



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

N. Cornejo Díaz<sup>a</sup>, A. Vergara Gil<sup>a</sup>, M. Jurado Vargas<sup>b,\*</sup>

<sup>a</sup> Centre for Radiological Protection and Higiene, P.O. Box 6195, Habana, Cuba

<sup>b</sup> Physics Department, University of Extremadura, 06071 Badajoz, Spain

### ARTICLE INFO

#### 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

### ABSTRACT

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.

© 2009 Elsevier Ltd. All rights reserved.

### 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](mailto:mjv@unex.es) (M. Jurado Vargas).

“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:

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

with the multiplier  $a=134775813$ , 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  $\approx 2^{19937} - 1$ .

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

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

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

This generator was implemented by us according to the work of Debord (2008), by combining two 16-bit MWC generators with multipliers  $a_1=18000$  and  $a_2=30903$ , 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:

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

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 ( $n=100000$ ), 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  $\chi^2$  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  $p_{\chi^2}(x \leq V)_{v=10}$  for each value of  $V$  was determined. The Kolmogorov–Smirnov test was finally applied to the 10 values of  $p_{\chi^2}(x \leq V)_{v=10}$  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  $\chi^2$  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  $((0,1) \times (0,1))$ , 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

the rest of the RNGs results in uniform distributions of the points in 2 dimensions.

Table 2 gives the results derived from applying the DIEHARD battery of tests (Marsaglia, 1995) to the RNGs considered in this work. As observed, the results of the individual tests OPSO, OQSO, and DNA show that in the random numbers obtained with the LM

generator only the least significant bits seem to be independent. This explains the poor behavior of this RNG in the aforementioned basic tests that evaluate the independence of real numbers, i.e., the “the sum of dice” and “the graphical correlation”.

The LC and XorS generators gave random sequences where the bit-independence was verified only in the most significant bits. However, for applications that require real random numbers, this could be irrelevant as non-randomness would appear until some places after the decimal point. Finally, it must be noted that the bit sequences obtained with the MWC, MT, and UVA generators passed all the tests in the DIEHARD battery of tests.

The results of the statistical empirical tests gave us some confidence regarding the possible use of the MT, MWC, UVA, LC, and XorS generators during the simulation of radiation transport in gamma-ray spectrometry. The results of empirical tests also suggest that LM might not be appropriate for our problem. In addition, other variables (the speed, the period, the requirements for the initialization of the generators, and the possibility of random sequence repetitions) need to be evaluated.

#### 4. Application-specific assessment

In a second step, the generators studied were implemented in the Monte Carlo program DETEFF, developed by us for detector efficiency calibrations in gamma-ray spectrometry, and experimentally validated using the Borland Delphi LC generator (Jurado Vargas et al., 2003; Cornejo Díaz and Jurado Vargas, 2008). The possible problems with the seeding of the sequences were taken into account for each RNG. The specific assessment was carried out by comparison of the full energy peak (f.e.p.) efficiency values determined with DETEFF using the different generators. For this task, we considered a typical HpGe type p detector and a water sample in a 250 mL Marinelli beaker which emits gamma rays between 60 and 1850 keV. The basic scheme of the source-detector system used is illustrated in Fig. 2, and the main characteristics and dimensions of the detector are given in Table 3. The calculations with the code DETEFF were performed with two different CPUs (Centrino Duo T5600 2 × 1.83 GHz and Intel Dual Core 2P4 3 GHz). The number of photons generated was set between  $1 \times 10^8$  and  $2 \times 10^8$  in order to keep the relative statistical standard deviations below 0.2%.

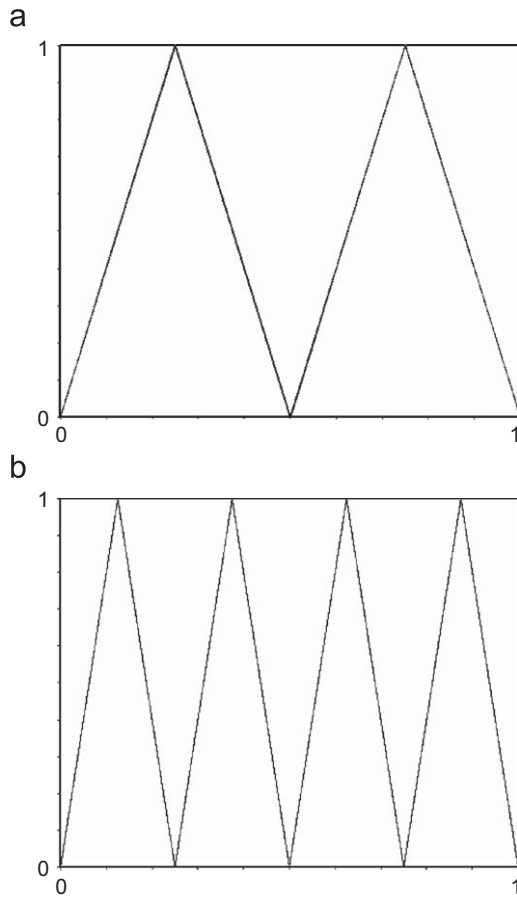


Fig. 1. Scatter plot of points  $(X_{j-1,k}, X_{j,k}) (1 \leq j \leq n; \kappa \in N)$  obtained with the LM generator: (a)  $k=2$  and (b)  $k=3$ . The number of peaks is equal to  $2^{k-1}$ .

Table 2 Results of the DIEHARD battery of tests (Marsaglia, 1995).

Test	Generators					
	LC	XorS	MWC	MT	UVA	LM
Birthday spacing	(1–24) to (6–29)	Yes	Yes	Yes	Yes	(4–27) to (9–32)
The overlapping 5-permutation	Yes	Yes	Yes	Yes	Yes	No
Binary rank $31 \times 31$	Yes	No	Yes	Yes	Yes	Yes
Binary rank $32 \times 32$	Yes	No	Yes	Yes	Yes	Yes
Binary rank $6 \times 8$	(1–8) to (14–21)	(1–8) to (17–24)	Yes	Yes	Yes	(15–22) to (25–32)
Bit stream	No	Yes	Yes	Yes	Yes	No
OPSO	(1–10) to (6–15)	(1–10) to (14–23)	Yes	Yes	Yes	(21–30) to (23–32)
OQSO	(1–5) to (5–9)	(1–5) to (19–23)	Yes	Yes	Yes	(22–26) to (28–32)
DNA	(1–2) to (5–6)	Yes	Yes	Yes	Yes	(19–20) to (31–32)
Count the 1's	(1–8) to (15–22)	(1–8) to (16–23)	Yes	Yes	Yes	(17–24) to (25–32)
Parking lot	Yes	Yes	Yes	Yes	Yes	No
Minimum distance	Yes	Yes	Yes	Yes	Yes	No
3D spheres	Yes	Yes	Yes	Yes	Yes	No
Squeeze	Yes	Yes	Yes	Yes	Yes	No
Overlapping sums	Yes	Yes	Yes	Yes	Yes	No
Runs	Yes	Yes	Yes	Yes	Yes	No
Craps	Yes	Yes	Yes	Yes	Yes	No

The numbers in parentheses represent the bits for which the test was satisfactory.

Table 4 gives the results as relative deviations between the f.e.p. efficiency values calculated with MT, MWC, UVA, XorS, and LM generators and those obtained with the LC generator, for the two CPUs used. The LC generator was taken as the reference because of its adequate results during the previous validation studies (Jurado Vargas et al., 2003; Cornejo Díaz and Jurado Vargas, 2008). The corresponding efficiency values for the LC and LM generators are plotted in Fig. 3 for the CPU Centrino Duo. The results with the Intel Dual Core are similar.

According to Table 4, the observed deviations between the efficiency values calculated with the MT, MWC, UVA, and XorS generators are not statistically significant. However, as was

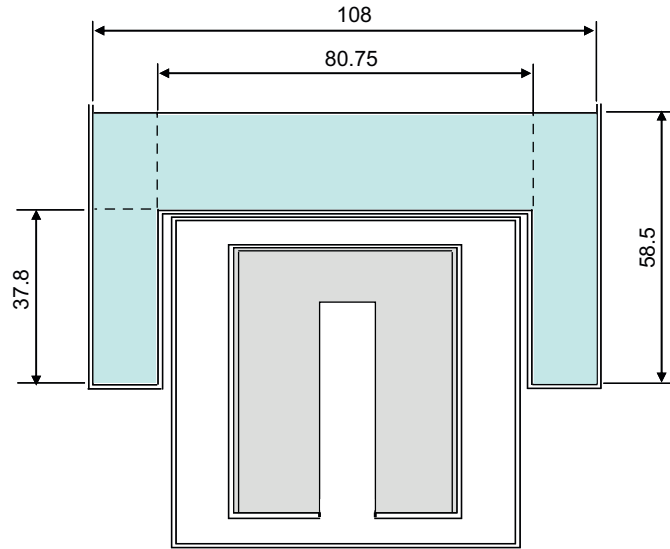


Fig. 2. Basic scheme of the source-detector system used during the specific assessment of RNGs.

Table 3 Main characteristics of the HPGe detector used for the specific assessment of RNGs.

Detector parameters	Values (mm)
Diameter	48.0
Length	57.0
Thickness of frontal dead layer	1.00
Thickness of lateral dead layer	1.00
Detector–cover distance	5.5
Inactive core diameter	12.0
Inactive core length	45.0
Thickness of Al lateral holder	1.0
Thickness of Al frontal holder	0.5
Thickness of Al frontal cover	0.5
Thickness of Al lateral cover	1.0
Cover diameter	75.0

Table 4 Relative deviations (expressed in %) between the values of the f.e.p efficiency calculated with various generators and those obtained with the LC, using two different CPUs.

Energy (keV)	Centrino Duo T5600 2 × 1.83 GHz					Intel Dual Core 2P4 3 GHz				
	MT	MWC	UVA	XorS	LM	MT	MWC	UVA	XorS	LM
60	−0.01	−0.01	−0.04	0.06	−4.46	−0.04	−0.01	−0.04	−0.04	−4.48
250	−0.15	−0.04	0.00	−0.07	−5.99	0.04	0.11	0.11	0.07	−5.89
662	−0.19	0.10	0.00	0.00	4.28	0.00	0.10	0.10	−0.10	4.28
1000	−0.03	−0.06	−0.16	−0.14	3.65	0.06	0.09	0.21	0.10	3.87
1300	−0.33	0.09	0.23	−0.09	−4.09	−0.23	0.09	0.09	0.09	−4.01
1850	0.27	0.09	0.00	0.18	0.66	0.16	0.11	0.11	0.23	0.63

expected, the LM generator introduces a significant bias in the f.e.p. efficiency values. Even when these deviations are significant in the context of the pertinent statistical uncertainties, they are not as high as would be expected taking into account the strong dependence of terms in the LM random sequences. This fact indicates that the simulation of radiation transport for efficiency calibrations in gamma-ray spectrometry is a “less demanding problem” regarding the independence of the terms in the random sequences produced by the RNG used. It seems to be a consequence of the diversity of variables and processes sampled during the simulation of each radiation path.

Fig. 4 shows the CPU times required for the specific energy of 661.66 keV, being the plots quite similar for the rest of energies. According to this figure, the speed is therefore not a limiting factor in choosing between the LC, MT, MWC, UVA, and XorS

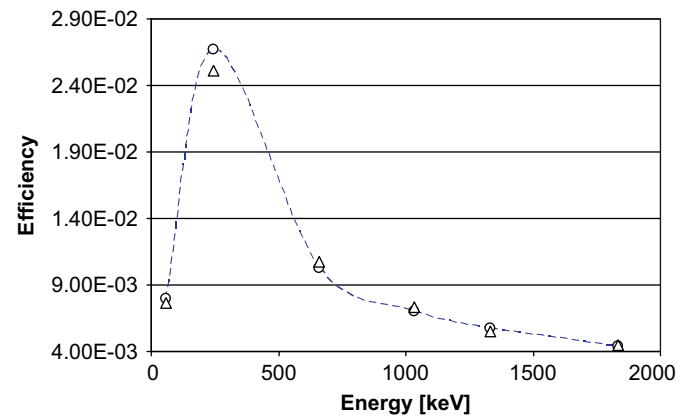


Fig. 3. Values of the full-energy peak efficiency calculated with the LC generator (dots) and with the LM generator (triangles). The dashed lines connect the LC values. The CPU used was the Centrino Duo T5600 2 × 1.83 GHz.

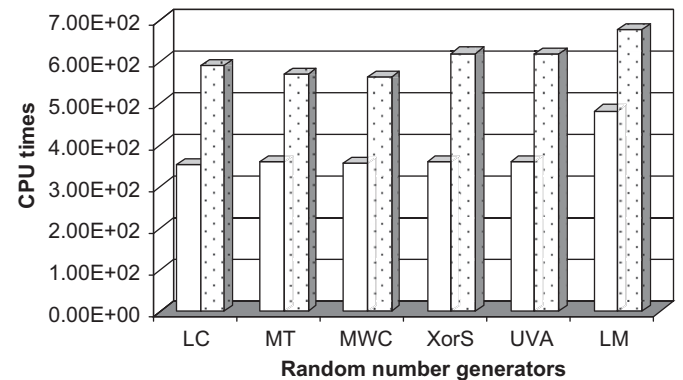


Fig. 4. CPU times for Centrino Duo T5600 2 × 1.83 GHz (white) and Intel Dual Core 3 GHz (dots) corresponding to calculations for 661.66 keV.

generators. It must be noted that, in general, the time required in each simulation with DETEFF to generate the random number sequence is just a small fraction of the total time consumed (about 1%). The periods of the studied random sequences are equal to or greater than  $2^{32}$ , the period of the MT generator being the greatest ( $2^{19937} - 1$ ). Anyway, in almost all the situations modeled with DETEFF,  $10^8$  random numbers are enough to obtain standard deviations  $< 0.2\%$  for the efficiency.

## 5. Conclusions

The statistical performance, the application-specific test, the speed, and the period of the LC, MT, MWC, UVA, and XorS generators studied indicate their suitability for Monte Carlo simulation applied to efficiency calculations in gamma-ray spectrometry, with the MT, MWC, and UVA generators being the most powerful. However, considering the possibility of repeating the random number sequences, the generators MT and MWC could be more convenient.

The *non-periodic logistic map based generator* (LM) implemented by us does not seem to be suitable for Monte Carlo simulations in gamma-ray spectrometry. The rejection of the first 300 numbers in each calculation would improve the statistical performance of the random sequence, but this implementation would be in clear disadvantage with the rest of the generators studied because of its speed.

Finally, the approach described in this work, combining the statistical empirical tests with the application-specific assessment could be applied to assess other generators intended for the simulation of gamma-ray spectrometry systems.

## References

- Barberis, G.E., 2007. Non-periodic pseudo-random numbers used in Monte Carlo calculations. *Physica B* 398, 468–471.
- Brown, F.B., Nagaya, Y., 2002. The MCNP5 random number generator. *Trans. Am. Nucl. Soc.* 87, 230–232.
- Cornejo Díaz, N., Jurado Vargas, M., 2008. DETEFF: an improved Monte Carlo computer program for evaluating the efficiency in coaxial gamma-ray detectors. *Nucl. Instrum. Methods Phys. Res. A* 586, 204–210.
- Debord, J., 2008. Math library for Pascal compilers. [Online] Available at: <http://sourceforge.net/projects/tpmath>.
- Geant4 Home Page. The GEANT4 Collaboration. [Online] Available at: <http://www.cern.ch/geant4/>.
- James, F., 1990. A review of pseudorandom number generators. *Comput. Phys. Commun.* 60, 329–344.
- Jurado Vargas, M., Cornejo Díaz, N., Pérez Sánchez, D., 2003. Efficiency transfer in the calibration of a coaxial p-type HpGe detector using the Monte Carlo method. *Appl. Radiat. Isot.* 58, 707–712.
- Knuth, D.E., 1998. *The Art of Computer Programming. Vol. II. Seminumerical Algorithms*. Addison-Wesley-Longman, ISBN: 0-201-89684-2.
- L'Ecuyer, P., 1988. Efficient and portable combined random number generators. *Commun. ACM* 31, 742.
- L'Ecuyer, P., Simard, R., 2009. TestU01: A software library in ANSI C for empirical testing of random number generators. [Online] Available at: <http://www.iro.umontreal.ca/~simardr/testu01/tu01.html>.
- Marsaglia, G., 1995. The Marsaglia random number CDROM, with the DIEHARD battery of tests of randomness. [Online] Available at: <http://stat.fsu.edu/pub/diehard>, <http://www.cYess.hku.hk/~diehard/cdrom>.
- Marsaglia, G., 2003. Random number generators. *J. Mod. Appl. Stat. Methods* 2 (1), 2–13.
- Matsumoto, M., 2002. Mersenne-twister random number generator. [Online] Available at: <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>.
- Matsumoto, M., Nishimura, T., 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. Modelling Comput. Simulation* 8 (1), 3–30.
- Salvat, F., Fernandez-Varea, J.M., Sempau, J., 2006. PENELOPE, A Code System for Monte Carlo Simulation of Electron and Photon Transport. OECD, ISBN: 92-64-02301-1.
- Vergara Gil, A., 2008. Pascal implementation of the Mersenne-twister random number generator. [Online]. Available at: <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/VERSIONS/PASCAL/Alex.pas>.