

NUMERICAL ANALYSIS OF THE PHASE TRANSITION IN THE ISING MODEL

FRANCISCO GARCÍA FLÓREZ

1 Introduction

2 General considerations

In this project we are going to consider the following Hamiltonian for the system

$$H = -J \sum_{\langle ij \rangle} s_i s_j - B \sum_i s_i , \quad (1)$$

where J and B are positive constants, referencing respectively the alignment energy and a possible external magnetic field, which we will consider zero for the first part of this study. Since this is a numerical simulation of the Ising model, there are several core remarks that we will explicitly state in the following.

2.1 System size, boundary conditions and dimensionality

Firstly, we will consider systems with different numbers of lattice points, defined as $N = L^d$, that we will use to increase the precision of some measurements and avoid size-specific effects. In relation to this remark, since we cannot simulate an infinite system we are imposing helical boundary conditions, defined as

$$s_i = s_{\text{mod}(i \pm L^n, N)} \quad \forall n \in [0, d) , \quad (2)$$

where d is the space dimension of the system. We are going to be considering both 2D and 3D systems only, but we will keep our expressions as dimension-independent as possible.

2.2 Monte-Carlo algorithms

This is a Monte-Carlo simulation of the Ising model, meaning we will compute several quantities like Energy or Magnetization using different configurations of the system. To find these configurations we will follow two different algorithms: Metropolis and Wolff's. We are not going to go into the details of how and why they work, just the basics:

- **Metropolis algorithm with single spin-flip dynamics.** As the name suggests, we will be performing single lattice point spin flips, N of them per simulation

step. Because it's a Metropolis algorithm the acceptance ratio is given by

$$A(\mu \rightarrow \nu) = \begin{cases} e^{-\beta(E_\nu - E_\mu)} & \text{if } E_\nu - E_\mu > 0 \\ 1 & \text{otherwise} \end{cases} . \quad (3)$$

Since each lattice point will be picked, on average, the same amount of times we will use simulation steps per lattice site as the unit of measure for time in every simulation generated with this algorithm.

- **Wolff's algorithm.** In this case we are not going to flip only one spin, but instead we will look for clusters of parallel spins and flip all of them with some probability. Using this algorithm near the critical point increases significantly the speed of the simulation. The probability of adding one extra spin to the cluster can be chosen to be

$$P_{\text{add}} = 1 - e^{-2\beta J} . \quad (4)$$

In this case we cannot use simulation steps per lattice point as a constant unit, so instead we will cluster flips as time for this algorithm. In the implementation we choose one cluster flip per simulation step, so in the following we will simply call t "Simulation steps".

2.3 Autocorrelation functions and Correlation time τ

Finally, for all these simulations to work we need to compute averages of quantities once the system has reached equilibrium, process also known as *thermalization*. There are several methods to check whether or not the system has equilibrated, but the simplest one is looking at the Magnetization and Energy as a function of simulation steps (per lattice point).

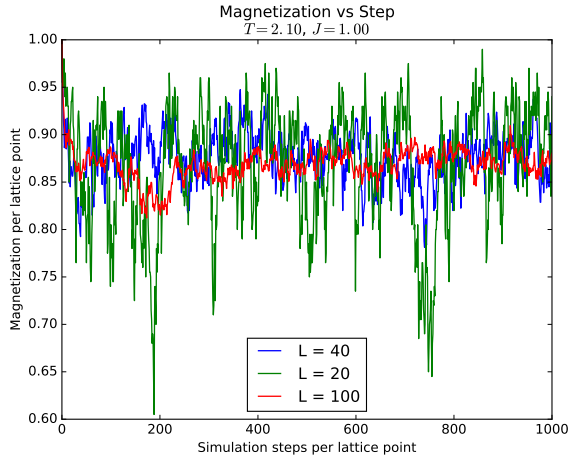


Figure 1: Plot of the magnetization per lattice point as a function of simulation steps per lattice point (sslp), using Metropolis algorithm.

In this plot we can easily see a thermalization time of around 100 sslp, however, as we could expect it depends heavily on T . In this simulation we are starting from a system with uniform spins, equivalent to a system at exactly $T = 0$, and thus the system equilibrates quicker at $T = 2.1$ (as shown in the plot) than at $T = 10$, after the phase transition. Later on we will take a second look at this.

Next, we can do the same with Wolff's algorithm to find:

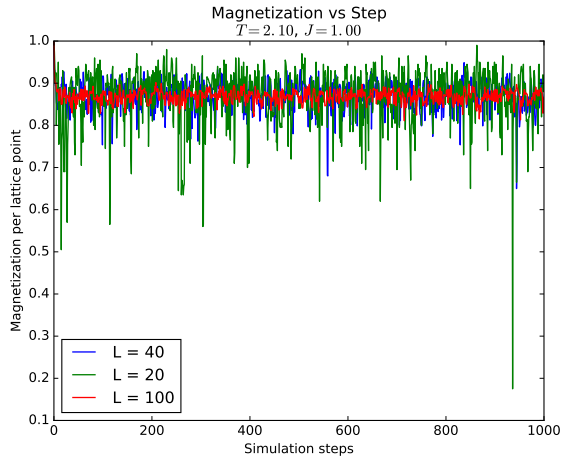


Figure 2: Plot of the magnetization per lattice point as a function of simulation steps, using Wolff's algorithm.

Of course, this algorithm provides significantly lower correlation times, as we will see later it's around 10 to 20 simulation steps. Plotting the energy as a function of steps would give a similar result, so we can omit the explicit plot. There is a better way to compute how much time should we wait, and it involves computing what is called autocorrelation function. We can compute this both using the energy or the

magnetization, using the following expression

$$\xi_M(t) = \frac{1}{t_{\max}-t} \sum_{t'=0}^{t_{\max}-t} M(t')M(t'+t) - \frac{1}{t_{\max}-t} \sum_{t'=0}^{t_{\max}-t} M(t') \times \frac{1}{t_{\max}-t} \sum_{t'=0}^{t_{\max}-t} M(t'+t) \quad (5)$$

And equivalently for the energy. From $\xi(t)$ we can compute the correlation time τ as

$$\tau = \frac{1}{\xi(0)} \int_0^\infty dt \xi(t), \quad \xi(0) = \langle M^2 \rangle - \langle |M| \rangle^2, \quad (6)$$

where we could change $M(t)$ by $E(t)$ and do the same with the energy. Firstly, using equation (5) we find the following results, using the Metropolis algorithm, for magnetization:

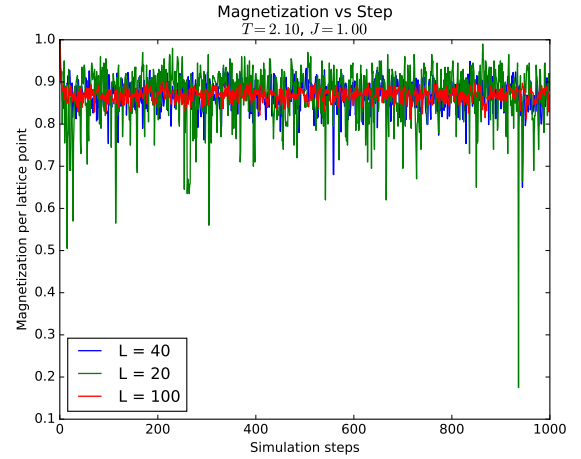


Figure 3: Plot of the autocorrelation function $\xi(t)$ using magnetization.

And energy:

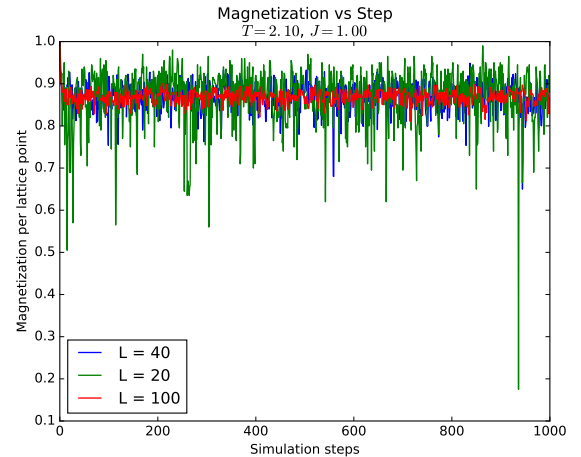


Figure 4: Plot of the autocorrelation function $\xi(t)$ using energy.

3 Phase transition

3.1 Energy $E(T)$ and Magnetization $|M(T)|$

3.2 Specific Heat $c_s(T)$ and Magnetic Susceptibility $\chi(T)$

3.3 Critical exponents