Reply to ''Comment on 'Algorithm for normal random numbers' ''

Julio F. Fernández
Instituto de Ciencia de Materiales de Aragón, Consejo Superior de Investigaciones Científicas and Universidad de Zaragoza, 50009-Zaragoza, Spain

Carlos Criado
Departamento de Física Aplicada I, Universidad de Málaga, 29071-Málaga, Spain
(Received 15 November 2000; published 9 April 2001)

We have recently proposed [Phys. Rev. E **60**, 3361 (1999)] an algorithm for the generation of normal pseudorandom numbers that is nearly 10 times faster than the Box-Muller algorithm. A flaw in this algorithm is pointed out in the preceding Comment [Probert, preceding Comment, Phys. Rev. E **63**, 058701 (2001)]. It turns out that significant correlations show up in strings of pseudorandom numbers generated by our algorithm if such strings are sufficiently long. A slightly modified algorithm that is free from this defect is proposed.

DOI: 10.1103/PhysRevE.63.058702 PACS number(s): 02.60.Cb

A flaw in our algorithm [1] for the generation of normal pseudorandom numbers (PRN) is reported in the preceding Comment [2]. Consider the sum

$$z=\sum_{i=1}^{M}\frac{x_i}{\sqrt{M}} \tag{1}$$

of M successively generated PRN, $x_1, x_2, \ldots, x_M$. The probability density distribution function (PDF) $p(z)$ that is obtained from our algorithm is not well behaved. Flaws in PDF distributions of some sufficiently large sums of PRN have been previously found in some well-known uniform random number generators [3].

Following Probert's preceding Comment, we have tested our algorithm by generating $p(z)$ distributions for various values of M and of the number N of molecules (registers, in a computer program) in the gas that is simulated by the algorithm. The algorithm we proposed in Ref. [1] gives $p(z)$ distributions that depend on M/N. The PDF's that are obtained are in general too wide. The standard deviation increases monotonically with M/N from the correct value for $M/N \ll 1$ up to a value that is approximately 2.5 times too large, for $M/N \gtrsim 10$. Unfortunately, this is so for arbitrarily long warming up times.

We report below a slightly modified version of our original algorithm [1], which, as far as we can tell, yields well-behaved $p(z)$ PDF's. They are free from the flaw reported in the preceding Comment. The improved algorithm is as follows:

$$\imath = U(1,N), \quad J = U_\imath(1,N), \tag{2}$$

$$R_{\mathrm{sign}} = 2U(0,1) - 1, \tag{3}$$

$$v_\imath \leftarrow R_{\mathrm{sign}}(v_\imath + v_J)/\sqrt{2}, \tag{4}$$

$$v_J \leftarrow -v_\imath + R_{\mathrm{sign}}\sqrt{2}v_J, \tag{5}$$

where $U(1,N)$ gives uniformly distributed random integers (UDRI) in the interval $[1,N]$, and $U_\imath$ gives UDRI that are different from $\imath$ in the interval $[1,N]$. The second line that defines quantity R_{sign} is new. Since $R_{\mathrm{sign}} = \pm 1$, the rotation defined by Eqs. (3) and (4) in this algorithm is either by $\pi/4$ or by $3\pi/4$. [As in Ref. [1], the updated value of $v_\imath$ from Eq. (3) is to be used in Eq. (4).]

The tests we have performed show that this algorithm meets all specifications given in Ref. [1] for the original one. In addition, Kolmogorov-Smirnov tests of the above algorithm for $N = 2^n$ and $n = 3,4,\ldots,9$ give no deviation from a Gaussian PDF for $M \lesssim 50N^2$, with 99% confidence. Finally, the above algorithm gives well-behaved $p(z)$ PDF's.

Table I shows numbers that we have obtained by making use of the algorithm given in Eqs. (3)–(5). These numbers were obtained as follows. We have generated the random values of indices $\imath$ and J in Eq. (2) and of R_{sign} in Eq. (3) with the GGL algorithm [4]. Initially, we set $v_\imath = 1$ for all $1 \leqslant \imath \leqslant N$ and generate $n_p \times 10^6$ PRN in order to warm up the algorithm, before each block of 10^6 sequential normal PRN

TABLE I. Number of times (referred to as ''failures'' below) the mean and the variance, given by our algorithm for 100 blocks of 10^6 sequentially generated numbers, deviated from their expected value by more than 2.576 times their standard deviations, for the shown number of molecules N and of warming up runs n_p. The expected number of times is 1 in both cases.

N	n_p	Failures of the mean	Failures of the variance
1 024	2	1	2
1 024	4	0	2
1 024	8	1	0
1 024	16	2	0
65 536	2	0	0
65 536	4	1	0
65 536	8	3	2
65 536	16	0	3
1 048 576	2	0	1
1 048 576	4	1	1
1 048 576	8	2	1
1 048 576	16	0	0

is generated. The starting seed for the GGL algorithm was chosen in each case to be a six digit number obtained from throwing six dice. Table I affords a comparison with the table of values given in the preceding Comment [2].

A portable FORTRAN code for Eqs. (2)–(5) may be downloaded from http://Pipe.Unizar.Es/ jff/code/rg.f or requested from one of us by electronic mail from jff@Pipe.Unizar.Es.

We are grateful to DGESIC of Spain for financial support through Grant No. PB99-0541.

[1] J.F. Fernández and C. Criado, Phys. Rev. E **60**, 3361 (1999).
[2] M. I. J. Probert, preceding Comment, Phys. Rev. E 058701 (2001).
[3] P. Grassberger, Phys. Lett. A **181**, 43 (1993); I. Vattulainen, T. Ala-Nissila, and K. Kankala, Phys. Rev. Lett. **73**, 2513 (1994).
[4] For details of the GGL algorithm $[x_{n+1}=16807x_n \mathrm{mod}(2^{31}-1)]$, see, for instance, P. Bratley, B. L. Fox, and L. E. Schrage, *A Guide to Simulation*, 2nd ed. (Springer-Verlag, New York, 1997), p. 215; A.M. Ferrenberg, D.P. Landau, and Y.J. Wong, Phys. Rev. Lett. **69**, 3382 (1992).