AD-A060 201 NAVY UNDERWATER SOUND LAB NEW LONDON CONN F/G 9/2

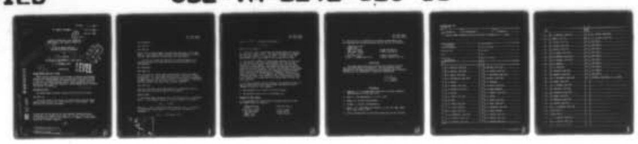
A RANDOM NUMBER GENERATOR FOR USF IN FORTRAN V PROGRAMS.(U)

SEP 68 R L GORDON

UNCLASSIFIED USL-TM-2242-320-68 NL

| OF |

AD A060 201



END

DATE FILMED

-12-78

DDC

U. S. NAVY UNDERWATER SOUND LABORATORY
FORT TRUMBULL, NEW LONDON, CONNECTICUT

A RANDOM NUMBER GENERATOR FOR USE IN FORTRAN V PROGRAMS*

by

R. L. Gordon

USL Technical Memorandum No. 2242-320-68

6 September 1968

INTRODUCTION

Random Number Generators (RNG)

Five external subroutines are available to generate uncorrelated, floating point random numbers which are uniformly distributed between 0.0 and 1.0. The subroutines are extremely fast (8 μs per word) and do not need an external starting number. Additional entry points are provided so that a generated sequence of random numbers may be continued from program to program.

Procedure for RNG

The Random Number Generator subroutines are called as follows:

CALL GAS1 (R)

This will result in the return of a floating point decimal random number between 0.0 and 1.0 to core location R. (See example). Each new call will provide a new number.

*A large part of the material for the assembly language generators comes from a Yale Computer Center Memo. No. 27G by J. Lach, and from an IEEE Spectrum Article in Feb. 1967 by R. P. Chambers, "Random Number Generation in Digital Computers".

The statements:

CALL GAS2 (R)

CALL GAS5 (R)

may be used to extract numbers from the other generators in the same fashion. Each generator has a different starting number and constant with the result that all five generators are uncorrelated.

In order to continue the sequence in an unbroken fashion in a subsequent program, two additional entry points are supplied with each generator.

The statements:

CALL SGAS1 (R)

will place the last random number generated as a fixed point number in core location R. This number may be printed or punched to restart the generator. To insure that all significant portions are transferred, a FORMAT designation of O12 should be used. If the number is read into the next program in the same format, then the statement:

CALL LGAS1 (R)

will store the number in the GAS1 generator so subsequent calls on GAS1 will keep the sequence unbroken (see example).

Special Notes:

1. The random number generators are of the multiplicative, congruential type.[1,5] This technique employs a recursion relation of the form:

$$R_{n+1} = F\ (R_n)$$

where R_{n+1} is the $(n+1)^{th}$ random number generated and F is a function of the previous random number, Rn. In particular, the function programmed is

$$R_{n+1} = C\ Rn\ (\text{Modulo } 2^{35})$$

where $C \approx \sqrt{2^{35}}$ and obeys the condition

$$C = 8m \pm 3$$

where m is an integer.[1,2]

2. It should be kept in mind that true random numbers can only be generated by a random process. The numbers generated by a computer are only approximations to such a process. The hope is that the approximations are close enough for the calculations at hand.

3. Two important tests[1,5] for a random number generator are the tests for uniformity of the distribution between 0 and 1 and for the correlation between samples. The generators used here have been tested for correlation[3] with 1000 samples and have been within the expected statistical accuracy. The test for uniformity has been made indirectly by a chi-square test for normality[4] on samples produced by this generator but transformed by a "Statpack"[6] routine, to a normal distribution. For 500 samples the probability of exceeding chi-square was 2% at a 95% confidence level. The generator included in the "Math-Pack"[6] library could not do better than 50% under the same conditions.

4. In order to obtain normal distributions this generator should be used in conjunction with an inverse normal distribution such as "Statpack"[6] provides. (See example).

5. Each generator uses a total of 19 core locations.

6. The generators are in File 3 of CUR Tape U183.

Examples of RNG Program

A. The generation of 100 uniformly distributed numbers with the last number saved for Program B.

```
    DIMENSION X(100)              @ Declare Array X
    DO 10I  =  1, 100
 10 CALL GAS1 (X(I))              @ Fill X Array
    CALL SGAS1 (FIN)              @ Save last No.
    PUNCH 1, FIN                  @ Punch last No.
  1 FORMAT (Ø12)
```

B. The generation of 100normally distributed random numbers with zero mean and unit variance starting from the last no. of Program A.

```
      DIMENSION X (100)
    1 FORMAT (Ø12)
      READ (3,1) START                @ Read starting No.
      CALL LGAS1 (START)              @ Enter starting No.
      DO 10I  =  1, 100
    2 CALL GAS1 (X(I))                @ Obtain uniform Nos.
   10 X(I)  =  TINORM6 (X(I), $2)     @ Obtain normally dist. No.
```

Conclusions

The random number generators presented are extremely fast and have excellent statistical properties. By using several of the generators in the same program even better properties may be obtained. The main use will probably be in Monte Carlo Simulations.

R. L. Gordon
R. L. GORDON

References

1. Chambers, R. P., "Random-Number Generation on Digital Computers", IEEE Spectrum, Feb. 1967, pp. 48-56.

2. Lach, J., "YCC Memorandum 2 G", Oct 1, 1964.

3. Newman, H., private correspondence.

4. Foreman, L., private correspondence.

5. Bauer, W. F., "The Monte Carlo Method", J. Soc. Ind. Appl. Math., Vol. 6, pp. 438-451, 1958.

6. Univac 1108 Math-Pack and Stat-Pack Subroutines UP-7542, UP-7502.

DISTRIBUTION LIST

NND-USNUSL-142

DATE	FILE	SERIAL
2 October 68	9-A-415-00-00/175	2242320-68

SUBJECT

A RANDOM NUMBER GENERATOR FOR USE IN FORTRAN V PROGRAMS by R. L. Gordon

TECHNICAL MEMORANDUM 2242-320-68	EVALUATION REPORT
CALIBRATION MEMORANDUM	SERVICE TEST REPORT
CALIBRATION REPORT	TEST REPORT
LETTER	OTHER
CLASSIFICATION UNCLASSIFIED	NUMBER OF COPIES MADE 71

COPY NO.		COPY NO.	
1	A. J. Alfiero, Code 2072	17	A. Nuttall, Code 2020
2	R. J. Almeida, Code 2072	18	R. M. Forget, Code 2072
3	C. R. Arnold, Code 2070	19	M. J. Goldstein, Code 2072
4	J. W. Askintowicz, Code 2072	20	S. Horvitz, Code 2072
5	J. A. Auwood	21	R. C. Jennings
6	M. D. Becker, Code 2072	22	J. F. Kelly, Code 2072
7	P. A. Breslin, Code 2072	23	J. R. Kelly (UNIVAC)
8	R. A. Brewer	24	R. J. Lambird, Code 2005
9	W. E. Cable, Code 2071	25	A. A. Lesick
10	D. L. Card	26	H. E Lundy, Code 2072
11	R. D. Clark, Code 2072	27	L. P. Onyx, Code 2072
12	W. R. Cleary (UNIVAC)	28	E. M. Podeszwa, Code 2040
13	J. S. Cohen, Code 2072	29	E. F. Poole (UNIVAC)
14	D. W. Counsellor, Code 2072	30	D. T. Porter, Code 2221
15	R. D. Drinkard, Code 2072	31	M. R. Powers, Code 2072
16	L. T. Einstein, Code 2071	32	D. L. Rawson, Code 2072

REMARKS

Copy. No.		Copy No.	
33	F. E. Robbins, Code 2072	59	R.L. Gordon, Code 2242
34	D. Hyde, Code 2020	60	R. L. Gordon, Code 2242
35	N. Owsley, Code 2242	61	Code 2064.2
36	N. Owsley, Code 2242	62	Code 2064.2
37	N. Owsley, Code 2242	63	Code 2064.2
38	G. R. Swope, Code 2072	64	Code 2064.2
39	M. A. Tucchio, Code 2333	65	Code 2064.2
40	T. T. Tylaska, Code 2072	66	Code 2064.2
41	A. R. Umphrey	67	Code 2064.2
42	J. F. Underwood	68	Code 2064.2
43	H. Weinberg, Code 2071	69	Code 2064.2
44	T. Wheeler, Code 2072	70	Code 2064.2
45	R. D. Whittaker, Code 2072	71	NavShips, Code 00V1C - J. D. Pope
46	E. Windt, Code 2072		
47	H. S. Newman, Code 2242		
48	R. Kneipfer, Code 2242		
49	J. Ianniello, Code 2213		
50	A. Quazi, Code 2242		
51	T. McAndres, Code 2241		
52	A. Krajec, Code 2241		
53	F. J. Kingsbury, Code 2242		
54	Code 2242-S		
55	Code 2242-S		
56	R. L. Gordon, Code 2242		
57	R. L. Gordon, Code 2242		
58	R. L. Gordon, Code 2242		