Comment on ''Algorithm for normal random numbers''

M. I. J. Probert*
Cavendish Laboratory, University of Cambridge, Madingley Road, Cambridge CB3 0HE, United Kingdom
(Received 11 August 2000; published 9 April 2001)

We have performed some simple statistical tests on the recently proposed algorithm for generating normally distributed random numbers and we find that it has some serious flaws with regard to practical application. In particular, we find that there is a statistically significant difference between the sample mean and that expected when using the original authors' recommended parameters of the algorithm. Only by significantly increasing these parameters, at the cost of considerable extra computational effort, do we get satisfactory statistics.

DOI: 10.1103/PhysRevE.63.058701 PACS number(s): 02.70.Rr, 02.90.+p

I. INTRODUCTION

In a recent paper, Fernández and Criado [1] published a new algorithm for generating normally distributed random numbers, which they claimed was considerably simpler and faster than the traditional Box-Muller algorithm [2].

The algorithm simulates N particles exchanging energy using a simple stochastic rule, and the corresponding velocities are used as a source of pseudorandom numbers. They provide a proof that, for any initial state of these N particles, the system is ergodic, and in the $N\rightarrow\infty$ limit it will tend to the Maxwell velocity distribution. In order to achieve this, the system must first be equilibrated by n_p passes through all N particles to remove any dependency on this initial state.

The authors state that ''$N\approx 10^4$ is sufficient in order to generate as many as 10^{15} pseudorandom numbers, with a roughly 10% error in the probability of the largest number in the sequence.'' and ''Deviations from equilibrium are statistically insignificant for $n_p\geqslant 2$ and $N=1024$ and $n_p\geqslant 4$ and $N=1\,048\,576$. Since n_p is expected to increase as $\ln N$, $n_p=8$ should provide ample warm up for any foreseeable applications.'' We have performed some simple tests of this new algorithm, and while we certainly agree that as presented it is faster than the Box-Muller algorithm and simpler to code, we do have some serious reservations about its use in practice that we wish to highlight here. In particular, we disagree with these two highlighted statements, which therefore raises basic questions about the applicability of this algorithm for general purpose usage.

II. METHOD

We implemented the algorithm as described [1], with the ''minimal standard'' uniform pseudorandom number generator of Park and Miller [3] combined with an additional Marsaglia shift [4]. This combined generator has been shown to be a very reliable generator of pseudorandom numbers uniformly distributed on [0,1].

We then calculated the sample mean $\overline{x}$ and sample variance s^2 of $N_{\rm rand}$ pseudorandom numbers x_i drawn from the standard normal distribution $N(0,1)$, i.e., with population mean $\mu=0$ and population variance $\sigma^2=1$. We compared the results to those generated by the standard Box-Muller technique and corresponding exact answers. We used the following standard statistics [5].

The expected sample mean is

$$\overline{x}=\mu=\frac{\sum x_i}{N_{\rm rand}},$$

and the expected variance of the sample mean is

$$V[\overline{x}]=\frac{\sigma^2}{N_{\rm rand}}.$$

Similarly, the expected sample variance is

$$s^2=\frac{N_{\rm rand}-1}{N_{\rm rand}}\sigma^2=\frac{\sum (x_i-\overline{x})^2}{N_{\rm rand}-1},$$

and the expected variance of the sample variance is

$$V[s^2]=\frac{1}{N_{\rm rand}}\left\{\frac{\sum (x_i-\overline{x})^4}{N_{\rm rand}}-\frac{N_{\rm rand}-3}{N_{\rm rand}-1}\left[\frac{\sum (x_i-\overline{x})^2}{N_{\rm rand}}\right]^2\right\}.$$

For the normal distribution $N(0,1)$ we have the following exact analytic results:

$$V[\overline{x}]=\frac{1}{N_{\rm rand}},$$

$$V[s^2]\sim\frac{2}{N_{\rm rand}},N_{\rm rand}\gg 1.$$

Our test procedure was as follows

(1) Initialize the uniform pseudorandom number generator, with a seed based upon the current system clock.

(2) ''Warm up'' the normal pseudorandom number generator, with n_p passes through all N particles.

(3) Generate $100*N_{\rm rand}$ pseudorandom normal deviates and calculate the above statistics for each block of $N_{\rm rand}$ numbers.

(4) We then test each block for statistically significant differences, by doing a hypothesis test at the 99% confidence

*Author to whom correspondence should be addressed

TABLE I. Failure rates at 99% confidence level for sample mean and variance of the pseudorandom number generator, calculated from 100 successive blocks of 10^6 numbers. The expected failure rate is 1%.

N	n_p	Failure rate of mean	Failure rate of variance
1024	2	33%	0%
1024	4	32%	0%
1024	8	35%	0%
1024	16	31%	0%
65536	2	24%	0%
65536	4	24%	0%
65536	8	26%	0%
65536	16	20%	0%
1048576	2	21%	3%
1048576	4	14%	0%
1048576	8	6%	0%
1048576	16	4%	0%

level to see if the sample mean is significantly different from the population mean, i.e., is $|\mu-\overline{x}|>2.575\sqrt{V[\overline{x}]}$? If so, then flag this as a significant difference. With 100 independent blocks of numbers and therefore 100 such tests, we can reasonably expect one failure.

(5) Repeat the hypothesis test for sample variance, i.e., is $|\sigma-s|>2.575\sqrt{V[s^2]}$? Again, with 100 such tests, we can reasonably expect one failure.

III. RESULTS

As we were only generating blocks of $N_{\text{rand}}=10^6$ pseudorandom numbers, we expected good results with $N=1024$ and $n_p=2$ given the statements made in the original paper [1]. This was not the case. We then extended the tests to consider other, larger values of N and n_p, and calculated the hypothesis-test failure rates at the 99% confidence level. We summarize the observed failure rates in Table I.

We also repeated this procedure to test the standard Box-Muller generator, and found that this did indeed give the expected statistics. In 100 tests, we found one failure in the mean test and no failures in the variance test at the 99% confidence level.

IV. CONCLUSIONS

The results clearly show that the recently proposed normal pseudorandom number generator requires much longer ''warm up'' times n_p than is suggested in the original paper [1]. Even with significantly extended ''warm up'' times, we still find that there are statistically significant differences in the expected behavior in the sample means, although the sample variance is behaving in an acceptable manner. This problem is reduced by going to significantly larger values of N than are to be expected on the basis of the original paper, but even here, the results are not satisfactory until considerably larger values of N and n_p are used. On the basis of the original paper, we expected satisfactory statistics for $N_{\text{rand}}=10^6$ from $N=1024$ and $n_p=2$. On the basis of the tests reported here, we find that even values of $N=1\,048\,576$ and $n_p=16$ are only marginally satisfactory.

The main advantage of the proposed method was its speed relative to the traditional Box-Muller method. We find that the need to increase n_p and N in order to get satisfactory statistics significantly reduces the speed advantage. For the case of $N=1024$ and $n_p=2$, we find that the ''warm up'' period takes 0.001 seconds and that it takes 57 seconds to calculate and test 100 blocks of $N_{\text{rand}}=10^6$ numbers. Increasing the values to $N=1\,048\,576$ and $n_p=16$, we find that the ''warm up'' period takes 38 seconds and that it now takes 115 seconds to calculate and test 100 blocks of $N_{\text{rand}}=10^6$ numbers. Note that there is an increased memory overhead required in storing the necessary data for a much larger value of N. As a comparison, using the same uniform random number generator and the traditional Box-Muller normal deviate generator, it takes 95 seconds to calculate and test 100 blocks of $N_{\text{rand}}=10^6$ numbers.

We therefore recommend that any proposed use of this normal pseudorandom number generator be approached with caution, particularly with regard to the basic parameters N and n_p of the algorithm.

ACKNOWLEDGMENTS

Financial support was provided by the Lloyd's of London Tercentenary Foundation. All calculations were performed on a Compaq Alpha XP1000 (500 MHz ev6) workstation.

[1] J.F. Fernández and C. Criado, Phys. Rev. E **60**, 3361 (1999).
[2] G.E.P. Box and M.E. Muller, Ann. Math. Stat. **29**, 610 (1958).
[3] S.K. Park and K.W. Miller, Commun. ACM **31**, 1192 (1988).
[4] G. Marsaglia, Linear Algebr. Appl. **67**, 147 (1985).
[5] G. Cowan, *Statistical Data Analysis* (Oxford University Press, Oxford, 1998), p. 68.