Finally, a new feature of Chapter G is several capabilities for generating stochastic processes and sequences. In Edition 7 the only capability provided was a standard routine for generating normal linear sequences (ARMA (p,q) processes) for Box-Jenkins time series analysis. This has now been supplemented by a subroutine (GGNPP) which generates non-homogeneous Poisson processes using a newly developed thinning algorithm due to Lewis and Shedler (1979). The subroutine generates the non-homogeneous Poisson process for any given rate function $\lambda(t)$; there is a built-in capability to generate either all the events which will occur in a fixed interval or else the successive times-between-events "one-at-a-time" for use in discrete event simulations.

The algorithm is as follows. Let $\lambda(t)$ be the rate function of the non-homogeneous Poisson process which it is desired to generate on some interval $(t_0,t']$ and let λ^* be the maximum of $\lambda(t)$ on this interval. Let $T_1^*,T_2^*,\ldots$ be the times-to-events from a homogeneous Poisson process with rate λ^*

on $(t_0,t']$. Now delete the events of T_i^* independently with probabilities $1-\lambda(T_i^*)/\lambda^*$ or else retain them. Then the retained (non-deleted) events are events from a realization of a nonhomogeneous Poisson process with rate function $\lambda(t)$.

In this algorithm a subroutine for computing the rate function $\lambda(t)$ must be supplied by the user; the homogeneous Poisson process with rate λ^* is generated using the fact that the times between events in this homogeneous process are independent exponential deviates with means $1/\lambda^*$.

Several other stochastic processes and sequences are under consideration for inclusion in later versions of Chapter G. These include analogs of normal ARMA (p,q) processes with exponential, Gamma and mixed exponential marginal distributions.

Not as easily classified are two features of the generation portion of Chapter G which have wide use in statistics. One is a subroutine (GGPER) to generate a random permutation of the integers 1 to K, and the other is a subroutine (GGSRS) to generate a simple random sample from a finite population. Random permutations are needed, for instance, in simulating the null-distributions which occur in non-parametric statistical methods.

REFERENCES

Chambers, J.M., C.L. Mallows, and B.W. Stuck (1976), "A method for simulating stable random variables," J. Am. Statist. Assoc., 71, 340-344.

Fishman, G.S. and L.R. Moore (1978), A statistical method for evaluation of multiplicative congruential generators with modulus 231-1, Technical Report 78-11, The University of North Carolina at Chapel Hill.

Kronmal, R.A. and A.V. Peterson, Jr. (1979), "On the alias method for generating random variables from a discrete distribution," The American Statistician, 33(4), 214-218.

Lewis, P.A.W., A.S. Goodman, and J.M. Miller (1969), "Pseudo-random number generator for the System/360," IBM Systems Journal, 8(2), 136-146.

Lewis, P.A.W. and G.S. Shedler (1979), "Simulation of non-homogeneous Poisson processes by thinning," Naval Research Logistics Quarterly, 26(3), 403-413.

Learmonth, G.P. (1976), "Empirical tests of multipliers for the prime-modulus random number generator $X_{i+1} = AX_i \pmod{2^{31}-1}$," Proceedings of the Ninth Interface Symposium on Computer Science and Statistics, Prindle, Weber and Schmidt, Boston.

Learmonth, G.P. and P.A.W. Lewis (1974), Statistical tests of some widely used and recently proposed uniform random number generators, Technical Report 55Lw73111A, Naval Postgraduate School, Monterey, California.