Simulating the Ising Model

Max Heimann 580560, Lyding Brumm 604522, Burak Varol 623941

January 2024

1 Introduction

In this report, we present our code which simulates the Ising model on a D dimensional lattice. The main challenge of the problem was the exponential scaling of the number of configurations: There are two spin values (-1 and 1) associated to each lattice point. For N lattice points, we have 2^N total configurations, which gives $\approx 10^{30}$ for 10×10 lattice.

The observation that not every configuration is equally important leads us to just include the configurations which gives the biggest contribution. This is done by the Metropolis Algorithm. We began the Markov Chain with a random initial configuration and each iteration in the chain refers to a Metropolis Iteration. Since the algorithm is probabilistic in nature, we use number of different Markov Chains (replicas) to calculate the statistical error. In the following, a list of modules can be found

Module Name	Summary
grid.c/grid_test.c	The grid which represents the Ising Model,grid related functions and their test
metropolis.c/metro_test.c	The Metropolis Algorithm and test
statistics.c/statistics_test.c	Statistical analysis functions calculating the mean values and errors; their tests
imdf.c/imdf_test.c	Data format which extracts the data from C and saves it in an analysable way and its test
main.c /test.c	The main code and its test
delta_h_test	the test of the delta h function

Table 1: Summary of modules

2 Problem

We have to calculate:

$$\langle M(s)\rangle = \frac{\sum_s M(s) e^{\frac{H(s)}{k_b T}}}{\sum_s e^{\frac{H(s)}{k_b T}}} \tag{1}$$

with

$$\frac{H(s)}{k_b T} = -\beta \sum_n \sum_k s(n)s(n+e_k) + b \sum_n s(n) \tag{2}$$

where $\beta = \frac{J}{k_b T}, b = \frac{\mu B}{k_b T}$.
This is however very hard to calculate and would take an insane amount of time. With the Markov chain,

$$s_1 \to s_2 \to s_3 ... \to s_{N_{thermal}} \to ... \to s_{N_{config}} \tag{3}$$

The observable for one replica is calculated as:

$$\langle M\rangle = \frac{1}{N_{config} - N_{thermal}} \sum_{k=N_{thermal}+1}^{N_{config}} M(s(k)) \pm \Delta M(s(N_{config}))) \tag{4}$$

The observable over all replicas can be calculated as (R = number of replicas):

$$\bar{\bar{M}} = \frac{1}{R} \sum_{r=1}^{R} \bar{M}^{(r)} \tag{5}$$

The error is

$$err(\bar{\bar{M}}) = \sqrt{\frac{1}{R(R-1)} \sum_{r=1}^{R} (\bar{M^{(r)}} - \bar{\bar{M}})^2)} \tag{6}$$

3 Experiments

We created 500 replicas and used the values $D = 2$, $N = 100$, $\beta = 0.43$, $b = 0.01$, $N_{config} = 1000$. For each replica, we use a different seed (from 1 to 501) and luxury number 2 to initialize the pseudo random number generator RANLUX in order to get statistically independent replicas. To find the $N_{thermal}$ value, we inspect the energy (1) and magnetization (2) graphs, look for the iteration at which the graph starts to oscillate and eyeball $N_{thermal}$ as $N_{thermal} = 200$.

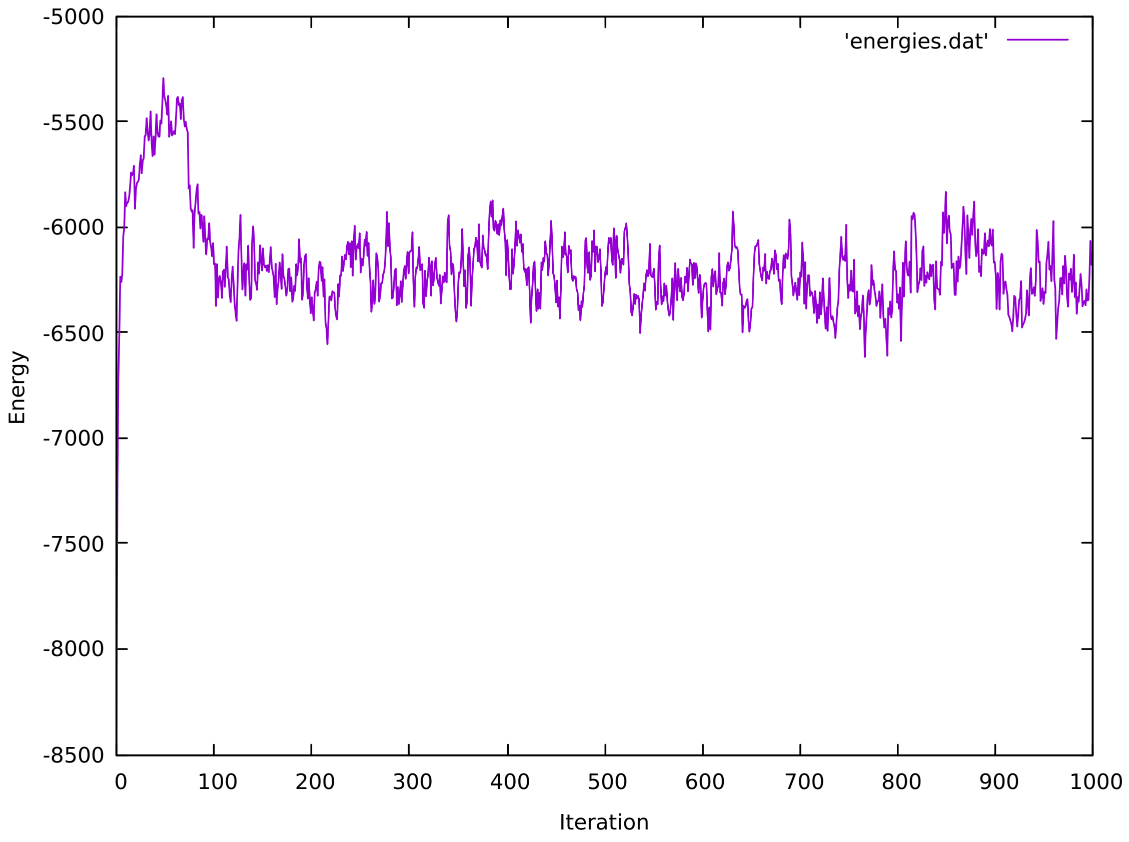

Figure 1: The Energy Thermalisation Graph

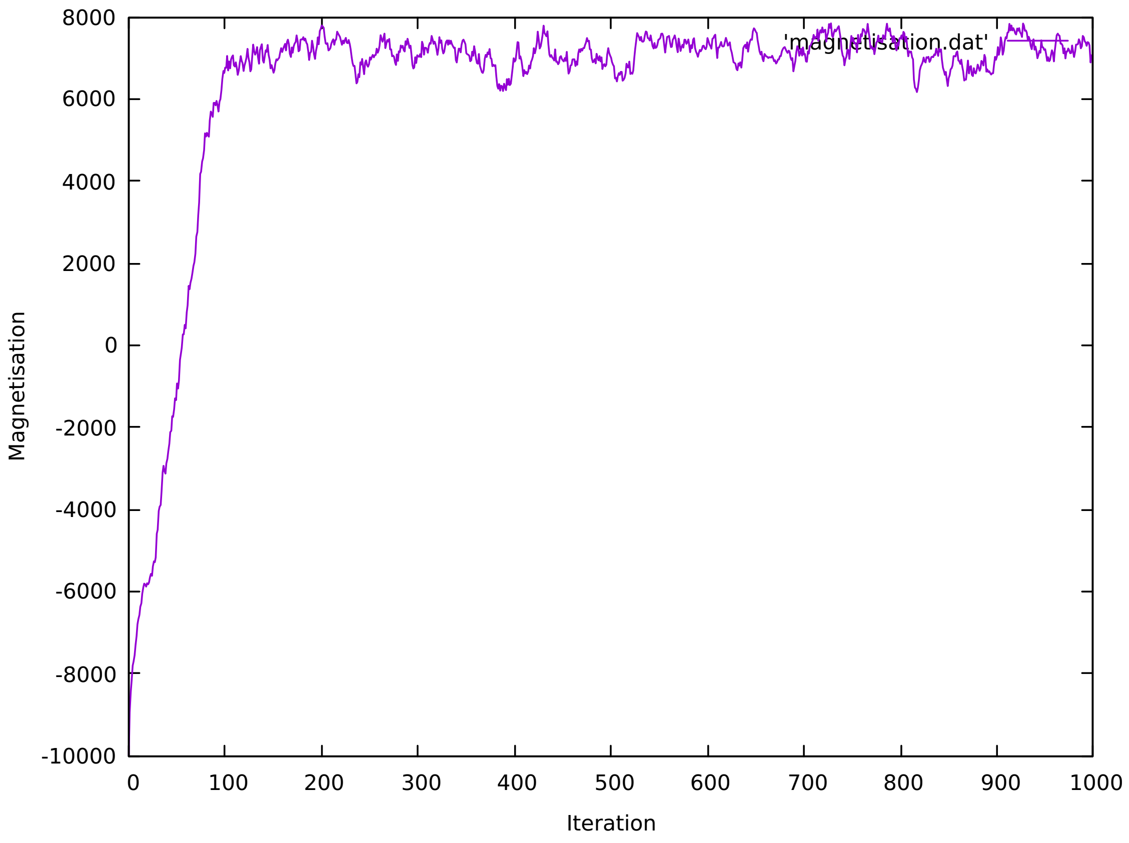

Figure 2: The Magnetization Thermalisation Graph

With these values in mind, we find the energy and magnetization as the following (using (5) and (6)):

$$E = (-6237.077049 \pm 0.628868)k_B T \tag{7}$$
$$M = 7184.701846 \pm 2.171860 \tag{8}$$

3.1 Large R and Large N_{config} limit

- If we send the number of replicas to infinity, the statistical error must go to zero. From the definition of standard deviation and the formula (6), we can write:

$$err = \frac{\sigma}{\sqrt{R-1}} \tag{9}$$

  For $R \to \infty$, we obviously expect the statistical error to vanish. This was experimentally verified see figure 5 and 6

- The Central Limit theorem says that, the sampling distribution of the mean will be always normal distributed with the $\sigma_{mean} = \frac{\sigma}{N}$ if the sampling size is large enough. We can apply the theorem to our case since the sampling size will be large, the samples are independent and identically distributed, and we have a finite variance. In our case, $N = N_{config}$ and the σ_{mean} is (roughly) the statistical error. We see that $\sigma_{mean} \to 0$ if N goes to ∞, meaning the statistical error will vanish for large enough N_{config}. This was experimentally verified see figure 3 and 4.

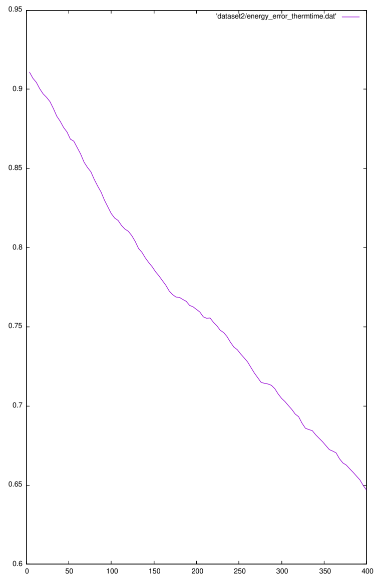

Figure 3: Energy Uncertainty as a Function of the Number of Datapoints

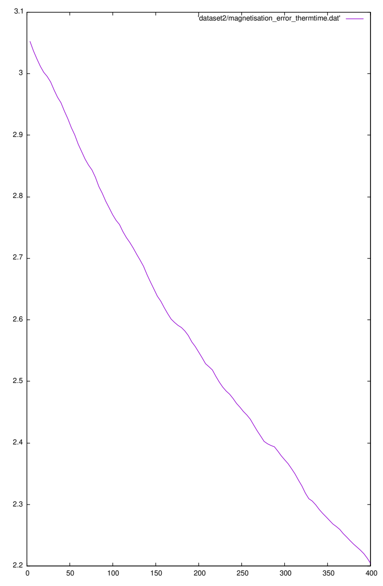

Figure 4: Magnetisation Uncertainty as a Function of the Number of Datapoints

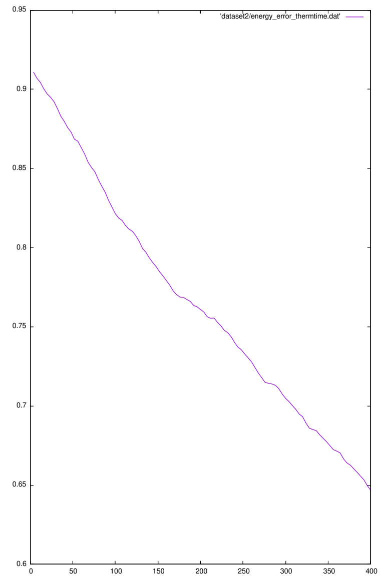

Figure 5: Energy Uncertainty as a Function of the Number of Datapoints

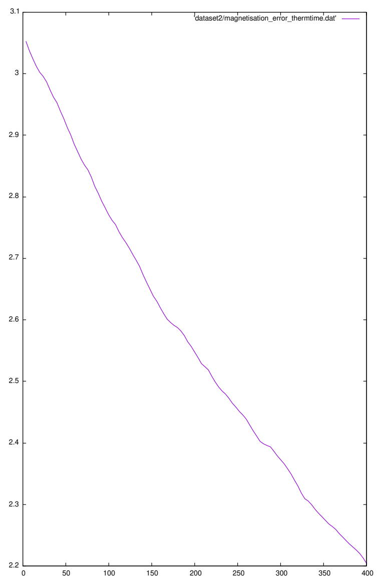

Figure 6: Magnetisation Uncertainty as a Function of the Number of Datapoints

We present our results for large R and large N_{config} in the following way: For the simulated data, we choose $N_thermal$ higher or consider less replica. By this method, we can demonstrate for increasing N and R the error decreases.

4 Probability distributions

To get an idea on how actual probability distribution looks like, we plotted the probability of the energy/magnetization is within one bin of a certain width. For the energy $\Delta E = 8k_BT$ and for the Magnetization $\Delta M = 30$. Plotting this against the normalized frequency we get the probability distribution. These are shown in 8 and 7.

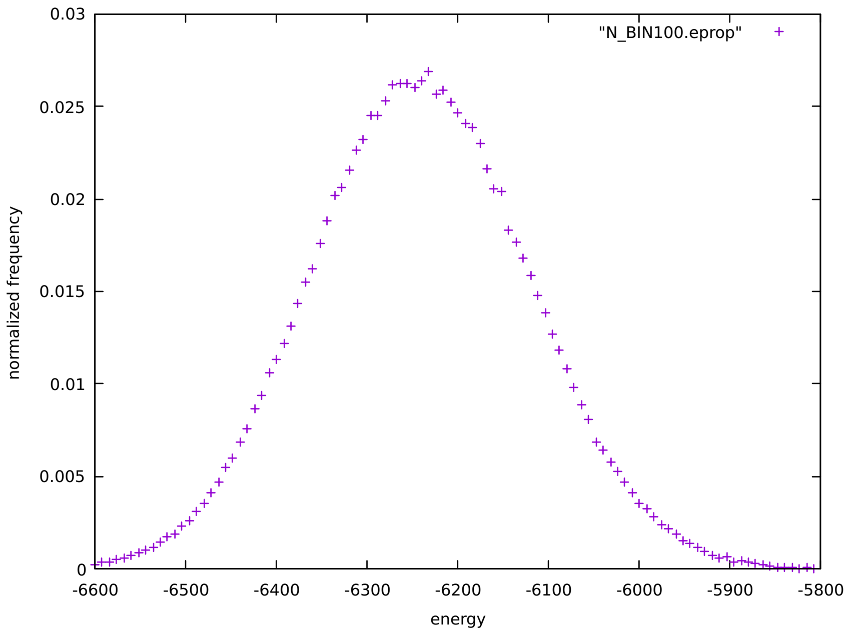

Figure 7: Probability distribution for the energy

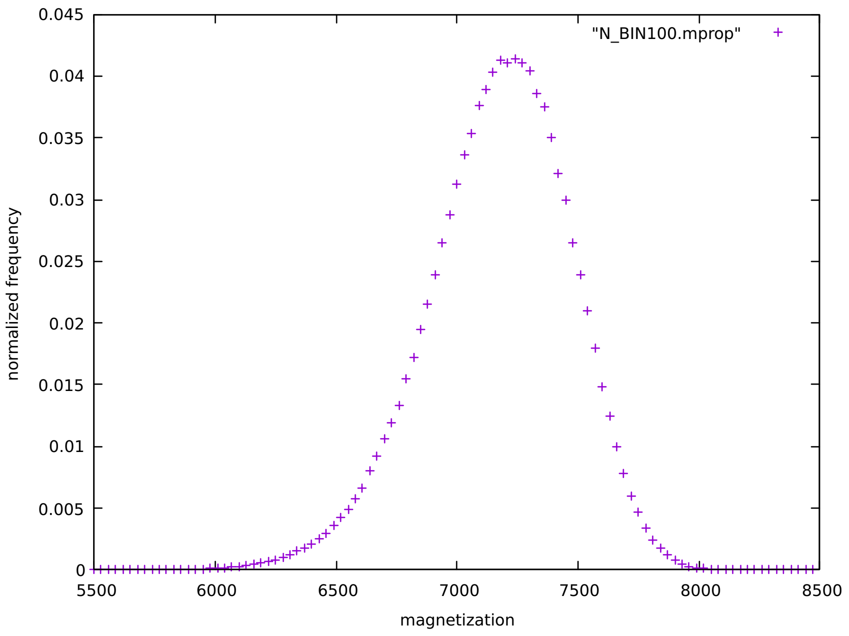

Figure 8: Probability distribution for the magnetization

5 Usage of the Code

The code is purely written in C and can be compiled with a Makefile. Be aware that the RANLUX library is present in the 'lib' subdirectory and compiled for the architecture of your machine. The main part of the code can be build with 'make && make analysation'. After a succesfull build the binary 'main' and 'analysation' will be present in the subdirectory 'app'. The executable 'main' implements the Metropolis algorithm and produces two files storing the energy and magnetisation of each iteration in a new-line seperated format, ready for gnuplot's plot function. Its parameters are:

- Dimension of the grid (integer)
- Number of Gridpoints per Dimension (integer)
- β (float)
- b (float)
- Number of iterations (integer)
- seed (integer)

It therefor calculates one replica. Running the main program therefor might look something like './app/main 2 100 0.43 0.01 1000 12345' The second executable takes two arguments, the path to a directory containing replica files and the thermalisation time. It calculates the expectation value of the energy and magnetisation as well as their error. Running the analysation program might look something like './app/analysation ./dataset2/ 300'.