Test of Random Number Generators with Wolf Algorithm for Ising Model

Paweł Grabiński

Abstract—Random Number Generators are of various quality. As we need to rely on them in computer simulation, the best performing should be chosen. Although, the methods used to test the Random Number Generators are not homogeneous in quality as well. Here, we test the three chosen generators Mersenne Twister, Ranlux with level 0 quality, and R250 by computing well known properties of the Ising Model in the critical temperature with usage of the Wolf algorithm.

I. Introduction

There are numerous random number generators (RNGs). The need of develop better generators comes from the fact that performing any simulation we need to strongly rely on them while computing new results with high precision. Below, I reproduce the results of the Monte Carlo Ising model test for RNGs performed in [2].
In the part II, we describe the Ising model, its Hamiltonian, and its observables. In the part III, there is described the Wolf algorithm and why it is valuable especially in the critical temperature regime. Finally in the part IV, details of the experiment are described. And in the part V, results for the chosen RNGs are shown and discussed. The code available for reproduction of the results is available at the repository[1].

II. Ising Model

Ising Model is a spin model. It consists of regular lattice of equally separated nodes representing the spins (s_i) which can take on values up (1) or down (-1). In the presence of the magnetic field $\mathbf{B}$, the directions are taken as one parallel to it and the other anti-parallel to it.

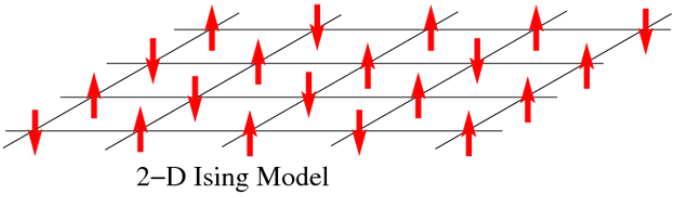


Figure 1. The 2-dimensional square lattice of spins (values ± 1) used in the Ising Model

The dynamics of the model is described by the following Hamiltonian function [4]:

$$\mathcal{H} = -J \sum_{<i,j>} s_i s_j - \mu \mathbf{B} \sum_i s_i. \tag{1}$$

[1]http://github.com/PGrabinski/SimulationMethods

Where the $< i, j >$ represents all the pairs of the nearest neighbors in the lattice. In one dimension (chain) it is 2 neighbors for a single spin, in two dimensions it is 4, and so on. $J = \pm 1$ is the coupling constant of the interaction. The positive value corresponds to the ferromagnetic and the negative to the ferromagnetic. In the below test, we consider $J = 1$ for the ferromagnetic system in presence of vanishing magnetic field $\mathbf{B} = 0$.

This model shows a phase transition in the $T_c \approx 2.269185$ between ferromagnetic phase where spontaneous magnetization of the material occurs and the paramagnetic phase where spins are distributed randomly unless the magnetic field is present.

Probability of a given configuration C in the fixed temperature is given as the softmax of the state's energy multiplied by appropriate factor:

$$P(C) = Z^{-1} \exp(-\beta \mathcal{H}(C)), \quad \beta = \frac{1}{k_b T} \tag{2}$$

Where the coefficient Z is the partition function defined as a sum over all possible states (configurations) of the system weighted by the Boltzman factor - its probability:

$$Z = \sum_i \exp(-\beta \mathcal{H}(C_i)) \tag{3}$$

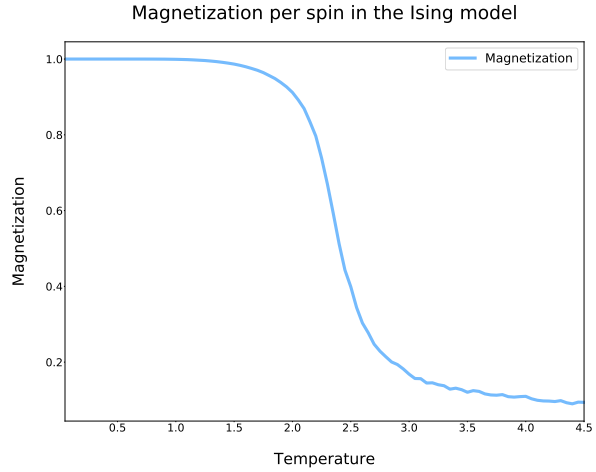
The thermal average of an observable A can be computed as:

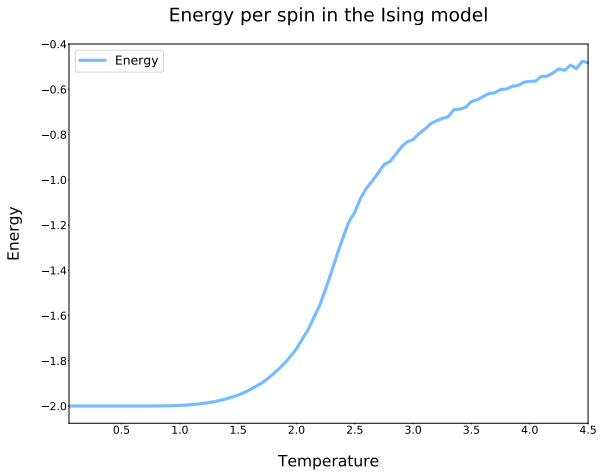
$$\langle A \rangle_T = Z^{-1} \sum_i A(C_i) \exp(-\beta \mathcal{H}(C_i)) \tag{4}$$

The main observables considered in the model are energy per spin, magnetization per spin, and specific heat per spin. The corresponding definitions are:

$$u = \frac{1}{N} \langle \mathcal{H} \rangle_T$$
$$m = \frac{1}{N} \langle \sum_{ij} s_{ij} \rangle = \frac{1}{N} \sum_{ij} \langle s_{ij} \rangle$$
$$c = \frac{\langle \mathcal{H}^2 \rangle_T - \langle \mathcal{H} \rangle_T^2}{N k_b T^2}$$

Due to the the exponential weighting, most of the configurations do not yield any impact on the calculated values while we sample from a simple distributions. To overcome this problem, usually, the Metropolis algorithm basing on the Monte Carlo Markov Chains is used. It is known to give proper results for the model except of the vicinity of the critical temperature T_c where it experiences so called critical slowing down - time needed to generate a physically relevant configurations grows to infinity.





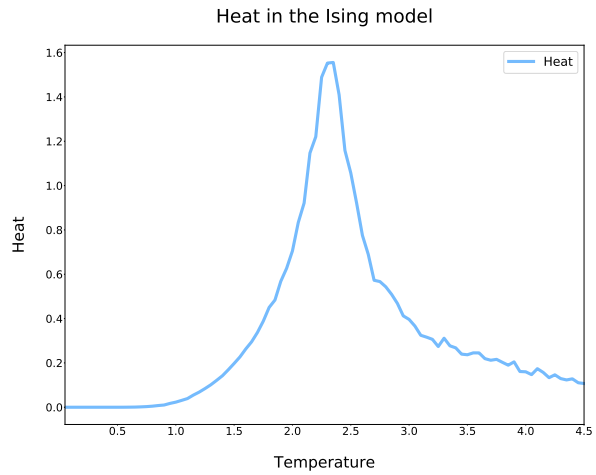


Figure 2. Results for a square lattice 16×16 as functions of temperature acquired with the Mersenne Twister generator and 10^4 samples per single temperature point.

III. Wolff Algorithm

Wolff Algorithm is a cluster algorithm. It does not consider single spins, but the bindings among them by building clusters of spins that are flipped together. For a fixed temperature, it consists of the following steps [1]:

1) Generate a random spin lattice configuration $\{s_i\}$.
2) For the given temperature compute the probability of the binding between two spins or equivalently the probability of adding a spin to the cluster:
$$p(T) = 1 - \exp(-\frac{2J}{k_b T}). \tag{5}$$
3) At random choose a spin s_i in the lattice,
4) With the probability $p(T)$ expand the cluster by adding the nearest neighbors (for $2d$ lattice 4 neighbors. In case of rejection, mark the binding between the spins as checked.
5) Recurrently for any added spin, add the nearest neighbors as in the step 4 if the bindings were not tested before.
6) After the recurrent adding, flip the spins of the cluster.
7) First, repeat the steps 3-6 for t_{therm} times to thermalize the lattice, and then repeat it for $N_{samples}$ times for different configurations.

Here, the Monte Carlo step is defined as a change of a single cluster.

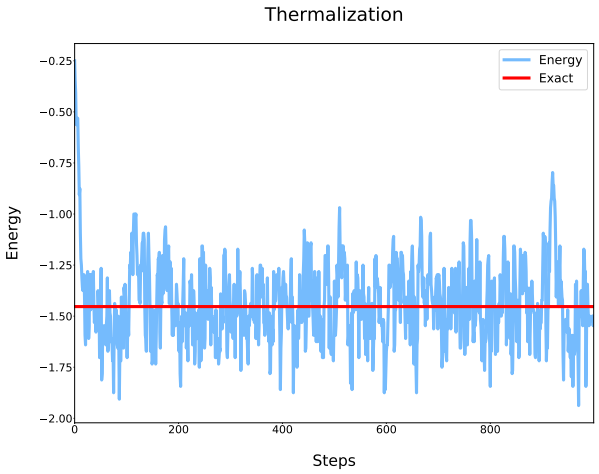


Figure 3. Thermalization achieved by flipping $t_{therm} = 1000$ clusters according to the Wolf algorithm. Here, the energies of the consecutive configurations (blue) are shown in contrast to the energy of the exact solution (red). Clearly, the system achieves the equilibrium in fraction of the steps assumed and then, it only fluctuates around the average value.

Main advantage of this algorithm is that it does not show the critical slowing down. In just a couple of steps, the equilibrium is achieved even in the critical temperature.

IV. Methods

In my experiment, I have tested the three RNGs: Mersenne Twister, Ranlux with level 0 quality, and R250.

Table I
Results for a square lattice 16×16 in the critical temperature $T_c = 2.269185$ acquired with three random number generators: Mersenne Twister (mt), Ranlux with luxury level 0 (rnlx0), and R250 (r250).

No.	RNG	10^N	Energy	Energy statistical error	Magnetization	Heat	Heat statistical error	Energy relative error	Heat relative error
1	mt	3	-1.45070	0.737843	0.710313	1.50607	-	0.003204	-
2	rnlx0	3	-1.46523	0.745378	0.721063	1.57804	-	-0.016322	-
3	r250	3	-1.45275	0.738532	0.715008	1.40929	-	0.000425	-
4	mt	4	-1.45157	0.233451	0.714325	1.49298	0.0126199	0.006400	-0.454837
5	rnlx0	4	-1.45200	0.233524	0.712613	1.49560	0.0181682	0.004556	-0.171729
6	r250	4	-1.45205	0.233500	0.711771	1.46915	0.0125929	0.004343	-2.34815
7	mt	5	-1.45250	0.073872	0.713161	1.49845	0.0059356	0.007635	-0.0454882
8	rnlx0	5	-1.45117	0.073808	0.712004	1.50497	0.00609168	0.025661	1.02599
9	r250	5	-1.45383	0.073926	0.715190	1.46037	0.00609557	-0.010362	-6.29145
10	mt	6	-1.45294	0.023367	0.713428	1.49730	0.00204259	0.005307	-0.695196
11	rnlx0	6	-1.45222	0.023357	0.712511	1.51139	0.00195741	0.036134	6.47284
12	r250	6	-1.45442	0.023387	0.715120	1.46539	0.00190334	-0.057980	-17.5113
13	mt	7	-1.45308	0.007390	0.713425	1.49731	0.000642402	-0.002165	-2.19489
14	rnlx0	7	-1.45235	0.007387	0.712611	1.50861	0.000636599	0.096658	15.5357
15	r250	7	-1.45431	0.007395	0.714950	1.46253	0.000613567	-0.168491	-58.983

The main task was calculating energy per spin, specific heat per spin, and their corresponding statistical errors in the point of the critical temperature $T_c = 2.269185$. Thermalization was done over $t_{therm} = 1000$ Monte Carlo Steps. After that every new configuration was considered for the computations. The number of samples ranged from 10^3 to 10^7. Every random number generated in the simulations were taken from a single RNG.

The specific heat was calculated from samples of 1000 energy values. Then, such a sample of the specific heat values was used to calculate its mean and variation.

V. Results and Discussion

The results are shown in the table I. For the case of 10^3 samples, the statistical error of the specific heat was not computed due to lack of statistical significance. The most important results are the last two columns "Energy relative error" and "Heat relative error". These values are relative error of the estimation defined as:

$$A_{diff} = \frac{\hat{A} - A_{exact}}{\hat{\sigma}}$$

Where exact values where taken according to [3]:

- energy per spin: $u_{exact} = -1.453064$,
- specific heat: $c_{exact} = 1.49872$.

As it is visible in the table I, the results for energy are reasonable for all examples as all of them are apart less than one standard deviation from the exact value. For the specific heat, the results are reasonable for all of the RNGs in the region up to 10^5 samples. For 10^5 samples, relative errors smaller than 10σ are reasonable, but the values over 5σ are alarming. The problem becomes visible for $10^6 - 10^7$ as we see that both Rnlx0 and R250 RNGs achieve errors many times higher. Only the Mersenne Twister generator remains stable for that big sample. This shows that from the three used RNGs only the Mersnne Twister should be used for the very precise computations of more complicated observables as the RNG of the highest quality.

References

[1] David, P., P. Landau, and K. Binder. guide to Monte Carlo simulations in statistical physics. Cambridge University Press, 2014.

[2] Ferrenberg, Alan M., D. P. Landau, and Y. Joanna Wong. "Monte carlo simulations: Hidden errors from "good"random number generators." Physical Review Letters 69, no. 23 (1992): 3382.

[3] Ferdinand, Arthur E., and Michael E. Fisher. "Bounded and inhomogeneous Ising models. I. Specific-heat anomaly of a finite lattice." Physical Review 185, no. 2 (1969): 832.

[4] Greiner, Walter, Ludwig Neise, and Horst Stöcker. Thermodynamics and statistical mechanics. Springer Science & Business Media, 2012.