

# Abstract

This work aim at provide the details of implementation of a simplified spin system called ISING model. After introducing the basic properties of ferromagnetic materials, the assumptions leading to the HEISENBERG Hamiltonian are stated. From here, the ISING Hamiltonian is derived. Then, a brief investigation of the Monte Carlo MARKOV chain method is proposed. The METROPOLIS update scheme is obtained and a careful attention is paid in determining averages as well as statistical errors. In the last part, we implement the one dimensional and the two dimensional square lattice models. In the latter case, we focus on the understanding of the critical behavior of the system near the phase transition. The CURIE temperature is finally extracted using the BINDER cumulant method.

---

# Contents

<b>I</b>	<b>Introduction</b>	1
I.1	The nail and the magnet . . . . .	1
I.2	Origins of ferromagnetism . . . . .	1
I.2.a	Exchange interaction . . . . .	1
I.2.b	Phase transition . . . . .	2
I.3	A simple model for ferromagnetism . . . . .	2
<b>II</b>	<b>Monte Carlo Simulation</b>	3
II.1	Generation of random numbers . . . . .	3
II.2	METROPOLIS algorithm . . . . .	4
II.3	Average and statistical error . . . . .	4
<b>III</b>	<b>Simulation of the ISING model</b>	6
III.1	One dimensional case . . . . .	6
III.1.a	Implementation . . . . .	6
III.1.b	Thermalization . . . . .	6
III.1.c	Results . . . . .	7
III.2	Two dimensional case . . . . .	8
III.2.a	Implementation . . . . .	8
III.2.b	Thermalization . . . . .	8
III.2.c	Results . . . . .	9
III.2.d	Determination of the critical temperature . . . . .	10
III.2.e	WOLFF algorithm . . . . .	11
III.3	Conclusion . . . . .	12
<b>References</b>		12

---

The code was written under Linux Ubuntu v.22.10 with Gedit v.42.2. Cmake v.3.24.2 was used to generate the `makefile`.

The report is written with L<sup>A</sup>T<sub>E</sub>X v.3.14159265. Every drawing are generated by the Ti<sub>K</sub>Z package and every plots are made with gnuplot v.5.4 using the cairolatex terminal.

# I Introduction

## I.1 The nail and the magnet

Consider a simple experiment where an iron nail is hanging on a wire. Alongside, there is a magnet whose magnetic field induces a magnetization inside the nail. At low temperature, the magnet will attract the nail but if the temperature increases until reaching 770°C the latter will no longer be attracted. If the experiment is repeated with a copper nail, nothing will happen. Materials that behave like the iron nail are called *ferromagnetic*. Contrarily to copper materials, iron materials possess aligned magnetic moments in the ground state [1].

It is possible to quantitatively explain this phenomenon by measuring the magnetization  $M$  of the iron nail as a function of the induced magnetic field  $B$ . In general,  $M = 0$  when  $B = 0$ . As  $B$  increases, the magnetization also slowly increases until reaching a maximum value called *magnetic saturation*  $M_s$ . This process is illustrated by the dashed line inside the loop of figure 1 and is called *initial magnetization curve*. If now  $B$  decreases, then  $M$  will also decrease but it will not cancel when  $B$  is zero again. The remaining magnetization at this point is called *saturation remanence*  $M_r$  and the material thus possesses un permanent magnetization. It is only after applying a negative magnetic field, called *coercivity*  $B_c$ , that we recover  $M = 0$ . The process is identical for negative values of  $M$  and  $B$  such that the plot of  $M(B)$  looks like a loop, called *hysteresis loop*. Further explanations have been provided by P. WEISS who emitted the assumption that below the critical temperature (e.g. 770°C for the iron), ferromagnetic materials are structured into magnetic domains [1]. Each of these domains has a non-zero magnetization but the latter varies between domains such that the magnetization of the whole material can be 0. When a magnetic field is applied, the spins inside the domains interact with it. When they are oriented in the same direction than the field, the domains become bigger whereas the domains with spins in the opposite direction become smaller. The way the spins interact with each other as well as with  $B$  is the subject of the next section.

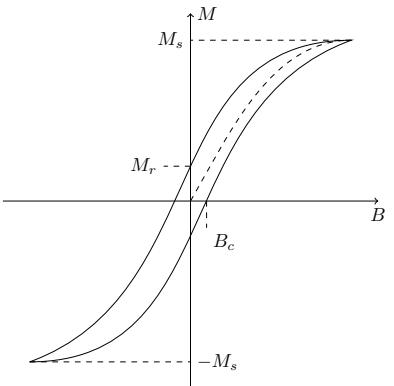


Figure 1: Hysteresis loop.

## I.2 Origins of ferromagnetism

### I.2.a Exchange interaction

Following [1] again for this section, we briefly investigate the interaction that describes the properties of the ferromagnetic materials. This naturally leads to the obtention of the HEISENBERG Hamiltonian.

**Magnetic dipole-dipole interaction** This is the interaction between the magnetic moments  $\mu$  of the material. Usually, these magnetic moments are of the order of the BOHR magneton  $\mu_B = e\hbar/(2m_e) \sim 10^{-23} \text{ A.m}^2$ . The magnitude of this interaction is thus approximately  $E_{\text{mag}} \simeq \mu_0 \mu^2 / (4\pi r^3) \simeq 10^{-4} \text{ eV}$ . If  $E_{\text{mag}}$  is responsible for the ferromagnetic properties of the material this means that it should be able to fight against thermal fluctuations. However, one find that  $E_{\text{mag}}/k \simeq 1 \text{ K}$ . This is completely incompatible with the critical temperatures observed (consider for instance the temperature of 770°C that the iron needs to undergo a phase transition). Nevertheless, we emphasize that this interaction plays an important role in the formation of magnetic domains.

**Exchange interaction** The exchange interaction is due to the COULOMB force combined to the PAULI exclusion principle. In order to explain it, we restrict ourselves to the study of two electrons of the cristal, both carrying a intrinsic magnetic moment. The wave function of the system is written  $\psi(1,2) \equiv \phi(\mathbf{r}_1, \mathbf{r}_2)\chi(1,2)$ ,  $\phi$  being the spatial wave function and  $\chi$ , the ‘spin wave function’. Since the electrons are fermions, they obey the PAULI exclusion principle and two cases can thus be distinguished. On the one hand, if the spins are parallel then  $\phi(\mathbf{r}_1, \mathbf{r}_2) = -\phi(\mathbf{r}_2, \mathbf{r}_1)$ . On the other hand, if the spins are anti-parallel then  $\phi(\mathbf{r}_1, \mathbf{r}_2) = \phi(\mathbf{r}_2, \mathbf{r}_1)$ . In the latter case, the spins can be neighbor without being bothered by the exclusion principle but if they are parallel the wavefunction cancels when  $\mathbf{r}_1 = \mathbf{r}_2$ . This means that two parallel spins will rarely be neighbor and so the COULOMB interaction will always be weaker than for two anti-parallel spins (because this interaction is proportional to  $1/r^2$ ). Consequently, we expect that two electrons with the same spin will have a lower energy than electrons with an opposite spin. The energy difference between these two cases is equivalent to a force between them and can be represented by an effective Hamiltonian  $H_{\text{int}} \equiv -J_{12} \mathbf{S}_1 \cdot \mathbf{S}_2$  where  $J_{12}$  is called *exchange integral*. From here, it is possible to show that (see e.g. [1, footnote 16 page 452] or [8, annexe A] for proof)  $J_{12} = E_s - E_t$  where  $E_s$  and  $E_t$  are the energy associated with the singlet and the triplet state, respectively.

**HEISENBERG model** In the HEISENBERG model, the assumption that the exchange integral is a constant is made. In other words, this means that from now on we assume that only nearest neighbor interact. However, the interaction is no longer restricted to only two electrons in order to study the properties of a real crystal. The effective Hamiltonian

becomes  $H_{\text{int}} \rightsquigarrow H'_{\text{int}} \equiv -J \mathbf{S}_i \cdot \mathbf{S}_j$ . Additionally, we suppose that a magnetic field  $\mathbf{B}$  is applied on the system. It will tend to align the spins with each other such that the HEISENBERG Hamiltonian reads as

$$H_H = -J \sum_i \sum_{\langle j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j - g\mu_B \mathbf{B} \cdot \sum_i \mathbf{S}_i , \quad (1)$$

where  $g$  is the LANDÉ factor and the sum over indices  $\langle j \rangle$  covers only the nearest neighbor of site  $i$ . Until now, it has not been possible to express the partition function associated to a system described by the HEISENBERG Hamiltonian. This result is still of major importance because, with additional assumptions, it leads to simplified models. These models are also able to report on interesting properties of ferromagnetic materials such as *phase transition* which is the subject of the next section.

### I.2.b Phase transition

We mentioned in section I.1 that after heating up an iron nail at a certain temperature, the hysteresis loop that he was previously following completely disappear. This temperature is known as *critical temperature* or CURIE temperature and separate the domain of existence of two phases. The ferromagnetic phase is ordered and possesses a non zero magnetization. The spins are aligned so we say that this phase is not symmetric. The paramagnetic phase is not ordered and has a zero magnetization. The spins take any direction, hence this phase is symmetric. The magnetization  $M$  is called the *order parameter* of the system because it completely characterize these two domains [1]. When the transition occurs, the direction of the spins steeply changes (ordered spins  $\leftrightarrow$  non ordered spins) and the system undergoes a *spontaneous symmetry breaking*. Generally, a transition followed by a symmetry breaking and whose order parameter is a continuous function of the temperature is called *second order phase transition*.

This kind of transition brought a great interest because the properties of the ferromagnetic crystals usually have a singular behavior near the critical point. Moreover, it seems that these properties mostly depend on the general characteristics of the phase transition rather than on the structure of the materials. Yet, it is often difficult to theoretically determine them (think of the HEISENBERG Hamiltonian for instance) and numerical simulations are thus needed.

## I.3 A simple model for ferromagnetism

**ISING Hamiltonian** The main difficulty of the HEISENBERG model lie in the fact that the three components of the spin operator do not commute. In the ISING approximation, only the  $z$  component of the magnetic field is kept which means that the Hamiltonian is now diagonal is the base of the  $s_i^z$  (because  $[H, s_i^z] = 0$ ). The problem then becomes a classical spin system. If we consider a general  $d$ -dimensional network where each site  $i$  is occupied by a spin that can take the values<sup>1</sup>  $\pm 1$ , the ISING Hamiltonian can be written in the following form

$$H_I = -J \sum_i \sum_{\langle j \rangle} s_i s_j - h \sum_i s_i , \quad (2)$$

where we have denoted  $s_i^z \equiv s_i$  and  $g\mu_B B^z \equiv h$ . We remark here that for  $h = 0$ , two possible configurations will give the lowest energy: every spins are up or every spins are down. However, when  $h \neq 0$  the lowest energy is obtained only when the spins are oriented in the same direction than the magnetic field.

**Exact results in one dimension** When  $d = 1$ , the network consists in  $N$  lattices, each containing one spin. Each of these spins has only one nearest neighbor and the ISING Hamiltonian thus reads as

$$H = - \sum_{i=1}^N (J s_i s_{i+1} + h s_i) . \quad (3)$$

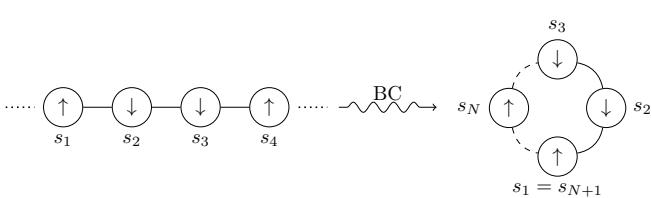


Figure 2: Boundary conditions of the one dimensional ISING model.

In order to calculate the partition function, one need to impose boundary conditions (BC). The usual way to do it is to say that the spin  $N + 1$  is equal to the spin 1. The linear network then becomes a closed chain as illustrated in figure 2. Using the transfer matrix method, we can factorize the partition function as follow

$$Z = \sum_{s_1=\pm 1, \dots, s_N=\pm 1} \prod_{i=1}^N e^{-\beta H_i} \quad (4)$$

$$= \sum_{s_1=\pm 1, \dots, s_N=\pm 1} t(s_1, s_2) t(s_2, s_3) \dots t(s_N, s_1) , \quad (5)$$

<sup>1</sup>Notice that in a real crystal we are interested in the study of the electrons of spin  $\pm 1/2$ . For sake of simplicity, this value of often ‘normalized’ to  $\pm 1$ .

where  $t(s_i, s_j) \equiv \exp\{\beta[J s_i s_j + h(s_i + s_j)/2]\}$ . Eqn. 5 is nothing except the trace of the matrix  $\mathcal{T} \equiv \begin{pmatrix} t(1/2, 1/2) & t(1/2, -1/2) \\ t(-1/2, 1/2) & t(-1/2, -1/2) \end{pmatrix}$ . After few lines of calculations, one find in the thermodynamic limit ( $N \rightarrow \infty$ )

$$Z = N e^{\beta J} \left[ \cosh(\beta h) + \sqrt{\sinh^2(\beta h) + e^{-4\beta J}} \right] . \quad (6)$$

The HELMHOLTZ free energy  $F(h, T) := -kT \ln Z$  is then easily found and from here every thermodynamic quantities are derived (the reader can find every proofs in [1, 5])

$$\begin{cases} M(h, T) = -\frac{\partial F}{\partial h} = \frac{N \sinh(\beta h)}{\sqrt{\sinh^2(\beta h) + e^{-4\beta J}}} \\ \chi(h, T) = \frac{\partial M}{\partial h} = \frac{N}{T} \cdot \frac{\cosh(\beta h) e^{-\beta J}}{[\sinh^2(\beta h) + e^{-4\beta J}]^{3/2}} \end{cases} , \quad \begin{cases} E(h = 0, T) = -T^2 \frac{\partial}{\partial T} \left( \frac{F}{T} \right) = -N J \tanh(\beta J/2) \\ C_v(h = 0, T) = \frac{\partial E}{\partial T} = N k (\beta J)^2 \operatorname{sech}^2(\beta J/2) \end{cases} . \quad (7)$$

These expressions will later be used to compare the results of the simulations to the theoretical ones.

## II Monte Carlo Simulation

### II.1 Generation of random numbers

During the simulation of the ISING model, sampling different configurations of the lattice is needed. For this purpose, one need to employ advanced techniques known as Monte Carlo MARKOV chain. These techniques will be presented in more details in the next section. However, we already argue that generating *variates* - that is, random numbers drawn from a probability distribution or a probability density function (PDF) - is a central task in Monte Carlo simulations. In most implementations, the generation of random variables is separated into two steps [6]. First, a sequence of random bits is generated by an *engine*. In the second step, these bits are converted into variates according to the specified probability distribution or PDF. The aim of this section is to give a brief overview of the available tools provided by C++ 11 to generate such variates.

**Random engines** The role of an engine is to output a completely deterministic series of bits (this is why we call them pseudo-random numbers) that depends on its internal state. This state may consist of a single integer or an array of integers. The initial value of this integer is called the *seed*, or *seeds* if this is an array. Once the engine is seeded, it will produce the exact same sequence after each compilation. Since C++ 11, the `<random>` library provides through the `std::mt19937` class an engine called Mersenne Twister. It has a non-repeating sequence of  $2^{19937} - 1$  which means that it can generates  $2^{19937} - 1$  different pseudo-random numbers before repeating the same pattern. This is an important point because we want our algorithm to be able to explore every microstates of the system. Another important point is to check for the reproducibility of the experiment. Indeed, it would be consistent to get the same average values even when the initial seed has changed. For this purpose, the `<random>` library also provides a *truly* random number generator through the `std::random_device` class. It uses the hardware entropy to generates the random numbers and can be used to seed a deterministic random engine.

**Generate variates** Once the engine is seeded, it is then easy to convert the bits into variates. The C++ `<random>` library propose several classes for that matter<sup>2</sup>. Three distributions have been used in the program (to be discussed in section III): `std::uniform_int_distribution`, `std::uniform_real_distribution`, `std::bernoulli_distribution`. The two first respectively output uniformly distributed integers and real numbers in a given interval while the latter returns a boolean according to a given probability (it is used to randomly fill the lattice with the values +1 or -1). Listing (1) below illustrates with a basic example how to actually write a program with every informations discussed in the current section.

```

1 #include <iostream>
2 #include <random>
3
4 double generate_random (std::mt19937 & gen) {
5     std::uniform_real_distribution<double> real_dist (0., 10.);
6     return real_dist(gen); // returns a uniformly distributed real random number in [0, 10[
7 }
8
9 int main (int argc, char ** argv) {
10
11     std::random_device rd;
12     std::mt19937 gen (rd()); // seeded with a true random number

```

<sup>2</sup>We refer the reader to the documentation for an exhaustive list of available distributions.

```

13     std::cout << generate_random(gen) << std::endl; // Possible output: 3.48
14
15     return 0;
16 }
```

Listing 1: Basic usage of the `<random>` C++ library.

## II.2 METROPOLIS algorithm

As already said, to sample different configurations of an ensemble according to a given probability distribution it is often convenient to use a MARKOV chain [6]. The idea of this method is to employ a stochastic process - meaning a process that represents the continuous or discrete evolution of a random variable - to generate random configurations according to a multivariate probability distribution  $P(x)$ . This distribution can be either continuous or discrete. However, we shall focus on the latter case to describe a lattice of spins. Indeed, the distribution must be discrete because the spins can take only two values ( $\uparrow$  and  $\downarrow$ ) and it is multivariate because it describes every sites on the lattice.

**MARKOV chain** A general MARKOV process describes the probability for the system to be in state  $x_t$  at time  $t$ . This probability depends on every previous states the system was in at times  $t_i, i \in \{1, 2, \dots\}$ . If we denote  $S_k$  the finite set of possible states for the system, then the probability that it is in state  $S_{k_n}$  at time  $t_n$  is

$$p(x_{t_n} = S_{k_n} | x_{t_{n-1}} = S_{k_{n-1}}, \dots, x_{t_1} = S_{k_1}) . \quad (8)$$

When this probability depends only on the immediate previous state, the corresponding sequence of states  $\{x_t\}$  is called a MARKOV chain [4]. In this case, the transition probability is simply  $p(x_{t_n} = S_{k_n} | x_{t_{n-1}} = S_{k_{n-1}}) \equiv p(x'|x)$  with  $p(x'|x) \geq 0$  and  $\sum_{\text{states}} p(x'|x) = 1$ . Note that we also demand the MARKOV chain to be *ergodic*, that is to say that for an infinitely long sequence all the states will appear and won't be stuck in subspaces of the system. Therefore, it must exist a stationary point distribution  $\pi(x)$  such that asymptotically we have  $\pi(x) = P(x)$ . A sufficient (but not necessary) condition to ensure ergodicity is to impose *detailed balance*, namely

$$\pi(x)p(x'|x) = \pi(x')p(x|x') . \quad (9)$$

**METROPOLIS algorithm** Now we have discussed the basic ideas behind a MARKOV chain, we would like to determine the *acceptance distribution*  $A(x'|x)$ , that is, the probability to accept the proposed state  $x'$ . A common approach consists into splitting the transition probability in two sub-steps such that  $p(x'|x) = R(x'|x)A(x'|x)$  where  $R(x'|x)$  is called the *proposal distribution* and represents the probability that a new state  $x'$  is proposed. Using the condition of detailed balance (eqn. (9)), we arrive at  $A(x'|x)/A(x|x') = P(x')R(x|x')/[P(x)R(x'|x)]$ . The next step is to choose an acceptance ratio that fulfills every conditions mentioned until here. METROPOLIS and HASTINGS proposed the following solution

$$A(x'|x) = \min \left[ r, \frac{P(x')R(x|x')}{P(x)R(x'|x)} \right] , \quad (10)$$

where  $r \in [0, 1]$  is a real random number (and can be generated with listing 1). If the proposal distribution is reversible, i.e.  $R(x'|x) = R(x|x')$ , then the acceptance probability simplifies to

$$A(x'|x) = \min \left[ r, \frac{P(x')}{P(x)} \right] , \quad (11)$$

and the algorithm is called METROPOLIS method [6]. Since the results for the ISING model are derived in the canonical ensemble of statistical mechanics, we shall now use its properties to obtain the expression of the ratio  $P(x')/P(x)$ . We recall that for a given temperature  $T$ , the probability that the system is in a certain state of energy between  $E$  and  $E + dE$  is weighted by a BOLTZMANN factor, namely  $P(E) = e^{-\beta E}/Z$ . This allows to express the acceptance probability as

$$A(E'|E) = \min \left[ r, \frac{P(E')}{P(E)} \right] = \min \left[ r, \frac{e^{-\beta E'}/Z}{e^{-\beta E}/Z} \right] = \min \left[ r, e^{-\beta \Delta E} \right] , \quad (12)$$

with  $\Delta E \equiv E' - E$ . We now have every ingredients in our possession to implement a METROPOLIS scheme. Algorithm (1) illustrates a basic update for a one dimensional ISING model ( $\Lambda$  denotes the chain of spins). This process has to be performed as many times as needed in order to explore every possible microstates.

## II.3 Average and statistical error

We are now able to update the lattice with the appropriate probability distribution. The next step is to derive the expression of the average value of the thermodynamics quantities (energy, specific heat, ...). The goal of this section is to provide the tools to obtain it as well as the associated statistical error.

**Algorithm 1** METROPOLIS update of a one dimensional ISING model.

---

```

1:  $\Lambda \leftarrow \{s_1, \dots, s_N\}$                                 ▷ Fill the lattice with values  $\pm 1$ 
2:  $x \leftarrow \text{random } (1, N)$                                ▷ Choose a spin at random
3:  $\Delta E \leftarrow 2s_x(h + Js_{x+1})$                          ▷ Compute the energy change undergoes by a spin flip
4: if  $\Delta E < 0$  then
5:    $s_x \leftarrow -s_x$ 
6: else if  $r < e^{-\beta\Delta E}$  then
7:    $s_x \leftarrow -s_x$ 
8: end if                                                 ▷ Repeat the process from line 2 as needed

```

---

**Average** Let  $I$  be a finite interval in  $\mathbb{N}$  and  $\mathcal{O}$  a physical observable whose (discreet) set of possible values is denoted  $\{\mathcal{O}_i\}_{i \in I}$ . To calculate the average of such an observable in the canonical ensemble, the following formula is used:  $\langle \mathcal{O} \rangle := \sum_{i \in I} \mathcal{O}_i P_i$  with  $P_i = e^{-\beta E_i}/Z$ . The probability  $P_i$  is rather difficult to evaluate with Monte Carlo simulations because the partition function is not usually known. However, this difficulty vanishes by generating a MARKOV chain. Indeed, in this case, the state of the system at time  $t_n$  depends only on the one at time  $t_{n-1}$  and the relative probability is thus the ratio of the individual probabilities such that the denominator cancels [4]. As a result, only the energy difference between the two states is needed. This is exactly what we have shown in the derivation of eqn. (12). Assuming now that the system has reached equilibrium after a certain algorithmic time  $\tau$  (this point will be discussed later, see e.g. figure 3), the average of any quantity  $\mathcal{O}$  simply becomes an arithmetic average, that is  $\langle \mathcal{O} \rangle = N^{-1} \sum_{i=1}^N \mathcal{O}_i$ . Here,  $N$  represents the number of METROPOLIS sweeps undergone by the system after equilibration. Higher order moments are then easily derived

$$\langle \mathcal{O} \rangle = \frac{1}{N} \sum_{i=1}^N \mathcal{O}_i, \quad \langle \mathcal{O}^2 \rangle = \frac{1}{N} \sum_{i=1}^N \mathcal{O}_i^2, \quad \langle \mathcal{O}^3 \rangle = \frac{1}{N} \sum_{i=1}^N \mathcal{O}_i^3, \quad \langle \mathcal{O}^4 \rangle = \frac{1}{N} \sum_{i=1}^N \mathcal{O}_i^4, \quad \dots \quad (13)$$

Note that when an attempt to a spin flip is rejected (this happens when  $\Delta E \geq 0$  and  $r > e^{-\beta\Delta E}$ ), the old state is counted again in the average.

**Statistical error** We must remark here that the results of eqn. (13) are valid only if ergodicity is fulfilled. This means that we should generate an infinitely long sequence of new configurations in order to obtain the exact value of  $\langle \mathcal{O} \rangle$ . It is obvious that this cannot be achieved. Therefore, every average value comes with its statistical error and the standard way to define it is [2]: error  $\equiv \sigma/\sqrt{N}$  where  $\sigma := \sqrt{\sum_{i=1}^N (\mathcal{O}_i - \langle \mathcal{O} \rangle)^2/N}$  is the standard deviation of the distribution. Using the results of eqn. (13), we arrive at

$$\langle \mathcal{O} \rangle = \frac{1}{N} \sum_{i=1}^N \mathcal{O}_i \pm \text{error}, \quad \text{with} \quad \text{error} = \frac{1}{\sqrt{N}} \sqrt{\frac{1}{N} \sum_{i=1}^N \mathcal{O}_i^2 - \left( \frac{1}{N} \sum_{i=1}^N \mathcal{O}_i \right)^2} = \sqrt{\frac{\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2}{N}} \quad . \quad (14)$$

We are now ready to start basic simulations of the ISING model (more informations are contained in section III). As discussed in section II.1, special attention should be paid to the reproducibility of the experiments. Table 1 below shows various results after five simulations. Errors are computed with eqn. (14) and a new truly random seed was used after each run. We see that the fluctuations are small. This means that the algorithm remains stable when the initial seed is changed. It is also important to remark that, even when considering the errors, the algorithm does not always recover the theoretical results. This is due to the finitude of the lattice size and the theoretical results will be found only if the latter tend to infinity (thermodynamic limit).

runs	$T = 0$				$T = 0.5$			
	$\langle M/L \rangle$	error	$\langle E/L \rangle$	error	$\langle M/L \rangle$	error	$\langle E/L \rangle$	error
1	0.034	0.003	-0.946	0.001	0.018	0.006	-0.764	0.002
2	0.174	0.006	-0.957	0.001	0.011	0.004	-0.745	0.002
3	0.054	0.006	-0.941	0.001	0.002	0.007	-0.749	0.002
4	0.214	0.008	-0.9535	0.0004	0.042	0.008	-0.748	0.002
5	0.000	0.003	-0.955	0.001	0.045	0.004	-0.753	0.002

Table 1: Average magnetization and energy per spin for 5 simulations on a one dimensional chain of spins. Parameters:  $L = 500, h = 0, J = 1, N = 1.3 \times 10^5$ . Theoretical results (from section I.3):  $\langle M/L \rangle = 0$  at  $T = 0$  and  $T = 0.5$ ,  $\langle E/L \rangle = -1$  at  $T = 0$  and  $\langle E/L \rangle = -0.762$  at  $T = 0.5$ .

Finally, we emphasize that eqn. 14 is valid only for statistically independent observations  $\mathcal{O}_i$ . This is almost true for the one dimensional ISING model because it is always in the paramagnetic phase (except at  $T = 0$ ) and so the new state depends only on the previous one. Nonetheless, when the dimension increases the Monte Carlo sampling become correlated near the phase transition and the error needs to be modified [4].

### III Simulation of the ISING model

#### III.1 One dimensional case

##### III.1.a Implementation

As we have seen in section I, an ISING model is described by four parameters: the temperature  $T$ , the magnetic field  $h$ , the coupling constant between nearest neighbor  $J$  and the size of the lattice that we will call  $L$  in the following ( $L$  is also equal to the number of spins). The goal is to generate new configurations when each of these parameters are fixed. Hence, it would be convenient to have a kind of ‘database’ that holds every informations on the lattice. We could also keep track of the magnetization and of the energy by updating their values each time a new configuration is generated. This way, for a given temperature, the average value of any physical quantity  $\mathcal{O}$  could easily be computed using eqn. 13. A practical way to handle these datas is to use object oriented programming (OOP). Because we do not want the parameters to change unexpectedly, we can set them as **private** members of a general **Ising1D** object. Once instantiated, this object will initialize the spins with the values  $\pm 1$  and run a METROPOLIS scheme (as implemented in algorithm 1) as long as intended. The details of implementation are illustrated in listing 2 hereinafter.

```

1 #include <random>
2
3 class Ising1D {
4 private:
5     int * m_spins; // An array that contains the value of the spins
6     double m_latticeSize;
7     double m_T;
8     double m_h;
9     double m_J;
10    double m_energy; // Energy after each new configuration
11    double m_magnetization; // Magnetization after each new configuration
12    double * m_datas_energy; // Stock the energy
13    double * m_datas_magnetization; // Stock the magnetization
14 public:
15    Ising1D (double latticeSize, double T, double h, double J); // Constructor
16    ~Ising1D (); // Destructor. Properly release the memory
17    void init (std::mt19937 & gen) noexcept; // Randomly initialize the spins
18    void metropolis (int iterations, std::mt19937 & gen); // Perform a Metropolis scheme
19    double * get_datas_energy () const noexcept; // Return the energy
20    double * get_datas_magnetization () const noexcept; // Return the magnetization
21 };

```

Listing 2: Implementation of the **Ising1D** class.

Additional methods (called *getters*) will allow the user to simply collect the magnetization and the energy the system was in at any time. The calculation of the average values is then trivial. Repeating this process for different temperatures allow to plot  $\langle \mathcal{O} \rangle$  versus  $T$  as shown in algorithm 2.

---

##### Algorithm 2 Computation of an average.

```

1: for  $T = T_{\min}$  to  $T = T_{\max}$  do
2:   ising1d  $\leftarrow$  Ising1D( $L, T, h, J$ )                                 $\triangleright$  Instantiate an Ising1D object
3:   ising1d  $\leftarrow$  init(gen)                                          $\triangleright$  Initialize the spins
4:   ising1d  $\leftarrow$  metropolis( $N$ , gen)                                      $\triangleright$  Apply a METROPOLIS scheme
5:    $\langle \mathcal{O} \rangle \leftarrow 0$ 
6:   for  $i = \text{equilibrium}$  to  $i = N$  do                                 $\triangleright$  When equilibrium is reached, average
7:      $\langle \mathcal{O} \rangle \leftarrow \langle \mathcal{O} \rangle + \text{get\_datas}[i]$ 
8:   end for
9:   output  $\langle \mathcal{O} \rangle$ 
10: end for

```

---

##### III.1.b Thermalization

The reader may have noticed that in algorithm 2, the loop over which the average is computed does not start at  $i = 0$  but at a certain value called **equilibrium**. In Monte Carlo simulations, the system is initialized with a certain configuration and needs a certain algorithmic time  $\tau$  (which here corresponds to the number of METROPOLIS loops) to ‘forget’ it. During this phase, the system is relaxing towards equilibrium and both the internal energy and the magnetization are changing [4]. This is usually not a good idea to compute averages during this phase because the fluctuations are too strong to correctly describe the system [6]. Once the system has reached equilibrium, it still fluctuates a bit but this time, around its mean value and the averages can then be calculated.

As discussed in section I.2.b,  $M$  is the order parameter of the system. It is thus of great help to visualize the fluctuations of the chain of spins. However, the one dimensional ISING model is always paramagnetic and so the magnetization is 0 at any temperature (this point is discussed in the next section). This makes it difficult to determine the time  $\tau$  after which the system has reached equilibrium (bottom curve on figure 3). A ‘trick’ to do it is to apply a non-zero magnetic field because the magnetization will then go to 1 at low temperature (top curve on figure 3). It is important to have in mind that because the parameters are different, this tip will *not* give the exact desired time  $\tau$  but it will, at least, give an order of magnitude. Finally, remark that we plotted the absolute value of the magnetization. Indeed, as mentioned in section I.3, two different configurations lead to the same energy when  $h = 0$ . It is thus likely that  $M$  becomes negative. This point has no importance here, but it can cause difficulties in determining the critical temperature in the two dimensional case.

### III.1.c Results

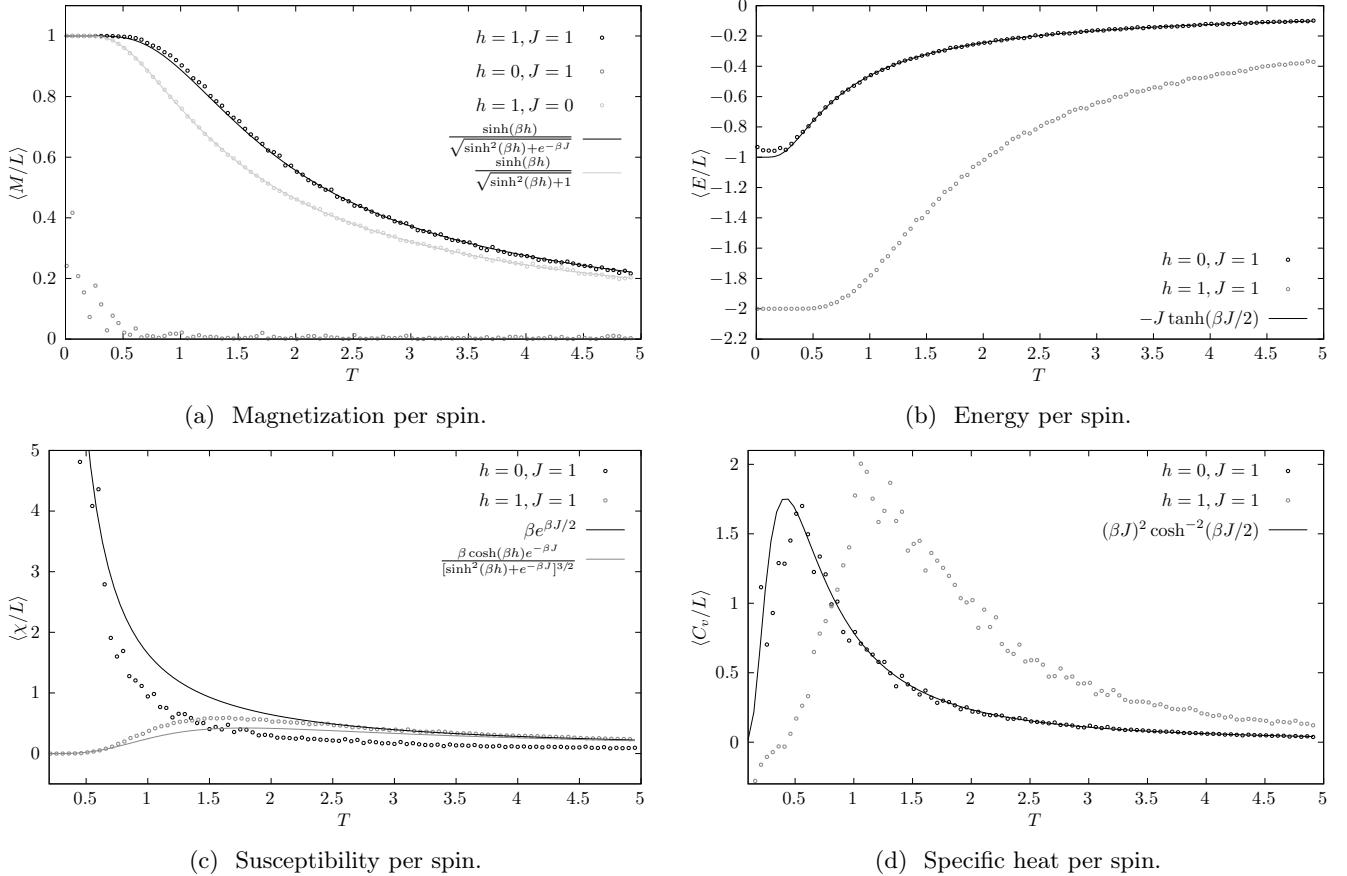


Figure 4: Each of the plots have been generated with a lattice of size  $L = 500$ . Between  $1.3 \times 10^5$  and  $2.9 \times 10^6$  METROPOLIS sweeps have been used to calculate the averages.

The plots hereinabove are produced with algorithm 2. In the following we let the BOLTZMANN constant  $k$  being

equal to unity such that the fluctuation relations take the form

$$\chi(h, T) = \frac{\langle M^2 \rangle - \langle |M| \rangle^2}{T} , \quad C_v(h, T) = \frac{\langle E^2 \rangle - \langle E \rangle^2}{T^2} . \quad (15)$$

These relations are convenient in our case because algorithm 2 just needs little adaptations to return the value of  $\langle M^2 \rangle$  and  $\langle E^2 \rangle$ . We see on figure 4 that after paying a careful attention in determining the equilibration time, the algorithm recovers quite well the theoretical results. One interesting point is presented on figure 4a were we see that when the magnetic field is turned off the magnetization is always 0 for  $T \neq 0$  which proves that the one dimensional ISING model is always paramagnetic and does not undergo a phase transition. (The large fluctuations when  $T \rightarrow 0$  are in fact the sign of a phase transition but this point will not be broached). When  $h \neq 0$  the spins are parallel to it at low temperature. However, when  $T$  increases, the thermal fluctuations dominate and  $M \rightarrow 0$ . The role of the coupling constant is also illustrated on this plot. When the spins are strongly interacting, a demagnetization delay is introduced because it is harder for the thermal fluctuations to surpass these interactions. These considerations also apply to the energy: when  $h = 0$  and at low temperature, all the spins are parallel and the ground state is  $-J = -1$ , as expected. When we put a magnetic field, all the spins are aligned between them and with  $h$  such that the energy of the ground state is doubled (see the discussion of section I.3). Another interesting point is to be denoted concerning the magnetic susceptibility<sup>3</sup> (see figure 4c). When there is no magnetic field,  $\chi$  follows a fast decay proportional to  $1/T$  (because  $\beta e^{\beta J/2} \sim_{T \rightarrow \infty} \beta \equiv 1/T$ ). This result is known as the CURIE law for paramagnetic materials and establish, again, that the one dimensional ISING model is a paramagnetic system. None of these quantities diverge (consider, for instance, the specific heat that follows a smooth curve) and so we conclude that there is no phase transition.

### III.2 Two dimensional case

#### III.2.a Implementation

In the two dimensional case, various lattice shapes can possibly be considered. In the following, we will use a square lattice of length  $L$  such that each spin possesses four nearest neighbor. In this case, the ISING Hamiltonian (eqn. 2) is written on the form

$$H = -J \sum_{i=1}^{L^2} \sum_{j=1}^4 s_i s_j - h \sum_{i=1}^{L^2} s_i . \quad (16)$$

and the energy difference under a spin flip is

$$\Delta E = 2s_{ij} [h + J(s_{i+1,j} + s_{i-1,j} + s_{i,j+1} + s_{i,j-1})] , \quad (17)$$

where  $s_{ij}$  denotes the spin at position  $(i, j)$ .

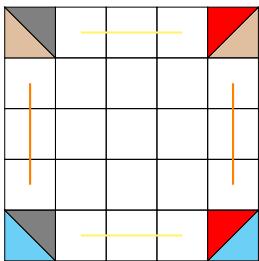


Figure 5: Boundary conditions for a square lattice.

in algorithm 2.

The implementation of the code is almost exactly the same as for the 1D case (see listing 2) except that the array that holds the value of the spins `int * m_spins` becomes a matrix, that is: `int ** m_spins`. An additional method is also written in order to return the value of the nearest neighbor in regard to the imposed boundary conditions. These latter are illustrated in figure 5. If a spin is on an edge, then one of its neighbor is not inside the lattice. Consider for instance a spin on the upper edge (yellow line on figure 5). The neighbor at position  $(i+1, j)$  is not inside the lattice and the boundary conditions consist in assuming that the latter is the spin on the opposite side (at position  $(1, j)$ ). If a spin is in a corner, two of its nearest neighbor are not inside the lattice. For example, the neighbor of the spin  $s_{11}$  (bottom left) will be the spins at positions  $(1, L)$  and  $(L, 1)$ . The method, called `double get_neighbor (int i, int j) const noexcept`, thus consider every special cases (eight in total) before returning the value of the spins. If the spin is not on an edge or in a corner, it simply returns the value of the spins accordingly to eqn. 17. Every other functions work exactly the same as for the one dimensional ISING model and are already implemented

#### III.2.b Thermalization

One advantage in the two dimensional case is that one can easily display the evolution of the system during the thermalization process (see figure 6). This allows to observe the relaxation towards equilibrium that we discussed previously (from figure 6a to figure 6d) and to obtain the equilibration time  $\tau$  ( $\tau \simeq 2 \times 10^7$  in this case). Notice how many METROPOLIS loops are needed to equilibrate the system when the dimension of the network increases. The compilation time becomes large and this show one of the limitations of the METROPOLIS algorithm. We also remark that, contrarily to what we have seen in figure 3, the magnetization of the system reaches 1 at low temperature even when no magnetic field is applied. This is the sign that the two dimensional ISING model is *ferromagnetic* at low  $T$  whereas it is *paramagnetic* at high  $T$ , as we will see in the next section.

<sup>3</sup>The algorithm partially recovers the theoretical result for this quantity. It is possibly due to an approximative determination of the equilibration point. Statistical errors should also be considered...

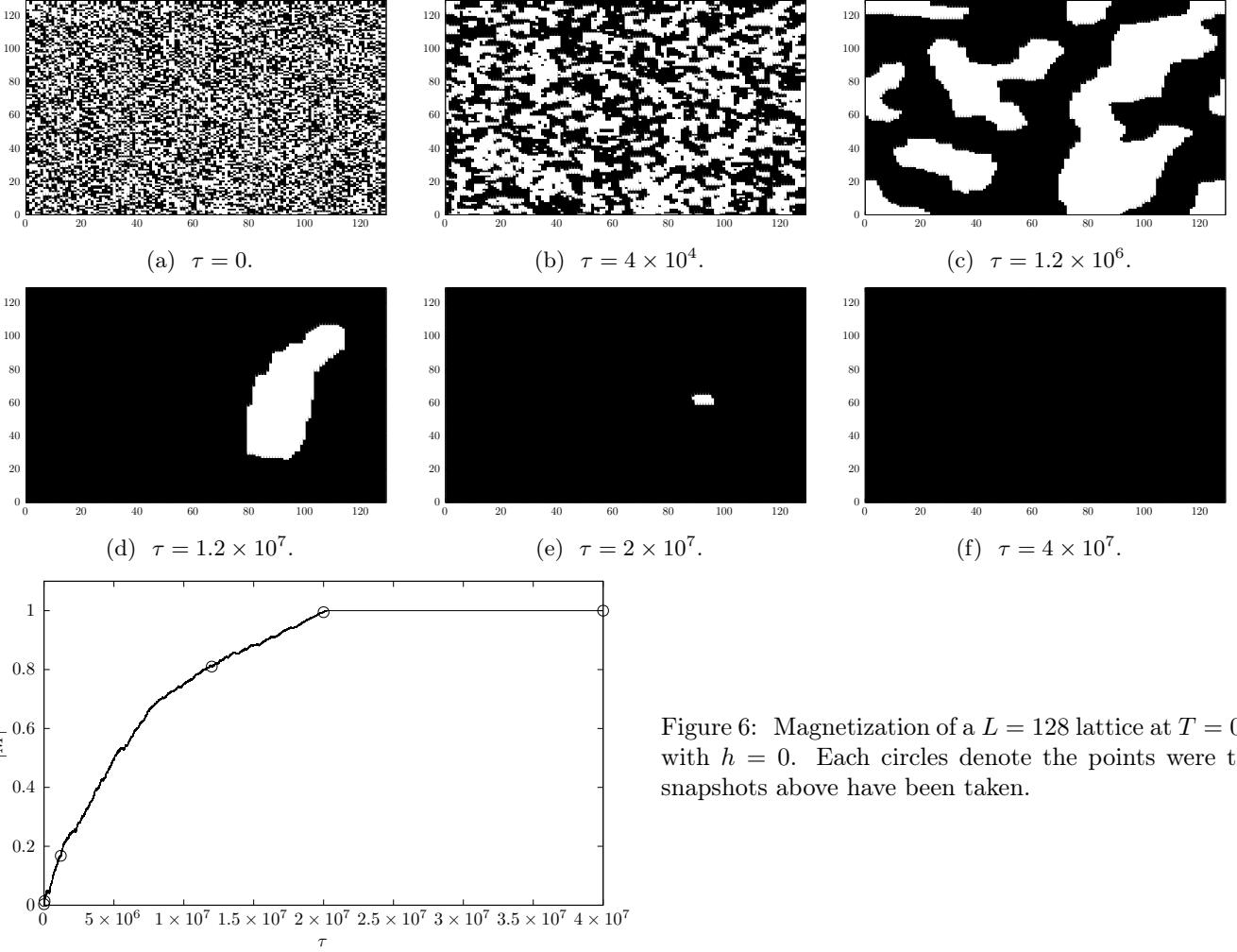


Figure 6: Magnetization of a  $L = 128$  lattice at  $T = 0.1$  with  $h = 0$ . Each circles denote the points were the snapshots above have been taken.

### III.2.c Results

In 1944, Lars ONSAGER proved in a mathematical ‘tour de force’ that the order parameter of a two dimensional square lattice obeys the following relations [6]

$$M(T) = \begin{cases} \left( 1 - \left[ \sinh \left( \frac{2J}{kT} \right) \right]^{-4} \right)^{1/8} & \text{for } T < T_c \\ 0 & \text{for } T \geq T_c \end{cases}, \quad (18)$$

where  $kT_c/J = \frac{2}{\ln(1+\sqrt{2})} \approx 2.269185$  is the critical temperature of the system.

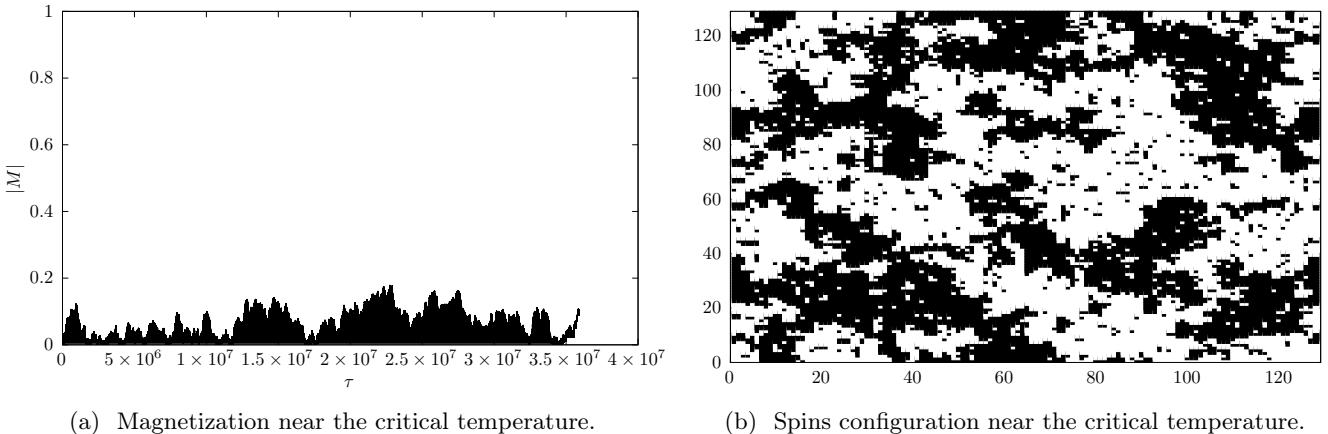


Figure 7: Behavior of the system ( $L = 128$ ,  $h = 0$ ) at  $T \approx T_c$ .

The snapshots on figure 6 were taken at really low temperature which, according to the ONSAGER’s result, explains why  $M \rightarrow 1$ . If the temperature increases, the magnetization will slightly decrease until it steeply goes to zero at

$T = T_c$ , as shown on figure 8a. At this point, the system undergoes large scale fluctuations because it goes from a non-symmetric phase (ferromagnetic) to a symmetric one (paramagnetic), hence, the algorithm becomes increasingly slow [6]. Even after  $4 \times 10^7$  sweeps, equilibrium is not obtained and approximately half the spins are up ( $\uparrow$ ) while the other half are down ( $\downarrow$ ). This phenomenon is known as *critical slowing down* and is illustrated in figure 7.

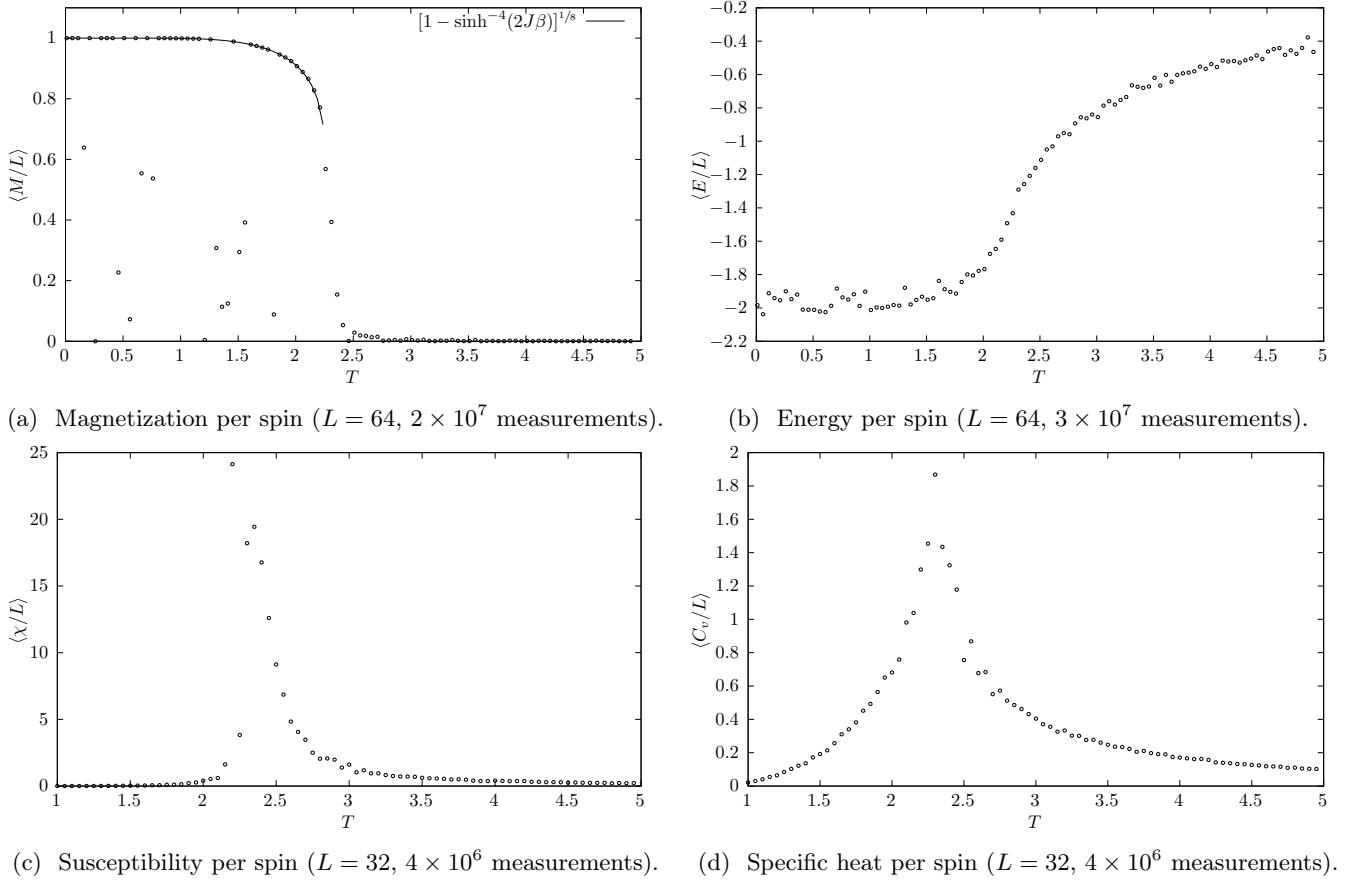


Figure 8: Average quantities versus temperature for the two dimensional square lattice with  $h = 0$  and  $J = 1$ .

One see on figure 8a that, even at low  $T$ , it is difficult for the algorithm to equilibrate the system in 2D. This is why we can see residual points that are not on the theoretical curve, plotted as reference. The problem becomes even bigger when the size of the lattice increases because more sweeps are needed to compute the average. A standard personal computer is thus usually stuck with lattices of size  $L \approx 64$  which is problematic since the theoretical result is derived for  $L = \infty$ . Nevertheless, the algorithm still recovers the ground state of the energy which is, for a  $d = 2$  dimensional square lattice,  $-dJ = -2$  in our case. After the phase transition, the spins are randomly aligned and hence, the energy increases. The magnetic susceptibility and the specific heat are calculated with eqn. 15 since using the absolute value of the magnetization is the only way to study the critical behavior of the system [3]. One clearly see a complete divergence near the phase transition because the points do not follow a smooth curve (compare with figures 4c and 4d). Notice that, bigger is the size of the lattice, and higher the ‘peak’ is<sup>4</sup>. So, again, it would be convenient to be able to perfom the calculations with a bigger value of  $L$  to determine the critical temperature. Here, we can only assume that  $T_c$  is approximately included in the interval  $T \in [2, 2.8]$ . Hopefully, another method have been proposed by Kurt BINDER and gives a more accurate result.

### III.2.d Determination of the critical temperature

In sections I.2.b and II.3 we briefly alluded to finite size scaling. The goal here is not to give a general overview of the subject because it is vast. Only two points will quickly be discussed. First, contrarily to the one dimensional case, the 2D ISING model is ferromagnetic at low  $T$ . All the spins are aligned and they are thus *correlated*. These correlations are usually characterized by a correlation length  $\xi$  (with  $\xi \rightarrow L$  for  $T < T_c$  and  $\xi \rightarrow 0$  for  $T > T_c$ ) which plays an important role in the critical slowing down near the phase transition [2]. To take them into account, the expression of the statistical error (eqn. 14 of section II.3) needs to be replaced by [4]

$$(\text{error})^2 = \frac{\sigma^2}{L} (1 + 2\tau_{\mathcal{O}}/\delta t) \quad , \quad (19)$$

<sup>4</sup>The other curves are not presented on the same plot because it was difficult to distinguish the differents points. However, the program can easily generate the datas.

where  $\tau_{\mathcal{O}}$  is the *correlation time* for the observable  $\mathcal{O}$  (we refer the reader to the cited references for more on the subject).

Second, it is important to have in mind that near the phase transition, the critical behavior of the system *in the thermodynamic limit* (i.e.  $L \rightarrow \infty$ ) can be extracted from the size dependence of the free energy  $F$ . In 1971, Micheal FISHER demonstrated that  $F$  can be expressed as a power law of the  $\alpha, \beta, \gamma, \nu$  coefficients [6], namely

$$F(T, h, L) = L^{-(2-\alpha)/\nu} \hat{F}(\varepsilon L^{1/\nu}, h L^{(\gamma+\beta)/\nu}) , \quad (20)$$

where  $\varepsilon \equiv (T - T_c)/T_c$  and  $\hat{F}$  is some function. These latter coefficients tend to depend only on global properties of the system and their values label the *universality class* to which the system belongs. A differentiation with respect to  $h$  and then setting  $h = 0$  leads to the finite size scaling laws

$$M = L^{-\beta/\nu} \hat{M}(\varepsilon L^{1/\nu}), \quad \chi = L^{\gamma/\nu} \hat{\chi}(\varepsilon L^{1/\nu}), \quad C_v = L^{\alpha/\nu} \hat{C}_v(\varepsilon L^{1/\nu}), \quad (21)$$

where  $\hat{M}$ ,  $\hat{\chi}$  and  $\hat{C}_v$  are scaling functions. Because of the generality of these coefficients, it is an important task (that we did not have time to investigate...) to determine them. However, in addition to these quantities, we may obtain important informations by examining higher order moments. For an ISING model in zero field, BINDER proposed a fourth order cumulant (or BINDER cumulant) defined as [4]

$$U_4 := 1 - \frac{\langle M^4 \rangle}{3 \langle M^2 \rangle^2} . \quad (22)$$

When  $L \rightarrow \infty$ ,  $U_4 \rightarrow 0$ , for  $T > T_c$  and  $U_4 \rightarrow 2/3$  for  $T < T_c$ . Furthermore, when the BINDER cumulant is plotted for different lattices sizes, the curves cross at fixed point and this point gives a good approximation of the CURIE temperature. Figure 9 below shows the plots of the fourth order cumulant for various sizes  $L$ . We obtained  $T_c \approx 2.26$  which is not too far from ONSAGER's result ( $T_c = 2.269$ ). Of course, this is an approximation and the exact result is only obtained in the thermodynamic limit. Considering that the plots are made with lattices of sizes from  $L = 4$  up to  $L = 32$  (which is obviously not in the thermodynamic limit), we can argue that with bigger sizes the result would have been even better.

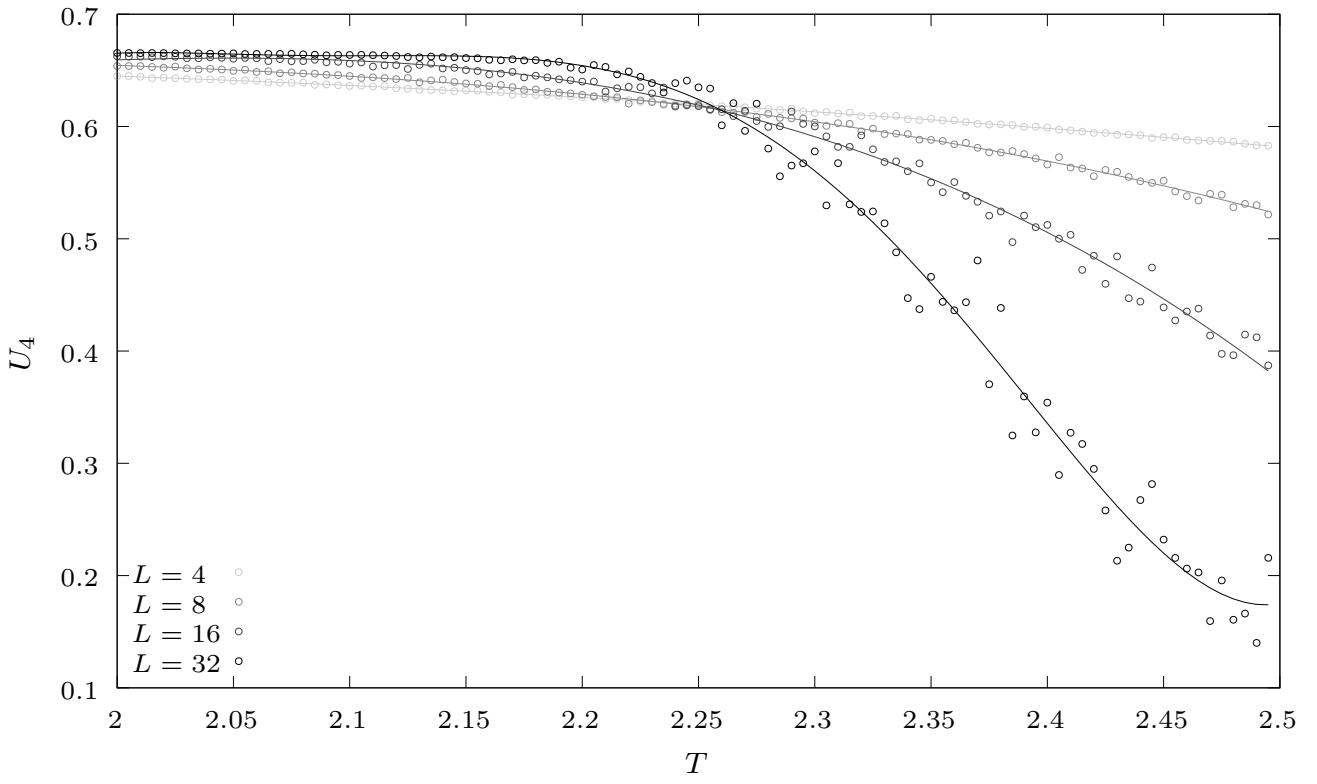


Figure 9: BINDER cumulant versus temperature for different lattices sizes. The straight lines are fittings.

### III.2.e WOLFF algorithm

Before concluding our work, we briefly give a word on improved update methods focusing on the WOLFF algorithm. We previously saw that the METROPOLIS scheme undergoes a critical slowing down when the critical point is approached. This problem is understood since the 80's [2]: at the phase transition, the algorithm does not explore every microstates, hence the ergodicity of the MARKOV chain is broken. We can explain it as follow: the fluctuations of the system become huge and a lot of attempts for a spin-flip are rejected which is why the algorithm becomes increasingly slow.

Since, other methods have been proposed. Some of them find their essence in the percolation problem (which is, roughly speaking, a counting problem). The update consists into randomly selecting one spin of the lattice and exploring its neighbor whose neighbor are then also explored. Each time a neighbor is parallel to the original spin, it is added to a cluster. At the end of the process (i.e. when there are no parallel spins anymore), all the cluster is flipped at once. Those algorithms respect detailed balance (eqn. 9) and thus, the MARKOV chain is ergodic [6]. One major advantage is that these schemes do not suffer of critical slowing down and so they are really fast. We rapidly implemented the WOLFF algorithm<sup>5</sup> and we see on figure 10 that the changes are huge. While we were previously stuck with a lattice of size  $L \approx 64$  (compare with figure 8a), here the system is correctly thermalized after only 30 sweeps even with a lattice of size  $L = 128$  ! (Be aware that this result has to be moderated because one WOLFF step is not equivalent to a METROPOLIS step since multiple operations are performed during the process). One can thus expect this kind of algorithms to give better results, specifically for the critical temperature (but this will maybe be the subject of another work).

### III.3 Conclusion

After giving a brief introduction to Monte Carlo MARKOV chain, we were able to derive the METROPOLIS update for a general  $d$ -dimensional spin system. We then provided an implementation of the one dimensional ISING model and tested it by varying every parameters. The results were analysed and we concluded that the chain of spins does not exhibit a second order phase transition. Considerations of the statistical errors and especially of the finitude of the lattice size allowed us to say that the algorithm recovers well the expected results, hence we moved on the two dimensional square lattice case. Thanks to the Ising1D class, few adaptations were needed and we rapidly obtained the plots of the various physical quantities. Comparison of them with the results found in the literature leads to the conclusion that the algorithm worked as excepted. We clearly obtained the divergence of the magnetic susceptibility and of the specific heat near the phase transition. Nevertheless, the determination of the CURIE was approximative and so we used the BINDER cumulant method. The latter gave  $T_c \approx 2.26$  which is in good agreement with the ONSAGER's result considering the finite size effects mentioned previously. Finally, we also want to emphasize here that we concentrated our efforts on the study of ferromagnetic materials. Yet, the ISING Hamiltonian can describe anti-ferromagnetic systems if the coupling constant  $J$  is set to negative. For the time being, the program can not help to determine the NÉEL temperature because it does not implement the calculation of the staggered magnetization [7]:  $M_s \equiv \sum_i (-1)^i M_i$  (but this can easily be done).

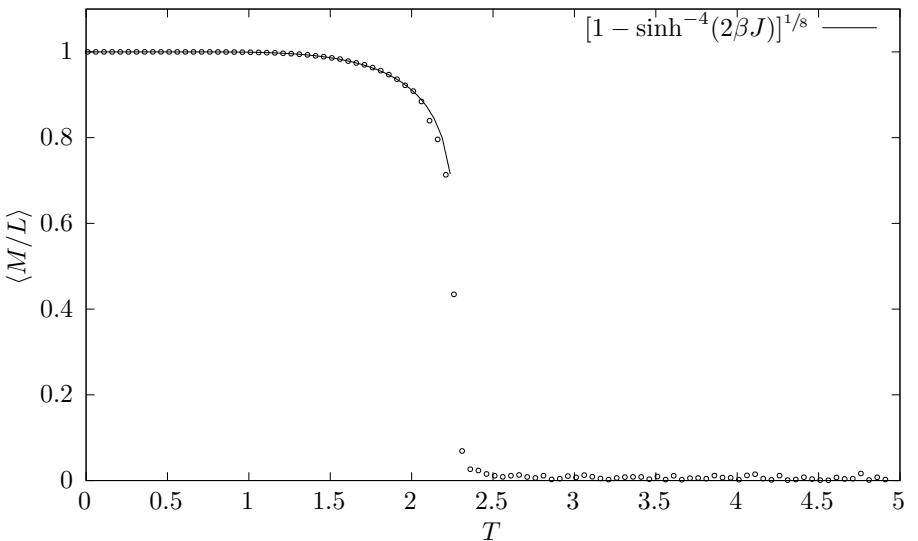


Figure 10: Magnetization versus  $T$  with the WOLFF algorithm. Parameters:  $L = 128$ ,  $h = 0$ ,  $J = 1$ .

## References

- [1] B. Diu et al. *Éléments de physique statistique*. Hermann, 2001. ISBN: 2 7056 6065 8.
- [2] Werner Krauth. *Statistical Mechanics: Algorithms and Computations*. Oxford University Press, 2006. ISBN: 0-19-851535-9.
- [3] Jacques Kotze. *Introduction to Monte Carlo methods for an Ising Model of a Ferromagnet*. Mar. 3, 2008. doi: 10.48550/arXiv.0803.0217. arXiv: 0803.0217[cond-mat]. URL: <http://arxiv.org/abs/0803.0217>.
- [4] David P. Landau and Kurt Binder. *A Guide to Monte-Carlo Simulations in Statistical Physics*. Third. Cambridge University Press, 2009. ISBN: 978-0-521-76848-1.

<sup>5</sup>We did not have time to do it in course but the implementation was fast because the update process was found in [6, p. 714]. The Ising2D class needed changes but the other already implemented functions work the same as before.

- [5] R. K. Pathria and Paul D. Beale. *Statistical Mechanics*. Third. Elsevier, 2011. ISBN: 978-0-12-382188-1.
- [6] Joseph F. Boudreau and Eric S. Swanson. *Applied Computational Physics*. Oxford University Press, 2018. ISBN: 978-0-19-870863-6.
- [7] Tasrief Surungan, Bansawang BJ, and Muhammad Yusuf. “Critical properties of the antiferromagnetic Ising model on rewired square lattices”. In: *Journal of Physics: Conference Series* 1011 (Apr. 2018), p. 012079. ISSN: 1742-6588, 1742-6596. doi: 10.1088/1742-6596/1011/1/012079. arXiv: 1804.09453[cond-mat]. URL: <http://arxiv.org/abs/1804.09453>.
- [8] Ashkan Shekaari and Mahmoud Jafari. *Theory and Simulation of the Ising Model*. Apr. 24, 2021. doi: 10.48550/arXiv.2105.00841. arXiv: 2105.00841[cond-mat]. URL: <http://arxiv.org/abs/2105.00841>.