Applied Radiation and Isotopes 68 (2010) 469–473

Contents lists available at ScienceDirect

Applied Radiation and Isotopes

journal homepage: www.elsevier.com/locate/apradiso

Assessment of the suitability of different random number generators for Monte Carlo simulations in gamma-ray spectrometry

N. Cornejo Díaz [a], A. Vergara Gil [a], M. Jurado Vargas [b,*]

[a] Centre for Radiological Protection and Higiene, P.O. Box 6195, Habana, Cuba
[b] Physics Department, University of Extremadura, 06071 Badajoz, Spain

A R T I C L E I N F O

Article history:
Received 20 July 2009
Received in revised form
3 November 2009
Accepted 13 November 2009

Keywords:
Random number generators
Gamma-ray spectrometry
Efficiency calibration

A B S T R A C T

The Monte Carlo method has become a valuable numerical laboratory framework in which to simulate complex physical systems. It is based on the generation of pseudo-random number sequences by numerical algorithms called random generators. In this work we assessed the suitability of different well-known random number generators for the simulation of gamma-ray spectrometry systems during efficiency calibrations. The assessment was carried out in two stages. The generators considered (Delphi's linear congruential, mersenne twister, XorShift, multiplier with carry, universal virtual array, and non-periodic logistic map based generator) were first evaluated with different statistical empirical tests, including moments, correlations, uniformity, independence of terms and the DIEHARD battery of tests. In a second step, an application-specific test was conducted by implementing the generators in our Monte Carlo program DETEFF and comparing the results obtained with them. The calculations were performed with two different CPUs, for a typical HpGe detector and a water sample in Marinelli geometry, with gamma-rays between 59 and 1800 keV. For the Non-periodic Logistic Map based generator, dependence of the most significant bits was evident. This explains the bias, in excess of 5%, of the efficiency values obtained with this generator. The results of the application-specific assessment and the statistical performance of the other algorithms studied indicate their suitability for the Monte Carlo simulation of gamma-ray spectrometry systems for efficiency calculations.

1. Introduction

Gamma-ray spectrometry has become one of the most widely used non-destructive procedures to quantify the activity of radionuclides. The analysis requires the knowledge of the peak efficiency at each photon energy, which can be determined by performing an efficiency calibration using standard samples with the same geometrical dimensions, density, and chemical composition as the sample of interest. These conditions often cannot be met because it is difficult to find adequate standards for all energies of interest and radionuclides with appropriate half-lives. Therefore, the use of interpolation procedures and a continuous renewal of spent standards are inherent shortcomings of this approach. Experimental calibrations are also time-consuming, especially when many different matrices or sample-detector geometries have to be measured. In order to overcome these difficulties, a powerful tool is the Monte Carlo simulation, which allows the peak efficiencies to be calculated taking into account the detailed characteristics of detectors and samples.

The Monte Carlo method is based on the generation of pseudo-random numbers by numerical algorithms called random number generators (RNGs). The ideal RNG does not exist, because no one generator is better than others for all purposes. The "quality" of a given generator is closely related to the problem to be solved. Therefore, despite the diversity of available statistical tests, it is also desirable to evaluate each random number generator according to the specific application it will be used for.

In this paper we describe the results of the studies carried out to assess the suitability of some of the most common RNGs for Monte Carlo simulation of gamma-ray spectrometry systems during efficiency calibrations. In Section 2 we provide a brief description of the RNGs considered. Section 3 summarizes the statistical empirical tests used for the preliminary assessment of the RNGs, while its application-specific assessment is described in Section 4. The main conclusions are provided in Section 5.

2. Random number generators considered

Discussion about which is the more appropriate term, "random number generator" or "pseudo-random number generator" is beyond the scope of this work, so that, for simplicity the term

* Corresponding author. Tel.: +34 924 289 537; fax: +34 924 289 651.
E-mail address: mjv@unex.es (M. Jurado Vargas).

0969-8043/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.apradiso.2009.11.037

"random number generator" will be used throughout. Although there are many random number generators that in principle could be used in a study of this type, we will focus on some of the simpler, faster, and widely known generators that have already been applied in stochastic simulations:

(a) *Linear congruential generator* (*LC*) (Knuth, 1998), provided with the Borland Delphi package. The sequence of random numbers is obtained by setting:

$$x_{n+1} = (a \cdot x_n + c) \bmod m (n \geq 0) \quad (1)$$

with the multiplier a=134 775 813, the increment c=1, and modulus $m=2^{32}$. The period of this generator is $2^{32}-1 \approx 4.29 \times 10^9$.

(b) *Mersenne twister generator* (MT), implemented by us (Vergara Gil, 2008). The algorithm is a twisted generalized feedback shift register of rational normal form, with state bit reflection and tempering (Matsumoto and Nishimura, 1998; Matsumoto, 2002). The period is $\approx 2^{19\,937}-1$.

(c) *Multiply with carry generator* (MWC) (Marsaglia, 2003), with the general algorithm:

$$N_n = (a \cdot N_{n-1} + c_{n-1}) \bmod m; n \geq 1 \quad (2)$$

$$c_n = \left[\frac{a \cdot N_{n-1} + c_{n-1}}{m}\right] \quad (3)$$

This generator was implemented by us according to the work of Debord (2008), by combining two 16-bit MWC generators with multipliers a_1=18 000 and a_2=30 903, respectively, and modulus $m=2^{15}$ to form 32-bit numbers. The period of this generator is $(a_1 \cdot 2^{15}-1)(a_2 \cdot 2^{15}-1) \approx 6 \times 10^{17}$. The algorithm used poses the problem of an additional requirement regarding the proper selection of the sequence seeds.

(d) *XorShift generator* (XorS) (Marsaglia, 2003), with a period of $2^{32}-1 \approx 4.29 \times 10^9$ and sequence:

$$N \ \mathbf{xor} \ (N \ \mathbf{shl} \ 1), \ N \ \mathbf{xor} \ (N \ \mathbf{shr} \ 3), \ N \ \mathbf{xor} \ (N \ \mathbf{shl} \ 10) \quad (4)$$

(e) *Universal virtual array generator* (UVA), implemented by us according to Debord (2008). This generator uses the content of the computer's random access memory to generate random sequences. This generator does not allow the repetition of random sequences and has no period.

(f) *Non-periodic logistic map based generator* (LM), implemented by us according to Barberis (2007) but without discarding the first 300 numbers in every calculation, as it was suggested by the author to improve the independence of the terms in the sequence, although with an obvious cost in time. This generator is based on the recurrence:

$$x_n = r \cdot x_{n-1} \cdot (1 - x_{n-1})(r = 4) \quad (5)$$

with the substitution $y_n = 2/\pi \cdot \arcsin\sqrt{x_n}$, where x_n and y_n are real numbers in the interval (0,1). The lack of periodicity was confirmed in Barberis (2007) up to 10^{13} numbers. Because of its simplicity, this generator seemed to be promising for our purposes.

Amongst the generators studied, the LC generator is one of the oldest and more used in Monte Carlo codes for radiation transport. The RNG implemented in the well known Monte Carlo code MCNP is based on a linear congruential scheme (Brown and Nagaya, 2002). Additionally, the subroutine RANECU, written by James (1990) from the algorithm proposed by L'Ecuyer (1988) and implemented in the Monte Carlo packages PENELOPE (Salvat et al., 2006) and GEANT4 (Geant4 Home Page), is the combination of two LC generators. The MT generator is another of the RNGs included in the GEANT4 toolkit.

3. Statistical empirical tests

The RNGs were first assessed with different basic statistical empirical tests according to Knuth (1998), which evaluate the moments, the uniformity, and the independence of terms in the sequences. In a second step, each RNG was checked with the DIEHARD battery of tests (Marsaglia, 1995), which contains 15 statistical tests covering among others independence between consecutive numbers, independence between bits and between sequences of bits, and uniformity in several dimensions. The DIEHARD battery of test is also included in the tests suite called TESTU01 (L'Ecuyer and Simard, 2009), which includes more sophisticated test for parallelization problems or larger random number sequences.

The results of statistical tests were evaluated using the student, chi-square, and Kolmogorov–Smirnov tests.

The random numbers obtained with the LM generator showed non-independence in the most significant bits. Therefore, this generator failed to pass some of the basic tests such as "the sum of dice" and "the graphical correlation" (Knuth, 1998). Regarding the "sum of dice" test, if we consider the experiment of throwing two dice (each of which is assumed to yield the values 1, 2, 3, 4, 5, or 6 with equal probability) the sum of dice must be an integer number in the interval [2, 12], with 36 different combinations. For each RNG, we simulated n throws (n=100 000), using $2n$ random numbers. The statistic V was then calculated according to

$$V = \sum_{k=2}^{12} \frac{(Y_k - n \cdot p_k)^2}{n \cdot p_k} \quad (6)$$

with Y_k being the observed number of times in which the sum of dice equals k $\{k \in N, 2 \leq k \leq 12\}$ and p_k the probability of obtaining the value k. V should follow the behavior of a stochastic variable with a χ^2 distribution (for 10 degrees of freedom) if the dice were "true", i.e., if the numbers obtained in the sequence were really independent. For each RNG the experiment was carried out 10 times and the corresponding probability $p_{\chi^2}(x \leq V)_{\nu=10}$ for each value of V was determined. The Kolmogorov–Smirnov test was finally applied to the 10 values of $p_{\chi^2}(x \leq V)_{\nu=10}$ obtained for each RNG, calculating the pertinent positive ($KS+$) and negative ($KS-$) maximum deviations from a uniform distribution in the interval (0,1). Table 1 presents the probability values $P_{KS}(x \leq KS+)$ and $P_{KS}(x \leq KS-)$ given for the differences $KS+$ and $KS-$, respectively, by assuming 10 degrees of freedom.

This "sum of dice" test was repeated by considering 3, 4, 5 and 6 dice, and the results were similar to those given in Table 1. As can be seen in that table, the differences $KS+$ and $yKS-$ for the LM generator are much greater than those expected if the statistics V followed a χ^2 distribution. As a consequence of this non-independence between consecutive elements in the sequence obtained with the LM generator, the "graphical correlation" test does not show points uniformly distributed in the area of interest ((0,1) × (0,1)), but a well-defined pattern, as shown in Fig. 1. On the contrary, the application of the "graphical correlation" test to

Table 1
Results of the Kolmogorov–Smirnov test.

RNG	$P_{KS}(x \leq KS+)$	$P_{KS}(x \leq KS-)$
LC	0.377	0.727
XorS	0.647	0.276
MWC	0.673	0.307
MT	0.417	0.321
UVA	0.673	0.307
LM	1.000	1.000