Statistics Canada Statistique Canada

Methodology Branch

Business Survey Methods Division

Direction de la méthodologie

Division des méthodes d'enquêtes entreprises

Canada

A Note on Various Methods for Generating Random Numbers With a Given Distribution

M.A. Hidiroglou
December, 1984

Working Paper No. BSMD 85-034E

Une Note sur quelques Méthodes Servant à Produire des nombres aléatoires ayant une distribution donnée.

par

M.A. Hidiroglou
(Division des Méthodes d'Enquêtes-Entreprises)

Sommaire

Il existe un besoin pour produire des nombres aléatoires suivant une distribution donnée. Pour des petits sondages, une table de nombres aléatoires peut suffir à obtenir une liste des unités à échantillonner. Cependant, à mesure qu'un sondage devient de plus en plus complexe, les nombres aléatoires doivent être produits à l'aide de l'ordinateur. Cette note décrit plusieurs méthodes qui peuvent être utilisées à ce but, et, donne aussi un aperçu des problèmes associées à la génération de nombres aléatoires par ordinateur.

A NOTE ON VARIOUS METHODS FOR GENERATING RANDOM

NUMBERS WITH A GIVEN DISTRIBUTION

by

M. Hidiroglou

Business Survey Methods Division

1. INTRODUCTION

There is a need to generate uniformly distributed random numbers when it comes to sample selection in a survey. For small surveys, a table of random numbers can be used to list the elements to be sampled. However, when the survey increases in level of complexity, such as is the case for a moderately large business survey, random numbers have to be computer-generated. In using a computer random number generator, one must bear in mind that the methods used in creating such numbers approximate a random sequence. Hence, what we are really generating are numbers which we refer to as pseudo-random numbers. There are several methods at hand for generating pseudo-random numbers.

Historically, Von Newman came up with a method known as the mid-square technique; here, the idea is to take the square of a random number and extract the middle digits. It is a poor random number generator, however,

because it may be short cycled. The most widely used method for generating random numbers is known as the congruential method; we will discuss its algorithm and its merits. Random number generators should be understood by the user in terms of their mathematical background and their validity. To quote Knuth (1970), "There is a tendency for people to avoid learning anything about random number generators; quite often, we find that some old method which is comparatively unsatisfactory has blindly been passed down from one programmer to another". To precisely illustrate the above comment, IBM users (see Programmer's Manual 1968) are often tempted to use the library routine RANDU. However, as we shall see later, RANDU is a poor way to generate random numbers. There are some excellent random number generators around. One of them, known as Super-Duper, has been written by an authority on random number generators, G. Marsaglia (1972). It is superior to RANDU in that it passes the tests of randomness and distributional uniformity much better.

A widely used application of random numbers is in the art of computer simulation or Monte-Carlo experiments. For this purpose, it is quite important to generate sequences which obey a distributional property. As we will see, the basic construction block for various distributional sequences is the uniform distribution. It is therefore quite important to start off right with a good uniform random number generator. Otherwise, inferences derived from the Monte-Carlo experiment may be biased.

2. GENERATING UNIFORM RANDOM NUMBERS

We wish to generate real numbers U_n, uniformly distributed between zero and one. We generate integers X_n between zero and some number m where m is the word size of the computer, and compute U_n as:

$$U_n = X_n/m$$

The algorithms for calculating X_n will depend on some earlier X_n's so that this calculated sequence must eventually be periodic. A successful method for generating pseudo-random sequences is known as the linear congruential method. This sequence is generated as follows:

$$X_{n+1} = (aX_n + c) \bmod m$$

where X_0 is the starting value $X_0 \geq 0$,

a is the multiplier $a \geq 2$,

c is the increment $c \geq 0$,

and m is the modulus $m > X_0$, $m > a$, $m > c$.

Choices of X_0, a, c and m have to be done carefully, otherwise the series will get into a cycle of short length. We refer to two types of congruential methods, the multiplicative and the mixed, depending whether $c = 0$ or $c \neq 0$, respectively.

Choices of the modulus m strongly influences the length of a generator's cycle and its speed of generation. Usually, m is in the form of a high power of 2, dependent on the machine's capacity for handling large integers. Also, the multiplier "a" can be chosen so as to give a period of maximum length; this is given in the following theorem provided by Greenberger.

The linear congruential sequence has a period of length m if, and only if,

i) c is relatively prime to m,

ii) $b = a-1$ is a multiple of p, for every prime p dividing m,

iii) b is a multiple of 4, if m is a multiple of 4.

A less used method is the quadratic congruential method.

$$X_{n+1} = (d\, X_n^2 + a\, X_n + c) \bmod m.$$

A theorem comparable to the previously stated one can also be obtained for this method. However, we just note in passing that there are several random number generating mechanisms, which essentially generalize the linear congruential method. Once a generator has been decided upon, its adequacy in terms of randomness must be statistically validated.

3. STATISTICAL TESTS

Does a given sequence behave randomly? This is the basic question with respect to random number generators. How does one define randomness? In a vague sense, it is intuitively appealing that such a sequence must not repeat itself in a regular pattern.

There are basically two methods for testing the randomness of a generator: empirical tests, for which the computer manipulates groups of numbers of the sequence and evaluates certain statistics; and theoretical tests, for which characteristics of the sequence are established by using number-theoretical methods based on the recurrence rule used to form the sequence.

Some of the most widely used general tests are the X^2 test and the Kolmogorov-Smirnov test. However, these two tests are like the first stepping stones to a fine sorting process when it comes to randomness testing. We will briefly describe the first of two clases of tests, the empirical tests. We will spend more on the second kind, the theoretical test, and more specifically, the spectral test. RANDU's performance as a random generator will become quite apparent when the spectral test is used.

(a) Empirical Tests

As was said in an earlier paragraph, groups of numbers of a random sequence are analysed statistically when this method is used. More

specifically, these tests are applied to the transformed integer sequences of the sequence generated between 0 and 1. We proceed to name and describe a few of these tests.

1. Equidistribution Test

   This test is applied to the transformed integer sequence using the Kolmogorov-Smirnov or the X^2 test.

2. Serial Test

   This is a two-dimensional (normed) test using the X^2 statistic. We measure the degree of independence between pairs of successive numbers in the sequence.

3. Gap Test

   This test examines the length of "gaps" between occurrences in the sequence. The X^2 test is used on a sequence of gaps.

4. The Run Test

   This is one of the strongest empirical tests. It is to be recommended for testing random numbers using the linear congruential generator. This test examines the length of monotone subsequences of the original sequence.

(b) **Theoretical Tests**

With this type of test the effects of the constraints a, m, and c which are part of the linear congruential generator is studied. One of the most basic theorems related to a theoretical test provided by Knuth (1970) is the following: Let X_0, a, c and m generate a linear congruential sequence with maximum period; let b = a-1, and let d be the greatest common divisor of m and b. Then,

$$P(X_{n+1} < X_n) = 1/2 + r$$

where $r = [2\ c(\text{mod } d) - d\ ]/2m$.

This is quite an important result, for it tells us that a random sequence is likely to oscillate quite frequently during the entire period of the generator. We refer to d as the potency of the series: series with potencies over 4 are desirable.

The serial correlation test may be applied over the entire period using generalized Dedekind sums. This correlation is defined as

$$C = \frac{\left[m \sum_{0 \le x < m} xs(x) - \left(\sum_{0 \le x < m} x\right)^2\right]}{\left[m \sum_{0 \le x < m} x^2 - \left(\sum_{0 \le x < m} x\right)^2\right]}$$

where $s(x) = (ax + c) \bmod m$.

Discarding terms of order 1/m, we have that the serial correlation can be expressed as $\sigma\ (a,\ m,\ c)$

where $$\sigma(a,m,c) = 12 \sum_{0 \le j < m} \left(\left(\frac{j}{m}\right)\right) \left(\left(\frac{aj + c}{m}\right)\right)$$

$$((x)) = Z - \lceil Z \rceil + 1/2 - 1/2\ \delta(Z)$$

$$\delta(Z) = \begin{cases} 1 \text{ if } Z \text{ is an integer.} \\ 0 \text{ if } Z \text{ is not an integer.} \end{cases}$$

Essentially $\sigma(a,m,c)$ is an orthogonal expansion. Examples as to how to compute $\sigma(a,m,c)$ are given in Knuth (1972).

We now turn our attention to a test formulated by Coveyou and MacPherson (1965); this test is known as the spectral test. This test is important to study the quality of linear congruential random number generators. It is by far the most powerful test known. The resulting expressions are usually evaluated with a computer program as they involve mimimization of quadratic forms over the integers.

The most important randomness criteria relies on the properties of t consecutive elements of the sequence, and the spectral test deals directly with this distribution. If we have a sequence $[U_n]$ of period m, the idea is to analyze the set of m points

$$(U_n, U_{n+1}, \ldots, U_{n+t-1})$$

in a t - dimensional space.

Let $1/\nu_t$ be the maximum distance between hyperplanes, taken over all families of parallel (t-1) dimensional hyperplanes that cover all points $\{ (x/m, s(x)/m, s^{t-1}(x)/m) \}$; we call ν_t the t-dimensional accuracy. The accuracy of a periodic sequence decreases as t increases while it remains the same in the case of a truly random sequence.

The spectral test consists of the determination of ν_t for small t, say 2 _ t _ 6. The spectral test rotates a t-dimendional hypercube and looks at the maximal distance between successively generated points in the sequence. ν_t is obtained by minimizing

$$(x_1^2 + x_2^2 + \ldots + x_t^2)^{1/2}$$

subject to $x_1 + ax_2 + \ldots + a^{t-1}\ x_t = 0 \pmod{m}$. Knuth (1972) has has given an algorithm which would solve this system.

A criterion which is relatively independent of m is obtained by normalizing by the volume of the ellipse in t-space defined by the relation

$$(x_1\, m - x_2\, a - \ldots - x_t\, a^{t-1}) + x_2^2 + \ldots\, x_t^2 \leq \nu_t^2 \quad .$$

The resulting coefficients of this normalized volume are,

$$C_t = \frac{\Pi^{t/2}\ \nu_t^t}{(t/2)!m}, \qquad t = 1, 2, \ldots;$$

$$\text{where} \left(\frac{t}{2}\right)! = \left(\frac{t}{2}\right) \left(\frac{t}{2} - 1\right) \ldots \left(\frac{1}{2}\right) \Pi^{1/2} \text{ for t odd}$$

$$\text{and } \left(\frac{t}{2}\right)! = \left(\frac{t}{2}\right) \left(\frac{t}{2} - 1\right) \ldots \left(1 \right) \text{ for t even.}$$

Table 1 gives the volume of the ellipses as the dimension of the hyperspace increases from 2 to 6.

Table 1 - Sample Results for the Spectal Test (Knuth 1972)

No.	a	m	C_2	C_3	C_4	C_5	C_6
1	23	10^8+1	0.000017	0.00051	0.014	0.343	4.6232
2	$2^{18}+1$	2^{35}	3.14	2×10^{-9}	2×10^{-9}	5×10^{-9}	10^{-8}
3	3141592221	10^{10}	1.44	0.44	1.92	0.07	0.08
4	3141592221	2^{35}	1.24	1.70	1.12	2.79	3.81
5	5^{15}	2^{35}	2.02	4.02	4.03	0.40	2.62
6	$2^{16}-3$	2^{32}	3.14	10^{-5}	10^{-4}	10^{-3}	0.02

Case No. 6 is the famous RANDU random number generator that is on the IBM 360 library program. As can be seen from the tabulated results, RANDU is good in two dimensions, however, it fails badly in the higher dimensions. It should be noted that RANDU obeys the following rule for three successively generated numbers of its sequence, that is

$$9X_n + 6X_{n+1} + X_{n+2} = 0 \bmod (2^{31}).$$

This automatically makes it fail in three dimensions.

4. TECHNIQUES TO GENERATE RANDOM NUMBERS WITH A GIVEN DISTRIBUTION

In the following sections, we provide methods that are used to generate random numbers with a given distribution. This is important, since Monte-Carlo techniques embody the generation of linear forms with a given stochastic disturbance.

(a) The Inverse Cumulative Function Rule

For the purpose of introducing random variables into a simulation model, one of the most important transformations of random variables is that which transforms a random variable X according to its own cumulative distribution function $F_X(a)$, that is, for a continuous random variable X, one can discuss the nature of the random variable U defined by

$$U = F_X (X)$$

a monotonically and continuously increasing function due to the nature of the cumulative distribution function. The resulting variate U is restricted to values between 0 and 1, although its distribution between those values may not be so apparent until one computes the cumulative distribution for U: letting F_U be the distribution (cumulative) function of U, we have

$$F_U\ (a) = P(U \leq a) = P[X \leq F_X^{-1}(a)\ ] \text{ for } 0 < a < 1,$$

where F_X^{-1} (a) is the inverse cumulative distribution function (CDF) for the random variable X. That is, F_X^{-1} (a) gives that value X, which when substituted into F_X produces a.

Hence, $F_U\ (a) = P[X \leq F_X^{-1}\ (a)\ ] = F_X\ [F_X^{-1}\ (a)\ ] = a.$

Consequently, the random variable $U = F_X\ (X)$, defined as the transformation of any arbitrary continuous random variable X according to its cumulative distribution function, is a uniformly distributed random variable.

This result is of special importance in that one can use the inverse of the CDF transformation, itself a monotone increasing function, in order to generate random variables having a particular cumulative distribution function. To accomplish this purpose, we first generate uniformly distributed random variables U; they are next transformed according to

$$X = F_X^{-1}\ (U).$$

This results in random variables whose cumulative distribution function is given by $F_X\ (x)$.

We proceed to provide some examples of this technique. For example, a commonly arising distribution is the negative exponential, which has probability distribution function of the form

$$f(x) = \lambda e^{-bx}.$$

and cumulative distribution function

$$F(x) = 1 - e^{-\lambda x} .$$

The inverse of this function is:

$$x = - \log_e (1-F(x)).$$

To generate x according to $F(x)$, simply choose a $U \sim U(0,1)$ random

$$x = -(1/\lambda) \log_e (1-U).$$

Another application of interest is the Weibull random variable with probability density function

$$f(x;\lambda,k) = \begin{cases} k\lambda x^{k-1} \exp(-\lambda x^k) & x > 0 \\ 0 & x \leq 0. \end{cases}$$

The corresponding cumulative distribution function is

$$F(x;\lambda,k) = \begin{cases} 1 - \exp(-\lambda x^k) & x > 0 \\ 0 & x \leq 0. \end{cases}$$

The inverse of this function is:

$$x = [(-1/\lambda) \ln (1 - F(x; \lambda,k)]^{1/k}.$$

To generate x according to $F(x;\lambda,k)$, simply choose a $U \sim U(0,1)$ random number

$$x = [-(1/\lambda) \ln (1-U)]^{1/k}.$$

(b) **Log-Trig Method for Generating Gaussian Variates**

For many density functions, it is not possible to obtain a closed form for the cumulative distribution function. Hence, other ways of generating random numbers have to be developed for these distributions.

In the case of the Gaussian or normal (0,1) distribution, the trig solution is at hand.

Supposing that.

$$Z_i \approx \text{NID}\,(0,\ \sigma^2),\ i=1,\ 2$$

then $f(z_1, z_2) = (2\pi\ \sigma^2)^{-1} \exp - \{z_1^2 + z_2^2)/2\sigma^2\ \}$.

This is equivalent to producing a random point (Z_1, Z_2) in the Euclidean space of two dimensions. We may transform our coordinates to polar coordinates $(R, \emptyset)$ as follows:

$$Z_1 = R \cos \emptyset$$

$$Z_2 = R \sin \emptyset.$$

The Jacobian of such a transformation is r, and hence the joint density function becomes

$$P(r, \emptyset) = (2\Pi\ \sigma^2)^{-1}\ r \exp\,(-r^2/2\sigma^2)$$

for $r > \emptyset$ and $0 \leq \emptyset \leq 2\pi$

The marginal distribution for Ø becomes the rectangular distribution.

Hence,
$$h_1(\emptyset) = \begin{cases} 1/2\pi & 0 \leq \emptyset \leq \pi \\ 0 & \text{other } \emptyset . \end{cases}$$

The marginal distribution for R becomes the Raleigh distribution of parameter σ.

$$h_2(r) = \begin{cases} \sigma^{-2} r \exp(-r^2/2\sigma^2), & r > 0 \\ 0 & , r \leq 0. \end{cases}$$

The inverse transformations for h_1 and h_2 are

$$\emptyset = 2\pi$$

$$r = [-2 \ln (1 - h_2(r)]^{1/2}.$$

Hence, pairs of the following will generate the pairs of variables whose parental distribution is the normal distribution:

$$Z_1 = [-2 \ln (1 - U_1)]^{1/2} \cos (2\pi U_2)$$

$$Z^2 = [-2 \ln (1 - U_2)]^{1/2} \sin (2\pi U_2)$$

where U_1 and U_2 are uniformly distributed random variables.

Random variables derived from a $N(\mu, \sigma^2)$ distribution can be generated from the N(0, 1) variables as follows:

$$W_1 = \sigma Z_1 + \mu$$

$$W_2 = \sigma Z_2 + \mu .$$

(c) <u>The Chi-Squared Family</u>

The probability density function of a χ^2 is

$$f(u) = \begin{cases} (1/2)^{1/2}\, u^{1/2-1}\, e^{-u/2} / \Gamma(1/2), & u > 0 \\ 0 & , u \leq 0 \end{cases}$$

where $\Gamma(n) = (n-1)\ \Gamma\ (n-1)$. The cumulative density function is

$$F(u) = \begin{cases} \int_0^u (1/2)^{1/2}\, U^{1/2}\ e^{-U/2}\, dU / \Gamma(1/2), & u > 0 \\ 0 & , u \leq 0. \end{cases}$$

This is not easily inverted; hence, one has to seek other techniques for producing random variables generated from such a distribution. The samples from this distribution may be obtained by using the fact that X^2 on n d.f. is the convolution of n variables distributed as a χ^2 with one d.f., which is in turn the square of a normal random variate. Hence, a sample value can be obtained from a X^2 with n d.f. by selecting n random normal deviate, squaring and adding them.

A quicker way to generate a X is as follows. Recalling that a X with 2 d.f. is an exponential distribution of mean 2, we use this fact. For an even degrees of freedom (n = 2p) we take

$$X_n^2 = \sum_{i=1}^{p} V_i$$

where V_1 is exponentially distributed. For odd degrees of freedom (n 2p+1), we take

$$X_n^2 = \sum_{i=1}^{p} V_i + Z^2$$

where Z is normally distributed N(0, 1).

VI REFERENCES

(1) CHEN, E.H. - "A Random Normal Number Generator for 32 bit-work Computers". Journal of the American Statistical Association, 66, (March 1971), 400-3.

(2) CALDWELL, Robert L. - Correlational Defects in the Standard IBM 360 Random Number Generator and the Classical Ideal Gas Correlation Function", Journal of Computational Physics, 14, 2 (February 1974), 223-6.

(3) COVEYOU, R.R. and MACPHERSON, R.D. - "Fourrier Analysis of Uniform Random Number Generators", Journal of the Association for Computing Machinery 14, (January 1967), 100-13.

(4) IBM Corporation: System/360 - Scientific Subroutine Package (360 A -CM -03X) Version III, Programmer's Manual, H20-0205-3, 1968.

(5) KNUTH, Donald E - "The Art of Computer Programming ", Vol. 2, Addison-Wesley, 1970.

(6) MACLAREN, M.D. and MARSAGLIA, G.- "Uniform Random Number Generators", Journal of the Association for Computing Machinery, 12, (January 1965), 83-9.

(7) MARSAGLIA, George. - "The Structure of Linear Congruential Sequences", in S.K. Zaremba, ed., Applications of Number Theory to Numerical Analysis, New York Academic Press, 1972.

(8) MIRHAM, Arthur G. - "Simulation - Statistical Foundation and Methodology", Academic Press, 1972.

(9) SOWEY, E.R. - "A Chronological and Classified Bibliography on Random Number Generation and Testing", International Statistical REview, 40,3 December 1972, 355-71.

(10) WELCH, Roy E - "Graphics for Data Analysis", to appear in Computers and Graphics, 1975.

(11) YUEN, Karen K. and DIXON, W.J. - "The Approximate Behaviour and Performance of Two-Sample Trimmed t", Biometrika 60, 2 August 1973, 309-75.

APPENDIX

A reservoir sampling Computer Program.

The following FORTRAN program should be used to generate uniform random (0,1) numbers. It has good properties under the spectral test.

```
DOUBLE PRECISION U1,U2,DLOG,DSIN,DCISm,DSQRT

REAL*8 E(2,100)

INTEGER*4 NA(128)

C       NSIZE = NO. OF ELEMENTS IN SAMPLE

C       NSAMP - NO. OF SAMPLES TO BE GENERATED


NSAMP=

NSIZE=

IN   =

IØUT =

INTEMP=63783

IX = 175632999

ISENI = 68593

ISEN2 = 63253723

SEN3 = 0.23283064E-9

DO 1 I = 1,128

IX = IX*ITEMP

NA(I) = IX

L = IX

M = L*(5**13)

KK = M*(5**13)

DO 2 JJI = 1,NSAMP

DO 3 JVR = 1,2

DO 3 JI = 1,MSIZE
```

```
L = L*ISEN1
M = M*ISEN2
J = 1+IABS(L)/1677216
JP = J
UI = 0.5+(FLOAT (NA(JP)+L+M)*SEN3
KK = KK+687471237
NA(JP) = KK
L = L*ISEN1
M = M*ISEN2
J = 1+IABS(L)/16777216
JP = J
U2 = 0.5+(FLOAT(NA(JP)+L+M)*SEN3
KK = KK*687471237
NA(JP) = KK
C     RANDOM NORMAL(0,1) DEVIATES
C     E(JVR,2*JI-1) = DCOS(6.28318*V2)*DSQRT(-2.*DLOG(U1))
E(JVR,2*J1) = DSIN(6.28318*V2)*DSQRT(-2.*DLOG(U2))
3 CONTINUE
2 CONTINUE
RETURN
END
```