MSc in Photonics PHOTONICS**BCN**

Universitat Politècnica de Catalunya (UPC)
Universitat Autònoma de Barcelona (UAB)
Universitat de Barcelona (UB)
Institut de Ciències Fotòniques (ICFO)

http://www.photonicsbcn.eu

Master in Photonics

MASTER THESIS WORK

Applications of a quantum random number generator to simulations in condensed matter physics

Guillem Guigó i Corominas

Supervised by Prof.Dr. Maciej Lewenstein (ICFO) &
Dr. Miguel Ángel Garcia-March (ICFO)

Presented on date 15th October 2018

Registered at

Escola Tècnica Superior
d'Enginyeria de Telecomunicació de Barcelona

Applications of a quantum random number generator to simulations in condensed matter physics

Guillem Guigó i Corominas

ICFO - The Institute of Photonic Sciences, Mediterranean Technology Park, Castelldefels (Barcelona), Catalonia, Spain

E-mail: guillemguigo@gmail.com

Abstract. We study the importance of the quality of random numbers in Monte Carlo simulations of 2D Ising systems. Simulations are carried out at critical temperature to find the dynamic scaling law of the linear relaxation time. Our aim is to show that statistical correlations that appear in large Ising simulations performed with pseudorandom numbers can be corrected using a quantum random number generator (QRNG). To achieve high speeds and large systems, Ising lattices are simulated on a field programmable gate array (FPGA) with an optical QRNG. Here we report on results on simulations with pseudorandom nunbers and first results with the QRNGs.

1 Introduction

Randomness is a very important concept in several fields such as philosophy, science and technology. Random processes can be used to extract random numbers, which have many applications in computation, simulations or cryptography, and are central to fundamental research and technological developments. Many efforts are being devoted towards developing efficient ways of generating large sequences of random numbers [1].

Defining randomness can be controversial, as it may have more than one interpretation depending on the field it concerns. In general, a sequence is considered to be random if it is unpredictable and follows a certain statistical distribution. Devices or methods that generate strings of random numbers are called random number generators (RNG) and can be built in several ways. In computing, it is important to distinguish between algorithmically generated number sequences, and numbers which are extracted from measurements of certain physical events. Methods that produce random numbers using arithmetic algorithms are said to be pseudorandom number generators (PRNG), as it is not possible to generate a true random sequence from a deterministic process. Physical or true random number generators (TRNG) measure a random, or at least apparently unpredictable physical process to extract random values and create a sequence of numbers that can then be accessed by a software [2].

PRNGs generate random numbers using an initial string of bits known as *seed*, which serves as input for a procedure which outputs a sequence of numbers that mimics the statistics of a random distribution. One of the most important aspects of a PRNG is its period. Each number in a pseudorandom sequence is determined by the current internal state of the generator; for a finite memory, there is a certain length after which the internal state will be the same as some previous state and the sequence will start repeating itself. For most purposes, PRNGs work just as fine as TRNG when it comes to statistical distributions, with the advantage that they can generate random numbers much faster than any other RNG and the sequences can be replicated if the seed is known, allowing for reproducibility. Nonetheless, the predictibility and deterministic nature of PRNGs make them unsuitable for some practical uses; they are not 100% cryptographically sercure, and although most PRNGs are designed to have very large periods, they are known to adversely affect simulations which require a high volume of data due to long range undetected correlations [3].

True random number generators are able to avoid these problems by measuring physical random processes from which independent, uncorrelated values are obtained. They rely on *entropy sources*, which consist of physical systems with some random quantity plus the instruments used to read them. The process of collecting unpredictable data is called *entropy gathering*. From the measurement of the random quantity a string of bits called the *raw bit string* is obtained. The raw bit string is often noisy and may have some degree of correlation, so it usually goes through a postprocessing stage referred to as *randomness extraction* (Figure 1). Randomness extractors transform the bits from the raw sequence into a shorter uniform random sequence at the output, which contains most of the randomness available in the system [2].

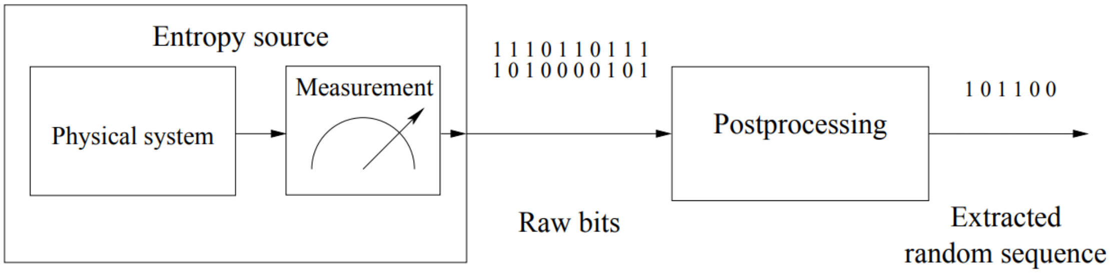

Figure 1: Block diagram of a typical physical random number generator. The raw bit string is obtained from the entropy source which consists of the physical system plus the measurement device, and then goes through a post-processing stage to remove any biases.

Quantum random number generators (QRNG) are a particular type of physical RNG in which data is gathered from the measurement of a quantum event. QRNG excel in generating random data due to the intrinsic randomness of quantum mechanics, where the outcome of a measurement is probabilistic even if we have complete knowledge of the system in consideration. This *intrinsic randomness* appears in contrast to the *apparent randomness* found in classical physics. Apparent randomness is the concept we use to express our lack of knowledge of the system and it implies the existence of the so called underlying *hidden variable theory*; there must exist some hidden variables that we cannot access, so we use probabilities and stochastic processes to partially describe the system. Had we known them, the illusion of randomness would disappear [4].

The first QRNGs were based on radioactive decay, but they were limited by the low bit generation bit rate. Since then, more efficient QRNGs have been developed. Optical QRNGs are among the most used nowadays, reaching speeds well above the megabit per second [5]. Although their generation rate is still several orders of magnitude lower than that of good PRNGs, these improvements have made QRNGs suitable for large-scale simulations that were previously limited to PRNGs [6].

One particular area in which PRNGs are known to adversely affect the results is on Monte Carlo methods [3, 7]. Monte Carlo simulations use stochastic methods to find solutions to complex problems in numerical integration and statistical physics, where most models cannot be solved analytically, by averaging over many random instances. If the random instances are uniformly distributed, the results are usually accurate. However, since the simulations require extremely large amounts of data, using PRNGs may result in correlated outputs, even if they are of good quality [8]. Several cases of such failures have been recorded in the Ising model and related problems [7, 9].

Monte Carlo simulations of Ising systems have been widely used to understand the properties of the model, some of which are still being studied. The dynamic scaling law of the relaxation time of 2D Ising lattices has been the focus of many studies.Data on linear relaxation studies has been limited by the extremely long simulated time required in equilibrium simulations due to the so-called ”critical slowing down” effect. Most of our current knowledge on the value of the dynamic exponent z for the 2D Ising model comes from short nonequilibrium simulations where relaxation is nonlinear, or from calculations at equilibrium on small lattices using the stochastic matrix method. The results from such simulations with short correlation lengths agree on the approximated value of z. Verifying this approximation in lattices with larger correlations lengths has not been an easy task [10].

Recent advancements in field programmable gate array processors (FPGA) have significantly reduced the computing time of Monte Carlo simulations. Dedicated to perform specific tasks programmed by users, digital circuits in FPGA contain multiple logical elements which can perform calculations independently and concurrently, allowing for massive parallel computing and speed-ups beyond reach of most CPU-based computers [11]. Using a FPGA-based device, Lin & Wang (2016) were able to study linear relaxation in large two-dimensional Ising lattices. Their main goal was to address whether critical Ising systems with longer correlation lengths conformed to the same dynamic exponent z found in previous simulations. While the value of z was consistent with studies of Ising lattices with shorter correlation lengths (with some statistical deviation), it was found that simulations in large lattices were very sensitive to statistical correlations between pseudorandom numbers [10].

In the current study, we test an ultrafast QRNG based on accelerated phase diffusion developed at ICFO in a FPGA-based computing system configured to perform

Monte Carlo simulations of 2D Ising models. As in [10], we run simulations in large square lattices (ideally up to 2048 × 2048 spins) at criticality and evaluate the dynamic scaling behavior of the linear relaxation time. Theoretically, statistical errors that appear in previous simulations carried with PRNGs should be able to be corrected using a QRNG, allowing for a more accurate calculation of z. Based on the results, we discuss whether quantum mechanics can be used as a benchmark for RNG.

2 Monte Carlo simulations of the Ising model

The Ising model is a mathematical model of a magnet used to study ferromagentism in statistical mechanics. Here we consider a finite 2D square lattice with N sites, where the spin at the ith site can be oriented either up or down ($s_i = \pm 1$). For the simplest ferromagnetic model without external magnetic field, where spins interact only with their nearest neighbors, the Hamiltonian of the system is

$$H = -J \sum_{\langle ij \rangle} s_i s_j, \tag{1}$$

where J is the interaction strength and s_i, s_j represent the spins at neighboring sites i and j. The model includes a thermal reservoir, an external system that acts as a source and sink of heat. The effects of the reservoir are incorporated in the calculations by giving the system a dynamics. We define a set of weights $\omega_\mu(t)$ which represent the probability that the system will be in the state or configuration μ at time t. We also define the transition rates $R(\mu \to \nu)dt$, which give the probability of going from state μ to state ν at each time interval dt. We can then write a *master equation* for the evolution of the weights in terms of the transition rates:

$$\frac{d\omega_\mu}{dt} = \sum_\nu [\omega_\nu(t) R(\nu \to \mu) - \omega_\mu(t) R(\mu \to \nu)] \tag{2}$$

Monte Carlo simulation of the system is carried out with the Metropolis algorithm. At each time step, the simulation updates the direction of a single spin according to the change in energy caused by the flipping of the spin, given by

$$\Delta E = E_\nu - E_\mu = 2J s_k^\mu \sum_{\langle ki \rangle} s_i^\mu, \tag{3}$$

where ΔE is the change in energy due to the spin flip, E_ν is the energy of the state after the spin flip, E_μ is the energy of the current state, s_k^μ is the spin that is to be flipped and s_i^μ are the nearest neighbour spins of s_k^μ. The flip move is accepted if it lowers the total energy of the system, or if the following condition is fulfilled,

$$R < \exp\left[-\frac{\Delta E}{kT}\right], \tag{4}$$

where R is a uniform random number between 0 and 1. The right hand side is the Boltzmann weighting factor, with k the Boltzmann constant and T the temperature. If the condition is not fulfilled, the spin is not flipped and the system stays in its current state. This criterion is chosen so that occupation probabilities p_μ of each state at equilibrium are proportional to their Boltzmann weight

$$p_\mu = \frac{1}{Z}\exp\Big[-\frac{E_\mu}{kT}\Big], \tag{5}$$

where $Z = \sum_\mu \exp\left[\frac{-E_\mu}{kT}\right]$ is the partition function. The probability distribution in Eq. (5) is known as Boltzmann distribution. In 1902, Gibbs showed that the occupation probabilities of a system in thermal equilibrium with a reservoir follow the Boltzmann distribution. To ensure that we obtain such distribution at equilibrium, we include the condition of *detailed balance*, which implies that the rate of change of any weight at equilibrium is zero, $\frac{d\omega}{dt} = 0$. This occurs when $p_\mu R(\mu \to \nu) = p_\nu R(\nu \to \mu)$ in Eq. (2)

The simulation starts with the system at a known configuration, either at $T = 0$ where all spins are aligned (all up or all down) or at $T = \infty$, where spins are randomly oriented. The desired temperature is then selected and the system is let to equilibrate. Once the system has reached equilibrium, the simulation runs for a certain number of steps in order to obtain values for the physical quantities to be studied. If repeated measurements are taken for a significant number of steps, the value of any quantity can be determined by averaging over all measurements. An important quantity considered here is the magnetization M, which for a given state μ is defined as the sum of all spin values divided by the total number of spins

$$M_\mu = \frac{1}{N}\sum_i s_i. \tag{6}$$

The 2D Ising model has a phase transition that takes place at the critical temperature $T_c = \frac{2J}{\log(1+\sqrt{2})} \simeq 2.269J$. Above this temperature the system is in the paramagnetic phase with zero average magnetization. Below T_c, the system is in the ferromagnetic phase (or antiferromagnetic if J is negative) and develops spontaneous magnetization, in which most of the spins are aligned (in the ferromagnetic case) and the magnetization is non-zero. The region near T_c is called the critical region, and the processes that occur in this region are called critical phenomena. It is important to define the *reduced temperature*, a dimensionless parameter t that measures how far away we are from T_c

$$t = \frac{T - T_c}{T_c}. \tag{7}$$

When approaching the critical temperature, the system tends to form large clusters of spins pointing in the same direction. These clusters contribute significantly to quantities such as the magnetization and energy, and they produce large fluctuations as they flip orientation, called critical fluctuations. This is caused by the divergence of the correlation length near T_c. The correlation length is a parameter that determines how fast the correlation length, a measure of how strongly correlated two spins at different sites are, vanishes. Near the phase transition, the correlation length diverges as

$$\xi \sim |t|^{-\nu}, \tag{8}$$

where ν is a positive quantity called a *critical exponent.* Critical exponents is the name given to the exponents that appear in the expression in terms of power laws of the quantities in which anomalous behavior is observed in the critical region. The value of critical exponents is a property of the Ising model itself, and independent of such things as the value of the coupling J or the shape of the lattice. In fact, physical systems of different nature and composition often show the same critical behavior, as long as they share the same symmetry group in the hamiltonian and the dimensionality of the lattice space. This property is known as *universality.*

Another quantity that diverges in the thermodynamic limit at T_c is the correlation or relaxation time τ of the system. The relaxation time of the Ising model is defined as the mean time-scale in which the magnetization autocorrelation falls off. The autocorrelation function gives us a measure of the correlation of the magnetization of the system at two different times, one a time interval t later than the other. The time-displaced autocorrelation $\chi(t)$ of the magnetization is given by

$$\chi(t) = \langle M(0)M(t)\rangle. \tag{9}$$

In our model, the autocorrelation is expected to fall off exponentially at long times as

$$\chi(t) \sim \mathrm{e}^{-t/\tau}, \tag{10}$$

where τ is the relaxation time, measured in Monte Carlo steps. The divergence of τ close to the phase transition is known as the *critical slowing down* effect, and goes as

$$\tau \sim \xi^z, \tag{11}$$

where z is the *dynamic exponent.* While z is still independent of the shape of the lattice, the spin-spin interaction J and so forth, it differs from other critical exponents in that its value is affected by changes in the dynamics of the system.

Many attempts have been made to obtain a good value of z for the 2D Ising model. The exact value cannot be calculated analitically, so numerical methods are used to obtain an approximated result. Different series-expansion methods give theoretical estimations between 2.0 and 2.50. Previous studies that used methods based on nonequilibrium relaxation simulations gave a value of $z \simeq 2.167$, in agreement with the value obtained using the stochastic matrix method to calculate the relaxation times of small 2D Ising lattices ($L \leq 16$) [10]. A value of $z \simeq 2.17$ is fairly common amongst Monte Carlo algorithms for the 2D Ising model [12]. Here we use a finite size scaling (FSS) approach to study Ising systems with longer correlation lengths. The calculations employed to obtain z are explained in more detail in section 4.

3 Simulation of the Ising model on FPGA

A field programmable gate array is a type of programmable logical device (PLD) in which circuits can be programmed by users to carry out specific calculations. A FPGA contains an array of logical elements (LE), individual components which perform simple logical operations (Figure 2). Each LE is made of a 4-input lookup table (LUT) and a flip flop. The lookup table is an array that can be configured to execute different kinds of 4-bit operations, and the flip flop is used to store a one bit value. Each LE has its four input and one output data channels connected to vertical and horizontal channels to which all other LE in the FPGA are connected. LEs can be programmed to couple to each other and build digital circuits using electronic design automaton (EDA). Multiple interconnected LE in FPGA can perform independent calculations, allowing for parallel data set processing. Unlike CPU based computers with separated memories and processing units, FPGA use data-stream-based algorithms that execute operations by flowing data through the appropriate circuits.

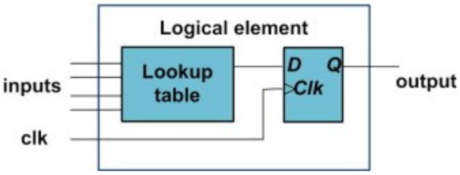

Figure 2: Internal structure of a logical element (LE) from [11].

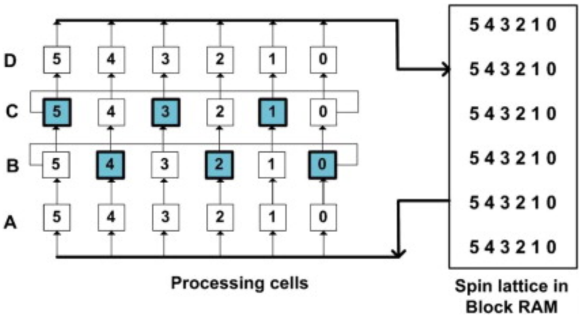

Figure 3: Processing matrix of a 6 × 6 Ising lattice from [11].

The Metropolis algorithm can be efficiently implemented in a FPGA and achieve significant speedups over devices that make use of CPUs. The procedure implemented in [11] is used here to update full rows of the lattice at each clock cycle. Figure 3 shows the functioning of a processing matrix. The matrix has two types of cells: the storage cells (blank) are used solely to store spin values, while the processing cells (blue) update the spin values following the Metropolis Algorithm. The processing cells are placed in an alternated way in the two middle rows of the 4-row sub-lattice of the matrix, and update their spin value according to the value in the neighboring storage cells. The spin values of the full lattice are stored in Block RAMs. The VERILOG statement that updates a single spin in the processing cells is

$$d <= ((m\char`^a + m\char`^b + m\char`^c + m\char`^d + R < p_1 + R < p_2) < 2)?m :\sim m. \tag{12}$$

Here m is the spin that is updated, a, b, c and d are the neighboring spins in the lattice, R is the register that stores the random number, $<=$ is the non-blocking assignment operator, ^ is the logical exclusive XOR operator, and p_1 and p_2 are the spin flipping probabilities

$$p_1 = \exp(-4/T) \text{ and } p_2 = \exp(-8/T), \tag{13}$$

where T is the temperature expressed in units of the interaction strength J. It can be verified that this particular update procedure fulfills the Metropolis algorithm [11].

A Xilinx Kintex-7 FPGA is used in this study. The FPGA is integrated in an Enclustra Mercury KX1 module inserted in a Mercury+ PE1 base board, which manages the power supply and communication with the computer and the QRNG. The QRNG used in this study is the one described in [5]. Simulations run for a total simulated time of 1300τ Monte Carlo Sweeps (MCS). Each *sweep* corresponds to a full update of the lattice, that is when each spin has been tested to flip at least once, on average. The system is let to equilibrate for 300τ, after which the magnetization is recorded every step for another 1000τ. Here τ is approximated with $\tau = L^z$, taking $z = 2.17$. This value of z is not obtained from our simulations, but is used as approximation to set an appropriate simulation length consistent with previous studies [10].

4 Results and discussion

Simulations were carried out in different Ising lattices, the smallest one consisting of 16×16 spins ($L = 16$) and the largest one 128×128, with 6 other lattices spaced by 16. In the following we show results of simulations performed in a conventional computer. Results from FPGA calculations are being performed by the company QuSide (https://www.quside.com). At this moment, only simulations with a lattice with $L = 32$ spins are finished. Once a similar range of lattice sizes as the one presented here is obtained, we will incorporate them to our study. Figure 4 shows an example of the (normalized) magnetization over time. Jumps in magnetization are caused by critical fluctuations at T_c and occur in all simulations.

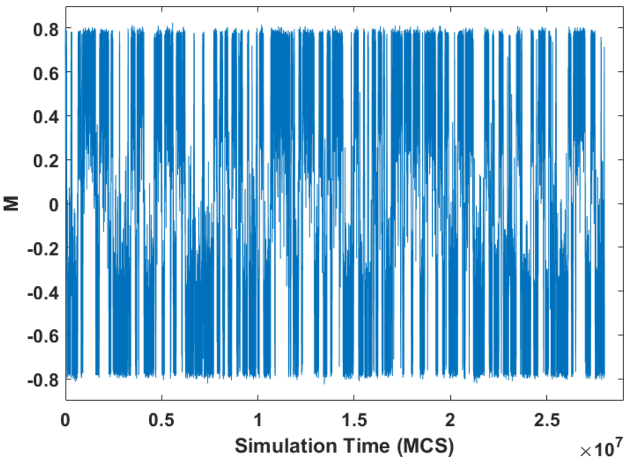

Figure 4: Magnetization of a 112×112 lattice as a function of simulation time t

From the magnetization, the time-displaced autocorrelation $\chi(t)$ of the system was obtained using Eq. (8). The time period over which the autocorrelation dropped off increased with the lattice size. The relaxation time τ for each lattice was then derived according to Eq. (10). Solving for τ, we get $\tau = -\frac{t}{\ln \chi(t)}$. Figure 5 shows the linear fit for autocorrelation in a 112×112 lattice, from which τ is obtained. Autocorrelation data was fitted between 0.3τ and 1.1τ. This fitting range was selected to avoid the initial nonlinearity while still retaining a good signal-to-noise ratio. The errors of the data points in figure 5 are smaller than the size of the symbols. Table 1 shows the linear relaxation time in MCS for the different lattices.

Lattice size L	16	32	48	64	80	96	112	128
Relaxation time $\tau(\times 10^4)$	0.1243	0.4555	1.3704	2.2903	3.9984	5.3615	7.2741	8.6739
$\ln(\tau/\tau_{FSS})$	0	-0.2034	0.0195	-0.0904	-0.0167	-0.1184	-0.1474	-0.2608

Table 1: Linear relaxation time for each lattice size. We compare τ to the theoretical τ_{FSS}.

The dynamic exponent z was obtained from τ using the finite size scaling (FSS) theory approach. In Eq. (8), an expression is given for the divergence of the correlation length near the critical temperature. For infinite lattices, ξ diverges to infinity at T_c; however, in finite simulations the correlation length is limited by the size of the lattice L. In fact, according to FSS theory the correlation length becomes *exactly* L at T_c, which, for a volume of L^d where d is the dimensionality of the system, it is the largest cluster of spins possible. Thus, Eq. (11) becomes

$$\tau_{FSS}(L) \sim L^z. \tag{14}$$

Here the subscript $_{FSS}$ is used to differentiate the theoretical τ_{FSS} from the simulated τ, but it is calculated in the same. The value of z can then be obtained by performing several simulations on lattices of different sizes and plotting τ against L on logarithmic scales. The slope of the resulting plot gives us the value of z.

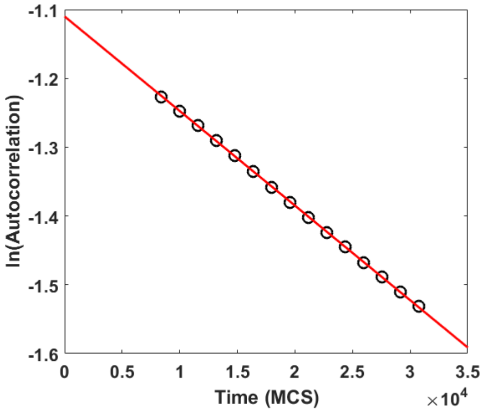

Figure 5: Log of the autocorrelation of the magnetization in a 112 × 112 Ising lattice as a function of the time delay t.

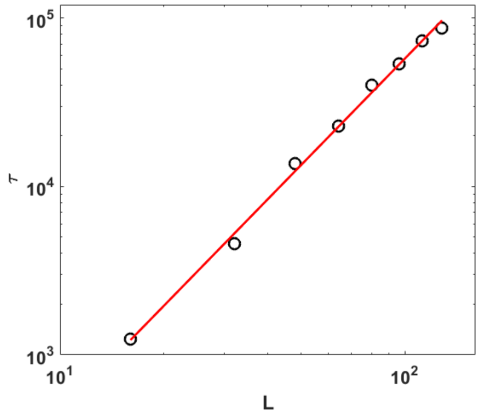

Figure 6: Log-log of the relaxation time τ and lattice size L. The slope gives a value of $z = 2.1008$.

The value of z was found to be 2.1008. To compare the obtained τ with the dynamic FSS theory, we plotted the log of τ/τ_{FSS} versus L (Figure 7). τ_{FSS} was calculated according to equation 14 by setting $z = 2.167$ and requiring that $\tau_{FSS} = \tau$ for $L = 16$. The constant of proportionality obtained was $a = 3.0559$.

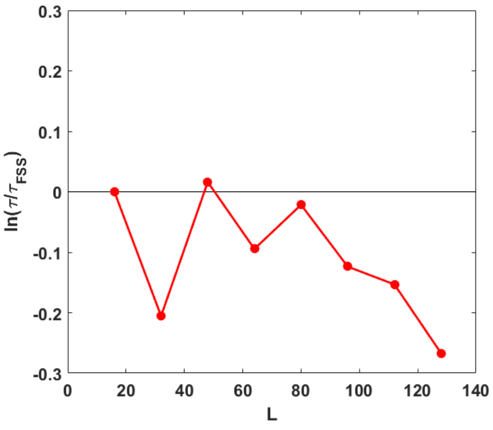

Figure 7: the log of the relaxation time τ over FSS theory $\tau_{FSS}(L)$.

The obtained z was not consistent with that of previous studies. While a relative error in the value of z was expected, it is difficult to quantify the effect of RNG in the outcome of the simulations due to the shortness of data. Nonetheless, figure 7 suggests that the error in τ increases as L grows. In fact, a more accurate estimation of z is obtained if data points from larger lattices are removed. A value of $z = 2.1341$ is obtained if the data point from $L = 128$ is removed, while removing data points $L = 112$ & $L = 128$ yields a value of $z = 2.1510$. The value closest to that of previous studies is obtained if only points 16 to 80 are used, with $z = 2.1713$. This may suggest that longer simulations are affected by correlations in the PRNG used by ordinary computers. Simulations with a QRNG should be free of such correlations.

Figure 8 shows the magnetization autocorrelation with $L = 32$ for the first 1000 MCS from a FPGA simulation using the QRNG. Autocorrelation shows an exponential drop-off, consistent with previous simulations. However, a single run is insufficient to estimate z, and the accuracy of the simulation cannot be evaluated without subsequent runs. More simulations on FPGA that are scheduled to be performed in the near future may help us obtain an accurate value of z and determine the quality of the QRNG employed.

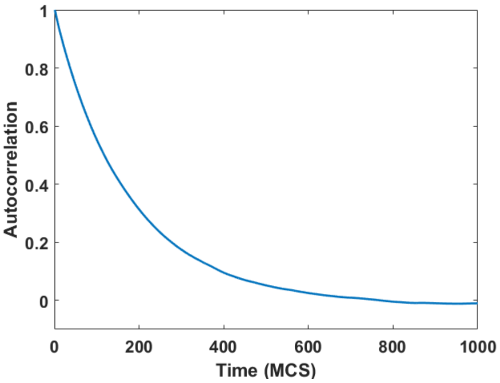

Figure 8: Autocorrelation in MCS of a $32x32$ lattice from a FPGA simulation.

5 Conclusions

We have illustrated that, for $L \leq 128$, the larger simulations with PRNG are prone to errors possibly due to the correlations in the random numbers. We presented a first calculation with the QRNG and we expect that upcoming simulations with TRN cure the problems in the determination of the dynamic critical exponent for the Ising model.

References

[1] Richard M Karp. An introduction to randomized algorithms. *Discrete Applied Mathematics*, 34(1-3):165–201, 1991.

[2] Miguel Herrero-Collantes and Juan Carlos Garcia-Escartin. Quantum random number generators. *Reviews of Modern Physics*, 89(1):015004, 2017.

[3] George Marsaglia. Random numbers fall mainly in the planes. *Proceedings of the National Academy of Sciences*, 61(1):25–28, 1968.

[4] Manabendra Nath Bera, Antonio Acín, Marek Kuś, Morgan W Mitchell, and Maciej Lewenstein. Randomness in quantum mechanics: philosophy, physics and technology. *Reports on Progress in Physics*, 80(12):124001, 2017.

[5] C Abellán, W Amaya, M Jofre, M Curty, A Acín, J Capmany, V Pruneri, and MW Mitchell. Ultra-fast quantum randomness generation by accelerated phase diffusion in a pulsed laser diode. *Optics express*, 22(2):1645–1654, 2014.

[6] V Du Preez, MGB Johnson, A Leist, and KA Hawick. Performance and quality of random number generators. In *International Conference on Foundations of Computer Science (FCS11)*, pages 16–21. CSREA, 2011.

[7] Alan M Ferrenberg, DP Landau, and Y Joanna Wong. Monte carlo simulations: Hidden errors from goodrandom number generators. *Physical Review Letters*, 69(23):3382, 1992.

[8] Makoto Matsumoto, Isaku Wada, Ai Kuramoto, and Hyo Ashihara. Common defects in initialization of pseudorandom number generators. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 17(4):15, 2007.

[9] Kenta Hongo, Ryo Maezono, and Kenichi Miura. Random number generators tested on quantum monte carlo simulations. *Journal of computational chemistry*, 31(11):2186–2194, 2010.

[10] Y Lin and F Wang. Linear relaxation in large two-dimensional ising models. *Physical Review E*, 93(2):022113, 2016.

[11] Y Lin, F Wang, X Zheng, H Gao, and L Zhang. Monte carlo simulation of the ising model on fpga. *Journal of Computational Physics*, 237:224–234, 2013.

[12] M Newman and G Barkema. *Monte carlo methods in statistical physics chapter 1-4*. Oxford University Press: New York, USA, 1999.