AD-773 747

STATISTICAL TESTS OF SOME WIDELY USED AND RECENTLY PROPOSED UNIFORM RANDOM NUMBER GENERATORS

G. P. Learmonth, et al

Naval Postgraduate School
Monterey, California

November 1973

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NPS55LW73111A

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
STATISTICAL TESTS OF SOME WIDELY USED AND RECENTLY PROPOSED UNIFORM RANDOM NUMBER GENERATORS

5. TYPE OF REPORT & PERIOD COVERED
Interim

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
G. P. Learmonth and P. A. W. Lewis

8. CONTRACT OR GRANT NUMBER(s)
NSF AG 476

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
--

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
November 1973

13. NUMBER OF PAGES
20

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Presented at Computer Science and Statistics, Seventh Annual Symposium on the Interface held at Iowa State University, Ames, Iowa, October 18 - 19, 1973.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Random number generator	Congruential generator
Pseudo-random numbers	Runs test
Shuffled random numbers	Serial test
Division simulation	Tausworthe generator

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Several widely used uniform random number generators have been extensively subjected to three commonly used statistical tests of uniformity and randomness. The object was i) to examine the power of these statistical tests to discriminate between "good" and "bad" random number generators, ii) to correlate these results with recently proposed mathematical characterizations of random number generators which might also be useful in such a discrimination, and iii) to examine the effect of shuffling on the random number generators. 13

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral M. B. Freeman
Superintendent

M. U. Clauser
Provost

ABSTRACT

Several widely used uniform random number generators have been extensively subjected to three commonly used statistical tests of uniformity and randomness. The object was i) to examine the power of these statistical tests to discriminate between "good" and "bad" random number generators, ii) to correlate these results with recently proposed mathematical characterizations of random number generators which might also be useful in such a discrimination, and iii) to examine the effect of shuffling on the random number generators.

Briefly the results show that the commonly used runs test has virtually no power to discriminate between "good" and "bad" generators, while serial tests perform better. Also shuffling does help, although much more needs to be done in this area. And finally, there is some utility to the mathematical characterizations, but many unanswered questions.

Prepared by:

Peter A. W. Lewis

Peter A. W. Lewis*
Department of Operations Research
and Administrative Sciences

Gerard P. Learmonth

Gerard P. Learmonth
Computer Center

Approved by:

J. R. Borsting, Chairman
Department of Operations Research
and Administrative Sciences

John M Wozencraft

J. M. Wozencraft
Dean of Research

*Partial support by the National Science Foundation through Grant AG476 is gratefully acknowledged

NPS55LW73111A

STATISTICAL TESTS OF SOME WIDELY USED AND RECENTLY PROPOSED UNIFORM RANDOM NUMBER GENERATORS

G. P. Learmonth and P. A. W. Lewis*
Naval Postgraduate School
Monterey, California

Abstract

Several widely used uniform random number generators have been extensively subjected to three commonly used statistical tests of uniformity and randomness. The object was i) to examine the power of these statistical tests to discriminate between "good" and "bad" random number generators, ii) to correlate these results with recently proposed mathematical characterizations of random number generators which might also be useful in such a discrimination, and iii) to examine the effect of shuffling on the random number generators.

Briefly the results show that the commonly used runs test has virtually no power to discriminate between "good" and "bad" generators, while serial tests perform better. Also shuffling does help, although much more needs to be done in this area. And finally, there is some utility to the mathematical characterizations, but many unanswered questions.

1. INTRODUCTION

The generation of pseudo-random numbers has been the subject of literally hundreds of papers in the computing and simulation literature (see the bibliography by Nance and Overstreet [13]). By far the most popular method of pseudo-random number generation has been the Lehmer congruential method:

$$X_{n+1} \equiv A \cdot X_n + C \qquad (\text{mod } P). \qquad (1)$$

The theory underlying the implementation and use of the congruence (1) is well described, see e.g. Knuth [5].

Recent attention has been given to alternatives to the Lehmer congruential method. Most notable are the feedback shift register generators [10, 15, 18]. These generators are based on irreducible polynomials over GF(2) and are capable of producing extremely long sequences relative to the word size of the computer.

The generation of pseudo-random numbers is itself a simulation, that is, we are attempting to simulate a random sequence of numbers. As in any simulation experiment, we apply some sort of test procedure to verify how well our simulation achieved the desired goal. Typically runs tests, serial tests, and various chi-square tests for independence are applied to relatively short sections of the pseudo-random sequence. Many of these tests have been outlined by Gorenstein [4] and Knuth [5]. Recent results have indicated that the power of these classical tests to detect "poor" generators is suspect.

We now have mathematical characterizations of pseudo-random sequences which should help one to discriminate between generators. Chief amongst these characterizations are the spectral structure described by Coveyou and MacPherson [1] and the lattice structure test advanced by Marsaglia [11] and Smith [17] for congruential generators. There are also presumably better statistical tests based on the properties of the periodogram, as outlined in Lewis, Goodman, and Miller [9]. All of these tests are more sensitive to subtle departures from randomness than some of the older tests, both mathematical and statistical, particularly with respect to the higher dimensional properties of the pseudo-random sequences. However, their interrelationship is not well understood. For example, Marsaglia [11] derides the utility of the Coveyou-MacPherson work. Moreover, in a real sense the proof of the pudding (here a pseudo-random number generator) is in the using and it is not clear how well the mathematical tests predict the results of the statistical tests,

*Support for part of this work under NSF Grant AG 476 is gratefully acknowledged.

i.e. the ability of the sequences to simulate independent uniform random numbers.

In this paper we propose to examine the performance of six pseudo-random number generators on the runs test and serial test for pairs and triples. Among these six generators are two known to be poor. Testing these serves to demonstrate the lack of discriminatory power of some of the statistical tests. In the case of one of them (RANDU) the mathematical characterizations clearly predict poor performance. The idea of shuffling the pseudo-random sequence will also be explored as a means of improving the statistical performance of a generator. This is a relatively new and untested method.

2. THE GENERATORS

The six generators used in this report were all designed for use on the 32-bit word IBM System/360. Table 1 summarizes the pertinent information about each of the six generators.

LLRANDOM is a version of the generator reported in Lewis, Goodman and Miller [9] (see Learmonth and Lewis [7] for details of LLRANDOM). Other versions of the Lewis, Goodman and Miller generator are used in the new IBM SL/MATH package and in the IBM version of the APL language. RANDU is the uniform random number generator provided with the IBM Scientific Subroutine Package [16] which, despite disclaimers of support from IBM, is still unfortunately widely used. TAUS is a feedback shift register generator proposed by Tausworthe. The present implementation was patterned after Payne [15].

GFSR is a FORTRAN implementation of a "generalized" feedback shift register generator by Lewis and Payne [10]. The algorithm employed in GFSR simplifies the coding of the feedback shift register and generalizes the implementation to virtually any word-size machine.

The last two generators listed in Table 1 were taken from the package "Super-Duper" by G. Marsaglia. The novelty of "Super-Duper" lies in the fact that it consists of combining (exclusive OR'ing) the results of both a Lehmer congruential generator and a feedback shift register generator. For one case (Super-Duper) we use this combination method. For the last generator we used only the congruential part of the Super-Duper package.

2.1 SHUFFLING.

The sequences produced by pseudo-random number generators are deterministic. In order to disguise this pattern, several techniques have been proposed to reorder, or shuffle, the sequence emanating from a generator. Marsaglia and Bray [13] have proposed maintaining a table of pseudo-random numbers within the generator. Two independent pseudo-random numbers are generated at each call. The first number, appropriately scaled, is used as a random index into the table. The tabled value is then returned with the number from the second sequence replacing it in the table.

Another procedure mentioned by Tukey [19] is to generate blocks of, say 1024, pseudo-random numbers and then shuffle them according to some random permutation. After all or some of these numbers have been used, generate another block and shuffle them according to the same permutation.

Marsaglia's [11] method of exclusive OR'ing the output of a Lehmer congruential generator and a feedback shift register generator is also aimed at breaking-up the basic deterministic pattern of the Lehmer generator. It produces a perfect (in the sense of unit cell volume) lattice structure.

For the purposes of the testing described here, the Marsaglia and Bray method was used with five of the six generators, the exception being GFSR. A table size of 128 was chosen and two sequences of the same generator were used, each with a different

NAME	TYPE	TRINOMIAL or MULTIPLIER	MODULUS	PERIOD	LANGUAGE
LLRANDUM	LEHMER CONGRUENTIAL	$7^5 = 16807$	$2^{31} - 1$	$2^{31} - 2$	SYSTEM/360 ASSEMBLER
RANDU	LEHMER CONGRUENTIAL	$2^{16} + 3 = 65539$	2^{32}	2^{29}	FORTRAN IV
TAUS	FEEDBACK SHIFT REGISTER	$X^{31} + X^{13} + 1$		$2^{31} - 1$	FORTRAN IV
GFSR	GENERALIZED FEEDBACK SHIFT REGISTER	$X^{124} + X^{37} + 1$		$2^{124} - 1$	FORTRAN IV
SUPER-DUPER	LEHMER CONGRUENTIAL WITH FSR	69069 $X^{17} + X^{15} + 1$	2^{32}	$2^{46} - 2^{29}$	SYSTEM/360 ASSEMBLER
SUPER-DUPER (CONGRUENTIAL ONLY)	LEHMER CONGRUENTIAL	69069	2^{32}	2^{29}	SYSTEM/360 ASSEMBLER

TABLE 1. SUMMARY OF THE SIX GENERATORS TESTED

starting value. To be consistent, this shuffling scheme was used in LLRANDOM even though LLRANDOM has the capability of shuffling directly incorporated into the generator. The shuffling scheme in LLRANDOM requires only one pseudo-random number since seven bits of that number are used to index the table of 128 ($2^7=128$). These seven bits are quite random due to our choice of modulus (see Knuth [5] page 12). Results of the statistical testing of the actual shuffled sequence produced in the LLRANDOM package are given in Appendix A.

As will be seen, the results of shuffling are encouraging when measured by statistical tests, even for poorly constructed generators. However, it is not entirely clear how shuffling changes the lattice structure of the sequence, and this raises important questions.

3. THE RUNS TEST

Pseudo-random number generators produce sequences which are inherently periodic although when P is prime and A is a positive primitive root of P, as in LLRANDOM, the numbers do not repeat until P have been generated and it is felt that this helps reduce local cyclic effects. The runs-up-and-down test is frequently applied to pseudo-random sequences to investigate the possibility of local (less than full period) cycles or stationary dependence.

Under the null hypotheses that a sequence is random the expected value and variance of the number of runs of given lengths in a sequence of length N are easily derived. Levene and Wolfowitz [8] have shown that for the observed number of runs of length d, N_d, the statistic

$$\frac{N_d - E[N_d]}{\{\mathrm{Var}[N_d]\}^{1/2}} \qquad (2)$$

is asymptotically normally distributed with mean 0 and variance 1 as the sample size, n, tends to infinity.

The literature on the power of the runs test is rather sparse, even though it is widely recommended as a test. The only significant result is due to F. N. David [2] which shows that the runs test has, asymptotically, the greatest power of any test of randomness against the alternative of first-order Markov dependence in a binary sequence.

3.1 TEST FORMULATION.

The runs-up-and-down test used here is patterned after Kendall and Stuart and was used and described fully by Lewis, Goodman and Miller [9]. Samples of size $N = 65{,}536$ were generated and runs counted for lengths $d = 1,2,\ldots,7$, with a last cell $d = 8$ collecting observed runs of length 8 or more. The expected number of runs is given by

$$E[N_d] = \frac{2(n-d-2)(d^2+3d+1)}{(d+3)!} \qquad d = 1,2,\ldots,7 \quad (3)$$

$$E[N_{8+}] = \frac{2n-7}{3} - \sum_{d=1}^{7} N_d. \qquad (4)$$

A chi square statistic is computed for the sample generated as

$$\chi_7^2 = \frac{N_d - E[N_d]^2}{E[N_d]} \qquad (5)$$

This statistic is only approximately distributed as a chi square variate with 7 degrees of freedom since the expected values in each cell are not equal, the last two cells both having expected values less than 5. Additionally, there is a certain lack of independence since a long run is usually followed by a short run.

The test procedure was to perform this runs test for samples of size $N = 65{,}536$. One hundred replications were then performed using different starting values (seeds).

3.2 RESULTS OF THE TESTS

Samples of one hundred chi square statistics (5) were obtained from each of the six generators. An additional sample of one hundred was taken for each of the generators, except GFSR, with shuffling implemented. Each set of one hundred was considered to be a sample from the distribution of this runs test statistic (5).

A preliminary analysis of the sample data showed that for some starting values, all six of the generators would fail the runs test at a 5% level based on a chi square distribution with 7 degrees of freedom. Even when shuffled, each generator produced values of the test statistic which were either too high or too low for acceptance. The question is, however, are the numbers of rejections consistent with the chi-square distribution theory, and if not, can we simulate the true (null) distribution with our results. The answer to the first question is that no generator produced results which were grossly different from that predicted by the chi-square distribution theory.

Thus, to further assess the ability of the runs test to discriminate "bad" generators from "good" generators, it was decided to compare sample distributions of the runs test statistic (5). A two-sample Kolmogorov-Smirnov test was performed on all pairs of samples (of size 100). The two-sample K-S test is distribution-free and we avoided the problem of having to specify an exact null distribution for the statistic (5). A two-sample K-S test subroutine was programmed using the algorithm presented by Kim and Jennrich [6].

Table 2 summarizes the results of these tests with the generators not shuffled. The pair of values given correspond to the sample K-S criterion, $c/(m\cdot n)$, and the $\Pr\{D(m,n) > c/(m\cdot n)\}$, respectively. The results indicate that RANDU produces runs test statistics which are not distributionally commensurate with the other five generators. This was to be expected since RANDU is one of the two "poor" generators among the six, but the indications are not strong. The somewhat surprising result is that TAUS appears, except for the case of RANDU, to be similarly distributed with the remaining four generators. TAUS is the other suspect generator referred to above. This is because the results of Toothill, Robinson and Adams [18] indicate that a feedback shift register generator with the primitive trinomial used in TAUS could not comply with the moment and marginal results of Levene and Wolfowitz [8].

Table 3 presents results of a similar test using the shuffled samples (with the exception of GFSR which was not shuffled). With the type 1 error at 5%, all generators appear to be distributionally commensurate. This result is encouraging since it was hoped that shuffling would improve the performance of relatively poor generators such as RANDU.

	RANDU		TAUSWORTHE		MARSAGLIA (MIXED)		MARSAGLIA (CONGRUENTIAL)		GFSR	
LLRANDOM	.23	.0062*	.13	.2820	.17	.0783	.17	.0783	.09	.7021
RANDU	--		.21	.0156*	.19	.0364*	.18	.0539	.24	.0038*
TAUSWORTHE	--		--		.09	.7021	.08	.8154	.13	.2820
MARSAGLIA (MIXED)	--		--		--		.12	.3682	.16	.1112
MARSAGLIA (CONGRUENTIAL)	--		--		--		--		.15	.1549

TABLE 2. KOLMOGOROV-SMIRNOV TWO SAMPLE TEST ON RUNS TEST STATISTICS. SAMPLE SIZE = 100. NOT SHUFFLED.

	RANDU		TAUSWORTHE		MARSAGLIA (MIXED)		MARSAGLIA (CONGRUENTIAL)		GFSR	
LLRANDOM	.12	.3682	.07	.9084	.09	.7021	.14	.2112	.13	.2820
RANDU	--		.09	.7021	.12	.3682	.18	.0539	.17	.0783
TAUSWORTHE	--		--		.08	.8154	.14	.2112	.16	.1112
MARSAGLIA (MIXED)	--		--		--		.13	.2820	.11	.4695
MARSAGLIA (CONGRUENTIAL)	--		--		--		--		.11	.4695

TABLE 3. KOLMOGOROV-SMIRNOV TWO SAMPLE TEST ON RUNS TEST STATISTICS. SAMPLE SIZE = 100. ALL GENERATORS EXCEPT GFSR ARE SHUFFLED.

As a last test, the samples of shuffled data from five of the six generators (the exception, again, was GFSR) were pooled to form a sample of 500 from the distribution of the runs test statistic. A K-S test was then performed matching the individual samples of 100 (not shuffled) against this composite. The results showed only RANDU to be distributionally different, again demonstrating the poor power of the runs test. Another result is that we have a simulation of size 500 of the distribution of the statistic (5) under (hopefully) the null independence hypothesis. For reference the order statistic estimate of the 0.95 quantile is 15.82; that for 0.99 is 22.05. For reference the corresponding quantiles for a chi-square variate with seven degrees of freedom are $\chi^2_{7,0.95} = 14.07$ and $\chi^2_{7,0.99} = 18.48$ respectively.

Although the runs test has been used frequently in the testing of pseudo-random number generators, we feel now that the test is of very doubtful use. The lack of distributional theory concerning the runs test statistic (5), combined with its empirical lack of ability to discriminate TAUS leads us to conclude that the runs test should not be employed in testing pseudo-random number generators. We hope we have laid it to rest forever. It is interesting to note that as late as 1971 Smith [17], in an excellent paper, recommended a generator on the basis of its lattice structure and then said "To check it, we carried out what is generally recognized as a sensitive test of uniformity and independence of a sequence, namely the runs test."

4. THE SERIAL TEST

In testing pseudo-random sequences, it is useful to test the uniformity of successive but not necessarily contiguous numbers taken as k-tuples for $k = 2,3,\ldots$ A standard test employed is the serial test.

Taking the sequence $\{U_i\}$ of uniform (0.0,1.0) deviates from a generator one divides the k dimensional unit hypercube into r^k smaller equisized hypercubes. For binary computers r is usually taken to be a power of 2 and determines the number of (leading) bits from each number which will be tested. The k-tuples themselves may be taken as overlapping or nonoverlapping, e.g. the sequence of 2-tuples $\{U_1U_2\},\{U_2,U_3\},\{U_3U_4\}$ or the sequence of 2-tuples $\{U_1U_2\},\{U_3U_4\},\{U_5,U_6\}$, ... In a sample of size n, the expected number in each of the small hypercubes is n/r^k under the null hypothesis of multidimensional uniformity. The following chi-square test statistic may then be computed:

$$S_k = \frac{r^k}{n} \sum_{j_1=1}^{r} \cdots \sum_{j_k=1}^{r} \left(f_{j_1,j_2,\ldots,j_k} - n/r^k\right) \qquad (6)$$

where $f_{j_1,j_2,\ldots,j_k}$ is the observed number of k-tuples in each small hypercube. For the case of nonoverlapping k-tuples, S_k is distributed asymptotically as chi square with $r^k - 1$ degrees of freedom. For overlapping k-tuples, a correction, due to Good [3], involving a chi square test for uniformity must be included for S_k to be approximately chi square.

The desirability for k-dimensional uniformity stems from the fact that in many stochastic variate generation algorithms, k-tuples of uniforms are used together to form the desired variate. For such an application, nonoverlapping k-tuples would be tested. The case for using overlapping intervals in a serial test is somewhat less clear. Aside from the distributional problems with the test statistic, it is hard to imagine a simulation which would require overlapping k-tuples, implying that some uniform deviate would be used more than once. The serial test when applied to overlapping intervals, however, can be considered as a test for serial independence in a pseudo-random sequence. Like the product form of the serial correlation (autocorrelation) statistic, this form of the serial test should be sensitive to lagged serial dependence in the pseudo-random sequence.

4.1 TEST FORMULATION

For samples of size $N = 65,541$, one hundred samples of the test statistic (6) were computed for each of the six generators. The tests were repeated for five generators with shuffling implemented. Two forms of the test were employed: the first set involved examining pairs of successive overlapped numbers and the second form involved overlapped triples. In both cases the first four bits were used.

The successive deviates were also lagged for $j = 1,2,\ldots,6$, so that, for example, for pairs we took contiguous numbers, numbers one apart, ... , numbers five apart (lag 6).

4.2 TEST RESULTS

Since the use of overlapped intervals invalidates comparing moments from the sample statistics (6) with expectation from a chi square distribution, we went directly to the two sample Kolmogorov-Smirnov test for distributions. Table 4 summarizes the results for the serial test for pairs on the six generators without shuffling. Only the first lag is shown here for brevity.

The results appear inconclusive. LLRANDOM has been rejected in spite of published results ([9] and [11]) indicating good two-space properties for this generator. In [9] LLRANDOM was tested by comparing the test results to the χ^2 (40 degrees of freedom) distribution, since no true distribution was known. It may be that what is showing up here is the relatively poor 2-lattice structure of the LLRANDOM generator, as shown in Table 8a, indicating correlation between the predictions of the statistical and mathematical tests.

Shuffling was then applied with little discernible improvement. Pairs of generators are still rejected, as shown in Table 5, but the shuffled LLRANDOM is not singled out. We return to this later.

In triples, Table 6 summarizes the results for the generators without shuffling. Only the first lag is shown. For the triples this means taking three successive and contiguous pseudo-random numbers, i.e. $\{U_1U_2U_3\},\{U_2U_3U_4\},\{U_3U_4U_5\},\ldots$ The only significant result here is that RANDU fails completely, and the departures, as predicted by the lattice structure and wave number tests, are gross. Although the distribution of the test statistic (6) is not chi square due to the overlapping intervals, the expected value of (6) is still $r^k - 1$ or 4095. While the test statistics for the other five generators were within two standard deviations of this expectation, RANDU's sample average test statistic was 28,787! We expected RANDU to perform badly here since RANDU is known to be poor in three-space.

With shuffling implemented, the results in Table 7 indicate that all of the generators, including RANDU are distributionally commensurate. Thus there is a very substantial improvement in this property of the sequences induced by shuffling.

5. CONCLUSIONS AND RECOMMENDATIONS

Many of today's statistical questions are being answered through large-scale simulation. The generation of good pseudo-random deviates for simulation and Monte Carlo experiments is of prime importance. Unfortunately many generators are being used whose statistical properties make them a hindrance rather than an aid in such experiments. Through personal communication, we have heard from several researchers who have had their experiments stymied due to poor pseudo-random number generators. The typical computer center will offer several generators of either unknown progeny or ones whose testing is rather weak. RANDU, which is one of the most widely used generators, is typical of this situation. It is supplied by IBM as part of the Scientific Subroutine Package and is therefore

	RANDU		TAUSWORTHE		MARSAGLIA (MIXED)		MARSAGLIA (CONGRUENTIAL)		GFSR	
LLRANDOM	.34	.0001*	.38	.0000*	.33	.0001*	.30	.0002*	.38	.0000*
RANDU	--		.11	.4695	.08	.8154	.15	.1549	.13	.2820
TAUSWORTHE	--		--		.13	.2820	.16	.1112	.07	.9084
MARSAGLIA (MIXED)	--		--		--		.12	.3682	.17	.0783
MARSAGLIA (CONGRUENTIAL)	--		--		--		--		.21	.0156*

TABLE 4. KOLMOGOROV-SMIRNOV TWO SAMPLE TEST ON SERIAL TEST STATISTICS FOR PAIRS. SAMPLE SIZE = 100. NOT SHUFFLED.

	RANDU		TAUSWORTHE		MARSAGLIA (MIXED)		MARSAGLIA (CONGRUENTIAL)		GFSR	
LLRANDOM	.17	.0783	.20	.0241*	.10	.5830	.08	.8154	.20	.0241*
RANDU	--		.10	.5830	.11	.4695	.19	.0364*	.11	.4695
TAUSWORTHE	--		--		.14	.2112	.25	.0023*	.10	.5830
MARSAGLIA (MIXED)	--		--		--		.14	.2112	.13	.2820
MARSAGLIA (CONGRUENTIAL)	--		--		--		--		.21	.0156*

TABLE 5. KOLMOGOROV-SMIRNOV TWO SAMPLE TEST ON SERIAL TEST STATISTICS FOR PAIRS. SAMPLE SIZE = 100. ALL GENERATORS EXCEPT GFSR ARE SHUFFLED.

	RANDU		TAUSWORTHE		MARSAGLIA (MIXED)		MARSAGLIA (CONGRUENTIAL)		GFSR	
LLRANDOM	1.0	.0000*	.19	.0364*	.15	.1549	.12	.3682	.12	.3682
RANDU	--		1.0	.0000*	1.0	.0000*	1.0	.0000*	1.0	.0000*
TAUSWORTHE	--		--		.15	.1549	.14	.2112	.13	.2820
MARSAGLIA (MIXED)	--		--		--		.12	.3682	.09	.7021
MARSAGLIA (CONGRUENTIAL)	--		--		--		--		.12	.3682

TABLE 6. KOLMOGOROV-SMIRNOV TWO SAMPLE TEST ON SERIAL TEST STATISTICS FOR TRIPLES. SAMPLE SIZE = 100. NOT SHUFFLED.

	RANDU		TAUSWORTHE		MARSAGLIA (MIXED)		MARSAGLIA (CONGRUENTIAL)		GFSR	
LLRANDOM	.12	.3682	.16	.1112	.18	.0539	.18	.0539	.16	.1112
RANDU	--		.07	.9084	.15	.1549	.13	.2820	.12	.3682
TAUSWORTHE	--		--		.11	.4695	.12	.3682	.12	.3682
MARSAGLIA (MIXED)	--		--		--		.17	.0783	.12	.3682
MARSAGLIA (CONGRUENTIAL)	--		--		--		--		.10	.5830

TABLE 7. KOLMOGOROV-SMIRNOV TWO SAMPLE TEST ON SERIAL TEST STATISTICS FOR TRIPLES. SAMPLE SIZE = 100. ALL GENERATORS EXCEPT GFSR ARE SHUFFLED.

convenient to use. Several reports are around which cite results of tests for uniformity, runs tests, and serial tests all confirming RANDU's adequacy. Users of other computers are in even worse shape since their random number generators are rarely documented and no one seems to know from whence they came!

The random number generators which have appeared in the literature rarely specify what tests, if any, have been performed. In the IMSL Library, for instance, not only are there no test results cited, but the user is given the option to use his own multiplier. Those algorithms which do acknowledge testing usually cite the runs tests and serial tests investigated here (cf. Smith [17] cited above). It was our aim in this paper to show whether these tests are sensitive to the type of departures from randomness which would be crucial to serious simulation efforts.

The most distinct conclusion one can draw from the results given in this paper is that the runs test has no utility as a test for randomness in pseudo-random number generators.

The serial test is somewhat more sensitive than the runs test and interesting conclusions can be drawn from the results obtained here, as follows:

Using pairs (Tables 4 and 5) a subtle departure in LLRANDOM was evidenced, but nothing startling showed up about RANDU. Shuffling did not clear up the situation completely. This will probably be used as the basis for further investigations to discriminate among shuffled generators. Using triples, gross departures in RANDU appear (of the hundred samples, all were rejected at a 5% level using the χ^2 approximation for the distribution of the statistic) and after shuffling, all generators (including RANDU) are statistically commensurate. The interesting point here is that it is not clear how much the shuffling we have used changes the lattice structure of RANDU, so that we may have a generator which passes the statistical tests but has "poor" structural characteristics. Thus there is evidence that Marsaglia's work has by no means provided the final answer on congruential and other generators. Also more needs to be known about the effect of various shuffling schemes on the mathematical characterizations.

A word about the inherent drawbacks of the serial test is in order here:

(i) A problem with any serial test is the need for a great deal of storage. In the tests reported here we were limited to looking at only the first four bits. (This takes 4096 memory cells; five bits takes 8 times as many.) If one takes only the first four bits one can hardly have a sensitive test for the whole pseudo-random word. Also, the lattice structure of the truncated sequence is different from that of the whole word, though the structures probably mimic each other to some degree.

(ii) There is a great problem of deciding what lags to take in serial tests. In the tests described above RANDU failed miserably on contiguous (1,1) triples, but not on other triples. Thus, if one looked at six lags, as we have done, it might be that departures would show up at other lags.

Fortunately, theoretical results obtained by Professor Murray Rosenblatt (personal communication) indicate that it is primarily the tests for contiguous pairs which will show up departures in congruential generators. His results also provide some theoretical basis for shuffling. Another possibility is to go to tests for randomness based on the empirical spectrum (Lewis, Goodman and Miller [9]). This test combines the serial tests of all lags.

For comparative purposes, Table 8a below shows the relative lengths of sides of unit cells for the three Lehmer congruential generators in this paper. These numbers were computed by Marsaglia [11]. Note that the one dimension of the 2-lattice for LLRANDOM is relatively large. Also, one dimension of the three-lattice of RANDU is very large.

Generator	2-lattice	3-lattice	4-lattice
LLRANDOM	1,7.6	1,1.6,3.9	1,1.09,1.69,2.07
RANDU	1,1.	1,1.1,1819	1,7.3,1856,1872
SUPER-DUPER (CONGRUENTIAL ONLY)	1,1.06	1,1.14,1.29	1,1.14,1.16,1.30

TABLE 8a.

Table 8b below lists the Coveyou-MacPherson wave numbers for LLRANDOM and RANDU. These numbers were computed and kindly supplied to us by Dr. L. Richard Turner of the NASA Lewis Research Center. As Smith [17] points out, there is a direct relationship between the 2-lattice and the 2 dimensional wave number. Small wave numbers correspond to poor n-space properties.

Dimension	LLRANDOM	RANDU
2	16807	23172
3	638.9	10.86
4	147.25	10.77
5	67.21	10.77
6	29.92	
7	16.55	

TABLE 8b.

Since congruential generators have inherent flaws (see Marsaglia [12]), the idea of shuffling the sequence as it leaves the generator seems to be a reprieve for these popular generators. Shuffling showed improvements in the tests performed here, particularly for RANDU in the serial test on triples. The effect of the various types of shuffling which have been proposed will be investigated in extensions of future work.

The case for feedback shift register generators is encouraging, but still unclear. After an initial flurry of activity in this field, the careful work of Toothill, Robinson, and Adams [18] showed that these types of pseudo-random number generators also had their flaws. Without a careful choice of primitive trimonial, a feedback shift register generator could be shown analytically to have bad statistical properties (generally runs properties, and these may be irrelevant). The generator, GFSR, has been constructed more carefully than TAUS but is rather costly in both initialization time and computer storage, making its utility doubtful. Until more extensive investigation on feedback shift registers is forthcoming, we cannot conclude that they are preferable to the shuffled congruential generators

Acknowledgment

We would like to thank LT D. W. Robinson for coding the generalized feedback shift register generator, GFSR, for inclusion with these results.

6. REFERENCES

[1] R. R. Coveyou and R. D. MacPherson, "Fourier analysis of uniform random number generators," JACM, Vol. 14, No. 1, 1967.

[2] F. N. David, "A power function for tests of randomness in a sequence," Biometrika, Vol. 34, 1947.

[3] I. J. Good, "On the serial test for random sequences," Annals of Mathematical Statistics, Vol. 38, 1957.

[4] S. Gorenstein, "Testing a random number generator," CACM, Vol. 10, No. 2, 1967.

[5] D. E. Knuth, The Art of Computer Programming: Seminumerical Algorithms, Vol. 2, Reading, Mass.: Addison-Wesley, 1969.

[6] P. J. Kim and R. I. Jennrich, "Tables of the exact sampling distribution of the two-sample Kolmogorov-Smirnov criterion D_{mn}, $(m<n)$," in Selected Tables in Mathematical Statistics, Vol. 1, H. L. Harter and D. B. Owen, eds., Chicago: Markham Publishing Co., 1970.

[7] G. P. Learmonth and P. A. W. Lewis, Naval Postgraduate School Random Number Generator Package: LLRANDOM, Research report NPS55LW 73061A, Naval Postgraduate School, Monterey, California, 1973.

[8] H. Levene and J. Wolfowitz, "The covariance matrix of runs up and down," Annals of Mathematical Statistics, Vol. 15, 1944.

[9] P. A. W. Lewis, A. S. Goodman and J. M. Miller, "A pseudo-random number generator for the System/360," IBM Systems Journal, Vol. 8, 1969.

[10] T. G. Lewis and W. H. Payne, "Generalized feedback pseudorandom number algorithm," JACM, Vol. 20, 1973.

[11] G. Marsaglia, "The structure of linear congruential sequences," in S. K. Zaremba, Ed., Applications of Number Theory to Numerical Analysis, New York: Academic Press, 1972.

[12] G. Marsaglia, "Random numbers fall mainly in the planes," Proc. National Academy of Sciences, Vol. 61, 1968.

[13] G. Marsaglia and T. A. Bray, "One-line random number generators and their use in combinations," CACM, Vol. 11, 1968.

[14] R. E. Nance and C. Overstreet, "Random number generation," Bibliography No. 29, Computing Reviews, Vol. 13, 1972.

[15] W. H. Payne, "FORTRAN Tausworthe pseudorandom number generator," CACM, Vol. 13, 1970.

[16] IBM System/360 Scientific Subroutine Package, GH20-0205.

[17] C. S. Smith, "Multiplicative pseudo-random number generators with prime modulus," JACM, Vol. 18, 1971.

[18] J. P. R. Toothill, W. D. Robinson and A. G. Adams, "The runs up-and-down performance of Tausworthe pseudo-random number generators," JACM, Vol. 18, 1971.

[19] J. W. Tukey, "How computing and statistics affect each other," to appear.

APPENDIX

The method of shuffling used throughout this paper was based on a proposal of Marsaglia and Bray [13]. Section 2 described this technique and mentioned that the package LLRANDOM had a shuffling scheme built directly into it.

The following table presents the results of performing the runs test, serial test for pairs, and serial test for triples using LLRANDOM's own shuffling scheme.

As before, 100 sample values of each test statistic were obtained from LLRANDOM using its own shuffling scheme. This sample was compared with the samples of the six generators included in the report (including LLRANDOM shuffled according to the Marsaglia-Bray scheme).

The results of the two sample Kolmogorov-Smirnov tests are shown. The pair of values in each cell are the observed value of the Kolmogorov-Smirnov criterion, $c/(m\cdot n)$, and the value of $\Pr\{D_{m,n} > c/(m\cdot n)\}$, respectively. With the exception of the LLRANDOM-Marsaglia (congruential) comparison on the serial test for pairs, the results indicate LLRANDOM to be distributionally commensurate with the other shuffled generators at the 5% confidence level.

GENERATOR	RUNS		SERIAL (PAIRS)		SERIAL (TRIPLES)	
LLRANDOM (MARSAGLIA-BRAY SCHEME)	.10	.5830	.16	.1112	.10	.5830
RANDU	.11	.4695	.08	.8154	.07	.9084
TAUS	.08	.8154	.11	.4695	.09	.7021
MARSAGLIA (MIXED)	.09	.7021	.10	.5830	.10	.5830
MARSAGLIA (CONGRUENTIAL)	.13	.2820	.19	.0364*	.13	.2820
GFSR	.15	.1549	.08	.8154	.12.	.3682

TABLE A. KOLMOGOROV-SMIRNOV TWO SAMPLE TEST COMPARING LLRANDOM (SELF-SHUFFLING) WITH SIX OTHER SHUFFLED GENERATORS.