A particular pseudo-random number generator is described that uses the full 31-bit capacity of the registers in the IBM SYSTEM/360 *computers.*

Experience with the generator in obtaining random permutations of sequences is discussed, and results of statistical tests applied to evaluate the generator are given. The generator has been found to be highly satisfactory.

An assembler language program of the generator is included.

A pseudo-random number generator for the System/360

by P. A. W. Lewis, A. S. Goodman, and J. M. Miller

The purpose of this paper is to describe a pseudo-random number generator that uses the full capacity of the 32-bit registers of IBM SYSTEM/360 computers, and to briefly report on and document the testing of and experience with the random number generator. The generator is a particular case of the sequence of numbers x_i generated by the equation

$$x_{i+1} \equiv Ax_i \pmod{p} \tag{1}$$

where p is a prime and A is a positive primitive root of p.

The generator was described by Hutchinson[1] and ascribed to Professor D. H. Lehmer. Hutchinson discussed a particular form of the generator for the IBM 7094, in which $p = 2^{35} - 31$ is the largest prime less than 2^{35} and $A = 5^5$. Unfortunately, his tests on this generator were not published; our own tests and use of the generator confirmed that it is an exceptionally good pseudo-random number generator.

Another and apparently independent description of the generator is given by Downham and Roberts,[2] who give a succinct description of the number theoretic concepts involved. They considered generators with relatively small values of p and came to the empirical conclusion that the positive primitive root A should be approximately $(p)^{1/2}$ to obtain good results. They used mainly serial correlation and runs tests, although these tests were not always properly applied. They concluded that a runs test, ascribed to Herman,[3] is the most sensitive test for randomness. The runs

test is, in fact, very old, and is known empirically to be powerful against a broad class of alternatives. A discussion of the runs test is given by Kendall and Stuart.[4]

Downham and Roberts' results on the relative size of A conform generally to the predictions of Coveyou and MacPherson,[5] who have given one of the few analytical evaluations of pseudo-random number generators in the literature (see also Zaremba[6] and Marsaglia[7]). Coveyou and MacPherson also concluded that it might be difficult or impossible to find "good" generators for the single-precision 32-bit word size of the SYSTEM/360. This theme was taken up by Whittlesey,[8] who discussed the generator of Equation 1 and gave additional references. However, Whittlesey apparently failed to realize that A should be a positive primitive root of p, if for no other reason than to obtain the full cycle of length p in the generator. Moreover, Whittlesey's reservations about generators of the form of Equation 1 for the SYSTEM/360 were based on a serial correlation test that, relative to other tests, has a rather dubious distribution theory. In addition this test, while it can detect gross departures, is probably not sensitive to small departures from randomness.

Our own experience with the generator of Equation 1 on the IBM 7094 led us to look for a similar generator for the SYSTEM/360. There are 31 bits available for computation in the 32-bit general register of the SYSTEM/360 (one bit is a sign bit), and the largest prime p less than or equal to $2^{31} - 1$ is very conveniently $2^{31} - 1$ itself. The factorization of $p - 1$ is

$$p - 1 = 2^{31} - 2 = 2 \times 3^2 \times 7 \times 11 \times 31 \times 151 \times 331$$

and $A = 7$ is a positive primitive root of $2^{31} - 1$. Any power (modulo p) of 7, say 7^k, where k is not a factor of $p - 1$, is also a positive primitive root of p. In this way, many A's could be generated, and it was confirmed empirically that an A approximately equal to $(p)^{1/2}$ is required to even begin to give good test results. Note that this is not an inclusive statement to the effect that all A's in the neighborhood of $(p)^{1/2}$ give good generators. In fact if a positive primitive root exists that is almost exactly equal to $(p)^{1/2}$, we can expect strong serial correlation of order zero in the generator.

The particular generator described in this paper uses $p = 2^{31} - 1$ and $A = 7^5 = 16807$. We describe here the tests used in evaluating this particular case of Equation 1 and give the test results obtained for a number of reasons:

- A well-tested and experientially acceptable generator for the SYSTEM/360 is not available.
- We have found experimentally that Coveyou and MacPherson's predictions are valid and that a pseudo-random number generator for a 31-bit machine has to be chosen carefully. In particular only two values of A of the many investigated gave test results as good as those obtained for $A = 7^5$.

- It has been our experience that many generators put forward without documentation turn out to be defective. This is because inadequate test statistics have been used, or because the test statistics have been misused, or because the series tested have not been long enough to detect subtle departures from randomness.
- Some of the tests described and advocated here use recent advances in statistical knowledge and are not well known to nonstatisticians. In fact these tests, and in particular those based on Fourier transforms, are of the type that Whittlesey[8] describes as not having yet been applied to evaluating the generator of Equation 1.

Random permutations

Our concern to obtain a "good" and relatively fast pseudo-random number generator arose because we were conducting large-scale synthetic sampling experiments on tests for serial dependence in time series. Let the observed series be $x_1, \cdots, x_i, \cdots, x_n$, and the rank of x_i be denoted by r_i. Let $a(r_i)$ be a monotone function of r_i, either the ranks themselves or scores (expected values of order statistics from some population). A test for lack of serial dependence in the time series can be based on the idea that under the null hypothesis all $n!$ orderings of the $a(r_i)$'s are equally likely. A commonly used test statistic for testing serial independence is the score product-moment statistic of lag one,

$$R(1) = a(r_1)a(r_2) + \cdots + a(r_{(n-1)})a(r_n)$$

or in the case of ranks

$$R(1) = r_1r_2 + r_2r_3 + \cdots + r_{n-1}r_n$$

There are $n!$ possible orderings of the ranks, and it is possible to compute the exact distribution of $R(1)$ under the null hypothesis in a reasonable amount of time on a SYSTEM/360 Model 91 computer only up to $n = 11$. Beyond that, synthetic sampling has to be used.

Our procedure for testing the generator for this particular purpose was to work out the exact distributions of the rank product-moment statistic for $n = 10$ and for lags 1, 2, and 3 and then compare these distributions with the estimated distributions obtained by generating random permutations of the numbers 1 to 10. A standard method[9] is used to generate random permutations (see also Reference 10). Up to 14,000,000 random permutations were generated and chi-square goodness-of-fit tests showed no discrepancy between exact and estimated distributions.

Generating permutations of the numbers 1 through 10 uses and tests only the first four bits of the random numbers. The remaining bits can, however, also be used. Some further experience is also relevant. The score product-moment statistics are known to have a normal distribution for large n under very weak conditions

on the numbers $a(r_i)$.[11] For normal scores the normal distribution was found to hold for $n = 50$ and for ranks for $n = 75$. Exponential scores,[12] however, produced a highly skewed distribution, which had not converged at $n = 9{,}000$. This immediately raised doubts about the random number generator; as a check, the distribution of the rank product-moment statistic was computed at $n = 9{,}000$. The latter distribution was found to be still normally distributed, indicating that the random number generator and permutation generating scheme held up at $n = 9{,}000$. Subsequent experience showed that the rate of convergence of the distribution of the score product-moment statistics to the asymptotic form depends critically on the skewness of the parent population of the scores.

Tests for randomness

Since the "random" numbers generated are specified to be uniformly distributed, as well as serially independent, it is necessary to test before anything else for a uniform one-dimensional marginal distribution. Provided sufficient divisions of the unit interval are used, a chi-square test of goodness-of-fit is adequate for this purpose. For very long series, the chi-square test is computationally much simpler than tests such as the Kolmogorov-Smirnov test.

Tests for serial independence in the random numbers are conveniently broken up into two types: direct tests on the raw data for the absence of serial correlation or "bunching," and tests on the Fourier-transformed data for a flat spectrum.

direct tests

For the direct tests we follow in part the discussion in MacLaren and Marsaglia,[13] relating their tests to their statistical antecedents. In essence, these tests of serial correlation or bunching check for uniformity of successive lagged pairs of random numbers. Thus, let ν be a power of two and determine the number of pairs $(x_i, x_{i+\ell})i = 1, 2, \cdots, N + \ell - 1$, for which the first $\log \nu$ bits of x_i had the value m and the first $\log \nu$ bits of x_{i+1} had the value n. Here N is the basic length of the sequence of random numbers (the additional $\ell - 1$ are used for computational convenience) and n and m run from 0 to $\nu - 1$, giving ν^2 possible pair values with frequency $f_{n,m}$. We compute the chi-square statistic

$$S^2(\ell) = \sum_{m,n,=0}^{\nu-1} \left(f_{n,m} - \frac{N}{\nu^2}\right)^2 \Big/ \left(\frac{N}{\nu^2}\right) \tag{2}$$

This is Good's serial test for sampling numbers.[14,15] Contrary to the usual assumptions, the statistic does not have a chi-square distribution with $\nu^2 - \nu$ degrees of freedom; in fact, its mean value is $\nu^2 - 1$ (see Reference 14). The exact distribution is not known.

Tests of n-tuples of various combinations of lags are possible but have not been applied.[13]

As indicated earlier, a useful test of randomness is the runs test. It is a nonparametric test, testing for serial independence per se. The exact definition of a run (or phase) up or down of

length d is given in Kendall and Stuart.[4] For example, in the sequence

$$5, 4, \underbrace{1, 2, 3, 4, 5}, \underbrace{4, 3, 2}, \underbrace{3, 4, 5, 6, 7, 8, 9}, \underbrace{6, 4}, \underbrace{5}, 3, 2$$

we have 5 runs, the lengths of successive runs being 5, 3, 7, 2, and 1. The expected number of runs in a sequence of length N is, for N large, $(2N - 7)/3$, and the expected number of runs of length d is

$$f(d) = \frac{2\{(N - d - 2)(d^2 + 3d + 1)\}}{(d + 3)!} \qquad d = 1, 2, 3, \cdots \tag{3}$$

Since this quantity decreases rapidly with d, it is usual to lump long runs together; in our case, we lumped together runs of 8 or longer. The expected number of such runs is[4]

$$f(8) = \tfrac{1}{3}(2N - 7) - \sum_{d=1}^{7} f(d) \tag{4}$$

The consistency with the hypothesis of serial independence is tested by a chi-square statistic. In a series of length N, let $n(d)$ be the actual observed number of runs of length d if d is less than 8, and the actual number of runs greater than or equal to d if $d = 8$. Let $\Sigma n(d)$ be the total number of runs observed in the sequence of length N. Then, if

$$f'(d) = f(d) \times \frac{\Sigma n(d)}{\frac{1}{3}(2N - 7)} \tag{5}$$

$$\chi^2(7) = \sum_{d=1}^{8} \frac{\{n(d) - f'(d)\}^2}{f'(d)} \tag{6}$$

Note that the statistic has the distribution of the chi-square test statistic only when N is large; even then it does not have a chi-square distribution with 7 degrees of freedom because the variance is inflated by the small cell frequencies for large d.[16]

tests for a flat spectrum

Tests of randomness based on the Fourier-transformed data have become practical with the advent of the fast Fourier transform algorithm;[17] the tests are discussed in References 18 and 19. The tests actually use the periodogram or estimated spectrum as follows:

Take the finite Fourier transform of the sequence x_j, $j = 0, 1, \cdots, N - 1$ to get

$$a_n = \frac{1}{N} \sum_{j=0}^{N-1} x_j e^{-2\pi i j n/N} \qquad n = 0, 1, \cdots, N - 1 \tag{7}$$

and compute the periodogram points

$$p_n = 2N\,|a(n)|^2 \qquad n = 1, 2, \cdots, M + 1 \tag{8}$$

and the (normalized) cumulative periodogram points

$$P_n = \sum_{r=1}^{n} p_r \Big/ \sum_{r=1}^{(N/2)-1} p_r \qquad n = 1, 2, \cdots, M + 1 \tag{9}$$

where $M = (N/2) - 2$ if N is even and $M = (N - 3)/2$ if N is odd.

For normally distributed, independent x_i (and asymptotically under fairly general conditions[20] for nonnormal, independent x_i), the p_n's are independent exponentially distributed variates and the P_n's are the order statistics from a uniform distribution. Thus the test for randomness has been transformed into a test for a Poisson process.[18] The alternatives are basically trends (nonflat spectra), and suitable test statistics should be fairly insensitive to small serial correlation and nonexponentiality in the p_n's. Following Cox and Lewis[18] and Durbin,[19] the test statistics used are the following:

- The median-spectrum test statistic

$$S = \frac{1}{M} \sum_{n=0}^{M} P_n \tag{10}$$

The normalized test statistic $U = [S - (1/2)](12M)^{1/2}$ has, under the null hypothesis, a unit normal distribution even for small M.

- The one-sided and two-sided modified Kolmogorov-Smirnov test statistics

$$\mathrm{KS}+ = \max_{1 \le n \le M} \left\{ P_n - \frac{n}{M+1} \right\} \tag{11}$$

$$\mathrm{KS}- = \min_{1 \le n \le M} \left\{ \frac{n}{M+1} - P_n \right\} \tag{12}$$

$$\mathrm{KS} = \max \{\mathrm{KS}+, \mathrm{KS}-\} \tag{13}$$

The asymptotic distributions of these statistics are given by Durbin.[21]

- A modification of Bartlett's test for variance heterogeneity can be used to test for a constant spectrum value.[22] Thus, we divide the M values of $p_n (n = 1, 2, \cdots, M)$ into k contiguous groups of size ν, where k is the largest integer such that $\ell = k\nu \le M$. Denote the sum of the p_n's in each group, divided by 2π, as

$$s_i^2 = \sum_{n=(i-1)\nu+1}^{i\nu} p_n/2\pi \qquad i = 1, \cdots, k \tag{14}$$

The test statistic is then

$$H(k) = \frac{\left\{ 2\ell \log \left[\frac{\sum_{i=1}^{k} s_i^2}{2\ell} \right] - \sum_{i=1}^{k} 2\nu \log \left(\frac{s_i^2}{2\nu} \right) \right\}}{(6\nu - 2)/(6\nu - 3)} \tag{15}$$

and has approximately a chi-squared distribution with $k - 1$ degrees of freedom if 2ν is greater than 5. The hypothesis of randomness is rejected for large values of $H(k)$.

A problem in using this test statistic is the arbitrary choice of k. Another point is that since the p_n's are approximately exponentially distributed, a k-sample Savage test statistic[23] may be more appropriate.

Tests results

To test the pseudo-random number generator of Equation 1 we generated 10 successive groups of numbers. The length of each sequence was $2^{16} + 5$, the five additional values being used in Good's serial tests for computational convenience. The value $N = 2^{16}$ was felt to be large enough for the power of the tests to be high enough to detect the presence of subtle departures from randomness. The use of 10 sequences enhances the reliability of the test results, particularly from a computational viewpoint. Thus, on an initial run, all values of $H(10)$ were found to be within the acceptance level of the test, but all values were below the median value of the test statistic and with rather small variability in the sample of size 10. Further checking indicated the necessity of doing the computations of the $H(k)$'s in double precision. The computations were performed on SYSTEM/360 Models 67 and 91 computers.

Table 1 gives the results of the direct tests of randomness, column two giving the initial value x_0 for each sequence of pseudo-random numbers. For the test of uniformity of the marginal distribution, $2^{12} = 4096$ cells were used, thus giving a test of the first 12 bits of the x_i's. The value of the chi-square goodness-of-fit

Table 1 Direct tests on the pseudo-random numbers

Run number	x_0	*Uniformity* $\chi^2(4095)$	*Goods' serial test* $S(1)$	$S(2)$	$S(3)$	$S(4)$	$S(5)$	$S(6)$	*Runs test* $\chi^2(7)$
1	12345678	4015.25 (−79.75)	263.67 (+8.67)	223.69 (−31.31)	266.62 (+11.62)	258.48 (+3.48)	269.05 (+14.05)	228.02 (−26.98)	16.18 (+9.18)
2	855998726	4112.12 (+17.12)	280.22 (+25.22)	267.45 (+12.45)	240.77 (−14.23)	254.55 (−0.45)	280.17 (+25.17)	222.44 (−22.56)	7.07 (+0.07)
3	745681489	4125.12 (+30.12)	253.34 (−1.66)	237.19 (−17.81)	227.50 (−27.50)	221.05 (−33.95)	238.95 (−16.05)	219.27 (−35.73)	12.15 (+5.15)
4	506104362	4113.50 (+18.50)	246.00 (−9.00)	235.98 (−19.02)	258.31 (+3.31)	258.02 (+3.02)	252.77 (−2.23)	251.91 (−3.09)	4.03 (−2.97)
5	236686234	4150.75 (+55.75)	246.39 (−8.61)	258.58 (+3.58)	221.28 (−33.72)	275.98 (+20.98)	286.08 (+31.08)	239.23 (−15.77)	12.10 (+5.10)
6	1912615462	4079.87 (−15.13)	241.45 (−13.55)	254.22 (−0.78)	289.19 (+34.19)	265.22 (+10.22)	266.31 (+11.31)	236.34 (−18.66)	5.39 (−1.61)
7	481694049	4268.87 (+172.87)	293.06 (+38.06)	263.89 (+8.89)	273.00 (+18.00)	246.66 (−8.34)	242.55 (−12.45)	277.56 (+22.56)	6.88 (−0.12)
8	785044942	4114.50 (+19.50)	238.02 (−16.98)	232.23 (−22.77)	272.69 (+17.69)	223.94 (−31.06)	283.42 (+28.42)	257.84 (+2.84)	9.94 (+2.94)
9	864268549	4058.37 (−36.63)	229.89 (−25.11)	201.48 (−53.52)	225.75 (−29.25)	275.25 (+20.25)	237.09 (−17.91)	221.17 (−33.83)	10.18 (+3.18)
10	13034519	4096.87 (+1.87)	246.97 (−8.03)	245.78 (−9.22)	232.42 (−22.58)	260.00 (+5.00)	277.53 (+22.53)	250.53 (−4.47)	3.31 (−3.69)

statistic is given in column three of Table 1. Seven values are above the mean value 4095; since the variance of the test statistic is 90.5, all deviations from the mean (shown in brackets below the actual values) are less than 1.9 standard deviations.

The results for Good's serial test for lags $\ell = 1, 2, 3, 4, 5, 6$, and $\nu = 16$ are given in columns four through nine of Table 1. There are $16^2 = 256$ cells and a mean value of 255; the exact distribution is not known but if a chi-squared distribution of 255 degrees of freedom is assumed with standard deviation of approximately 23, then the maximum and minimum values of the six columns of $+38.06$ and -53.52 are within 2.3 standard deviations.

The last column of Table 1 gives the results for the runs tests; the chi-squared statistic has a mean of seven but a standard deviation highly inflated by the unequal cell frequencies. No inordinately large deviations from the mean were obtained.

There is, therefore, no evidence of departures from randomness in the direct tests.

Results of tests on the transformed numbers are given in Table 2. For the median test, U is a unit normal deviate and the test results do not give any indications of departures from randomness. The values of KS+, KS−, KS for the ten series are given in

Table 2 Tests on the transformed pseudo-random numbers

Run number	*Median test* U	*Goodness-of-fit tests*			*Variance heterogeneity test*	
		KS+	KS−	KS	$H(10)$	$H(20)$
1	−1.069	0.455	1.006	1.006	4.442 (−)	10.340 (−)
2	−1.310	0.162	0.870	0.870	8.874 (+)	18.393 (+)
3	−1.425	1.334	0.208	1.334	5.992 (−)	20.325 (+)
4	−0.014	0.501	0.715	0.715	9.056 (+)	18.355 (+)
5	−1.265	0.346	1.026	1.026	4.845 (−)	7.116 (−)
6	+0.047	0.496	0.510	0.510	4.177 (−)	13.938 (−)
7	−1.105	0.113	0.733	0.733	9.423 (+)	13.948 (−)
8	+0.650	0.711	0.532	0.711	9.139 (+)	15.408 (−)
9	−0.104	0.482	0.544	0.544	10.120 (+)	14.769 (−)
10	−0.275	0.322	0.567	0.567	4.350 (−)	17.572 (−)

Table 3 Pseudo-random number generator

```
RANDOM   CSECT
         USING  *,15          INITIAL LINKAGE
         STM    2,5,28(13)
         LM     2,3,0(1)      LOAD ADDRESSES OF
                              VARIABLES PASSED
         L      5,A           COMPUTE NEXT INTEGER
         M      4,0(2)        RANDOM NUMBER WITH
                              X(I+1)=AX(I) (MOD P)
         D      4,P
         ST     4,0(2)
         SRL    4,7           COMPUTE NEXT REAL
                              RANDOM NUMBER
         A      4,CHAR
         ST     4,0(3)
         LM     2,5,28(13)    TERMINAL LINKAGE
         BR     14
CHAR     DC     F'1073741824' CONSTANTS. CHAR FIRST
A        DC     F'16807'      SO A IS ON DOUBLE WORD
P        DC     F'2147483647' BOUNDARY. MAKES LM
         END                  INSTRUCTION FASTER.
```

columns three through five in Table 2. The statistic KS has a one-sided upper 5 percent point of approximately 1.3 and a 1 percent point of approximately 1.6. No significantly large deviations occur in column five.

The variance heterogeneity test was applied to the transformed sequences with $k = 10$ and $k = 20$, giving variates with chi-squared distributions of 9 and 19 degrees of freedom, respectively. These values are given in columns six and seven. The plus and minus signs in brackets indicate whether the variates were below or above the median value of the distribution. The only abnormally large deviation occurs in series 5 for $H(20)$; the probability of a single value less than 7.116 is approximately 0.04. Taken as one of ten independent variates, this is not significantly small. Note, however, that the average of the variate values for $H(10)$ and $H(20)$ are well below the true means, indicating the possibility of a very subtle departure from randomness in the generator. It would be surprising, however, if some such departure did not show up in pseudo-random sequences of length 2^{16}. The overall conclusion, however, from the tests and the experiences in generating random permutations is that for a 32-bit word size the pseudo-random number generator is remarkably good.

Generator program

A program to implement the algorithm of Equation 1 is shown in Table 3. The program is written in SYSTEM/360 basic assembler language. This generator can be used by any program that con-

forms to the SYSTEM/360 FORTRAN linkage conventions. In particular, it may be invoked in a FORTRAN program (compiled on SYSTEM/360) by the statement:

```
CALL RANDOM (INT, REAL)
```

where INT is any full-word integer variable and REAL is any full-word real variable (single precision). The integer variable, INT, should be given an initial value before the first use of the generator. The generator returns an integer random number in INT and a real random number between 0 and 1 in REAL.

The program was run and timed internally on a SYSTEM/360 Model 67 computer. The generator was called 1,000,000 times within a FORTRAN "DO LOOP"; execution of the loop took 31,162,846 microseconds. Thus, we have an upper bound of $\sim$31.2 microseconds on the time to call a random number on the SYSTEM/360 Model 67. Faster times can be obtained using subroutines that generate *sequences* of random numbers and using subroutines that generate only integer random numbers. Another scheme has been proposed by Payne, Rabung, and Bagyo.[24] This scheme may be slightly faster on SYSTEM/360 computers. On the Model 91, however, the loop made may make the two methods comparable.

ACKNOWLEDGMENTS

We are indebted to Dr. B. Tuckerman, IBM Research Center, for his invaluable help in finding the positive primitive root of the prime $p = 2^{31} - 1$, and to Dr. F. G. Gustavson, also of IBM Research, for his helpful comments on the assembler language program. Mr. John R. B. Whittlesey of Mandrel Industries, Inc., has also provided some valuable criticisms.

CITED REFERENCES

1. D. W. Hutchinson, "A new uniform pseudo-random number generator," *Communications of the ACM* **9**, No. 6, 432–433 (1966).
2. D. Y. Downham and F. D. K. Roberts, "Multiplicative congruential pseudo-random number generators," *Computer Journal* **10**, No. 1, 74–77 (1967).
3. R. G. Herman, *The Statistical Evaluation of Random Number Generating Sequences for Digital Computers*, Office of Technical Services, U. S. Department of Commerce, Washington, D. C. (1961).
4. M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics* **3**, page 353, Charles Griffin and Company, London (1966).
5. R. R. Coveyou and R. D. MacPherson, "Fourier analysis of uniform random number generators," *Journal of the Association for Computing Machinery* **14**, No. 1, 100–119 (1967).
6. S. K. Zaremba, "The mathematical basis of Monte Carlo and quasi-Monte Carlo methods," *SIAM Review* **10**, 303–314 (1968).
7. G. Marsaglia, "Random numbers fall mainly in the planes," *Proceedings of the National Academy of Sciences* **61**, No. 2, 25–28 (1968).
8. J. R. B. Whittlesey, "A comparison of the correlational behavior of random number generators for the IBM SYSTEM/360," *Communications of the ACM* **11**, No. 9, 641–644 (1968).

9. L. E. Moses and R. F. Oakford, *Tables of Random Permutations*, Allen and Unwin, London (1963).
10. E. W. Page, "A note on generating random permutations," *Journal of the Royal Statistical Society C*, **16**, 273–274 (1967).
11. K. Jogdeo, "Asymptotic normality in nonparametric methods," *Annals of Mathematical Statistics* **39**, 905–922 (1968).
12. D. R. Cox and P. A. W. Lewis, *The Statistical Analysis of Series of Events*, page 54, Methuen, London; Dunod, Paris; and Barnes and Noble, New York (1966).
13. M. D. MacLaren and G. Marsaglia, "Uniform random number generators," *Journal of the Association for Computing Machinery* **12**, No. 1, 83–89 (1965).
14. I. J. Good, "The serial test for sampling numbers and other tests of randomness," *Proceedings of the Cambridge Philosophical Society* **49**, 276–284 (1953).
15. I. J. Good, "The generalized serial test and the binary expansion of $\sqrt{2}$," *Journal of the Royal Statistical Society A*, **130**, 102–107 (1967).
16. M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics* **2**, page 462, Charles Griffin and Company, London (1961).
17. J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation* **19**, 297–301 (1965).
18. D. R. Cox and P. A. W. Lewis, *The Statistical Analysis of Series of Events*, Chapter 6, Methuen, London; Dunod, Paris; and Barnes and Noble, New York (1966).
19. J. Durbin, "Tests of serial independence based on the cumulated periodogram." *Bulletin of the International Institute of Statistics* (1967).
20. R. A. Olshen, "Asymptotic properties of the periodogram of a discrete stationary process," *Journal of Applied Probability* **4**, No. 3, 508–528 (December 1967).
21. J. Durbin, "The probability that the sample distribution function lies between two parallel straight lines," *Annals of Mathematics and Statistics* **39**, No. 2, 398–411 (1968).
22. D. R. Cox and P. A. W. Lewis, *The Statistical Analysis of Series of Events*, page 168, Methuen, London; Dunod, Paris; and Barnes and Noble, New York (1966).
23. A. P. Basu, "On a generalized Savage statistic with application to life testing," *Annals of Mathematical Statistics* **39**, No. 5, 1591–1604 (1968).
24. W. H. Payne, J. R. Rabung, and T. P. Bogyo, "Coding the Lehmer pseudo-random number generator," *Communications of the ACM* **12**, No. 2, 85–86 (February 1969). See also W. Liniger, "On a method by D. H. Lehmer for the generation of pseudo-random numbers," *Numerische Mathematik* **3**, 265–270 (1961).