ACTA TEHNICA CORVINIENSIS – Bulletin of Engineering
Tome VII [2014] Fascicule 3 [July – September]
ISSN: 2067 – 3809

[1.] Mihaela OSACI

STUDIES ABOUT INFLUENCE OF THE PSEUDO-RANDOM NUMBER GENERATORS IN SIMULATION MODELS FOR MAGNETODIELECTRIC NANOCOMPOSITES

[1.] Department of Electrical Engineering and Industrial IT, 'Politehnica' University of Timisoara, Revolutiei street, no. 5, Hunedoara, ROMANIA

Abstract: *Interest for magneto-dielectric nanocomposites is motivated by their interesting properties [1]: high electrical resistivities, low losses in turbionar currents and hysteresis, endurance at high temperature, chemical stability. Such materials offer many appealing technological properties: can easily be fabricated in various shapes and dimensions with low technological losses and at low prices. These properties open the possibility of using these materials instead common magnetic materials for electric machines and other electromagnetic devices. In this context the simulation problems of this materials behaviour in electromagnetic fields is very important. This paper proposes a study on how the PRNG type influences the results obtained through a simulation models for magnetodielectric nanocomposites behavior in electromagnetic fields. The case study model allows the calculation of the Néel time in nanoparticles systems with magnetic dipole interactions.*

Keywords: *superparamagnetism, magnetic nanoparticles system, stochastic model, pseudo-random number generator, simulation model, nanoparticle system*

INTRODUCTION

Nowadays, in nanomagnetism the effort is becoming greater for understanding the phenomena at nanoscopic scale. For this purpose, the modelling and simulation, especially the stochastic modelling and simulation play a major role. So, random numbers are being used more and more in computational nanomagnetism. To simulate the random components, we use random numbers, with theoretical or empirical distributions. For generating these numbers through algorithms, we use random numbers uniformly distributed in the range (0, 1) generated by computer. The algorithm used for generating is called pseudo-random number generator (PRNG).

The pseudorandom numbers are generated by deterministic algorithms, PRNG. An acceptable PRNG must yield sequences of numbers that are uniformly distributed, statistically independent, reproductible and non-repeating for any desired length. Thus, they are subjected to empirical and statistical tests, as follows: frequency test, serial test, autocorrelation test, run test, and chi-square test for goodness of fit [2]

The quality requirements for a PRNG include a huge period length, good statistical properties, high speed, low memory usage, repeatability of jumping ahead and splitting facilities [2].

The pseudorandom numbers used in simulation algorithms in science and engineering [3], [4], [5], [6].

The random number generators (RNGs), like those in MATLAB, are algorithms for generating pseudorandom numbers with a specified distribution [2]. A given number may be repeated many times during the sequence, but the entire sequence is not repeated.

A pseudorandom sequence is described by several properties [7]. A random stream is the sequence of values that are returned by a generator. The period of a sequence is its length, which is the number of values that it generates before the entire sequence is repeated. The state is the information that the generator keeps internally in order to create

successive values in the stream. The seed of a sequence is a single value defining its starting point.

The MATLAB software offers six generator algorithms [7]. Table 1 summarizes the key properties of the available generator algorithms and the keywords used to create them [3].

Table 1. Generator algorithms in Matlab [7]

Keyword	Generator	Multiple Stream and Substream Support	Approximate Period In Full Precision
mt19937ar	Mersenne twister (used by default stream at MATLAB startup)	No	$2^{19936}-1$
mcg16807	Multiplicative congruential generator	No	$2^{31}-2$
mlfg6331_64	Multiplicative lagged Fibonacci generator	Yes	2^{124}
mrg32k3a	Combined multiple recursive generator	Yes	2^{127}
shr3cong	Shift-register generator summed with linear congruential generator	No	2^{64}
swb2712	Modified subtract with borrow generator	No	2^{1492}

Some of the generators (mcg16807, shr3cong, swb2712) provide for backwards compatibility with earlier versions of MATLAB [7]. Two generators (mrg32k3a, mlfg6331_64) provide explicit support for parallel random number generation. The remaining generator (mt19937ar) is primarily designed for sequential applications. Depending on the application, some generators may be faster, or return the values with more precision.

THE CASE STUDY SIMULATION MODEL

The case study simulation model is a 3D model (arrangement of the nanoparticle in the system), phenomenological-stochastic, that allows the simulation of the average Néel relaxation time in a system of nanoparticles dispersed in dielectric matrix.

In this model [10], we consider a superposition of the nanoparticle size distributions, of the effective anisotropy constant distributions, and the nanoparticle disposal distributions in the sample. We take into account the dipolar magnetic interaction between the nanoparticles. With this model, we will realize advanced studies on understanding the phenomena at nanoscopic level and their implications in nanoparticles system applications, especially in electromagnetic devices.

Our model [10] is reported to the display of the nanoparticles in a given quantity, and we assume that it is aleatory (Gaussian distribution). We consider a statistical angular distribution for the angles made at a given moment by the other j nanoparticles with the uniaxial anisotropy axis, and we are going to simulate a 3D initial geometrical state after the dispersion of the nanonanoparticles in the basic matrix. These distributions will be simulates by stochastic methods.

So, for the N nanoparticles of the system, we simulate the distribution of diameters and anisotropy constants. The nanoparticle diameters are simulated as aleatory variables generated by a lognormal repartition law. The effective magnetic anisotropy constants of the nanoparticles, due to the surface effects, can be also simulated using a lognormal distribution. The variation of the anisotropy constants is justified [8] by the appearance of a surface anisotropy of the nanoparticles (a structural anisotropy resulted from the discontinuity of the magnetic interactions between individual spins found on the particle surface).

The distributions related to the nanoparticle geometrical arrangements and magnetic moment orientations are Gaussian distributions simulated through the Box Mueller method [9].

The model [10] starts from the idea of an Ising two-level model [11] of a spherical nanonanoparticle system with distribution of diameters and effective magnetic anisotropy effects. The energy of an i nanoparticle of the system in the $\vec{H}_i$ *local magnetic field, oriented along its easy-magnetisation axis, is a function of* θ_i, $E_i = f(\theta_i)$, *where* θ_i *is the angle between the direction of the i nanoparticle magnetic moment and the direction of the easy-magnetisation axis, along which the external magnetic field acts. In the* E_i *expression [6], we find the* M_{pi} *- i nanoparticle magnetic moment,* v_i *- volume of the i nanoparticle,* K_{ieff} *- the effective anisotropy constant of the nanoparticle, and* H_i*, local magnetic field in direction of easy magnetisation axis of nanoparticle. If we take into account the interactions of nanoparticles, we can consider that the* H_i *field is made of two contributions: the external magnetic field H that acts along the easy-magnetisation direction, and the projection along the external field of the dipolar magnetic field created on the i nanoparticle, due to the dipolar magnetic interactions of the nanonanoparticles.*

$$H_i = H + H_{di} \qquad (1)$$

The magnetic behaviour of the nanoparticles in a dielectric matrix is determined by the Néel relaxation processes (nanoparticle magnetic moment rotation), characterised by the Néel relaxation time τ_N, whose expression for the non-interactive case is given, for example, in [12], because the Brown relaxation processes (nanoparticle rotation in the solid dielectric matrix), characterised by Brown relaxation time τ_B [12] in blocked.

In these conditions, the M_{pi} magnetic moment of a given i nanoparticle can be in one of the two equilibrium states, with the minimum energies E^i_{min1} and E^i_{min2} determined by $\theta_i = 0$ and $\theta_i = \pi$. These minimums are separated by the maximum energy E^i_{max}.

In the presence of thermal fluctuations, the magnetic moment of an i nanoparticle in steady state, with minimal energy E^i_{min1}, can spontaneously change its direction in the minimum energy state E^i_{min2}. The energy barriers for these re-orientations are E^i_{b12} and E^i_{b21}.

We consider that the system is in thermal equilibrium. The average number of nanoparticles that pass in time unit from a minimum to another minimum is proportional with $N_k \exp\left(-\frac{E_{bk}}{k_B T}\right)$, where k_B is the Boltzmann constant, and T is the temperature. If N_1 is the average number of nanoparticles in the state with minimum average energy $<E_{min1}>$, N_2 is the number of nanoparticles in the minimum average energy $<E_{min2}>$, the average energy barriers $\langle E_{b12}\rangle = < E_{max} > - < E_{min1} >$ and $\langle E_{21}\rangle = <E_{max}> - <E_{min2}>$, then the equilibrium condition shall be:

$$N_1 \exp\left(-\frac{<E_{b12}>}{k_B T}\right) = N_2 \exp\left(-\frac{<E_{b21}>}{k_B T}\right) \qquad (2)$$

$<x>$ is the notation for the statistical average for x. For a very large number N of nanoparticles, the arithmetical average estimator of x tends to the statistical average of x.

If N is the total number of nanoparticles of the system:

$$N = N_1\left[1 + \exp\left(-\frac{<E_{b12}> - <E_{b21}>}{k_B T}\right)\right] \qquad (3)$$

The difference $n = N_1 - N_2$ determines the resultant magnetic moment of the system. The total magnetic moment of the system in a given moment is proportional with n. Within an infinitesimal time period, the n difference becomes:

$$\frac{\partial n}{\partial t} = f\left[N_1 \exp\left(-\frac{<E_{b12}>}{k_B T}\right) - N_2 \exp\left(-\frac{<E_{b21}>}{k_B T}\right)\right] \qquad (4)$$

where f is a factor measured in s^{-1}.

Having in view the relations between N_1, N_2 and n, and applying a simple calculation artifice, we can write:

$$\frac{\partial n}{\partial t} = -f \cdot \exp\left(-\frac{<E_{b12}> + <E_{b21}>}{2k_B T}\right) \cdot \left[nch\left(\frac{<E_{b12}> - <E_{b21}>}{2k_B T}\right) + Nsh\left(\frac{<E_{b12}> - <E_{b21}>}{2k_B T}\right)\right] \qquad (5)$$

Starting from the equation (5), at usual temperatures much higher than the blocking temperature and very close to the equilibrium, the magnetic relaxation process of the system is governed by a differential equation which offers the solution that represents the exponential evolution in time of the residual magnetisation of the system with the Néel relaxation time:

$$\tau_N = \frac{\exp\left(\frac{<E_b>}{k_B T}\right)}{f_0 \cdot ch\left(\frac{<E_{b12}> - <E_{b21}>}{2k_B T}\right)} \qquad (6)$$

where $<E_b>$ is the average energy barrier and $f_0 = 2f$.

In the expression of energy barriers for the magnetic moments, we find re-orientations of the dipolar field of the nanoparticle H_{di}. To calculate the H_{di} and the average dipolar field $<H_d>$, we use a stochastic method [13].

Using this model, we will perform a series of advanced studies to understand the effects of the distribution of particle sizes, the surface anisotropy and the anisotropy distribution, as well as the role of the dipole interaction on the magnetic properties. These studies will enhance our understanding of the magnetic properties of the nanoparticle systems, which is now far from being complete.

RESULTS AND INTERPRETATIONS

The simulations were made for a system of 14 335 spherical nanoparticles of magnetite, with the spontaneous magnetization $M_s = 4.46 \cdot 10^5\ A/m$,

dispersed in solid dielectric matrix, with average diameter of 10 nm, dispersion of the aleatory variable lnd 0.0625, average effective magnetic anisotropy constant of 10000 J/m^3, and dispersion of the aleatory variable ln K_{eff} 1.21. The spatial distribution is simulated with Box Muller transformation [13], [14]. The spatial distribution parameters are: angular distribution dispersion for magnetic moments orientation: 0.25, dispersion of the spatial distribution of the nanoparticles: 0.01 (uniform arrangement of the nanoparticles in solution).

To generate series of random numbers with various distributions (lognormal, Gaussian) used in the model, we applied stochastic methods, and we checked the generated series of numbers with the concordance statistical criterion Kolmogorov-Smirnov [9], using from Matlab the function kstest. In case the series passes the test, the parameter H of this function returns the value 0; otherwise, the value is 1.

For simulations, we used 3 types of pseudo-random number generators: Mersenne Twister (mt19937ar), multiplicative congruential generator (mcg16807) and shift-register generator summed with linear congruential generator. Thus, we simulated the average Néel relaxation time for the nanoparticle system and the average effective magnetic relaxation time at various volume fractions of nanoparticles in solution, for seed=5. The graphical results are presented in Figure 1.

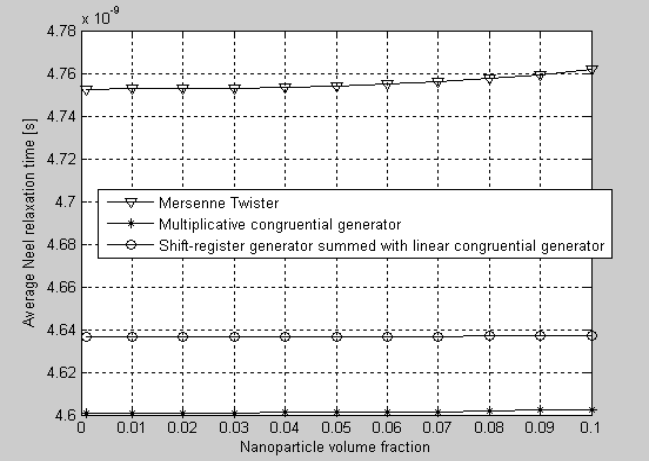

Figure 1. The average Néel relaxation time versus the nanoparticle volume fraction, for generator seed=5

In the Figure 1 can be seen that the type of pseudo-random number generators influences the average value of the Néel relaxation time and the average effective magnetic relaxation time of the nanoparticles. Closer to reality are the results simulated with the Mersenne Twister generator. Most of the literature in the field [16], [17] claims an increase of the magnetic relaxation time with increasing the concentration of nanoparticles, at low concentrations.

CONCLUSIONS

A way to check the results of a stochastic simulation is to rerun the simulation with two or more different generator algorithms, and the MATLAB software's generator choice provides you with the means to do that. It can be seen that the results do not differ greatly in case we work with different PRNGs, but a more realistic dependence can be seen when working with the Mersenee Twister generator.

According to the literature [see the References] and the researchers in the field, the current PRNG's state-of-the-art for simulation work is the so-called "Mersenne Twister" (MT).

It's hard to imagine any scientific application failing with this generator, although it has a huge period. It has also been adapted to many languages, and it is the default RNG in research.

References

[1.] *M. Osaci, M. Panoiu, I. Muscalagiu, Comparative study of the models for simulating the behavior of the ferritic nanocomposites within radiofrequency magnetic fields, Romanian Journal of Physics, vol. 49, nr. 9-10, 2004*

[2.] *H. Perros, Computer Simulation Techniques: The definitive introduction, 2009, http://www4.ncsu.edu/~hp/simulation.pdf*

[3.] *M. Panoiu, C. Panoiu, I. Sora, M. Osaci, About the Possibility of Power Controlling in the Three-Phase Electric Arc Furnaces Using PSCAD EMTDC Simulation Program, Advanced in Electrical and Computer Engineering, 7(1), pp.201-207, 2007*

[4.] *M. Pănoiu,C. Pănoiu, M. Osaci, I. Muscalagiu, Simulation Result about Harmonics Filtering for Improving the Functioning Regime of the UHP EAF, Proceedings of tha 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, pp.75-80, 2007*

[5.] *M. Pănoiu, C. Pănoiu, M. Osaci, I. Muscalagiu, Simulation Results for Modeling the AC Electric Arc as Nonlinear Element Using PSCAD*

EMTDC, WSEAS Transactions on Circuits and Systems, 6(1), pp. 149-156,

[6.] M. Pănoiu, C. Pănoiu, I. Şora, M. Osaci, Simulation Results on the Reactive Power Compensation Process on Electric Arc Furnance Using PSCAD-EMTDC, International Journal of Modelling, Identification and Control, 2(3), pp. 250-257, 2007

[7.] http://www.mathworks.com/help/techdoc/ref/randstream.list.html

[8.] M. Knobel, W.C. Nunes, L.M. Socolovsky, E. De Biasi, J.M. Vargas, J.C. Denardin, Superparamagnetism and other Magnetic Features in Granular Materials: A Review on Ideal and Real Systems, Journal of Nanoscience and Nanotechnology, vol. 8, 2836-2857, 2008.

[9.] E. Petrisor, Simulare Monte Carlo, Politehnica Publishing House, Timisoara, 2006, pp. 67.

[10.]M. Osaci, Study about the possibility to control the superparamagnetism-superferromagnetism transition in magnetic nanoparticle systems, Journal of Magnetism and Magnetic Materials, 343, 189-193, 2013

[11.]R. Prozorov, Y. Yeshurun, Magnetic irreversibility and relaxation in assembly of ferromagnetic nanoparticles, Physical Review B, vol. 59, issue 10, pp. 6959-6965

[12.]R.E. Rosensweig, Journal of Magnetism and Magnetic Materials, 252 (2002) 370–374.

[13.]Manuela Pănoiu, Caius Pănoiu, Mihaela Osaci, Ionel Muscalagiu, Modeling and Simulating the AC Electric Arc Using PSCAD EMTDC, Proceedings of the 5th WSEAS Int. Conf. on System Science and Simulation in Engineering, Tenerife, Canary Islands, Spain, December 16-18, 2006

[14.]Manuela Pănoiu, Caius Pănoiu, Mihaela Osaci, Ionel Muscalagiu, Simulation Results for Modeling the AC Electric Arc as Nonlinear Element Using PSCAD EMTDC, WSEAS Transactions on Circuits and Systems,Vol 6, 2007, Pp 149-157

[15.]Mihaela Osaci, Manuela Pănoiu, Adela Diana Berdie, Stela Rusu-Anghel, Three-dimensional stochastic model for simulation of the nanoparticles geometry in nanoparticles dispersed magnetic media, Proceeding ICCOMP'09, Proceedings of the WSEAS 13th International Conference on Computers, 2009, pp. 179-183.

[16.]Ainhoa Urtizberea, Eva Natividad, Ana Arizaga, Miguel Castro and Arturo Mediano, J. Phys. Chem. C, 2010, 114 (11), pp 4916–4922

[17.]J.L. Dormann, L. Bessais, D. Fiorani, J. Phys. C: Solid State Phys, 21, (1988), 2015

ACTA Technica CORVINIENSIS
BULLETIN OF ENGINEERING
ISSN:2067-3809