# Ising Model and its simulation using Monte Carlo methods with Metropolis sampling

Íris Damião e João Campagnolo
Faculdade de Ciências, Universidade de Lisboa
Criticalidade e Sistemas Complexos, Professora Patrícia Faísca

## 1 Introdução

The mechanism by which some materials form permanent magnets is called Ferromagnetism. Among the several types of magnetism, (paramagnetism, diamagnetism, antiferromagnetism...) Ferromagnetism is the strongest type. The term Ferromagnetism was originally used for any material that could exhibit spontaneous magnetization: a net magnetic moment in the absence of an external magnetic field. There are two levels of magnetic alignment that result in this behavior. One is **ferromagnetism** in the strict sense, where all the magnetic moments are aligned. The other is **ferromagnetism**, where some magnetic moments point in the opposite direction but have a smaller contribution, so there is still a spontaneous magnetization.

The simplest description of this phenomenon is the Ising model, which consists, very succinctly, in a system represented by $N$ spins that can be in one of two microstates $spinup = +1$, $spindown = -1$ under the influence of an external magnetic field,$H$. This model is extremely important not only because there is a point where spontaneous magnetization occur by the alignment of all the spins (even when a magnetic external field is not present), but also because is essential in the study of second order phase transitions. Therefore, in this project we will try to simulate computationally the Ising Model using Monte Carlo methods with Metropolis sampling, hoping in that way be able to illustrate the physical phenomenon of ferromagnetism.

### 1.1 Origin of Ferromagnetism

Magnetism arises due to quantum-mechanical effects: **spin** and the **Pauli exclusion principle**. An electron, due to its spin, carries a magnetic dipole moment, producing a magnetic field. The spin of the electron can only be in a state: "up" or "down". Besides the spin of the electron, the orbital angular momentum of the electron about the nucleus also contributes to the ferromagnetic behaviour. The alignment of these magnetic dipoles in matter generates a larger macroscopic magnetic field.

Because the electrons all exist in pairs with opposite spin, every electron's magnetic moment is cancelled by the opposite moment of the second electron in the pair. Only atoms with unpaired spins can have a net magnetic moment. The unpaired spins tend to align in parallel to an external magnetic field (paramagnetism). However, in ferromagnetism, the dipoles tend to align spontaneously, giving rise to a spontaneous magnetization, even without an external field.

For a ferromagnet, an increase in temperature means an increase in thermal motion, or entropy, that competes with the ferromagnetic tendency for dipoles to align. Beyond a certain temperature, the Curie temperature, there is a second-order phase transition and the system can no longer sustain a spontaneous magnetization, although it still responds paramagnetically to an external field. Below

that temperature, there is a spontaneous symmetry breaking and magnetic moments become aligned with their neighbors. The Curie temperature itself is a critical point where the magnetic susceptibility is theoretically infinite and domain-like spin correlations fluctuate at all length scales.

The study of ferromagnetic phase transitions, via the **Ising model**, came to show that **mean field theory** approaches fail to predict the correct behaviour at the critical point, and these have to be replaced by **renormalization group theory**.

## 1.2 Ising Model

Experiments conducted in the neighbourhood of critical points suggest that critical exponents assume the same universal values, contradicting the classical theories. The values of the critical points in this context depend on: the dimension of the physical systems, the dimension of the order parameter and the range of microscopic interactions. In order to construct a microscopic theory of the critical behavior, we analyze very simple models, such as the Ising model. The Ising spin Hamiltonian is given:

$$H = -J \sum_{<ij>} s_i s_j - h \sum_i s_i \tag{1}$$

where $s_i$ is a random variable assuming the values $\pm 1$ on the sites $i = 1, 2, ..., N$ of a d-dimensional lattice and $J$ is the coupling constant. The first term represents the interaction energies introduced to bring about an ordered ferromagnetic state (if $J > 0$). The second term involves the interaction between the applied field $h$ and the spin system. In order to solve the Ising problem, we have to obtain the canonical partition function:

$$Z_N = Z(T, h, N) = \sum_{s=\pm 1} exp(-\beta H), \tag{2}$$

where the sum is over all configurations of spin variables. From tils partition function, we have the magnetic free energy per site: (demonstrated in the 2nd class, using the transfer matrix method)

$$g = g(T, h) = lim_{N\to\infty}\Big[ -\frac{1}{\beta N} ln Z_N \Big] \tag{3}$$

this one-dimensional solution sees the free energy as an analytic function of $T$ and $h$, which precludes the existence of a spontaneous magnetization (and of any phase transition) (Pierls argument). Onsager obtained an analytical solution for the Ising model on a square lattice, with nearest neighbours interactions and no external field. For $T \to T_C$ the specific heat diverges according to a logarithmic asymptotic form:

$$C_H = 0 \sim ln|T - T_C| \tag{4}$$

with a well-defined critical temperature, $\frac{k_B T_C}{J} = \frac{2}{ln(1+\sqrt{2})}$, meaning that the free energy is not analytic at $T_C$.

The solution of the Ising model in three dimensions remains an open problem. There have been many efforts to obtain long series expansions for several thermodynamic quantities associated with the three-dimensional Ising model. From refined asymptotic analyses of these series, we obtain a range of values for the critical exponents in agreement with experimental measurement, represented in Table 1.

| | Landau | Ising ($d = 2$) | Ising ($d = 3$) | Experiments |
|---|---|---|---|---|
| $\beta$ | 1/2 | 1/8 | $\approx 5/16$ | $0.3 - 0.35$ |
| $\gamma$ | 1 | 7/4 | $\approx 5/4$ | $1.2 - 1.4$ |
| $\delta$ | 3 | 15 | $\approx 5$ | $4.2 - 4.8$ |
| $\alpha$ | 0 | $0(log)$ | $\approx 1/8$ | $\approx 0$ |

Table 1: Table with the values for the critical exponents $\beta$, $\gamma$, $\delta$ e $\alpha$ obtained from an asymptotic analysis for theirs series expansions, and by experiments (last column).

### 1.3 Mean Field Theory

In this Section, we are discussing various approaches to obtain a mean-field solution to the Ising model, carrying out the approximations that are commonly referred to as "mean-field approximations".

First, we decompose the spins $s_i$ into their mean value (the magnetization), and fluctuations around it $s_i = m + \delta s_i$. Then we apply two approximations:

1. We **neglect the fluctuations**: $\delta s_i = 0$, meaning that $\delta s_j \delta s_i = 0$. So, Equation 1 becomes:

$$H = -J \sum_{<ij>} (m^2 + m\delta s_i + m\delta s_j + \delta s_i \delta s_j) - h \sum_i s_i \tag{5}$$

2. We assume the system to be **invariant under translation** (big aproximation), meaning that $\delta s_i = \delta s_j$. So, from the previous equation we get:

$$H = -J \sum_{<ij>} m^2 - Jm \sum_i 2\delta s_i - h \sum_i s_i \tag{6}$$

The coordination number, $z$, describes how many pairs of interactions exist in a lattice (1D: $z = 2$; 2D:$z = 4$; 3D:$z = 6\ldots$). There are $Nz/2$ pairs of interactions in an $N$-dimensional lattice. So, we have:

$$H = -\frac{JNz}{2} m^2 - Jmz \sum_i \delta s_i - h \sum_i s_i = -\frac{JNz}{2} m^2 - \sum_i s_i (h - Jmz) \tag{7}$$

Which implies that:

$$H \approx -\sum_i s_i (h + Jmz) \tag{8}$$

Where $h$ corresponds to the external field and $Jmz$ to an internal field, which, added together, correspond to the **Mean Field**. Now we can proceed to calculate the Mean Field magnetization, using the partition function. We find that:

$$tanh[\beta(Jmz + h)] = m \tag{9}$$

This is a transcendental equation, that can be solved using graphical methods: When $\beta Jz = 1 <=> Jz = k_B T <=> T_C = Jzk_B$. Note that $T_C$ depends only on the coordination number, $z$, of the lattice. However, $z$ alone cannot define a lattice uniquely. For example, triangular lattice in 2 dimensions and simple cubic lattice in 3 dimensions both have coordination number $z = 6$. To understand why the predictions for $T_C$ get better as $z$ increases, we must take into account the approximations that were made. Namely, the neglecting of the fluctuations. This approximation has

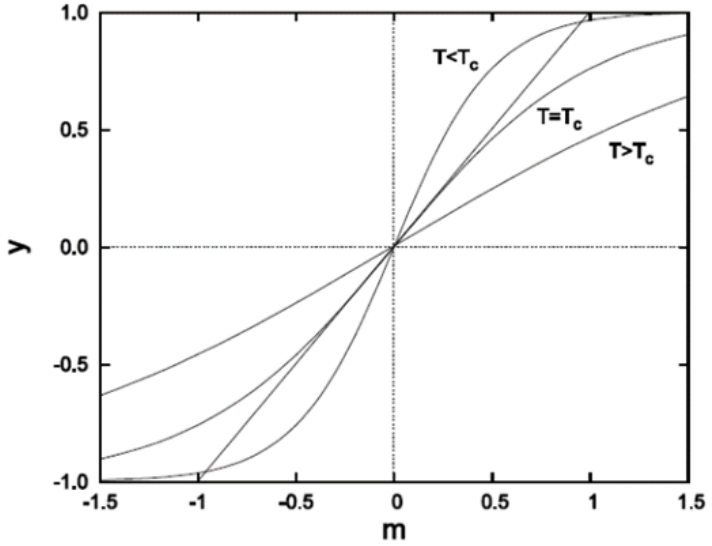


Figure 1: Graphic solution for Equation 10 for the 3 different cases $T < T_C$, $T = T_C$ and $T > T_C$.

a bigger influence when there are fewer first neighbors (the only ones that are considered in the Ising model). Since the fluctuations correspond to a random variable, that deviates $s_i$ from $m$, the more first neighbors the less the value of $s_i$ will strand away from the average value, $m$.

Landau was motivated to suggest that the free energy of any system should be analytic and that it obeys the symmetry of the Hamiltonian. Given these two conditions, one can write down (in the vicinity of the critical temperature, $T_C$) a phenomenological expression for the free energy as a Taylor expansion in the order parameter. For example, expressing the free energy in even powers of the magnetization, and retaining the terms of 2nd and 4th order:

$$f(m) = f_0 + am^2 + bm^4 \tag{10}$$

Where the coefficient of the highest even power of the order parameter ($b$, in this case) must be positive, so that the free energy corresponds to a minima in an equilibrium point. The parameter $a$ measures the proximity to the critical point.

### 1.4 Renormalization Group Theory

RG theory is useful since the Mean Field Theories don't account for fluctuations, and these cannot be neglected for systems with dimensions that are inferior to the critical dimension ($d = 4$). The renormalization group is intimately related to *scale invariance*: symmetries in which a system appears the same at all scales. The RG theory can be applied to the Ising model, according to the following procedure: 1) Coarse Graining; 2) Obtain a partition function that resembles in shape to the original Ising model partition function for $N$ spins; 3) Rescaling. This theory explains the scaling hypothesis, the universality classes and correctly predicts critical exponents.

## 2 The Ising Model and The Metropolis Algorithm

Previously we introduced the Ising Model and we outlined its importance in the description of Ferromagnetism. However, the Ising Model can often be very difficult to evaluate numerically since typically there are many states in the system. For instance, considering a system with $L$ spins where

each spin can assume the values $-1, 1$, then there are $2^L$ possible configurations for it. Which even for a small lattice represents a huge number. This motivates using Monte Carlo methods to simulate the Ising Model.

The basic idea behind Monte Carlo algorithms is the simulation of the random thermal fluctuation of a system from state to state over the course of an experiment [1]. In order to do so in a Monte Carlo calculation, a model of the system is created and then it passes through a variety of states in such a way that the probability of the system being in any particular state at a given time $t$ is equal to the weight $w_u(t)$ which that state would have in the real system. To achieve this we have to choose a dynamics for the simulation, i.e. a rule for transiting from one state to another with an appropriate probability, according to the real system. The choice of these dynamics distinguish the Monte Carlo methods. In this work, we approached the Ising model with the Metropolis algorithm, which is described in the following section.

## 2.1 Metropolis Algorithm

The method starts by choosing a set of selection probabilities $g(\mu \to \upsilon)$, that represent the probability of transition from state $\mu$ to state $\upsilon$, $\mu \to \upsilon$. Then it uses acceptance probabilities $A(\mu \to \upsilon)$ to make the decision of moving the system to the new state or not. This translates the idea that if we start off in a state and our algorithm generates a new state $\upsilon$ from , then we should accept that state and change our system to the new state $\upsilon$ a fraction of the opportunities, $A(\mu \to \upsilon)$. The rest of the chances, we keep the state . Note that, we can choose any value for the acceptance ratio between 1 and 0, however, $A(\mu \to \upsilon) = 0$ means that $P(\mu \to \upsilon) = 1$, therefore we should leave this case. In addition, we should remember that the acceptance ratio must have such a value that the detailed balance is satisfied. This is equivalent to say that the Equation 11 must be fulfilled.

$$\frac{P(\mu \to \upsilon)}{P(\upsilon \to \mu)} = \frac{g(\mu \to \upsilon)A(\mu \to \upsilon)}{g(\upsilon \to \mu)A(\upsilon \to \mu)} \tag{11}$$

Consequently, if the new state is accepted, the algorithm moves to the next step and repeats the process by choosing a new state with probability $g(\mu \to \upsilon)$, and then deciding to accept it or reject it according to the acceptance probability, $A(\mu \to \upsilon)$. If the new state is accepted the computer changes the system to it, else, it stays in the original state . Posteriorly, the process is repeated again and again.

Regarding the selection probabilities $g(\mu \to \upsilon)$, this should be chosen so that the condition of **ergodicity**, the requirement that every state must be available from every other in a finite number of steps, is met. Which still leaves quite a space to decide how $g(\mu \to \upsilon)$ is chosen. However, it is known for systems in thermal equilibrium, that the energy fluctuations are small in comparison with the energy of the entire system. The real system spends most of its time in a subset of states with a narrow range of energies and rarely makes transitions that change the energy of the system dramatically. This is essential, so that we don't spend to much time of simulation considering transitions to states with an energy that is very different from the energy of the present state. The simplest way of achieving this in the Ising Model is to consider only those states which differ by the flip of a single spin, **single-spin-flip dynamics**. Doing this, we guarantee that only small changes of energy take place and at the same time that from any state of the system we can get to any other state.

Actually, by implementing these dynamics, the amount of energy change between the current state, $E_\mu$, and any possible new state's energy, $E_\upsilon$, is $2J$ (between the spin we choose to flip and each one of its neighbors). Therefore, in a lattice with coordination number $z$, the maximum difference in energy would be $2Jz$. Note that in the method, all sites of the lattice have the same number of neighbors due to the implementation of **periodic boundary conditions**.

In the Metropolis algorithm the selection probabilities $g(\mu \to \upsilon)$ for each of the possible states $\upsilon$ are all chosen to be equal. Thus if there are $N$ sites on the lattice, using single-spin-flip as the

only way of transition between states, there are $N$ flips we can perform and therefore $N$ possible states,$\upsilon$, which we can reach from a given state, $\mu$. Being the selection probabilities equal to all spins, consequently there are $N$ selection probabilities $g(\mu \to \upsilon)$ which are non-zero and take the value,

$$g(\mu \to \upsilon) = \frac{1}{N} \tag{12}$$

Fulfilling the detailed balance condition presented in Equation x, with these selection probabilities the follow equation must hold:

$$\frac{P(\mu \to \upsilon)}{P(\upsilon \to \mu)} = \frac{g(\mu \to \upsilon)A(\mu \to \upsilon)}{g(\upsilon \to \mu)A(\upsilon \to \mu)} = \frac{A(\mu \to \upsilon)}{A(\upsilon \to \mu)} = e^{-\beta(E_\upsilon - E_\mu)} \tag{13}$$

Thus, we want to select the acceptance probability for our algorithm to satisfy:

$$\frac{A(\mu \to \upsilon)}{A(\upsilon \to \mu)} = e^{-\beta(E_\upsilon - E_\mu)} \tag{14}$$

To maximize the acceptance ratios, Metropolis methods sets the larger of $A(\mu \to \upsilon)$ or $A(\upsilon \to \mu)$ to be 1 (largest value possible) and then adjusts the other to satisfy the constraint presented in Equation 14. By this reason the acceptance algorithm is:

$$A(\mu \to \upsilon) = \begin{cases} e^{-\beta(E_\upsilon - E_\mu)}, & \text{if } E_\upsilon - E_\mu > 0 \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

Which means that if we select a new state which has an energy lower than or equal to the present one, we should always accept the transition to that state. If it has a higher energy then we maybe accept it, with the probability given above. In conclusion, this is the Metropolis algorithm for the Ising model with single-spin-flip dynamics which is characterized by the Equation 15.

To summarize, the Metropolis algorithm to simulate the Ising model goes succinctly through the following steps:

1. Pick a spin site of the lattice with size $L$, using selection probability $g(\mu \to \upsilon)$ and calculate, according to its neighbors, the contribution to the energy involving this spin,$E_\mu$;
2. Flip the value of the spin and calculate the new contribution, $E_\upsilon$;
3. If the new energy is smaller than the energy associated to the previous state, this is if $E_\upsilon - E_\mu <= 0$, then we keep the spin flipped;
4. By other hand if the energy of the new state is greater, $E_\upsilon - E_\mu > 0$, then we only keep the flipped state with probability $A(\mu \to \upsilon) = e^{-\beta(E_\upsilon - E_\mu)}$.
5. The same process is repeated until the thermal equilibrium had been reached.

## 2.2 Limitations of the Metropolis algorithm: Critical slowing down

It is important to note that the Metropolis algorithm does not perform well around the critical point due to the phenomena of critical slowing down.

When we are near a phase transition, the autocorrelation time, say $\tau$, is very long (according to Landau theory of phase transitions, the correlation length (in the disordered phase) is $\epsilon \sim \frac{1}{\sqrt{a}}$, and at the phase transition $a \to 0$ (Equation 9, so $\epsilon$). In a context where we are doing Monte Carlo simulations on a ferromagnet near $T_C$, where $T_C$ is the temperature of phase transition, we are studying the observable $M$, which is the total magnetisation. If we are in a microstate characterized by a magnetization far from the average magnetization, a long $\tau$ means that it will take a long time

(a lot of iterations, then) for the simulation to access microstates with a magnetization that is near the average magnetization. The microstates with a magnetization nearer to the average value of the magnetization have greater statistical relevance. So, for microstates with $M$ far from its average value, a longer time of sampling is required in order to gather good statistics with MC simulations (slowing down). Other techniques, such as cluster algorithms, are required in order to resolve the model near the critical point.

# 3 Our implementation of the Metropolis algorithm for the Ising Model

Above we explained the ideas, assumptions and the methodology behind the Metropolis algorithm in the simulation of the Ising Model. Therefore, we are now in conditions to explain how we implemented our own Metropolis algorithm, step by step, and present our results and compare them with the exact solutions for the Ising model.

## 3.1 Metropolis simulation

The main features of the Metropolis simulation were already pointed out. We will now explain the particular approaches that were performed in order to accomplish a correct implementation of this methodology. Before we start, it is important to clarify that for simplicity reasons, we assumed the external magnetic field, $B$, to be zero. Even though that the case $B \neq 0$ is not much harder to simulate, using $B = 0$ allows us to reach the results we are looking for. Consequently, the Hamiltonian of our simulated system is then given by:

$$H = -J \sum_{<ij>} s_i s_j \tag{16}$$

The first step of the simulation consisted in creating a lattice of size $L$ with $LxL = N$ sites, with each one being able to take one of the values $\pm 1$. We did this by creating a single integer array with size $N$. As mentioned previously, it important for the simulation to fix the same number of neighbours for all the spins in the lattice. For this reason, usually periodic boundary conditions are imposed to the array. This is, we specify that the spins on one edge of the lattice are neighbours of the corresponding spins on the other edge. However, to save computational space and time we used a single coordinate $i$ to create our lattice; and therefore to guarantee the periodic boundary conditions we used a variation of these, the **helical boundary conditions**. To understand how these conditions work and how they are implemented, let's consider the case of a two-dimensional $LxL$ lattice, just like the one we simulated. If this is the case then $i$ (counter variable) would run from 0 to $L^2 - 1$. Which means that regardless the actual dimension of the lattice it can always be represented by a one-dimensional array, and the value of the spin $i$ can is simply given by $s[i]$. And then, also the wrapping around the lattice, to ensure that all the spins have the same number of neighbours, becomes easily given for each spin $i$ by the following expressions:

$$(i \pm 1) mod L^2 \tag{17}$$

$$(i \pm L) mod L^2$$

Next we needed to decide the starting value for each one of the spins in our lattice. Once again to simplify we decided to start our system at temperature $T = 0.1$, that all the spins were aligned to the top (all up) and consequently all the sites in the lattice had value $+1$. In other words, even though that we did not start our system with a grounded temperature ($T = 0$ or $T = \infty$) we assumed

the configuration of nule temperature as our initial state. The justification for doing this is clear: we hope to save time by making the system reaching its final equilibrium state faster with this initial state than with either a $T = 0$ or $T = \infty$. And at the same time is right to consider that from all the range of temperatures to which we will submit our system to, the initial one will be the closer state to this type of grounded equilibrium, all spins aligned.

With the lattice generated and each of the neighbours to all the spins in it defined, as well their initial values, we started our simulation. Here the first step was to generated a new state, called $\upsilon$ in the previous explanations, by exclusively making a spin flip. To implement this, then we just generated a random number $k$ between 0 and $N-1$, that would correspond to the lattice site to be flipped. As explained before, then we needed to calculate the difference between this new state and the actual one to decide if the flip would actually occur or not. Considering that flipping the spin $k$ would only change its value, $s_k$, and not the value of its neighbours, $s_i$, so that $s_i^\upsilon = s_i^\mu$. Thus to calculate this energy difference we performed the following approximation:

$$E_\upsilon - E_\mu = -J \sum_{<ij>} s_i^\upsilon s_j^\upsilon + J \sum_{<ij>} s_i^\mu s_j^\mu = -J \sum_{in.ntok} s_i^\mu (s_k^\upsilon - s_k^\upsilon) \tag{18}$$

Note that if $s_k^\mu = +1$, than after spin $k$ has been flipped we must have $s_k^\upsilon = -1$,so that $s_k^\upsilon - s_k^\mu = -2$. On the other hand, if $s_k^\mu = -1$ then $s_k^\upsilon - s_k^\mu = +2$, therefore we can write:

$$E_\upsilon - E_\mu = 2J s_k^\mu \sum_{in.ntok} s_i^\mu \tag{19}$$

When we perform these approximations instead of summing over all the $\frac{1}{2}Nz$ spins in the system to calculate the energy difference, we are exclusively summing over all z terms, which is obviously a much more efficient way of doing it. After this calculation, the implemented code performs an evaluation on keeping the new state or the previous one by applying the set of rules mentioned above:

- If $E_\upsilon - E_\mu \leq 0$ we definitely accept the move and flip the spin $s_k \to s_k$;
- If $E_\upsilon - E_\mu > 0$ we flip it with a probability of $e^{-\beta(E_\upsilon - E_\mu)}$

For the last situation, the way we found to perform the evaluation of flipping or not the spin, was by generating a random number r between 0 and 1 and then evaluating if the number was greater or smaller than the probability $A(\mu \to \upsilon) = e^{-\beta(E_\upsilon - E_\mu)}$. If $r < A(\mu \to \upsilon)$ then we flip the spin, if it is not, we leave the spin alone.

And this is our complete algorithm, that repeats these same steps over and over again for a huge number of times ($\approx 10000000$) in order guarantee that the equilibrium is reached, before performing measurements of the system. After attaining equilibrium, the probability of finding the system in any particular state proportional to the Boltzmann weight $e^{-\beta E_\mu}$.

### 3.2 Results

Our Ising model simulation using Monte Carlo methods with Metropolis sampling was developed in C++ and the code can be found in the Annexes 1. Also, in the Annexes we find a table with the variables and what they represent, to make the code interpretation easier.

In this work we performed the explained simulation for different sizes lattices $L = 6, 12, 18$ and we calculated values of the energy, $E$, magnetization,$M$, specific heat, $C$, and magnetic susceptibility,$\chi$ , (all calculated per unit of spin) of the system, through the following expressions:

1. $E = \frac{<H>}{N}$ where $H$ where H is the energy of the system given by the Equation 16. However, performing the Hamiltonian calculation to every state of the system is not a very efficient way

of calculation the energy of the system. Therefore, the clever way of doing it is to calculate the energy of the Hamiltonian, $H$, at the very start of the simulation, and then every time we flip a spin the energy of the new system's state can be calculated by making use of the $\Delta E$ perform in each step.

$$E_\upsilon = E_\mu + \Delta E \tag{20}$$

Then,

$$E = \frac{< E_\upsilon >}{N} \tag{21}$$

2. $M = \frac{<M_\upsilon>}{N}$ where $M_\mu = \sum_i s_i^{-mu}$. Note that once again calculate the sum over all spins to calculate the magnetization in every iteration step is not a very efficient way of measuring it. Consequently, one more time what we indeed calculate was

$$\Delta M = M_\upsilon - M_\mu = \sum_i s_i^\upsilon - \sum_i^\mu = s_k^\upsilon - s_k^\mu = 2s_k^\upsilon \tag{22}$$

to every single time a spin was flipped. And then this value is added to the initial magnetization of the system.

$$M_\upsilon = M_\mu + \Delta M = M_\mu + 2s_k^\upsilon \tag{23}$$

3. $C = \frac{<H^2>-<H>^2}{NT^2}$;
4. $\chi = \frac{<S^2>-<S>^2}{NT}$.

It is important to note, that the averages $< A >$ are calculated after leaving the system relaxate for $nrelax = 10^7$ iteration steps ("Monte Carlo steps") and measuring during the next $nmonte = 10^7$ steps according to the expression:

$$< A >= \frac{1}{nmonte} \sum_{n=1}^{nmonte} A_n \tag{24}$$

The obtained results are represented in the graphics presented below. All quantities were manipulated in normalized units, $J = 1$ and $K_B = 1$. All the data was collected for the same range of temperatures, $T = [0.1, 5]$. In the Figure 2 are presented the results for the Energy per spin in function of the Temperature, $E(T)$.

In Figure 3 we have the results for the Magnetization per spin in function of the Temperature, $M(T)$.

In Figure 4 we found the results for the specific heat per spin also in function of the temperature, $C(T)$. And for last, in Figure 5 we can analyse the values for the magnetic susceptibility per spin in function of the temperature,$\chi(T)$.

Knowing that both the magnetic susceptibility, $\chi(T)$, and the specific heat, $C(T)$, diverge in the critical point ($T_C$) for the thermodynamic limit $L \to \infty$. Therefore, approximations for the exact value of the critical point for the Ising model were calculated in function of the lattice size, $T_C(L)$, by calculating the maximums for both parameters. The estimations found for these values are presented in the Table 2.

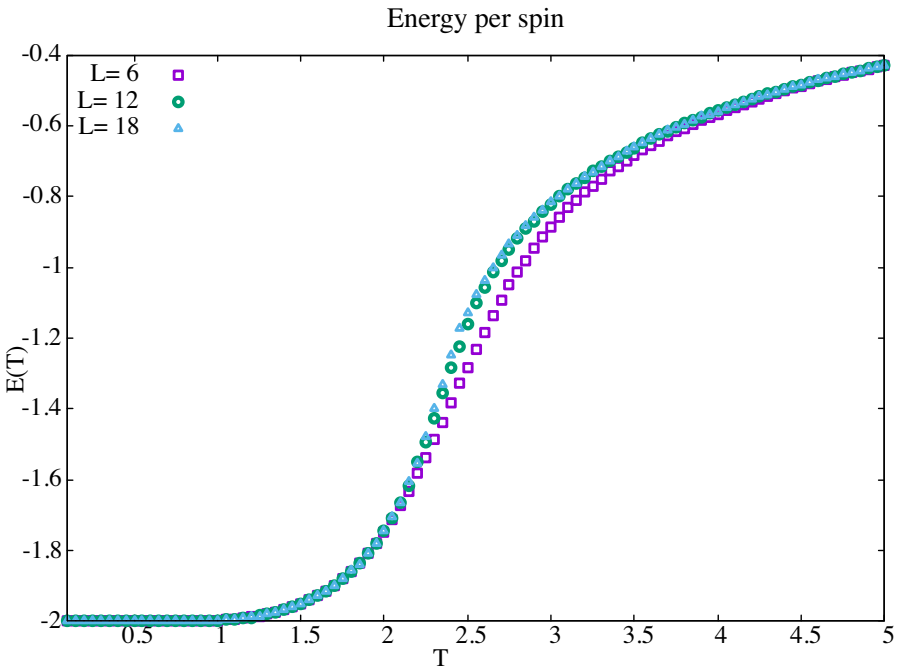


Figure 2: Energy per spin,$E(T)$ in function of the temperature, $T = [0.1, 5]$ for 3 different lattice sizes $L = 6, 12, 18$

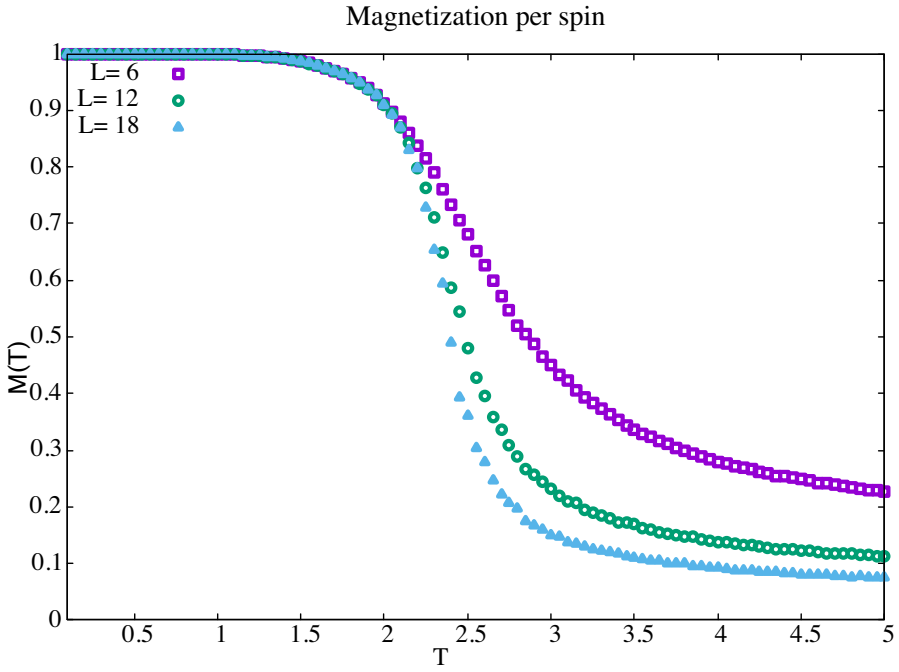


Figure 3: Magnetization per spin,$M(T)$ in function of the temperature, $T = [0.1, 5]$ for 3 different lattice sizes $L = 6, 12, 18$

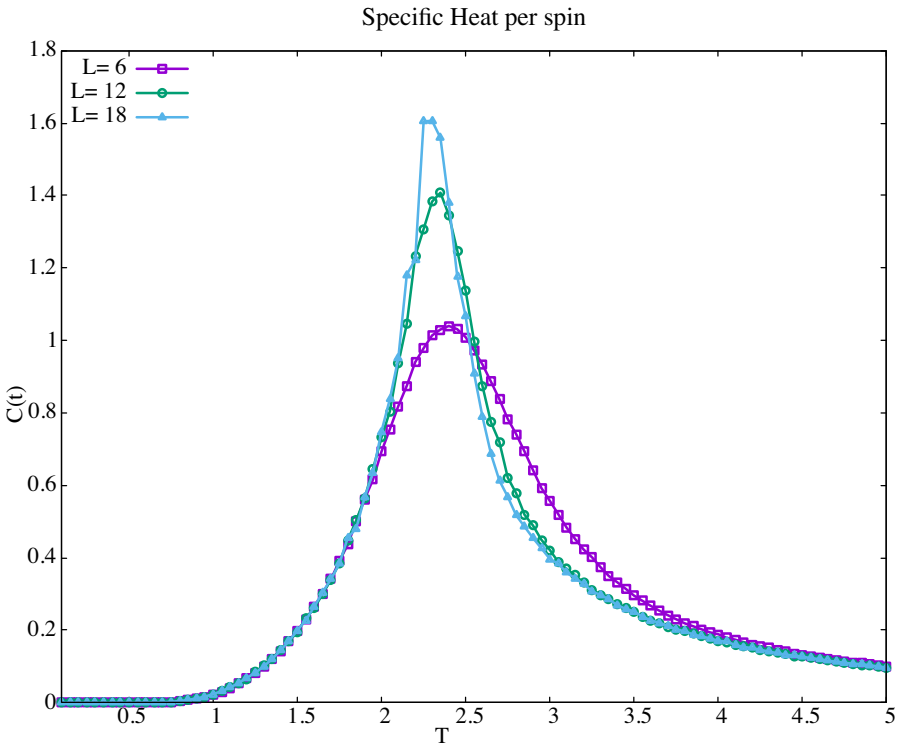


Figure 4: Specific heat per spin,$C(T)$ in function of the temperature, $T = [0.1, 5]$ for 3 different lattice sizes $L = 6, 12, 18$

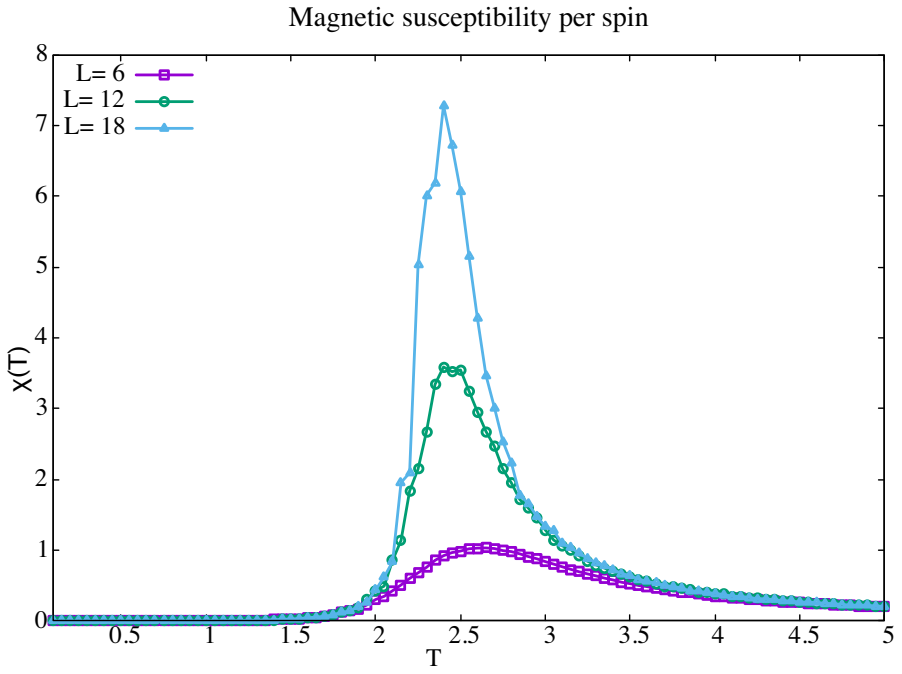


Figure 5: Magnetic susceptibility per spin,$\chi(T)$ in function of the temperature, $T = [0.1, 5]$ for 3 different lattice sizes $L = 6, 12, 18$

| $L$ | $C$ | $\chi$ |
|---|---|---|
| $L = 6$ | $T_C = 2.40$ | $T_C = 2.60$ |
| $L = 12$ | $T_C = 2.35$ | $T_C = 2.45$ |
| $L = 18$ | $T_C = 2.30$ | $T_C = 2.40$ |

Table 2: Table with the values for the critical temperature obtained by our Ising Model simulation with Metropolis algorithm for lattice sizes $L = 6, 12, 18$. The values were estimated by finding the temperature for the maximum values of magnetic susceptibility $/chi(T)$ and specific heat $C(T)$.

From the Figures above we can observe a decrease in the value of the critical point $T_C$ when the lattice size, $L$, decreases bringing these values closer to the theoretical value for the critical point, $T_C = 2.269$.

# 4 Discussion

# 5 Conclusion

The simulation of the Ising model confirmed the existence of a phase transition in the absence of an external field $h$. This observation becomes particularly evident by the significant change in the magnetization that constitutes the order parameter of the system. Additionally, our simulation was successful in proving that this phase transition is a second order one, since for increasing lattice sizes a discontinuity starts to show up in the second derivative properties of the system, Magnetization per spin, $\chi(T)$" and Specific Heat $C(T)$. Also, the values predicted for the critical temperature, $T_C$, were relatively close to the theoretical value $T_C == 2.269$. However, the distance between this exact value and the experimental one decreases when we increase the lattice size, since the theoretical value corresponds to the value found for an infinite size value.

Regarding the implementation of the Metropolis Monte Carlo algorithm we were able to confirm its efficiency and adequacy to simulate the Ising Model. This conclusions are supported by the rapid convergence of the simulated system to equilibrium, taking only a few minutes of personal computation for small size lattices to obtain this state. It also stands out, the algorithm ability to predict equilibrium values with great precision that is very evident in the close overlap between the theoretical and simulation based results for the several system properties. Although, near a second order phase transition the correlation length is very large, and therefore there are fluctuations on all scales. So, our local algorithm had difficulty in sampling the space of relevant configurations efficiently. The mean magnetization may actually be more or less correct, but more complicated observables (higher moments of M, correlation functions, etc) are difficult to compute. Fluctuations modify the simple mean field scaling, and $\epsilon$. Cluster algorithms perform updates on all scales, and capture these physical phenomena better.

# 6 References

[1]

# 7 Annexes

## 7.1 Annexes 1

```
//
//  main.cpp
//  Ising Model
//
//  Created by Íris on 23/06/2018.
//  Copyright © 2018 Íris e João. All rights reserved.
//

#include <iostream>
#include <vector>
#include <sstream>
#include <iostream>
#include <array>
#include <math.h>
#include <iomanip>
#include <algorithm>
#include <stdio.h>
#include <stdlib.h>
#include <ctime>
#include <fstream>
#include <time.h>
#include <random>

using namespace std;
using std::vector;
using std::cout;
using std::endl;



int main(int argc, const char * argv[]) {


    ofstream file1("/Users/Damiao/Desktop/energia1_L18.txt");
    ofstream file2("/Users/Damiao/Desktop/magnetização1_L18.txt");
    ofstream file3("/Users/Damiao/Desktop/calor1_L18.txt");
    ofstream file4("/Users/Damiao/Desktop/qui1_L18.txt");

    std::random_device rd;    // Generates a random number
    std::mt19937 gen(rd());


    int L = 6; // Size of the lattice
    int N = pow (L,2);
    int z = 4; // número de vizinhos
```

```
double r;



// Inicializa a rede de dimensão N.

int matriz_inicial [N];
int nn [N][z];

// Helical- bouundary conditions - Identificação dos vizinhos para cada spin i

for (int i=0; i<=(N-1);i++){
    matriz_inicial [i] = 1;

    //Vizinhos do spin i

    nn[i][0] = (i+1)%(N);
    nn[i][1] = ((i-1)%(N)+(N))%(N);
    nn[i][2] = (i+L)%(N);
    nn[i][3] = ((i-L)%(N) + (N))%(N);
}



////////////////////////////////////////////////////////////////////////////////////////
// IMPLEMENTAR MÉTODO DE MONTE CARLO METROPOLIS
////////////////////////////////////////////////////////////////////////////////////////

double J = 1;

//Intervalo de temperatura considerado

double tStep=0.05;
double Tmax=5;

int nmonte = 10000000;
int nrelax = 10000000;

int r1; //dois números aleatórios
std::uniform_int_distribution<> dis_int(0,N-1);
double r2;

int energias [z+1];
double d_e;
double A;


for(double T=0.1;T<=Tmax;T=T+tStep){
     cout << T << endl;
```

```
// Relaxação
for (int n=0; n<=nrelax; n++){

    d_e = 0;
    r1 = dis_int(gen); // 1 passo gerar um flip aleatório
    d_e = 2*J*matriz_inicial[r1]*(matriz_inicial[nn[r1][0]]+matriz_inicial[nn[r1][1]]+
    if (d_e <= 0){
        matriz_inicial [r1] = (-1)*matriz_inicial[r1];
    }
    else {
        A = exp((-1/T)*(d_e));

        r2 = std::generate_canonical<double, 10>(gen);

        if (r2<A){
            matriz_inicial [r1] = (-1)*matriz_inicial[r1];
        }
    }
}

// Equilibrio - Relaxação - Onde se calculam as variáveis termodinâmicas

double m=0; //Magnetização total
double h=0; //Energia Total - Hamiltoniano

// Para calcular os valor de energia do sistema inicial assim como de magnetização

for (int i=0; i<=(N-1);i++){
    h = h -((0.5)*matriz_inicial[i]*(matriz_inicial[nn[i][0]]+matriz_inicial[nn[i][1]]
    m = m + matriz_inicial[i];
}


double H = 0;
double M = 0;
double H2 = 0;
double M2 = 0;

for (int n=0; n<nmonte; n++){
    d_e=0;
    r1 = dis_int(gen); // 1 passo gerar um flip aleatório
    d_e = 2*J*matriz_inicial[r1]*(matriz_inicial[nn[r1][0]]+matriz_inicial[nn[r1][1]]+
    if (d_e <= 0){
        matriz_inicial [r1] = (-1)*matriz_inicial[r1];
        h = h + d_e;
         m = m + 2*matriz_inicial[r1];

    }
    else {
```

```
                A = exp((-1/T)*(d_e));

                r2 = std::generate_canonical<double, 10>(gen);
            if (r2<A){

                matriz_inicial [r1] = (-1)*matriz_inicial[r1];
                h = h + d_e;
                m = m + 2*matriz_inicial[r1];
                                }
        }
        H = (H + h);
        M = (M + fabs(m));

        H2 = H2+h*h;
        M2 = M2+m*m;
    }

    double E_av = (H/N)/(nmonte); //Energia média por spin
    double M_av = ((M)/N)/(nmonte); // Magnetização média por spin

    H2 = H2/(nmonte);
    M2 = M2/(nmonte);

    double E1 = (H)/(nmonte);
    double M1 = (M)/(nmonte);

    double  c =(H2-(E1*E1))/(T*T*N); // Calor especifico por spin
    double  Xi = (M2-(M1*M1))/(T*N); // Susceptibilidade por spin


    file1 << T << "  " << E_av << endl;;
    file2 << T << "  " << M_av<< endl;
    file3 << T << "  " << c << endl;
    file4 << T << "  " << Xi << endl;



}
return 0;
}
```

## 7.2 Annexes 2