N72-18192

NASA TECHNICAL MEMORANDUM

NASA TM X-64637

CASE FILE COPY

UNIFORM RANDOM NUMBER GENERATORS

By William R. Farr
Preliminary Design Office

November 10, 1971

NASA

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
TM X-64637		

4. TITLE AND SUBTITLE	5. REPORT DATE
Uniform Random Number Generators	November 10, 1971
	6. PERFORMING ORGANIZATION CODE: PD-DO-PF
7. AUTHOR(S): William R. Farr	8. PERFORMING ORGANIZATION REPORT #
9. PERFORMING ORGANIZATION NAME AND ADDRESS: George C. Marshall Space Flight Center, Marshall Space Flight Center, Alabama 35812	10. WORK UNIT NO.
	11. CONTRACT OR GRANT NO.
12. SPONSORING AGENCY NAME AND ADDRESS: National Aeronautics and Space Administration, Washington, D. C. 20546	13. TYPE OF REPORT & PERIOD COVERED: Technical Memorandum
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

Prepared by the Flight Mechanics Branch, Flight Performance and Mission Analysis Division, Preliminary Design Office, Program Development.

16. ABSTRACT

Methods are presented for the generation of random numbers with uniform and normal distributions. Subprogram listings of Fortran generators for the Univac 1108, SDS 930, and CDC 3200 digital computers are also included. The generators are of the mixed-multiplicative type, and the mathematical method employed is that of Marsaglia and Bray.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
random, uniform, computer, subprogram, Fortran	STAR Announcement
	ERICH E. GOERNER, Director, Preliminary Design Office

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	17	$3.00

MSFC - Form 3292 (May 1969)

TABLE OF CONTENTS

	Page
INTRODUCTION	1
THEORETICAL CONSIDERATIONS	1
FORTRAN CONGRUENT GENERATORS	4
GENERATION OF NORMAL VARIABLES	6
CONCLUSION	7
APPENDIX A: FORTRAN LISTING OF UNIVAC 1108 UNIFORM GENERATOR	8
APPENDIX B: FORTRAN LISTING OF CDC 3200 UNIFORM GENERATOR	9
APPENDIX C: FORTRAN LISTING OF SDS 930 UNIFORM GENERATOR	10
REFERENCES	11
BIBLIOGRAPHY	12

TECHNICAL MEMORANDUM X-

UNIFORM RANDOM NUMBER GENERATORS

INTRODUCTION

With the increased use of simulation and Monte Carlo methods in all the various disciplines of engineering and science, the need for sequences of numbers that appear to be drawn from particular probability distributions has become increasingly more important. The production of these sequences of numbers, or so-called "pseudo-random" numbers, must be simple, fast, accurate, and, most importantly, reproducible. The purpose of this paper is to present automated procedures for the production of these pseudo-random numbers consistent with the above criteria. In automating the random number generators, emphasis was placed upon the uniform distribution.

Most computer library subprogram generators are coded in complex machine language and therefore tend to increase, rather than alleviate, any confusion existing on the generation of random variables. Presented herein are Fortran subprograms for the generation of uniform pseudo-random numbers on the unit interval [0, 1). The generators are of the mixed-multiplicative congruential type, and the method is that of Marsaglia and Bray [1]. The subprograms described are for the Univac 1108, the CDC 3200, and the SDS 930 digital computers. A method for generating normal random variables from uniform variables is also included.

THEORETICAL CONSIDERATIONS

There are many methods for generating uniform random variables, each having inherent advantages and disadvantages depending upon its utilization [2]. The mixed-multiplicative congruential generator is discussed herein. Because of varied opinions concerning the appropriateness of this type of generator for certain Monte Carlo applications [3-5], the validity of these particular applications will not be addressed. This discussion will consist of the basic theory of this type of generator and how it may be used conveniently via Fortran subprograms by the method of Marsaglia and Bray [1].

The congruential method is basically an arithmetric recurrence relation involving integers in which each new number is generated from the previous number by some deterministic approach. The recurrence relation is initiated by some initial value and, at some subsequent point, will redevelop forming a closed loop. The length of this closed loop is the period of the generator and, hopefully, is nearly equal to the total integer population of the machine, which is designated by m.

The deterministic approach of the congruential generators is the relation,

$$X_{i+1} = aX_i + c \bmod (m) \quad (0 \leq X_i < m) ,$$

which means that the expression $aX_i + c$ is to be divided by m and X_{i+1} is set equal to the remainder. To illustrate this, let m (modulus) = 25, a (multiplier) = 7, and c = 1, and let $X_0 = 3$ be the initial value for

$$X_1 = 7 \times 3 + 1 \bmod (25) \qquad X_1 = 22 ,$$

$$X_2 = 7 \times 22 + 1 \bmod (25) \qquad X_2 = 5 ,$$

$$X_3 = 7 \times 5 + 1 \bmod (25) \qquad X_3 = 11 , \text{ etc.}$$

Numbers on [0, 1) can be obtained by dividing by m. The method usually is called multiplicative if $c = 0$ and mixed if $c \neq 0$. The modulus m is normally taken as 2^n for an n-bit binary machine and 10^n for an n-digit decimal machine. The constants a and c are chosen to provide speed, a long period, and good statistical results [6].

A basic problem inherent in congruence method generators is that choice of X_0, a, and c which will insure a maximum period. The following theorem presented by Hull and Dorbell [2] solves this problem.

Theorem: The sequence defined by the congruence relation

$$X_{i+1} = aX_i + c \bmod (m)$$

has full period m, provided

1. c is relatively prime to m
2. $a = 1 \bmod (p)$ if p is a prime factor of m,
3. $a = \bmod (4)$ if 4 is a factor of m.

Thus, if m is a power of 2, as on a binary machine, we need only to have c an odd number and $a = 1 \bmod (4)$. The proof of this theorem is included in Reference [2].

The most favorable aspect of the congruence generators is the characteristic that a sequence of random digits may be reproduced by simply starting the generator with the same initial value. In the paper by Stockmal [7], algorithms are presented that evaluate

$$X_i = f(i)$$

or

$$i = f^{-1}(X_i) ,$$

where X_i is the ith element of a congruential random number sequence. These algorithms can be very useful when only certain parts of a Monte Carlo simulation need to be repeated and the generation of the entire sequence of random variables is not required.

Congruence generators have been widely used, and the results have been favorable. An extensive list of references may be found in Reference 2.

FORTRAN CONGRUENT GENERATORS

Marsaglia and Bray [1] developed a method of generating uniform random variables by the congruence method utilizing a single Fortran statement. They also describe Fortran programs that mix several such generators.

Initially, consider the handling of integers by the SDS 930 digital computer. Here, Fortran integers are stored in 24 bits, and the multiplication of 2 integers produces a 24-bit integer mod, 2^{24}. When used in algebraic expressions, the sign on an integer I is determined by the relation

$$m(I) = \begin{cases} I & \text{if } 0 \le I < 2^{23} \\ -2^{24} + I & \text{if } 2^{23} \le I \le 2^{24} - 1 \end{cases},$$

which has a range from -2^{23} to $2^{23} - 1$. We may therefore use the single Fortran statement I = I*A for each random integer on $-2^{23} < I < 2^{23} - 1$ to produce a new random integer thus giving the congruence relation $X_{i+1} = a X_i \mod (2^{24})$. Finally dividing by 2^{24} and adding 1/2 will produce a uniform variate on [0, 1),

$$U = 1/2 + I/2^{24} .$$

The process is more complicated in the Univac 1108 digital computer where 1's complement arithmetic is used. Here, multiplication of two 36-bit integers yields the product mod (2^{36}) but the sign is handled by

$$m(I) = \begin{cases} I & \text{if } 0 \le I < 2^{35} \\ -2^{36} + I + 1 & \text{if } I \ge 2^{35} \end{cases} .$$

Therefore, steps must be taken to subtract the integer 1 at certain times to represent the proper remainders of 2^{36}. The following instruction, which is a correction by Grosenbaugh [8] to Marsaglia and Bray's original Fortran instruction, handles this requirement satisfactorily:

```
I = I * K + MINO ( 0, ISIGN (K-1, L) )
```

The two functions MINO and ISIGN are standard Fortran library routines. MINO determines the minimum value of a series of integer quantities, and ISIGN transfers the sign of the second argument to the absolute value of the first argument. The constant K can be chosen for maximum period as shown by Van Gelder [6].

The CDC 3200 digital computer handles integers exactly as does the Univac 1108. However, the CDC 3200 stores integers as 24 bits, therefore the only change required is the power of 2 in all equations.

These one-line Fortran generators may be incorporated directly into programs easily as they are, but Marsaglia and Bray obtained better statistical results upon combining several generators. As an example, consider the following SDS 930 generator:

```
L    =  L * ML
M    =  M * MM
J    =  1 + IABS (L)/2**16
U    =  1/2 + [N(J) + L + M]/2**24
K    =  K * MK
N(J) =  K
```

The array N is a 128-element array filled with random numbers previously assigned by a one-line generator. In the procedure, J is used to choose from the N array; J comes from the random integer L after division by the appropriate power of 2. The desired random variable U is formed from the

sum of the randomly chosen N array element, the random integer L used to find J, and a third additional random integer M. The used element of N is replaced by the random integer K. Similar procedures are used for the Univac 1108 and CDC 3200 generators.

Subprogram listings are given for the Univac 1108, CDC 3200, and SDS 930 generators in Appendices A, B, and C, respectively. All three subprograms are called in the exact same manner, so that a computer program may be converted to run on several machines by simply inserting the appropriate generator. Initially, the subprograms must be given a non-zero integer to establish the internal 128-cell array. An odd number in the millions gives excellent statistical results. The initial call to a generator does not produce a useful result. Uniform random variables on the unit interval [0, 1) may be obtained by calling the generator with a zero integer argument. To repeat a sequence of variables, the generator is simply reinitialized with the same initial value.

GENERATION OF NORMAL VARIABLES

As in the case of the uniform pseudo-random number generator, there are a variety of ways to generate normally distributed random variables. A method of normal variable generation by Box and Muller [9, 10] is discussed here. Their method is simple, fast, and requires very little memory storage.

The following method is used to generate a pair of random deviates (X_1, X_2) from the same normal distribution starting from a pair of uniformly distributed random variables (U_1, U_2) distributed [0, 1).

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

$$X_2 = (-2 \ell n\, U_1)^{\frac{1}{2}} \sin (2 \pi U_2) \quad .$$

The pair (X_1, X_2) will be normally distributed [0, 1] and are very reliable in the tails of the distribution. The estimated speed is 6.01 milliseconds per variable on a CDC 3200 digital computer.

CONCLUSION

Methods have been shown and subprograms have been developed for the fast, simple, and reproducible generation of pseudo-random numbers for uniform and normal probability distributions. The accuracy of the numbers must not be assumed to be 100 percent for all utilizations, because potential error sources do exist. Marsaglia [5] presents results that indicate that every multiplicative generator has a defect that makes it unsuitable for certain Monte Carlo applications. Other remote error possibilities are stated in the references. However, for most applications the generators presented herein give accurate results.

APPENDIX A

FORTRAN LISTING OF UNIVAC 1108 UNIFORM GENERATOR

```
      FUNCTION RAN(JJ)                                                  RAN  10
C                                                                       RAN  20
C       UNIFORM RANDOM NUMBER GENERATOR FOR THE UNIVAC 1108             RAN  30
C                                                                       RAN  40
C       FIRST CALL MUST BE OF THE FORM   X = RAN(J),  WHERE J IS AN     RAN  50
C       INITIAL INTEGER VALUE.  SUBSEQUENT CALLS MUST BE OF THE FORM    RAN  60
C       X = RAN(0).                                                     RAN  70
C                                                                       RAN  80
      DIMENSION N(128)                                                  RAN  90
      J=JJ                                                              RAN 100
      IF (J.EQ.0) GO TO 2                                               RAN 110
      DO 1 I=1,128                                                      RAN 120
      J=J*227157+MIN0(0,ISIGN(227156,J))                                RAN 130
    1 N(I)=J                                                            RAN 140
      L=JJ                                                              RAN 150
      M=JJ                                                              RAN 160
      K=JJ                                                              RAN 170
    2 L=L*274693+MIN0(0,ISIGN(274692,L))                                RAN 180
      M=M*243133+MIN0(0,ISIGN(243132,M))                                RAN 190
      I=IABS(L)/268435456+1                                             RAN 200
      RAN=.5+FLOAT(N(I)+L+M)*,145519152E-10                             RAN 210
      K=K*249149+MIN0(0,ISIGN(249148,K))                                RAN 220
      N(I)=K                                                            RAN 230
      RETURN                                                            RAN 240
      END                                                               RAN 250-
```

APPENDIX B

FORTRAN LISTING OF CDC 3200 UNIFORM GENERATOR

```
      FUNCTION RAN(JJ)                                                  RAN  10
C                                                                       RAN  20
C        UNIFORM RANDOM NUMBER GENERATOR FOR THE CDC 3200               RAN  30
C                                                                       RAN  40
C        FIRST CALL MUST BE OF THE FORM   X = RAN(J),  WHERE J IS AN    RAN  50
C        INITIAL INTEGER VALUE.   SUBSEQUENT CALLS MUST BE OF THE FORM  RAN  60
C        X = RAN(0).                                                    RAN  70
C                                                                       RAN  80
      DIMENSION N(128)                                                  RAN  90
      J=JJ                                                              RAN 100
      IF (J.EQ.0) 3,1                                                   RAN 110
    1 DO 2 I=1,128                                                      RAN 120
      J=J*227157+MIN0(0,ISIGN(227156,J))                                RAN 130
    2 N(I)=J                                                            RAN 140
      K=JJ                                                              RAN 150
      M=JJ                                                              RAN 160
      L=JJ                                                              RAN 170
    3 L=L*274693+MIN0(0,ISIGN(274692,L))                                RAN 180
      M=M*243133+MIN0(0,ISIGN(243132,M))                                RAN 190
      I=IABS(L)/65536+1                                                 RAN 200
      RAN=.5+FLOAT(N(I)+L+M)*.596046447E-07                             RAN 210
      K=K*249149+MIN0(0,ISIGN(249148,K))                                RAN 220
      N(I)=K                                                            RAN 230
      RETURN                                                            RAN 240
      END                                                               RAN 250-
```

APPENDIX C

FORTRAN LISTING OF SDS 930 UNIFORM GENERATOR

```
      FUNCTION RAN(JJ)                                                    RAN  10
C                                                                         RAN  20
C        UNIFORM RANDOM NUMBER GENERATOR FOR THE SDS 930                  RAN  30
C                                                                         RAN  40
C        FIRST CALL MUST BE OF THE FORM    X = RAN(J).  WHERE J IS AN     RAN  50
C        INITIAL INTEGER VALUE.    SUBSEQUENT CALLS MUST BE OF THE FORM   RAN  60
C        X = RAN(0).                                                      RAN  70
C                                                                         RAN  80
      DIMENSION N(128)                                                    RAN  90
      J=JJ                                                                RAN 100
      IF (J) 1,3,1                                                        RAN 110
    1 DO 2 I=1,128,1                                                      RAN 120
      J=J*65539                                                           RAN 130
    2 N(I)=J                                                              RAN 140
      L=JJ                                                                RAN 150
      M=JJ                                                                RAN 160
      K=JJ                                                                RAN 170
    3 L=L*4357                                                            RAN 180
      M=M*9197                                                            RAN 190
      I=1+IABS(L)/65536                                                   RAN 200
      RAN=0.5+FLOAT(N(I)+L+M)*0.59604644E-07                              RAN 210
      K=K*10757                                                           RAN 220
      N(I)=K                                                              RAN 230
      RETURN                                                              RAN 240
      END                                                                 RAN 250
```

REFERENCES

1. Marsaglia, George; and Bray, T. A.: One-Line Random Number Generators and Their Use in Combinations. Comm. ACM, vol. 11, no. 11, 1968, pp. 749-751.

2. Hull, T. E.; and Dorbell, A. R.: Random Number Generators. SIAM Review, vol. 4, no. 3, 1962, pp. 230-248.

3. Coveyou, R. R.; and MacPherson, R. D.: Fourier Analysis of Uniform Random Number Generators. J. ACM, vol. 14, no. 1, 1967, pp. 100-119.

4. Greenberger, Martin: Method in Randomness. Comm. ACM, vol. 8, no. 3, 1965, pp. 177-179.

5. Marsaglia, G.: Random Numbers Fall Mainly in the Planes. Proc. Natl. Acad. Sci, vol. 60, no. 5, 1968.

6. Van Gelder, A.: Some New Results in Pseudo-Random Number Generation. J. ACM, vol. 14, no. 4, 1967, pp. 785-792.

7. Stockmal, F.: Calculations with Pseudo-Random Numbers. J. ACM, vol. 11, 1964, pp. 41-52.

8. Grosenbaugh, L. R.: More on Fortran Random Number Generators. Comm. ACM, vol. 12, no. 11, 1969, p. 639.

9. Box, G. E. P.; and Muller, Mervin E.: A Note on the Generation of Random Normal Deviates. Annals of Mathematical Statistics, vol. 29, 1958, pp. 610-611.

10. Muller, Mervin E.: A Comparison of Methods for Generating Normal Deviates on Digital Computers. J. ACM, vol. 6, 1959, pp. 376-383.

BIBLIOGRAPHY

Chambers, R. P.: Random-Number Generation on Digital Computers. IEEE Spectrum, vol. 4, no. 1, 1967, pp. 48-56.

Kronmal, Richard: Evaluation of a Pseudorandom Normal Number Generator. J. ACM, vol. 11, no. 3, 1964, pp. 357-363.

Marsaglia, G.: Expressing a Random Variable in Terms of Uniform Random Variables. Annals of Mathematical Statistics, vol. 32, no. 3, 1961, pp. 894-898.

Marsaglia, G.; and Bray, T. A.: A Convient Method for Generating Normal Variables. SIAM Review, vol. 6, no. 3, 1964, pp. 260-264.

Meyers, Paul L.: Introductory Probability and Statistical Applications. Addison-Wesley Publishing Company, Reading, Mass., 1970, pp. 250-255.

APPROVAL

UNIFORM RANDOM NUMBER GENERATORS

By William R. Farr

The information in this report has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

This document has also been reviewed and approved for technical accuracy.

Robert M. Croft

ROBERT M. CROFT
Acting Chief, Flight Mechanics Branch

William D. Goldsby, Jr.

WILLIAM D. GOLDSBY, JR.
Deputy Chief, Flight Performance and Mission Analysis Division

Erich E. Goerner

ERICH E. GOERNER
Director, Preliminary Design Office

James T. Murphy

JAMES T. MURPHY
Acting Director, Program Development

MSFC—RSA, Ala

Scientific and Technical Information
Facility (25)
P. O. Box 33
College Park, Maryland 20740
Attn: NASA Representative (S-AK/RKT)

DISTRIBUTION

INTERNAL

DIR
Dr. Rees

DEP-T
Dr. Lucas

AD-S
Dr. Stuhlinger

PD-DIR
Mr. Murphy

PD-SA-DIR
Mr. Huber

PD-SA-V
Mr. Orillion

PD-DO-DIR
Mr. Goerner
Dr. Thomason
Mr. Heyer
Mrs. Andrews

PD-DO-E
Mr. Digesu

PD-DO-ES
Mr. Cole

PD-DO-S
Mr. Marshall

PD-DO-SI
Mr. Butler

PD-DO-P
Mr. Goldsby

PD-DO-PF
Mr. Croft
Mr. Sumrall
Mr. Lowery
Mr. Perkinson
Mr. Solmon
Mr. Farr (10)

PD-DO-PM
Mr. Leonard
Mr. Gold
Mr. Wheeler
Mr. Bradford
Mr. Young

PD-DO-PA
Mr. Allen
Mr. Jump
Mr. Burton
Mr. French
Mr. Zimmerman
Mr. Sullivan
Mr. Sparks

PM-PR-M

S&E-AERO-DIR
Dr. Geissler

S&E-AERO-D
Mr. E. Deaton

S&E-AERO-F
Mr. Horn
Mr. Mabry

S&E-AERO-Y
Mr. Vaughn

S&E-AERO-YA
Mr. Kaufman

S&E-COMP-SD
Mr. Zeanah

A&TS-MS-H

A&TS-MS-IL
Miss Robertson (8)

A&TS-MS-IP
Mr. Ziak (2)

A&TS-PAT
Mr. L. D. Wofford, Jr.

A&TS-TU (6)

EXTERNAL

Scientific and Technical Information
Facility (25)
P. O. Box 33
College Park, Maryland 20740
Attn: NASA Representative (S-AK/RKT)

NASA-Manned Spacecraft Center
Houston, Texas 77058
Attn: Library (5)

Northrop Space Laboratories
Technology Drive
Huntsville, Alabama 35805
Attn: Library (5)

Lockheed Missiles and Space Company
Technology Drive
Huntsville, Alabama 35805
Attn: Library (5)

NASA Headquarters
Washington, D. C. 20546
Attn: Library (5)

NASA-Ames Research Center
Mountain View, California 94035
Attn: Library (5)

NASA - Langley Research Center
Hampton, Virginia 23365
Attn: Library (5)

Jet Propulsion Laboratory
Pasadena, California 91103
Attn: Library (5)

Tech Information Division (5)
NASA-Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

Computer Sciences Corporation
S&E-COMP
Attn: Mr. Ron Hall
Attn: Mr. Emmett Jones