A Comparison of Methods for Generating Normal Deviates on Digital Computers*

MERVIN E. MULLER

Princeton University, Princeton, N. J., and International Business Machines Corp., New York, N. Y.

Abstract. Two methods recently developed for generating normal deviates within a computer are reviewed along with earlier proposals. A comparison of the various methods for application on an IBM 704 is given. The new direct method gives higher accuracy than previous methods of comparable speed. The detailed inverse technique proposed yields accuracy comparable with, or better than, most previous proposals using about one-quarter the computing time.

1. *Introduction*

Many applications of electronic computers require the efficient generation of large numbers of pseudo random normal deviates. Tables of pseudo random normal deviates are of course available, for example [15, 19], but they are not sufficiently extensive for many purposes and an outside source of this kind cannot usually be used effectively by the computer. What is required is some method of generation which can be rapidly carried out by the machine itself. From independent random normal deviates well-known methods can of course be used to generate n-dimensional normal deviates with arbitrary means and variance-covariance matrix.

A number of different ways of generating pseudo random normal deviates is known, for example [1, 7, 12, 17]. The most recent methods are [1] and [12]. All of these approaches have the common feature that they require the use of pseudo random numbers.

Methods are available by which pseudo random numbers may be produced within the machine, see for example [4, 6, 7, 9, 10, 11, 17, 18]. Judging by the results given for example by [6, 7, 14, 17], the most satisfactory procedure, now in use for generating random numbers is the one based on residue class techniques, see for example [11, 16]. We shall not consider here the validity of employing deterministic methods for generating random numbers but assume that some machine method of satisfactorily producing random numbers is available from which random normal deviates are to be produced.

The purpose of the paper is to review the several methods for generating pseudo random normal deviates within a large-scale computer. Certain specific comparisons of these methods for an IBM 704 are given in table 2.

* Received October, 1958. Prepared in connection with research sponsored by the Office of Ordnance Research, U. S. Army; Statistical Techniques Research Group, Princeton University, Contract No. DA 36-034-ORD 2297.

2. *Methods*

2.1 *A Direct Approach.* The following approach which was recently developed by Box and the present writer [1][1] may be used to generate a pair of random deviates from the same normal distribution starting from a pair of random numbers.

METHOD: Let U_1, U_2 be independent random variables from the same rectangular density function on the interval [0, 1]. Consider the random variables:

$$X_1 = (-2 \log_e U_1)^{\frac{1}{2}} \cos 2\pi U_2$$

$$X_2 = (-2 \log_e U_1)^{\frac{1}{2}} \sin 2\pi U_2$$

Then (X_1, X_2) will be a pair of independent random variables from the same normal distribution with mean zero, and unit variance.

2.2 *An Inverse Approach.* In principle, the inverse method of generating a normal deviate X from a uniform deviate U is well known. The problem is to find the inverse relationship $X = X(U)$ given that

$$U = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{X} e^{-t^2/2}\, dt.$$

The actual determination of $X(U)$ offers certain difficulties when it is desired to generate reliable normal deviates, especially for large values of X. The details of the procedure have been carried out for application on a large size binary machine by the author [12].

The relation $X = X(U)$ is approximated stepwise. The interval [0, 1] for U is subdivided so that over each sub-interval it is possible to obtain a reliable and fast procedure for computing X. Over most of [0, 1], $X = X(U)$ is approximated by Chebyshev-type polynomials. As X becomes large in absolute value it is necessary to increase the degree of the approximating polynomial. However, even though the degree of the polynomial increases, the frequency with which these approximations are needed decreases, hence this method will use, on the average, a low order of approximating polynomial for $X = X(U)$. Due to symmetry it is actually only necessary to study $X = X(U)$ for $\frac{1}{2} \leqq U \leqq 1$. For $127/128 \leqq U \leqq 1$, and by symmetry $0 \leqq U \leqq 1/128$, $X = X(U)$ has a singularity of logarithmic type, consequently for U in this subinterval an approximation of more subtle type than Chebyshev polynomials is needed. Here a satisfactory rational approximation is obtained by using a truncated continued fraction expansion. The necessary coefficients for the approximations for $X = X(U)$ are given in [12].

Having specified the two most recent methods we will now review the earlier methods.

2.3 *Central Limit Approach.* By appealing to the central limit theorem of probability, e.g. [3], we know what sums of an arbitrary number of U's will be

[1] As pointed out by the referee of this paper, $X_1 + X_2$ in the third equation on page 611 of [1] should read $X_1^2 + X_2^2$.

asymptotically normally distributed. Though this method is easy to use, it has certain limitations which will be considered in section three.

2.4 *Rejection Approach.* This technique is attributed to von Neumann. The presentation here follows essentially that in [17]. One proceeds to generate normal deviates in the truncated region $-b \leqq X \leqq b$ as follows: Generate uniform deviates U_1 and U_2. Each time compute $Y = -2b^2(U_1 - \frac{1}{2})^2$. If $\log_e U_2 \leqq Y$, then the value $X = b(2\,U_1 - 1)$ is used as a normal deviate. If $\log_e U_2 > Y$ then one rejects the pair (U_1, U_2) and repeats the above process.

2.5 *Hastings' Approach.* Pade-type or rational approximations to transform a uniform deviate to a normal deviate have been suggested by several people. The best known version is due to C. Hastings [5, p. 192]. Using this approach one obtains a normal deviate X from a uniform deviate $U = q$ as follows:

$$X = X^*(q) = \eta - \left\{ \frac{a_0 + a_1\eta + a_2\eta^2}{1 + b_1\eta + b_2\eta^2 + b_3\eta^3} \right\}$$

where

$$\eta = \sqrt{\ln 1/q^2}, \qquad q = \frac{1}{\sqrt{2\pi}} \int_{X(q)}^{\infty} e^{-(1/2)t^2}, \qquad 0 < q \leqq .5,$$

and

$$\begin{aligned} a_0 &= 2.515517 & b_1 &= 1.432788 \\ a_1 &= 0.802853 & b_2 &= 0.189269 \\ a_2 &= 0.010328 & b_3 &= 0.001308 \end{aligned}$$

2.6 *Teichroew's Approach.* A fixed number of uniform deviates is summed, then using an interpolating Chebyshev polynomial an improved approximate normal deviate is obtained. The complete details have been obtained by Teichroew [17]. Teichroew calls this the method of "Approximation by Curve Fitting". Since his paper has had limited circulation the details are reviewed here. He proceeds to find a normal deviate as follows:

It is desired to find $y = m(\Theta)$ where

$$\int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\, dt = \int_0^{\Theta} \phi_\gamma(t)\, dt,$$

and where ϕ_γ is the density function of the sum of γ uniform deviates. This approach requires that the range of Θ be restricted. For the restricted range $\Theta_L \leqq \Theta \leqq \Theta_U$, $y = m(\Theta)$ is approximated by determining an interpolating polynimial. A Chebyshev polynomial of degree $k - 1$ is fitted so that its values coincide with the values of $m(\Theta)$ at the k roots of the Chebyshev polynomial of degree k.

Though Teichroew has obtained the coefficients of the Chebyshev polynomials for $\gamma = 6$, 8 and 12, we shall describe the case $\gamma = 12$ since this value of γ gives the most satisfactory results. As before, U_i denotes a uniform deviate, and let $\Theta = \sum_{i=1}^{12} U_i$ so that $0 \leqq \Theta \leqq 12$. However, Θ must be restricted; here Teichroew chooses $\Theta_L = 2$, $\Theta_U = 10$, where $\text{Prob}\,\{\Theta_L \leqq \Theta \leqq \Theta_U\} \geqq 1 - (2)10^{-5}$, so there is a very small probability of being outside this range. His program was prepared for SWAC and was arranged so that if Θ fell outside (2, 10) the machine would halt and then type out this fact. (This procedure has been adopted by this writer for the IBM 704 and will appear in the near future as a SHARE program.) Let $r = (\Theta - 6)/4$. Then an approximate normal deviate X is obtained as

$$X = \sum_{j=0}^{9} d_{2j+1} T_{2j+1}(r).$$

For machine computation it is not economical or convenient to find X in the above form; so a rearranged series in r is obtained. If we truncate the series after the term $T_9(r)$, this series will be

$$X = a_1 r + a_3 r^3 + a_5 r^5 + a_7 r^7 + a_9 r^9,$$

where the coefficients are as follows:

$$a_1 = 3.949846138$$

$$a_3 = 0.252408784$$

$$a_5 = 0.076542912$$

$$a_7 = 0.008355968$$

$$a_9 = 0.029899776$$

3. *Appraisals of the Methods*

3.1 *The Direct Approach.* This method developed from the desire to have a way of generating normal deviates which would be reliable in the tails of the distribution. Mathematically this approach has the attractive advantage that the transformation for going from uniform deviates to normal deviates is exact. Whereas the accuracy of other methods is not easy to analyze, or to change, the accuracy obtainable here depends essentially on the precision of the necessary function subroutines. Further, since most computing centers have library programs to compute values of trigonometric functions, logarithms, and square roots this approach requires little additional machine program writing.

Inspection of table 2 in section 4 shows that this method gives higher accuracy than previous methods of comparable speed and it indicates the amount of memory space required.

3.2 *The Inverse Approach.* This method has the advantage that if a reasonable level of accuracy is desired over a given range of X it is possible to achieve a very fast procedure for generating X at the expense of utilizing memory space in the computer. If this approach is to be efficient the approximations to $X = X(U)$ should be designed to work over subintervals of U whose lengths are either a negative power of two or a negative power of ten depending upon whether the computer operates in the binary or decimal mode; see [12]. If the approximation of $X = X(U)$ is to be valid when X is allowed to take values beyond three standard deviations from the mean, then this approach may require too many memory locations to be practicable on a small computer if a very high level of accuracy is required.

This procedure has been developed by the author [12] for use on a large scale binary computer. This procedure then yields accuracy comparable with, or even better than most previous proposals using about one-quarter of the computer time. It was developed so that the maximum absolute error in X should be less than 4×10^{-4} in the range $-5 \leqq X \leqq 5$, where Prob $\{-5 \leqq X \leqq 5\} > 1 - 6 \times 10^{-7}$, so that X is correct to within 4×10^{-4} except for an event of probability less than 6×10^{-7}. Inspection of table 2 in section 4 shows that at least for a machine the size of the IBM 704 the increase in computing speed possible with this method has not been acquired at the expense of an excessive amount of additional storage. It should also be kept in mind that if the range of X is decreased then the computing time and storage requirements can be considerably reduced.

3.3 *The Central Limit Approach.* This approach has the advantage of requiring little memory space and it appears to be reasonably fast. However, if the values of X can be restricted to be within three standard deviations of the mean, then for the accuracy attainable here the speed of this procedure is easily matched or improved upon by other methods. Further, if values of X beyond three standard deviations from the mean are required this approach is unreliable; see for example [8]. Finally, some objection to this procedure has been raised in the past due to the fact that here one requires that several uniform deviates be generated for each approximate normal deviate. The problem of comparing the accuracy of this approach with others is complicated to the extent that the Central Limit Theorem is concerned with an asymptotic convergence in probability. Consequently in comparing this procedure with others it is necessary to see how the actual distribution function of a finite number of uniform deviates compares to the limiting normal distribution function. If one takes sums of twelve uniform deviates, then its distribution about its mean of six will have the advantage of having a unit variance.

Table 1 gives an indication of the errors one makes in using this approach. For example, inspection of the table shows that the probability of the sum of 12 uniform deviates being greater than 3.2000 above its mean is 0.455824×10^{-3}. Yet this probability point for normally distributed deviates with the same mean and variance gives a value of 3.3165. Thus the difference is -0.1165.

3.4 *Rejection Approach.* For some distributions rejection-type techniques, see for example [2, 8, 13], are acceptable. However, for the normal distribution this approach is very inefficient, especially if precise tail values are important. This method is mentioned here only for the sake of historical completeness. If one wanted to generate normal deviates in the truncated region $-b \leqq X \leqq b$, then the inefficiency of the process can be appreciated by realizing that the probability that a pair (U_1, U_2) will be used to generate a normal deviate, namely Prob $\{U_2 \leqq \exp(-2b^2(U_1 - \frac{1}{2})^2\}$, equals $\int_0^1 \exp(-2b^2(U - \frac{1}{2})^2\, dU$, which is

TABLE 1

Normal deviates exceeded with certain probabilities compared with sums of 12 uniform deviates exceeded with the same probabilities (means and variances equated)

Sum of 12 deviates	Probability of a larger deviate	Normal deviates with same mean and variance with same probability	Difference of deviates
0.0000	0.500000	0.0000	0.0000
0.2000	0.421711	0.1975	0.0025
0.4000	0.346338	0.3952	0.0048
0.6000	0.276483	0.5933	0.0067
0.8000	0.214180	0.7920	0.0080
1.0000	0.160727	0.9915	0.0085
1.2000	0.116639	1.1912	0.0088
1.4000	0.817077×10^{-1}	1.3937	0.0063
1.6000	0.551457×10^{-1}	1.5969	0.0031
1.8000	0.357846×10^{-1}	1.8018	−0.0018
2.0000	0.222756×10^{-1}	2.0089	−0.0089
2.2000	0.132681×10^{-1}	2.2183	−0.0183
2.4000	0.754029×10^{-2}	2.4304	−0.0304
2.6000	0.407497×10^{-2}	2.6458	−0.0458
2.8000	0.208611×10^{-2}	2.8648	−0.0648
3.0000	0.100700×10^{-2}	3.0882	−0.0882
3.2000	0.455824×10^{-3}	3.3165	−0.1165
3.4000	0.192173×10^{-3}	3.5505	−0.1505
3.6000	0.748223×10^{-4}	3.7917	−0.1917
3.8000	0.266137×10^{-4}	4.0410	−0.2410
4.0000	0.852607×10^{-5}	4.3004	−0.3004

asymptotically $(1/b)\sqrt{\pi/2}$. It is possible to introduce stratified sampling techniques in an attempt to improve this procedure, but to date no efficient improvement for this procedure is known.

3.5 *Hastings' Approach.* This elegant set of approximations is very reliable. Except for certain important subintervals such as $(1.48 \leqq X \leqq 2.42)$, the absolute value of the error is less than 4×10^{-4}, and even on these intervals the absolute values of the error are less than 6×10^{-4}. However, faster procedures requiring less memory space are now available.

3.6 *Teichroew's Approach.* This is a very neat procedure requiring a very small amount of memory space. However, as it now exists it will not generate deviates much beyond four standard deviations from the mean. One might also object to the fact that it uses more than one uniform deviate (in fact either eight or twelve) when generating a normal deviate. Neither of these objections is serious and there is much to recommend this procedure, especially on small machines.

4. *Some Comparisons*

In the comparisons (table 2) it has been assumed that there is no error introduced in generating a uniform deviate. In fact, the truncation error (the error

TABLE 2

Timing, precision, and memory space comparisons for generating normal deviates on the IBM 704

Method	Time per deviate (milliseconds)	Precision		Memory space (reusable temporary locations)
		in units of X	except for probability less than	
Inverse	(Average) = 1.395 (Standard deviation) = 1.261 (87.5% of cases) ≦ 0.996	4×10^{-4}	6×10^{-7}	202 (4)
Sum of 12 uniform deviates	5.052	See table 1	See table 1	25 (4)
Direct	6.601	5×10^{-7}	4×10^{-8}	175 (7)
Teichroew's				
$\gamma = 12$	6.948	2×10^{-4}		46 (4)
$(\gamma = 8)$	6.278	2×10^{-4}	2×10^{-5}	49 (4)
Hastings'	6.968	6×10^{-4}	4×10^{-8}	104 (8)
Rejection	(Average) = 16.360 (Standard deviation) = 28.201	5×10^{-7}	6×10^{-7}	76 (5)

introduced from working with a discrete number of digits for U) in U is not serious until X is desired beyond five standard deviations from its mean. If high precision is required beyond this range it will be necessary to generate U with double precision in order to have enough significant bits in U for those procedures which utilize $U < 5 \times 10^{-6}$. However, this situation can be handled by generating a second random number in order to supply the additional significant bits.

Though the timing and memory space considerations given in table 2 may be valid for other computers, they have been derived for a floating point binary machine with index registers, in this case the IBM 704. Further, the timing and memory space requirements have been evaluated subject to existing library subroutines which are available for the 704 through the SHARE organization. It is possible that some computing time or memory space could be saved if specialized function subroutines were written; however the present approach has the advantage that if a given sampling, or Monte Carlo problem requires some of the same library function subroutines they will be already available in the memory. The memory space requirements indicated include the necessary function subroutines. For each method a normal deviate is formed for use as a "floated" normal deviate.

Acknowledgements

The author wishes to express his appreciation to Drs. G. E. P. Box and John W. Tukey for their interest and generous comments concerning this work. The author is also indebted to Mrs. A. Schay for her excellent help with the numerical computations.

REFERENCES

1. G. E. P. Box and M. E. Muller, A note on the generation of normal deviates. *Ann. Math. Stat. 28* (1958), 610–611.
2. James W. Butler, Machine sampling from given probability distributions. *Symposium in Monte Carlo Methods*, H. A. Meyer, ed., Wiley & Sons, New York, 1956, pp. 249–264.
3. H. Cramer, *Mathematical Methods of Statistics*. Princeton University Press, Princeton, N. J., 1946.
4. George E. Forsythe, Generation and testing of random digits at the National Bureau of Standards, Los Angeles. *Monte Carlo Method*, National Bureau of Standards AMS 12 (U. S. Government Printing Office, Washington, D. C., 1951), pp. 34–35.
5. C. Hastings, *Approximations for Digital Computations*. Princeton University Press, Princeton, N. J., 1955.
6. D. L. Johnson, Generating and testing pseudo random numbers on the IBM Type 701. *Math. Tables Aids Comp. 10* (1956), 8–13.
7. M. L. Juncosa, Random number generation on the BRL high-speed computing machines. Ballistic Research Laboratories, Aberdeen Proving Ground, Md., Report 855 (1953).
8. Herman Kahn, Applications of Monte Carlo. RAND Report No. RM 1237 AEC, (1954).
9. D. H. Lehmer, Mathematical methods in large scale computing units. *Proceedings of Second Symposium on Large-Scale Digital Calculating Machinery*, Harvard University Press, Cambridge, Mass., 1951, pp. 141–146.
10. D. H. Lehmer, Description of "Random number generation on the BRL high-speed computing machines", *Math. Rev. 15* (1954), 559.
11. J. Moshman, The generation of pseudo-random numbers on a decimal calculator. *J. Assoc. Comp. Mach. 1* (1954), 88–91.
12. M. E. Muller, An inverse method for the generation of random normal deviates on large-scale computers. *Math. Tables Aids Comp. 12* (1958), 167–174.
13. J. von Neumann, Various techniques used in connection with random digits. *Monte Carlo Method,* National Bureau of Standards AMS 12 (1951), pp. 36–38.
14. *Symposium in Monte Carlo Methods,* H. A. Meyer, ed. (Gainesville). John Wiley & Sons, New York, 1956.
15. The Rand Corp.: *One Million Random Digits and 100,000 Normal Deviates*. The Free Press, Glencoe, Illinois, 1955.
16. O. Taussky and J. Todd, Generation and testing of pseudo random numbers. *Symposium in Monte Carlo Methods,* H. A. Meyers, ed., John Wiley & Sons, New York, 1956, pp. 15–27.
17. D. Teichroew, Distribution sampling with high speed computers. Ph.D. Thesis, University of North Carolina, 1953.
18. D. F. Votaw and J. A. Rafferty, High speed sampling. *Math. Tables Aids Comp. 5* (1951), 1–8.
19. H. Wold, *Random Normal Deviates*, Tracts for Computers No. XXV, Cambridge University Press, 1948.