JOURNAL OF COMPUTATIONAL PHYSICS 24, 450–454 (1977)

# Note

## Correlational Defects in the Standard IBM 360 Random Number Generator and the Classical Ideal Gas Correlation Function

RÜDIGER AMADORI

*Institut für Festkörperforschung, Kernforschungsanlage Jülich, Jülich, Germany**

Received September 8, 1976; revised December 14, 1976

In a previous note [1] with the above title, Coldwell demonstrated the manifestation of a defect in a special random number generator and discussed methods [2] to circumvent this defect. The purpose of this note is to draw attention to other devices which may serve for the same intent.

In [1], particle configurations had been generated by taking successive pseudorandom numbers as particle coordinates. With respect to the two-body correlation function, configurations generated in this way by RANDU turned out to behave more like a crystal than like an ideal gas. This behavior might have been expected from Marsaglia's theorems [3, 4] in cases where the number of hyperplanes is far from the upper limit, even if each pseudorandom number is used only once as a particle coordinate as done in [1], whereas in [3, 4], each pseudorandom number is used $n$ times where $n$ is the dimension of the problem.

In the following, three different approaches will be discussed to avoid harmful correlations introduced by the pseudorandom number generator in cases like that presented in [1]: (i) taking no additional precautions but minimizing correlations, (ii) destroying correlations by additional manipulations, (iii) using a generator producing points uniformly distributed in $n$-space.

(i) Since one cannot get rid of the correlations if particle coordinates are formed from successive pseudorandom numbers of a generator of congruential type, one might try to make these correlations as small as possible. In the notation of [5], "oscillations" should be as rapid as possible, and exact prescriptions to this end concerning the choice of the multiplier value are given for three-space. Another measure of independence of pairs, triples,... of pseudorandom numbers is given by the spectral test [6, 7]. In the notation of [7], quantities $C_n$ ($n = 2, 3, 4,...$) represent such a measure. If, for a particular generator, the $C_n$ value is close to the theoretical upper limit ($C_2 = 3.63$, $C_3 = 5.92$, $C_4 = 9.87$), then maximum independence (minimum correlation) of $n$-tuples has been achieved. There is no general prescription on how to choose multipliers in order to attain these maximum $C_n$ values. According

* Now with Universität Dortmund, Rechenzentrum, D-4600 Dortmund 50, Germany.





ISSN 0021-9991

to [7], a generator has passed the spectral test, if $C_n > 0.1$, and passed the test with flying colors, if $C_n > 1.0$ ($n = 2, 3, 4,...$). The extremely small value of $C_3 = 2.0 * 10^{-5}$ of the original version of RANDU as opposed to the good $C_2 = 1.96$ value indicates extreme correlations among triples generated by RANDU, as found in [1] and elsewhere. Multiplier value 65549 suggested in [8] to improve high order bit statistics is somewhat better in $C_3 = 3.9 * 10^{-2}$ (retaining a good $C_2 = 3.13$), but does not pass the spectral test according to [7], and in [5], this value is shown to "produce sufficiently rapid oscillations for some purposes." This was found to be true in the test represented by the $\chi^2$ value of the configurations in [1], see Table I, in which relevant information from Table I of [1] is reproduced and augmented by the results obtained with other generators and methods, along with additional information.

The spectral test is known to be one of the most stringent tests for pseudorandom number generators of the congruential type. Although generally applicable, the $\chi^2$ value of configurations in [1] as a special test is much less sensitive to correlations if these are not extremely bad. This is demonstrated by the fact that only the $\chi^2$ value of the configurations generated by the original RANDU is undoubtfully bad. For other multiplier values or methods, meaningful conclusions cannot be drawn from the single $\chi^2$ value given in [1], in spite of the fact that 1000 configurations contribute to that value. This may easily be seen by using different initial values for complete runs of 1000 configurations. Thirty complete runs have been done by the author of this note, the range of $\chi^2$ values encountered is given in Table I, and serves *only* to demonstrate this heavy dependence on the initial value.

(ii) To destroy correlations across dimensions arising if successive pseudorandom numbers of the same generator are taken as particle coordinates, the following method [2] had been tested in [1]: Particle coordinates are still formed from successive pseudorandom numbers, but these numbers are generated by first filling a table of numbers produced by RANDU and then picking numbers from this table with 65549 substituted for the multiplier value. To make coordinates even more random, a two-table fill had also been tested. Since multiplier values different from the original value 65539 had been chosen in the table pick, one might ask what kind of results would have emerged if these multiplier values had been used in the first place? The answer is that in the particular example presented in [1], the correlations would have been invisible.

Using successive pseudorandom numbers as coordinates of the same particle is simple and seems to be "natural," but nobody is forced to do this. Decoupling the coordinates by using separate calling streams with different initial values for each of the coordinates is an alternative method which does not increase the amount of coding or cpu-time required, as opposed to the methods discussed in [1]. Although an unfortunate selection of the initial values (such that one of the initial values is a predecessor or successor of a few steps of one of the other initial values) may introduce undesirable correlations, this was not encountered in the 30 complete runs conducted by the author. The problem of choosing "correct" initial values avoiding overlap in the generated sequences is considered in [9] and requires solution of a

TABLE I

Range of $\chi^2$ Values of Two-Body Correlation Functions for a Classical Ideal Gas of 32 Particles as Obtained by Simulation with Different Pseudorandom Number Generators in 30 Runs of 1000 Configurations According to [1]

| Generator name | RANDU | RANDU | RANDU | RANDU with table fill | RANDU with two-table fill | RANDU, multiple calling streams | Seraphin | GGL | Tausworthe Kendall | GFSR |
|---|---|---|---|---|---|---|---|---|---|---|
| cpu-time[a] | 0.33 | 0.33 | 0.33 | 1.37 | 2.57 | 0.33 | 0.16 | 1.50[c]<br>0.77[d]<br>GGL1:0.19[e]<br>GGL2:0.18[e] | 0.22[e] | 0.73<br>0.29[e] |
| Modulus | $2^{31}$ | $2^{31}$ | $2^{31}$ | $2^{31}$ | $2^{31}$ | $2^{31}$ | $2^{32}$ | $2^{31}-1$ | ($N=31, M=6$) | ($p=98, q=27$, delay = 9800) |
| Multiplier(s) | 65539 | 65549 | 331804469 | 65549<br>65539 | 65549<br>65525<br>65539 | 65539 | 32781 | 16807 | — | — |
| Reference | [8] | [8] | [17] | [1] | [1] | — | [10] | [18] | [14] | [16] |
| $\chi^2$ range[b] | 1189–1583 | 18–50 | 20–47 | 23–45 | 17–49 | 20–52 | 16–49 | 21–47 | 24–49 | 15–53 |

[a] In seconds, for generation of 93,000 pseudorandom floating point numbers in (0, 1) interval on IBM/370-168, MVT, FORTRAN H EXTENDED, OPT(2).

[b] Initial values taken from successive values produced by GGL generator with initial value 123456789.

[c] Original version.

[d] Upgraded by author.

[e] Assembler coded.

congruence equation to determine the relative shift of sequences with different initial values.

Using calling streams with different initial values for the $x$, $y$, and $z$ coordinate of course results in correlation of *each* of the three coordinates of successive particles, within a configuration and across configurations in the example of [1], if no new initial values are chosen for each configuration. But this correlation is restricted to one dimension, namely, $x$ coordinates of successive particles are correlated according to $x_3 = (6x_2 - 9x_1) \bmod(1)$ [5] in the case of RANDU, and the same holds for the $y$ and $z$ coordinates. These correlations of each of the particle coordinates are harmless, since independent from each other (as long as the initial values are not predecessors or successors of a few steps of each other), contrary to the correlations across dimensions if one calling stream is used and successive pseudorandom numbers taken as coordinates of the *same* particle, resulting in the arrangement of all particles in planes.

Using a different multiplier value in the separate calling streams as suggested by one of the referees will destroy correlations even more thoroughly, however, choosing alternate multiplier values may be hazardous if performance of these multiplier values is unknown and could be obtained only by additional extensive tests.

The Seraphin generator [10] is an extremely fast generator. Although originally published in assembler language, it may easily be implemented in FORTRAN. Furthermore, values of floating point pseudorandom numbers produced by the Seraphin generator are related to the corresponding integer values in an unconventional way due to the outstanding feat of the Seraphin generator for floating and transforming integer numbers to the unit interval in a single machine instruction.

(iii) It should be kept in mind that all methods discussed so far only minimize or destroy correlations, but do not guarantee $n$-dimensional uniformity of pseudorandom points generated by these methods. A generator guaranteeing $n$-dimensional uniformity of pseudorandom $n$-tuples is the Tausworthe generator [11]. However, as in the case of congruential generators where the multiplier value has to be chosen carefully to avoid harmful correlations, choosing "good" feedback tabs of the Tausworthe generator is of vital importance [12, 13]. Coding implementations of the Tausworthe generator are given in [14–16]. Although only one-dimensional uniformity is predicted in using all 31 bits for a single number in the Kendall algorithm [14] with parameters as given in Table I, in this particular example, the effects of possible nonuniformity are inevident.

## Conclusions

Avoiding harmful correlations introduced by a pseudorandom number generator may be done by various methods. These methods may be rated according to the cpu-time required and the performance in different tests. The test represented by the $\chi^2$ value of configurations discussed in [1] is applicable only if the performances of the various methods are differing by orders of magnitude, due to the heavy dependence of the $\chi^2$ value on the initial value in the particular example given in [1].

## ACKNOWLEDGMENTS

The author would like to thank G. Schulte and U. Lipeck for supplying the spectral test results.

## REFERENCES

1. R. L. COLDWELL, *J. Computational Phys.* **14** (1974), 223.
2. M. D. MACLAREN AND G. MARSAGLIA, *J. Assoc. Comput. Mach.* **12** (1965), 83.
3. G. MARSAGLIA, *Proc. Nat. Acad. Sci.* **61** (1968), 25.
4. G. MARSAGLIA, *Numer. Math.* **16** (1970), 8.
5. F. A. BLOOD, *J. Computational Phys.* **20** (1976), 372.
6. R. R. COVEYOU AND R. D. MACPHERSON, *J. Assoc. Comput. Mach.* **14** (1967), 100.
7. D. E. KNUTH, "The Art of Computer Programming," Vol. 2, Chap. 3, Addison–Wesley, Reading, Mass., 1968.
8. IBM Corp. System/360 Scientific Subroutine Package, H20-0205-3, New York, 1968.
9. G. CENACCHI AND A. DE MATTEIS, *Numer. Math.* **16** (1970), 11.
10. D. S. SERAPHIN, *Comm. ACM* **12** (1969), 695.
11. R. C. TAUSWORTHE, *Math. Comp.* **19** (1965), 201.
12. B. M. FELLEN, *Comm. ACM* **12** (1969), 413.
13. J. P. R. TOOTILL *et al.*, *J. Assoc. Comput. Mach.* **18** (1971), 381.
14. J. R. B. WHITTLESEY, *Comm. ACM* **11** (1968), 641.
15. W. H. PAYNE, *Comm. ACM* **13** (1970), 57.
16. T. G. LEWIS AND W. H. PAYNE, *J. Assoc. Comput. Mach.* **20** (1973), 456.
17. J. H. AHRENS, U. DIETER, AND A. GRUBE, *Computing* **6** (1970), 121.
18. IBM Corp. System/370 Subroutine Library Mathematics, SH12-5300-0, New York, 1971.