Uniformity of Congruential Pseudorandom Number Generators. Dependence on Length of Number Sequence and Resolution

LEIF HOLMLID AND KJELL RYNEFORS

Department of Physical Chemistry, University of Göteborg, Fack, S-402 20 Göteborg, Sweden

Received May 10, 1976; revised March 30, 1977

Congruential pseudorandom number generators have been used for a long time in Monte Carlo sampling of initial values for trajectory calculations in chemical reactive scattering studies. Difficulties with standard generators of the type normally used for such work (multiplicative generators with modulus 2^n or 10^n), i.e., a nonrandom behavior of the sampled distributions, are reported here. This behavior is attributed to the existence of a long-range structure not previously well known in the number sequences from such generators. This structure is thoroughly investigated for one generator with modulus 10^8 previously well tested and is shown to give much too smooth or much too rough distributions in frequency tests, depending on the resolution in the random numbers and the fraction of a period used in the test. For a resolution of 10^{-n} only a fraction of the period considerably less than 10^{-n} should be used (e.g., for $n = 2$, less than one-hundredth of the period). It is concluded, that generators with modulus a prime will be preferable in this respect, and that the methods often used to break up the short-range correlation (shuffling, etc.) are not efficient for destroying this long-range structure.

1. INTRODUCTION

The choice of a random number generator for Monte Carlo trajectory calculations of reactive collisions between molecules is, unfortunately, not simple. The generator supplied by IBM in the SSP library (RANDU) was the one most easily accessible, when a study of absolute complex formation cross sections in alkali–alkali halide reactive scattering [1] was started. RANDU is of the congruential multiplicative type,

$$A_{k+1} = XA_k \bmod M, \qquad k = 0, 1, \ldots, \tag{1}$$

where X (the multiplier), M (the modulus) and A_k are all integer numbers. A_{k+1} is an integer pseudorandom number, $0 < A_{k+1} < M$. Similar generators have been used for a long time [2–8], also in Monte Carlo trajectory calculations [9–12], and have been tested thoroughly with many types of tests [2–7]. Several monographs on the subject exist [2–4].

Since the final results of our trajectory studies on absolute cross sections [1] were to be tested by statistical methods for deviations from the theoretically expected behavior, the sampled distributions of the initial values also had to be tested (by χ^2

0021-9991/78/0263–0297$02.00/0

tests). Very poor test results were found when RANDU was used, far outside what could be tolerated. Tests directly on the generator also showed poor behavior.

In Monte Carlo trajectory work, congruential multiplicative and "mixed" [2–4] generators are used almost exclusively. Thus, it was considered very important to investigate whether other generators of this type behaved poorly during our tests, and to attempt to find the reasons for the poor test results. RANDU was not considered a good candidate for closer investigation, since no previous trajectory work using RANDU was known. It seemed reasonable that for one certain application (i.e., one special "test") one special type of generator would give the best performance, in this case probably the congruential type, as judged from its extensive use. Thus another congruential generator, more often used in trajectory work and possibly also well tested, was sought.

A nonexhaustive literature study of Monte Carlo trajectory studies showed that in most cases no explicit description of the generators used was given, but only inconclusive references. In some cases the ultimate sources appeared to be one of Refs. [6, 11, 12]. Since only Ref. [6] contained real tests, e.g., frequency tests, one of the "satisfactory" generators from Ref. [6] ($A_0 = 1, X = 1003, M = 10^8$; Generator 1 (Table I)) was chosen for continued work.

This generator was found to be better than RANDU, but the distributions were still far from sufficiently random for our testing purposes (by simple χ^2 tests). A study of the generator and the sampling procedure was then started. It was found that the period of the sequence was so short (5×10^6) that more than one period was used. Even after a strong reduction of the length of the number sequence used, a distinct nonrandom behavior remained in the tests. The results of these tests, χ^2 tests of the distributions of frequency tests, are collected in Table I, where a generator with a much longer period (Generator 2; $M = 2^{31} - 1$, a prime, $X = 7^5$ and $A_0 = 2147483$ [7]) is included for comparison. See also the Appendix.

Detailed studies of the generator revealed an unexpected structure in the number sequence, in the form of a steady repetition of differences between numbers located far apart in the sequence. Frequency tests applied to different fractions of the loop and with several levels of resolution in the numbers have given very poor test results (probability less than 0.0005 for randomness in 4 tests out of 11). The relation between these test results and the fine structure has been clarified. The results indicate that most common congruential generators may give nonrandom results outside the acceptable limit for distribution sampling, if applied without consideration of the limits on resolution and fraction of the loop to be used. These limitations on the use of RNGs have, as far as we know, not been described previously.

In an early report, a fine structure similar to the one discussed here was found in "mixed" generators with $M = 2^n$ [13]. It has also been observed [14, 15] that the digits in numbers formed by generators with modulus 2^n have short periods, as proved in [3]. Tests on the separate digits in the numbers have shown nonrandom behavior [14]. However, the implications of the short periods of the separate digits for the uniformity of the pseudorandom numbers was not reported. Marsaglia [8] has described a block structure, giving an effective period often considerably smaller

TABLE I

The Probability $P(\chi_t^2)$ at Which the Distributions of Goodness-of-Fit χ^2 Values Are Found in a Comparison with a Random Distribution[a]

Initial parameters	No. of classes	Generator 1			Generator 2
		1.5 × period	1.0 × period[b]	0.24 × period	6×10^{-4} × period
Rectangular distributions					
Vibrational phase	20	0.99 S, R	0.97 S	0.53	0.36 R
Polar orientation angle	20	1.00	0.86	0.93 S	0.70
	12	0.28	0.04	0.72 S	0.93
Angular velocity vector orientation angle	20	0.91	0.63	0.73	0.79
	12	1.00 S	0.99 S	0.72 S	0.58
Other distributions					
Impact parameter	12	0.12	0.04	0.75	0.82 S
Azimuth orientation angle	20	0.94	0.83	0.48	0.29

[a] For each generator, 1.2×10^6 numbers evenly distributed over the region were used (however, see footnote *b*). R (too rough) and S (too smooth) distributions of pseudorandom numbers indicate a failure at a 90% confidence level for a χ^2 test applied to two classes, either P = 0.0–0.5 and 0.5–1.0 or P = 0.0–0.1 and 0.1–1.0 of the distribution of primary χ^2 values. See also the Appendix.

[b] 1.0 × period run is part of 1.5 × period run.

than the period of the loop. This block structure is a large structure in the sequence, inside which the regularities described here develop. The "waves" in [16] and the results in [17] concern *n*-tuples and have no direct relation to the results here. Previous correlation studies concern only very short range correlation, and have not detected the long-range fine structure. The uniformity tests applied to random number generators have also failed, apparently since too few numbers [5, 6] or unsuitable procedures have been used.

2. Results of Frequency Tests

The procedure used to test the uniformity of Generator 1 within fractions of the period is described here for one specific example (see Fig. 1a). A frequency test has been applied to each hundredth of the period (5×10^4 numbers) using 10^4 classes (resolution 10^{-4} in the numbers). The first 50 values of the resulting χ^2 values are identical with the last 50 values, since the effective period [8] is equal to 2.5×10^6. Thus, only the values from the first effective period are used. The 50 χ^2 values are transformed to probabilities and plotted in a histogram. A final χ^2 test of this distribution (Fig. 1a) against a random distribution gives a value of χ_f^2, which shows that the distribution of χ^2 values from the uniformity tests is random with a probability

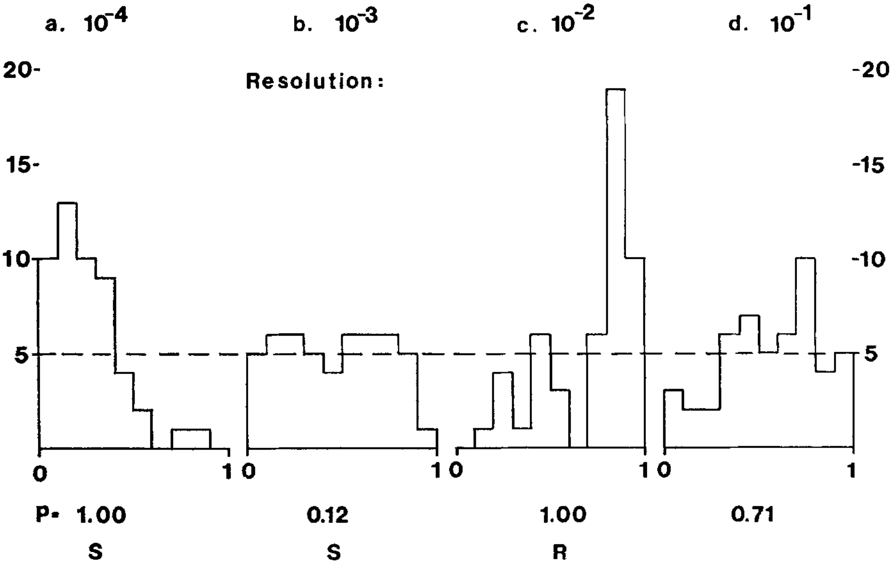

FIG. 1. Distributions of probability $P(\chi^2)$ values from frequency tests on Generator 1, using variable resolution (10^{-1}–10^{-4}) in the tests with the same sample size (10^{-2} of period $= 5 \times 10^4$), i.e., 50 tests in one effective period. p is the probability of finding a value of χ^2 less than that found from the distribution in the figure, if the distribution was random. S indicates too smooth, R too rough distributions of pseudorandom numbers, with the same criterion as in Table I.

<0.0005. Thus, the pseudorandom numbers in this case do not behave like a truly random number sequence.

In Fig. 1, a few more distributions of the same sample size but for different resolutions in the numbers are shown. In Fig. 2, the results of a larger number of tests are summarized. High test values, indicating nonrandom distributions of the numbers, as well as more normal values, are found. Too smooth or too rough distributions are also indicated in Fig. 2.

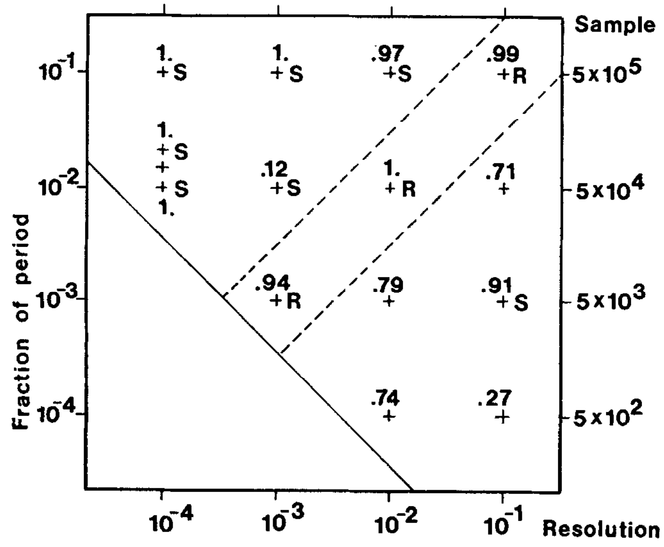

FIG. 2. Probabilities for randomness of the distributions in Fig. 1 and other similar distributions, as a function of resolution and fraction of period used in Generator 1. In the low left-hand corner the number in each class in the frequency test is too low for a simple χ^2 test. In the top row χ_t^2 tests have not been possible because of too few values of χ^2, but probabilities from a binomial distribution are given. For the other distributions, S and R indicate too smooth and too rough distributions of the pseudorandom numbers, using the same criterion as in Table I.

3. DISCUSSION

Figures 3 and 4 demonstrate a few remarkable features of Generator 1.

1. A fine structure in the random number sequence (with period 5×10^6) exists, such that the differences $A_{k+500} - A_k$ and $A_k - A_{k-500}$ are identical, independent of location within the loop of the generator.

2.1. $A_k \bmod 10^4 = A_{k-500} \bmod 10^4$. Thus, only 500 different values of $A_k \bmod 10^4$ exist in the entire loop, i.e., only a fraction $500/10^4$ (loop length/M) of the possible values.

Fraction	No	0	1	2	3	4
	0	1	1003	1006009	9027027	54108081
	Δ	61110000	93330000	9990000	19970000	29910000
1×10^{-4}	500	61110001	93331003	10996009	28997027	84018081
	Δ	-38890000	-6670000	9990000	19970000	-70090000
2×10^{-4}	1000	22220001	86661003	20986009	48967027	13928081
	Δ	61110000	-6670000	9990000	19970000	29910000
3×10^{-4}	1500	83330001	79991003	30976009	68937027	43838081
	Δ	-38890000	-6670000	9990000	19970000	29910000
4×10^{-4}	2000	44440001	73321003	40966009	88907027	73748081
	Δ	-38890000	-6670000	9990000	-80030000	-70090000
5×10^{-4}	2500	5550001	66651003	50956009	8877027	3658081
	Δ	61110000	-6670000	9990000	19970000	29910000
6×10^{-4}	3000	66660001	59981003	60946009	28847027	33568081
	Δ	-38890000	-6670000	9990000	19970000	29910000
7×10^{-4}	3500	27770001	53311003	70936009	48817027	63478081
	Δ	61110000	-6670000	9990000	19970000	29910000
8×10^{-4}	4000	88880001	46641003	80926009	68787027	93388081
	Δ	-38890000	-6670000	9990000	19970000	-70090000
9×10^{-4}	4500	49990001	39971003	90916009	88757027	23298081
	Δ	-38890000	-6670000	-90910000	-80030000	29910000
1×10^{-3}	5000	11100001	33301003	906009	8727027	53208081

FIG. 3. Numbers A_k from different parts of the first 10^{-3} fraction of the loop of Generator 1, and differences $\Delta = A_k - A_{k-500}$ between them. In each row, numbers A_k up to A_{k+4} are given. Note that, e.g., in the first column $61110000 = 10^8 - 38890000$.

2.2. The whole 5×10^6 loop is generated by a group of 500 numbers by repeated addition of 500 different numbers with four trailing zeros.

3. Complete uniformity of the digit 10^n exists in a fraction of 10^{n-7} of the loop, $n \geqslant 4$. All digits $10^n, n \geqslant 5$, are "pseudouniformly" distributed in a fraction of 10^{m-7} of the loop, $m < n$, due to the constant differences $(A_k - A_{k-500})$.

4. Each one of the small groups of 500 numbers *does not* have to be completely uniform, since the digit 10^4 in $(A_k - A_{k-500})$ never contains a factor of 10 (never even or 5).

The first result in (3) is related to a proof in [3, p. 12], which shows that $A_k \bmod 10^n$ has a period of 10^n or less, since $10^n, n \leqslant 7$, is a factor of M.

The results in Figs. 1 and 2 may be understood in the following way. The integers $A_k/10^4$ are distributed too smoothly in fractions 10^{-2} and 10^{-1} of the loop, since several groups of the basic block of 500 numbers are studied and the digit 10^0 in the integers is uniformly distributed in 10^{-3} of the loop. In the case of integers $A_k/10^5$ the basic block consists of 5000 numbers, which is 10^{-3} of the loop. Complete uniformity of the

Fraction	No	0	1	2	3	4
	0	1	1003	1006009	9027027	54108081
Δ		*11100000*	*33300000*	*-100000*	*-300000*	*-900000*
1×10^{-3}	5000	11100001	33301003	906009	8727027	53208081
Δ		*11100000*	*33300000*	*-100000*	*-300000*	*-900000*
2×10^{-3}	10000	22200001	66601003	806009	8427027	52308081
Δ		*11100000*	*33300000*	*-100000*	*-300000*	*-900000*
3×10^{-3}	15000	33300001	99901003	706009	8127027	51408081
Δ		*11100000*	*-66700000*	*-100000*	*-300000*	*-900000*
4×10^{-3}	20000	44400001	33201003	606009	7827027	50508081
Δ		*11100000*	*33300000*	*-100000*	*-300000*	*-900000*
5×10^{-3}	25000	55500001	66501003	506009	7527027	49608081

	0	1	1003	1006009	9027027	54108081
Δ		*11000000*	*33000000*	*-1000000*	*-3000000*	*-9000000*
1×10^{-2}	50000	11000001	33001003	6009	6027027	45108081
Δ		*11000000*	*33000000*	*99000000*	*-3000000*	*-9000000*
2×10^{-2}	100000	22000001	66001003	99006009	3027027	36108081
Δ		*11000000*	*33000000*	*-1000000*	*-3000000*	*-9000000*
3×10^{-2}	150000	33000001	99001003	98006009	27027	27108081
Δ		*11000000*	*-67000000*	*-1000000*	*97000000*	*-9000000*
4×10^{-2}	200000	44000001	32001003	97006009	97027027	18108081
Δ		*11000000*	*33000000*	*-1000000*	*-3000000*	*-9000000*
5×10^{-2}	250000	55000001	65001003	96006009	94027027	9108081
Δ		*11000000*	*32000000*	*-1000000*	*-3000000*	*-9000000*
6×10^{-2}	300000	66000001	98001003	95006009	91027027	108081
Δ		*11000000*	*-67000000*	*-1000000*	*-3000000*	*91000000*
7×10^{-2}	350000	77000001	31001003	94006009	88027027	91108081
Δ		*11000000*	*33000000*	*-1000000*	*-3000000*	*-9000000*
8×10^{-2}	400000	88000001	64001003	93006009	85027027	82108081
Δ		*11000000*	*33000000*	*-1000000*	*-3000000*	*-9000000*
9×10^{-2}	450000	99000001	97001003	92006009	82027027	73108081
Δ		*-89000000*	*-67000000*	*-1000000*	*-3000000*	*-9000000*
1×10^{-1}	500000	10000001	30001003	91006009	79027027	64108081

FIG. 4. Numbers A_k and differences $A_k - A_{k-5000}$ (top) from the first 10^{-2} fraction of the loop; numbers A_k and differences $A_k - A_{k-50000}$ (bottom) from the first tenth of the loop. See also Fig. 3. Note, that, e.g., $A_k - A_{k-5000} = 10 \times (A_k - A_{k-500}) \bmod 10^8$.

digit 10^0 in this case is reached only in 10^{-2} of the loop. No strict conclusion is possible in this case for a sample of 10^{-3} of the loop, but too rough distributions are found. If still smaller fractions of the loop are used, which has been possible here for smaller resolutions (integers $A_k/10^6$ and $A_k/10^7$), approximately random distributions are found. Thus, it is only in these cases that the expected pseudorandom nature of the number sequence is borne out.

The results in Table I for Generator 1 can now be discussed. This case is much more complicated than the other tests described here. The sampling for each parameter was made in 1.7×10^4 different locations in the sequence and employed in the mean 3×10^{-2} of the loop with a resolution of 10^{-1}–5×10^{-2}. Since numbers are sampled over the whole period or a large part of it, the results are expected to fall into the "too smooth" region in Fig. 2. As seen in Table I, this is generally correct. In the cases where neither too smooth nor too rough distributions have been found, the "distributed" sampling procedure has often given many tests with similar values of χ^2, thus the high probability limits in many such cases.

The generators in [6], the generator used in [11] and described in [9, 10], and the generator RANDU all have structures of the type described here, since the modulus is 2^n or 10^n. Thus, they all must be used taking into account the restrictions in resolution and length of sequence. The method proposed in [9–11], choosing a new A_0 to avoid repetition, will not improve the sequence. All these generators have several loops. The mixed generator in [12] appears to have only one loop. However, the same structure will be found since the modulus is 2^{35}.

4. Conclusions

The results from the frequency tests show that for a resolution in the pseudorandom numbers of 10^{-n} only a fraction of the period considerably smaller than 10^{-n} must be used. The shuffling methods used to destroy the short-range order in random number sequences [3, 5, 18] do not appear promising for improving the randomness for larger fractions of the period, owing to the large amount of numbers that must be stored if the long-range pattern is to be destroyed. A "mixed" generator is believed to behave in the same way as the "multiplicative" type tested here (cf. [8, 13]) even if the corresponding pattern may be more difficult to find. An acceptable procedure is to choose a generator with a very long period and with many random digits in the numbers, i.e., with M large and with loop length as close to M as possible [3, 18].

It also appears necessary to choose M to be a prime [7, 19, 20], to avoid all problems associated with short periods of the less significant digits. The more well behaved Generator 2 is of this type. However, testing of the final random quantities appears to be necessary in all critical applications.

In Monte Carlo trajectory work, the use of generators with modulus prime, which are not expected to show any structure of the type reported here, does not seem to be common (no references were found). It is possible that the "rough" distributions sometimes found in the final distributions have been caused at least partly by poor randomness in the initial values sampled. In our trajectory work [1], there were some indications in this direction, but because of the complex calculations, no strict conclusion has been possible. The results of tests on the initial values for trajectory work collected in Table 1 demonstrate that even if the pseudorandom numbers are used in "short, independent bursts of fluctuating length" [10] as in Monte Carlo trajectory work, the results may be sensitive to the detailed nature of the generator. Thus, careful statistical analysis of initial parameter values and a choice of generators with modulus prime [7], also in other respects well tested, are recommended for such work.

APPENDIX: Realistic Tests of Distribution Sampling

The results from tests of sampled distributions of initial parameter values for a Monte Carlo trajectory study of chemical reactive scattering [1] are collected in Table I. Vibrational and rotational number distributions are omitted because of their special shape. The last distribution in the table was found by rejection sampling [4, 9, 10], and required several random numbers for each parameter value. The values of χ^2 were found from comparison with the true distributions. Each value of χ_f^2 used 50 values of χ^2; each of those was computed from a sample of about 340 parameter values. Both high and low values of χ_f^2 and the corresponding probabilities are suspect. The distributions of χ_f^2 for all three runs of Generator 1 are clearly nonrandom, while the run of Generator 2 does not depart significantly from randomness. (Generator 2 is not expected to show a fine structure of the type described here; Table I shows only that it is possible to find random distributions in this test by a good choice of number sequence.) The very different behaviors of the cases using Generator 1, where two different resolutions (12 and 20 classes) have been employed, should be noted.

Acknowledgments

We would like to thank the reviewers of this paper for referring us to some interesting work on RNGs, especially the work of Marsaglia [8].

References

1. L. Holmlid and K. Rynefors, *Chem. Phys.* **14** (1976), 403.
2. B. Jansson, "Random Number Generators," pp. 39–55, Victor Pettersons Bokindustri AB, Stockholm, 1966,
3. D. E. Knuth, "Seminumerical Algorithms," Vol. 2 of "The Art of Computer Programming," Addison–Wesley, Reading, Mass., 1969.
4. T. G. Lewis, "Distribution Sampling for Computer Simulation," Lexington Books, Lexington, Mass., 1975.
5. M. D. MacLaren and G. Marsaglia, *J. Assoc. Comput. Mach.* **12** (1965), 83.
6. A. Van Gelder, *J. Assoc. Comput. Mach.* **14** (1967), 785.
7. P. A. W. Lewis, A. S. Goodman, and J. M. Miller, *IBM Systems J.* **8** (1969), 136.

8. G. Marsaglia, *in* "Applications of Number Theory to Numerical Analysis" (S. K. Zaremba, Ed.), Academic Press, New York, 1972.
9. D. L. Bunker, *in* "Molecular Beams and Reaction Kinetics" (Ch. Schlier, Ed.), p. 362, Academic Press, New York, 1970.
10. D. L. Bunker, *in* "Methods in Computational Physics" (B. Alder, S. Fernbach, and M. Rotenberg, Eds.), Academic Press, New York, 1972.
11. D. Bunker, *J. Chem. Phys.* **37** (1962), 393.
12. M. Karplus, R. N. Porter, and R. D. Sharma, *J. Chem. Phys.* **43** (1965), 3259.
13. P. Peach, *J. Amer. Statist. Assoc.* **56** (1961), 295.
14. O. Taussky and J. Todd, *in* "Symposium on Monte Carlo Methods," University of Florida (H. A. Meyer, Ed.), Wiley, New York, 1956.
15. V. D. Barnett, *Math. Comp.* **16** (1962), 63.
16. R. R. Coveyou and R. D. MacPherson, *J. Assoc. Comput. Mach.* **14** (1967), 100.
17. G. Marsaglia, *Proc. Nat. Acad. Sci. U.S.A.* **61** (1968), 25.
18. G. P. Learmonth and P. A. W. Lewis, Research Report NPS 55LW 73111A, Naval Postgraduate School, Monterey, Calif., 1973.
19. C. S. Smith, *J. Assoc. Comput. Mach.* **18** (1971), 586.
20. W. H. Payne, J. R. Rabung, and T. Bogyo, *Comm. ACM* **12** (1969), 85.