Contents

List of Figures

List of Tables

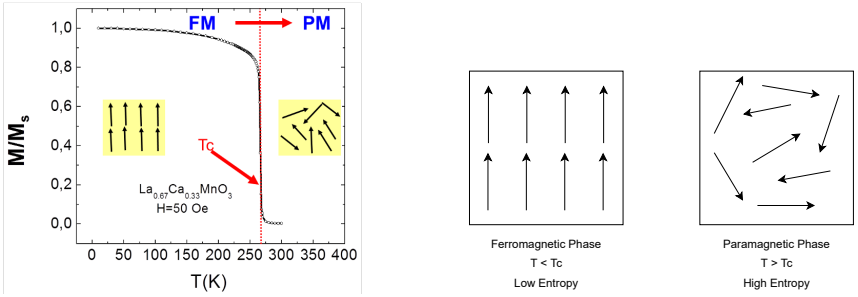
Chapter 1

Ferromagnetism and the Ising Model

In this Chapter a brief introduction to ferromagnetism is presented, from the history and significance of the Ising model, to the joint density of states and relevant thermodynamic relations and properties obtained from it.

1.1 Ferromagnetism

Magnetic materials are critical in our modern society, since they have very broad applications in our daily lives. For instance, there are magnetic materials in every speaker and microphone, hard-drive disks on our computers, they play a big role in medicine where they are used in body scanners (usually in MRI machines) and together with applications in electric motors, transformers and generators [1]. Magnetic materials can be classified in terms of their magnetic properties, such as diamagnetism, paramagnetism, ferromagnetism, and so forth [2].



(a) (b)

Figure 1.1: (a) Magnetization curve for $La_{0.67}Ca_{0.33}MnO_3$ as a function of temperature. Taken from [3]. (b) Spins diagram for the ferromagnetic phase (left) and for the paramagnetic phase (right).

A ferromagnetic material exhibits spontaneous magnetization in the absence of an applied external magnetic field. This way, the magnetic moments of the atoms that compose the material are all naturally aligned along one direction [4]. One of the key features of ferromagnetic materials is that they only display this property below a certain well-defined critical temperature, T_C, usually called the Curie temperature. This temperature defines a phase transition between a ferromagnetic and a paramagnetic state, Figure 1.1(a). Above the Curie temperature, the entropy within the material becomes too strong and the magnetic moments start to dis-align, thus the materials looses its spontaneous magnetization, Figure 1.1(b). In other words, the effect due to thermal disorder overwhelms the ferromagnetic order.

This kind of nature is common to various transition metals, Iron, Cobalt, Nickel, and so forth, and to some rare earth materials, such as Gadolinium. One application of ferromagnets is magnetic refrigeration. This is based on the magnetocaloric effect (MCE), which consists of a change in temperature of a magnetic material by the application and removal of an external magnetic field. For a ferromagnetic material, the MCE is more prominent when the magnetic field is applied for temperatures around the Curie temperature. High performance magnetic refrigerants include the LaFeSi 1:13 family [5] and MnFeP based intermetallics [6].

In recent years, computational materials design is becoming the new norm to discover new materials by combining computer science with quantum and statistical mechanics. It is more advantageous than experimental sciences since it is not as time-consuming and bounded by high costs in equipment as running an experiment [7, 8, 9], however it still can not totally replace the experimental sciences.

1.2 Ising Model

In 1920 Wilhelm Lenz, a German physicist, gave Ernest Ising, his PhD student, an exercise that considered a one-dimensional chain of spin-1/2 particles that can be either pointing up or down. The spins can only interact with their neighbours. Later in 1925, Ising solved this problem, awarding him a doctorate in physics, and concluded that there was no phase transition in the one-dimensional case and wrongly extrapolated that there was no phase transition in higher dimensions [10]. It was not until 1944 that a Norwegian-born American physicist, Lars Onsager, solved the much harder two-dimensional case analytically, in a square lattice [11]. In this case, Onsager showed that there is a well defined phase transition from a ferromagnetic to a paramagnetic state at a critical temperature (T_C), therefore proving Ising's extrapolation wrong.

Since then, the Ising Model has become one of the most studied and published physical models, since it has a non-trivial phase transition while being able to have an analytical solution, at least for dimensions lower or equal to two. As of yet, there are no exact solutions for higher dimensions.We are only able to estimate thermodynamic properties from three dimensional lattices by numerical simulation.

The Ising Hamiltonian describes a system of lattice of atomic spins which can have two spin directions, up (+1) and down (−1) with neighbour to neighbour interactions. So, it is written as

$$\mathcal{H} = -\sum_{<i,j>} J_{ij} S_i S_j - H \sum_i S_i, \tag{1.1}$$

where J is the interaction for constant between two neighbouring particles, S_i is the spin value of the particle at the site i, $< i, j >$ denotes that the sum is conducted over all neighbouring particles. The second term represents the interaction with an external magnetic field, H, which may be zero. If $J > 0$, at $T = 0K$, the system will have a ferromagnetic behaviour since parallel spins are energetically more favourable, and if $J < 0$, the system will behave in an anti-ferromagnetic way.

In this work, I will not evaluate the Ising Model with an applied magnetic field, $H = 0$, and consider $J_{ij} = 1$ to simplify the computations. With this being said, the Hamiltonian treated in this work goes as follows,where ij represent the sum over all lattice sites.

$$\mathcal{H} = -\sum_{<i,j>} S_i S_j \equiv -\frac{1}{2} \sum_{ij} S_i S_j \tag{1.2}$$

To simulate real materials we can obtain the values of the interaction constant for each atom in our lattice through Density Functional Theory (DFT) calculations [12, 13].

1.2.1 Joint Density of States

It is mentioned in any undergraduate course in statistical mechanics that the density of states (DoS), $g(E)dE$, is the number of states that have a certain energy between E and $E + dE$ [14]. For discrete systems, like the Ising Model, the DoS evaluated at a certain energy is the exact number of microstates that have a specific energy E. Through the DoS we can obtain the partition function $Z(T)$, as a function of temperature, and then compute energy related thermodynamic variables, such as the mean energy $E(T)$, specific heat $C(T)$, Helmholtz free energy, $F(T)$. As we are dealing with magnetic systems, our natural interest is to study the magnetization throughout various temperatures and applied magnetic field intensity. For this, we need the number of microstates with a certain pair (E, M). This is the exact description of the Joint Density of States (JDoS), a multi-variable histogram with information about the number of microstates with a certain energy and another parameter, like density, ρ, number of particles, N, or, in this case, magnetization, M. This way we can compute the partition function as a function of both temperature and magnetization, $Z(T, M)$ and the Helmholtz free energy as $F(T, M)$.

Table 1.1 represents the JDoS for the Ising Model in a simple square (SS) lattice with length 2 and periodic boundary conditions (PBC). The extreme magnetization configurations appear only once in the JDoS, no matter the size of the lattice, since they represent all up or all down spins configurations.

Table 1.1: Joint Density of States for the SS lattice with $L = 2$ Ising Model. The rows are the values of energy, E, and the columns the values of magnetization, M.

E / M	-4	-2	0	$+2$	$+4$
-8	1	0	0	0	1
-4	0	0	0	0	0
0	0	4	4	4	0
$+4$	0	0	0	0	0
$+8$	0	0	2	0	0

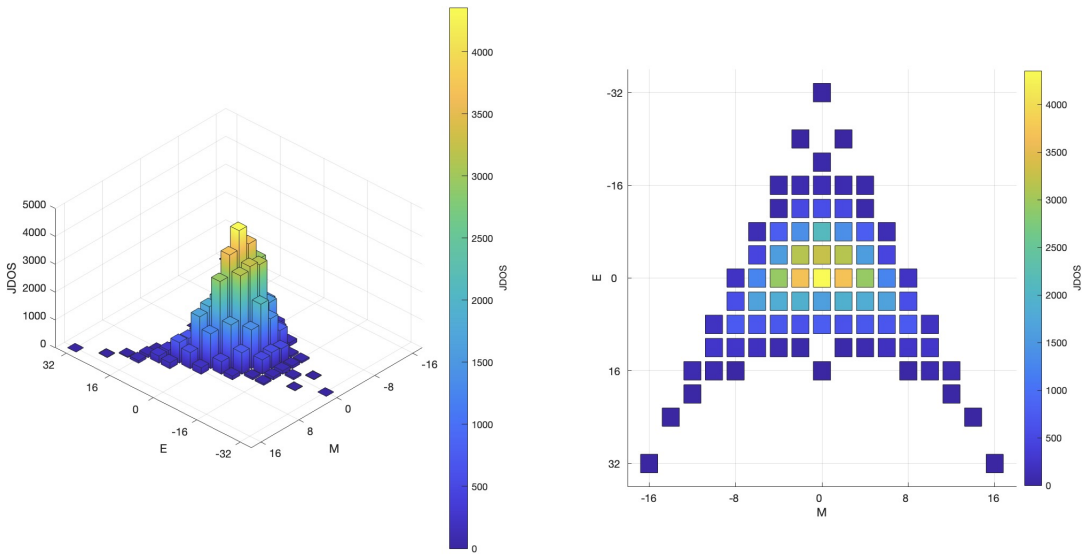


Figure 1.2: Plot of the exact JDoS for the SS L4 Ising Model with PBC. This was obtained by visiting each microstate available to the system.

There are three configurations worth noting.The first is the zero energy and zero magnetization state. This is the state with the most micro configurations since it represents the macrostate in which is likely to find the system in, when $T \gg T_C$ (has the highest entropy). The second is the macrostate with the highest energy and zero magnetization. For every system size and lattice this macrostate has always 2 microstates, since the spins are sorted in a chess/checker board pattern. The last configuration is the state with the least energy and zero magnetization. Its value is always going to be equal to $2L$, since the spins are sorted in rows and columns of alternating positive and negative spins, similar to slices. Throughout this work these configurations will appear again and they will be referred as Zero Zero, Checker Board and Slice, respectively.

The simplest way to compute the JDoS is to visit all of the possible microstates. It is easy to do this when we have 16 spins ($L = 4$ in a SS lattice, Figure 1.2), since the number of possible configurations is still quite small, $2^{16} \approx 65E3$ configurations. But, for instance, for systems with 256 spins ($L = 8$ in a SS lattice) it becomes computationally impossible to randomly or sequentially visit all of the microstates since we have to sample $2^{256} \approx 1E77$ configurations. Instead we can use various clever numerical methods methods to get an estimation of the JDoS in a much more reasonable time with fairly good precision. I will present widely studied methods in the next Chapter and a new unpublished method in the third Chapter.

1.2.2 Thermodynamics

From the JDoS we can obtain all of the thermodynamic quantities when the system is in an equilibrium state. In this section I will present some useful formulas to compute those variables from the JDoS. The probability of a given state with energy E_i, is given by

$$P_i = \frac{\sum_q g(E_i, M_q) e^{-\beta E_i}}{Z}, \tag{1.3}$$

where β is defined as $\beta \equiv 1/k_B T$ and Z is the canonical partition function given by

$$Z = \sum_q Z(T, M_q) = \sum_q \sum_i g(E_i, M_q) e^{-\beta E_i}. \tag{1.4}$$

From this we can obtain mean thermodynamic variables,

$$\langle E \rangle = \frac{1}{Z} \sum_i \sum_q g(E_i, M_q) E_i e^{-\beta E_i}, \tag{1.5}$$

$$\langle M \rangle = \frac{1}{Z} \sum_q \sum_i M_q g(E_i, M_q) e^{-\beta E_i} \equiv \frac{1}{Z} \sum_q M_q Z(T, M_q). \tag{1.6}$$

$\langle E \rangle$ is related to the specific heat by

$$\langle C \rangle = \frac{\langle E^2 \rangle - \langle E \rangle^2}{(k_B T)^2}, \tag{1.7}$$

and finally to obtain the mean entropy we use the second law of thermodynamics,

$$\langle S \rangle = \int \frac{\langle C \rangle}{T} dT. \tag{1.8}$$

However there is another way to estimate thermodynamic properties, from the computed JDoS. Using the Helmholtz free energy, defined as

$$F(T, M) = -k_B \ln(Z(T, M)) \equiv U - TS, \tag{1.9}$$

and the principle of minimum energy, we can extract $F_{min}(T) = \min(F(M, T))$, which is defined as the lowest temperature-dependent free energy that can be reached by the system. From it we can obtain the magnetization and energy for that value of free energy minima, $M_{F_{min}}$ and $E_{F_{min}}$, respectively.Through F_{min} we can use the following thermodynamic relations to compute the specific heat and entropy of the system:

$$C = -T\frac{\partial^2 F_{min}}{\partial T^2}, \tag{1.10}$$

$$S = -\frac{\partial F_{min}}{\partial T}. \tag{1.11}$$

These two different methods of obtaining thermodynamic information about the system, in the thermodynamic limit ($N \to \infty$), converge to the same solution.

1.2.3 Relevance

Despite its simplicity and age, the Ising Model is used in a multitude of research fields within the physical and social sciences.

Within the physical sciences the Ising Model is used to simulate not only magnetic materials but also systems that are characterized by nearest-neighbour interactions and undergo a phase transition Ising-like, this means systems that go from an ordered-low entropy phase to a disordered-high entropy phase at a specific critical temperature [15]. An example would be liquid vapour transitions, where the order parameter is the density, ρ, binary liquid mixtures, where the order parameter is the concentration and the transition corresponds to the mixing of the two liquids.

In the social sciences, the Ising Model has been used to describe a plethora of theoretical models of social behaviour and model financial markets [16]. Topics ranging from the study of racial segregation in certain communities [17], to a demonstration that a community that speaks only one language can start speaking another one without any outside bias [18].

In economics Monte Carlo methods are widely used to model financial markets due to its randomness, and the Ising Model is the standard model that those methods are applied to. In theses case studies the spins are the option of buying or selling a certain stock through neighbour to neighbour communication or external factors. The "magnetization" represents the average actions of the stock market agents [19, 20].

Chapter 2

Monte Carlo Methods Applied to the Ising Model

A short review of the Monte Carlo (MC) methods used to solve the Ising Model in the current paradigm is presented. There will be a focus on the famous Metropolis Method and the Wang-Landau sampling.

2.1 Metropolis Method

The classic Metropolis method, introduced in 1953 by Metropolis et al. [21], belongs to the Markov chain Monte Carlo (MCMC) class of algorithms. These algorithms exploit the fact that if we construct a Markov chain that has a specific equilibrium distribution one can obtain samples recording the states generated by the Markov chain.

In a brief fashion, a Markov chain is a stochastic model that describes a sequence of possible events, in this case microstates,in which the probability of transiting to another state depends on the current state. This way the probability of the next state, S_j, given the current state, S_i, can be written as

$$P(S_j, t) = \sum_i W(S_i \to S_j) P(S_i, t), \tag{2.1}$$

where $W(S_i \to S_j) \equiv W_{ij}$ is the transition probability to move from the state i to j. We require that

$$W_{ij} \geq 0 \qquad\qquad \sum_j W_{ij} = 1. \tag{2.2}$$

The master equation considers the change of the probability of the next state with time, t,

$$\frac{dP(S_j, t)}{dt} = \sum_i \left[W_{ij} P(S_i, t) - W_{ji} P(S_j, t) \right]. \tag{2.3}$$

In the equilibrium regime, the master equation has to equal 0, and we get the detailed balance condition for the equilibrium probability $P_{eq}(S_j)$,

$$W_{ji} P_{eq}(S_j) = W_{ij} P_{eq}(S_i). \tag{2.4}$$

The objective of Metropolis sampling is to generate canonical configurations with an equilibrium probability

$$P_{eq}(E_i) = \frac{\exp(-\beta E_i)}{Z}. \tag{2.5}$$

Here Z is the partition function, however this is usually not know before hand. When considering a Markovian process we generate each new configuration from the preceding one avoiding this problem. As a result the difference of energy between the two states is needed, $\Delta E = E_i - E_j$ and the transition probability of given as

$$W_{ij} = \begin{cases} \tau_0^{-1} \exp(-\beta \Delta E) & \text{if } \Delta E \geq 0 \\ \tau_0^{-1} & \text{if } \Delta E < 0 \end{cases} \tag{2.6}$$

where τ_0^{-1} is the time required to attempt a spin-flip. We often set this time unit to one.

This way the Metropolis method applied to the Ising Model [22], with a fixed external magnetic field and a fixed temperature, goes as follows:

1. Choose an initial state;
2. Choose a spin i and perform a spin-flip;
3. Calculate the energy change from that spin-flip, ΔE;
4. Accept the flip with a probability $\min(1, \exp(-\beta \Delta E))$;
5. When the system reaches equilibrium, measure any thermodynamic quantity needed;
6. Go back to (2) and repeat until there is enough samples of the thermodynamic variables.

Note that we accept the spin flip if a given uniformly random number $r, r \in [0, 1]$, is less or equal to the acceptance criteria. Typically in Monte Carlo simulations we define the MC time as the amount of trial flips equal to the number of spins in our lattice, N.

2.1.1 Success and Limitations

The Metropolis sampling proposed by Metropolis et al. [21], can be successfully applied to an array of models, ranging from widely studied quantum ensembles and gases simulations to state-of-the-art protein and peptide simulations and to machine learning and neural networks. The following paragraphs will be directed to magnetic systems, like the Ising model, but they can be extrapolated to others physical systems.

When estimating thermodynamic variables the simulation must reach the equilibrium stage, where the probability distribution takes the form of Equation 2.5. Then we take a measurement at each MC time, resulting in R total values for that variable. At the end the average of that variable, A, is taken $\langle A \rangle = \frac{1}{R} \sum_i A_i$. For large systems the time taken to reach equilibrium stage is often very long thus making the simulation time consuming. This is worsened by the fact that to study how some thermodynamic variable A changes over a wide range of temperatures or applied fields intensities, we need to run multiple Metropolis simulations for each temperature and field intensity values, making this process very time-consuming.

Lastly there is another shortcoming known as critical slowing down. In short, for computations where the temperature is near the critical temperature, T_C, the sampling slows down, meaning that it is more time-consuming for the computations to reach the equilibrium stage, thus slowing down the overall simulation [22].

2.2 Wang-Landau Sampling

Since its introduction, the Metropolis sampling was the go-to method to study phase transitions and critical phenomena in condensed matter physics and statistical mechanics. In the final decades of the 20th century scientists were committed to develop new methods that could overcome the shortcomings of the Metropolis sampling. Various methods were proposed such as the cluster flip algorithms, where Swendsen and Wang where pioneers, and the multicanonical ensemble method [23]. The first solved the critical slowing down present in the Metropolis and the second could sample rough energy landscapes with ease.

In 2001, Fugao Wang and David P. Landau [23, 24] proposed a new Monte Carlo method, now called the Wang-Landau (WL) method or sampling. The goal of this method diverges from the goal of previous methods. Instead of generating configurations with a canonical probability, this method tries to estimate the canonical partition function

$$Z = \sum_E g(E) \exp(-\beta E), \tag{2.7}$$

through the estimation of the density of states $g(E)$ from a flat histogram in the phase space.

The main difficulty of a simple random walk in the energy space is that the walker would spend most of its time in the highest probable states thus making the random walk ineffective. The idea of the Wang-Landau sampling is to do a random walk,Metropolis like, and accepting the new states with a probability proportional to the inverse to their DoS $\frac{1}{g(E)}$. Doing this we obtain a flat histogram meaning that each macrostate has the same probability of being visited. Since $g(E)$ is unknown *à priori*, in each step of the random walk, $g(E)$ is also being constructed.

The proposed algorithm computed the DoS but, it can also estimate the JDoS by performing the random walk in the phase space composed by energy, E, and the second order parameter, in our case, the magnetization of the system, M. This comes with a downside, since the JDoS has much more information than the DoS it takes much longer compute. Later in 2006, Landau et al. [25], presented a modification of the WL called the global updates method. This is a much harder version to implement but is much more efficient when sampling 2D discrete phase spaces and continuous phase spaces [26].

2.2.1 Algorithm

First we start with an arbitrary configuration of spins and a guess for the density of states. Usually,this guess is $g(E, M) = 1$. Choosing a random site in our lattice, we perform a trial flip and compute the energy-magnetization pair before the trial, (E_i, M_i), and after, (E_j, M_j). The new configuration is accepted with a probability

$$P((E_i, M_i) \to (E_j, M_j)) = \min\left(1, \frac{g(E_i, M_i)}{g(E_j, M_j)}\right). \tag{2.8}$$

Whether the configuration is accepted or rejected, we have to update the histogram and refine the DoS estimation. This way, being the system in the state (E, M),

$$H(E, M) = H(E, M) + 1,$$

and we multiply the current value of the DoS by a modification factor, $f > 1$,

$$g(E, M) = f \times g(E, M).$$

A reasonable choice for the initial value of the modification factor if $f_0 = e$. If f_0 is too small, the simulation will take a very long time to reach all of the possible macrostates, (E, M). If it is too large, then we will have large statistical errors. This process is repeated until the histogram is considered "flat", all of the possible energies have been visited the same number of times. As it is impossible to obtain a 100% flat histogram, we define a rule for flatness as $\min(H(E,M)) > \langle H(E,M)\rangle \times p$; p is chosen according to the size of the problem. For small cases, such as the two dimensional Ising model, p can be set as high as 0.95, but for larger systems the flatness condition may never be satisfied if p is near unity. Once a flat histogram is reached, we set $H(E,M) = 0$, keep the estimation of the DoS and reduce the modification factor, $\sqrt{f_i} \rightarrow f_{i+1}$ and continue the random walk. We stop the simulation if $f < f_{final}$, where f_{final} is a number very close to one (often $f_{final} \sim 1 + 1E - 8$).

At the end, the method gives us the relative density of states. To determine the normalized DoS we can use the fact that

$$\sum_{E,M} g(E,M) = 2^N. \tag{2.9}$$

We can also use number of configurations that have a designated magnetization $\Omega(M)$ to normalize the JDoS. This way we normalize the JDoS in a sequential manner magnetization through magnetization. $\Omega(M)$ is just the possible combinations that exist given the number of spins down $N_\downarrow$ and the number of spins up $N_\uparrow$ ($N_\uparrow + N_\downarrow = N$). So $\Omega(M)$ can be easily calculated by

$$\Omega(M) = \frac{N!}{N_\downarrow!(N - N_\downarrow)!}, \tag{2.10}$$

with

$$N_\downarrow = \frac{N - M}{2}. \tag{2.11}$$

2.2.2 Success and Limitations

The modification factor controls the accuracy and the steps taken to reach a flat histogram in our computations. As it approaches unity, the number of iterations goes to infinity. This way, at the beginning of the simulation, when the modification factor is large, the estimation of the DoS is quite bad, since there we are taking fewer samples with a high weight. At the later stages, the modification factor is lower therefore we have more samples of each macrostate with a lower overall weight. Therefore, the initial stages of the simulation are characterized by the accumulation of low precision statistics and the later stages by the refining of the firsts samples.

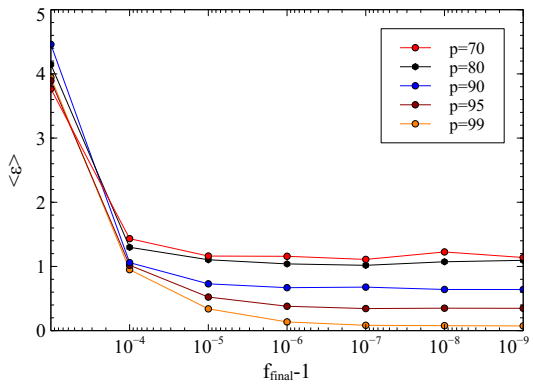
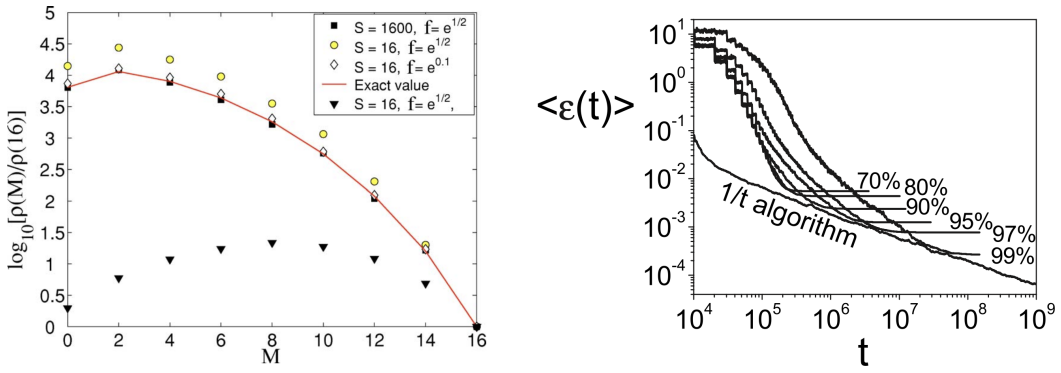


Figure 2.1: Mean absolute error of the JDoS for the Ising model computed by the Wang-Landau sampling for a L4 SS lattice plotted against $f_{final} - 1$. The values were averaged over 100 simulations.

In originally proposed WL method, due to the biased samples at the initial stages of the computations, the final estimation of the JDoS will converge, with high precision, to a inaccurate solution independent of the flatness criteria and f_{final} value used, shown in Figure 2.1.

A few modifications to the original WL were proposed throughout the years, but the most noteworthy are the modifications from Chenggang Zhou et al. [27] and Belardinelli et al. [28]. Zhou has studied the method in great detail [27] and was a co-author of the global updates method [25]. He proposed the introduction of a parameter S, defined as the separation between successive records in the histogram. Meaning that in our random walk we would only update the histogram each S steps. This will diminish the correlation between successive samples and improve accuracy, Figure 2.2(a). The modification proposed by Belardinelli et al. is more complex and introduces a new way of changing the modification parameter during the computations, and named it the $1/t$ time-dependent algorithm. The method starts as the WL does, however, if $f_{i+1} \leqslant 1/t$, where t is the MC time, the modification factor becomes $f_{i+1} = f(t) = 1/t$ and we discard the histogram and update f each MC time step. The simulation only stops when $f(t) < f_{final}$. This ensures that the mean error vanishes, Figure 2.2(b).



(a) (b)

Figure 2.2: (a) $g(M)$, the density of states for magnetization M of L4 SS Ising lattice normalized. The solid line connects the exact values, and the symbols were obtained with different parameters f and S. Data was averaged over 100 measurements, so the statistical errors are smaller than the symbols. Taken from [27]. (b) Comparison between the mean error computed by the original WL for different flatness criteria and the mean error calculated using the $1/t$ time-dependent algorithm for a L8 SS Ising lattice. Taken from [28].

Either in its original forms, or using these more recent improvements, the WL method has become one of the go-to methods for DoS and JDoS estimation, because of its efficiency and ability to sample the whole phase space even if the estimation of the JDoS is not the most precise [27, 28]. It has been applied to magnetic systems like the Ising model, Lennard-Jones fluid simulations, biologic processes, etc [29, 30, 31].

Chapter 3

Flat Scan Sampling

Background for the Flat Scan Sampling (FSS) method and a general description are presented along with a C++ single core implementation details,the method's validation and proof of convergence, a short analysis of the skip parameter and a comparison with WL sampling.

3.1 Background

The author of FSS, João Amaral, had previously, in 2014, proposed a new method to estimate the JDoS, Random Path Sampling (RPS) [32]. The RPS was implemented in high performance languages and extensively studied by Nuno Fortunato [33, 34, 35].

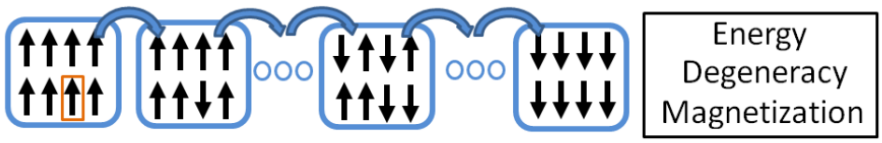


Figure 3.1: Scheme of how the Random Path Sampling method works. Taken from [33].

The RPS method departs from the premise that by starting on an extreme magnetization point in the phase space, generally all spins up (M), and successively flipping one spin down at each step of the random walk, we arrive at the other end of the phase space $(-M)$, as illustrated in Figure 3.1. Performing R sweeps of the phase space generate a histogram that is flat in magnetization and we can obtain the JDoS by

$$H(E,M)/R = P(E,M), \tag{3.1}$$

$$\Omega(M) \times P(E,M) = g(E,M). \tag{3.2}$$

$\Omega(M)$ is defined in Equation 2.10.

The idea for FSS departs from the basic mechanism of RPS, in the sense that it is a method that estimates the JDoS by a sequential sweep of the phase space magnetization by magnetization. However the way of that both methods sample the phase space is completely different. The FSS takes a similar approach to the WL method, in the sense that a random walk with probability proportional to the inverse of the DoS is performed, a flat energy random walk.

3.2 Algorithm

The Flat Scan Sampling method stems from the observation that if the DoS at a certain magnetization M_q is known, by performing a random walk the energy space (E, M_q), with a probability proportional to the inverse of the DoS $\frac{1}{g(E)}$, called a flat energy random walk, and by sampling a set number of statistically diverse configurations for each energy value the DoS at the next magnetization M_{q+1} can be estimated. This is possible because at each step of the random walk we perform a scan, i.e., in a sequential manner, we flip and unflip each spin in our configuration obtaining information about the DoS at the next magnetization, $g(E, M_{q+1})$. The value of $g(E_j, M_{q+1})$ is then computed through the value of $g(E_i, M_q)$ by the following equation

$$g(E_j, M_{q+1}) = g(E_i, M_q) \times \text{fraction of configurations.} \tag{3.3}$$

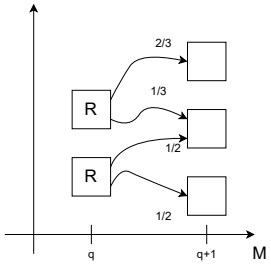


Figure 3.2: Scheme of how the Flat Scan Sampling works.

Shown in Figure 3.2, the fraction of configurations corresponds to the fraction of the scanned configurations in the random walk that contributed to the estimation of the DoS at the next magnetization. This way, the JDoS is computed sequentially by starting at a known DoS in the phase space, such as all of the spins up (M), and sweeping the whole phase space magnetization by magnetization until we arrive at the configuration where all of the spins are down $(-M)$.

The principal parameter of the method is the maximum number of samples for each point in the energy space, known as REP. In later section new parameters will be introduced to try to make the estimation more accurate while sacrificing some performance. There will be also an extensive study of how this parameters affects the precision of the JDoS and wall time.

The algorithm can be written in the following steps:

1. Choose a magnetization value where $g(E, M_q)$ is known, usually magnetization where the spins all up/down;
2. Generate a certain configuration in that magnetization and compute its energy, E_i;
3. Choose a spin to flip down and another one to flip up and compute the energy of the new configuration, E_j;
4. Accept the new configuration with a probability $\min(1, g(E_i)/g(E_j))$;
5. Sequentially flip each spin in the configuration, taking the system from the state (E_i, M_q) to (E_j, M_{q+1}) and accumulate a histogram $H(E_i, E_j)$ and unflip the spins;
6. If the all of the number of sampled states per (E, M_q) pair is equal to REP, stop the simulation and compute the DoS at $q + 1$ by using Equation 3.3. Where the fraction of configurations is now equal to $H(E_i, E_j)/\sum_j (H(E_i, E_j))$.

3.3 Implementation

C++ was the preferred language because of it's speed and optimization over python or Matlab, and modularity, over C. The random number generator (RNG) used was the xoshrio256**.

Before the description of the implementation let us define the function that handles the scan. This should be performed each step of the simulation. The scan is defined as flipping each spin in the configuration, measuring the new energy and registering the change in state $E_i \to E_j$ in the histogram. Thus, this operation can be implemented with a for cycle running from 0 to N, the number of spins.

```
1: function SCAN(configuration)
2:     for idx = 0,1,..., N-1 do
3:         flip the spin idx spin down
4:         compute the new energy Ej
5:         H(Ei, Ej)++
6:         flip the spin idx spin up
7:     end for
8: end function
```

The actual implementation follows the base-line algorithm described in the last section. By knowing that the JDoS is symmetric, to save computing time, we can estimate only half of the JDoS and after the simulation mirror it. Here `qmax` is the index of the last magnetization in our computation. The variable `hist(E)` is used to count how many configurations were sampled in each point of the energy space. We only stop when every point has sampled REP microstates. In pseudo-code it can be written as follows

```
1: for q=0,1,...,qmax do
2:     set hist(E) = 0 and H(Ei,Ej)=0
3:     generate random configuration with M=Mq and compute its energy Ei
4:     scan(configuration)
5:     hist(Ei)++
6:     while min(hist(E) < REP) do
7:         flip one random spin down
8:         flip one random spin up
9:         compute the energy of the new configuration Ej
10:        set ratio = min(g(Ei)/g(Ej))
11:        if rand() < ratio then
12:            accept new configuration
13:        else
14:            reject new configuration
15:        end if
16:        if hist(Ei) < REP then
17:            hist(Ei)++
18:            scan(configuration)
19:        end if
20:    end while
21:    set g(E,Mq+1) = g(E,Mq) * H(Ei, Ej) / sum(H(:,Ej))
22: end for
```

We sample successive configurations, thus reducing statistical accuracy and increasing correlation between scans. This way, similar to the use of the S parameter in the WL method, we can improve sampling correlation by introducing a new parameter called skip. This way, we only sample configurations that are distanced by skip steps in the random walk. Only line 17 is modified by the addition of "and `k % skip = 0`". Usually this value is set equal to the number of spins in the system, N, since, in theory, given N steps in the random walk it might be enough to shuffle the whole configuration. As we will see in the next section, this might not be the best case, and keeping increasing this value might have diminishing returns.

3.4 Validation and Convergence

We will validate FSS in two steps. The first is to show that the method is able to explore the phase space of the system correctly as it is imperative that it does. The second is to verify that JDOS values and sampled (E,M) phase space are reasonable estimation. This can be verified by computing the mean absolute error of the estimated JDoS against the exact JDoS, given by the formula

$$\langle|\epsilon|\rangle = \sum_{E,M} \frac{|g(E,M) - g_e(E,M)|}{g_e(E,M)}. \tag{3.4}$$

Here $g_e(E,M)$ represents the exact joint density of states.As the largest system that we have access to the exact solution is the L4 SS lattice, all of the following studies will be done for that system.

During the flat energy random walk, there is no guarantee that the method will be able to find all of the energies correctly. For low enough values of REP, the random walk might end before the whole energy space is discovered. When just one point in the energy space, during one iteration of the method, is not discovered, the method might not converge to a stable solution, since we use the DoS computed in the last iteration to estimate the new DoS for M_{q+1}. For reasonable REP values, the method does explore the whole phase space available to the system.

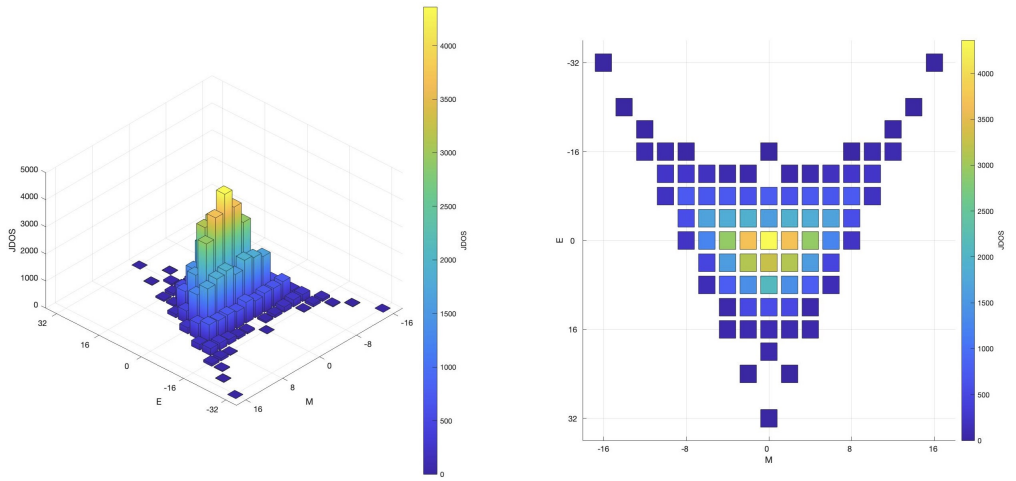
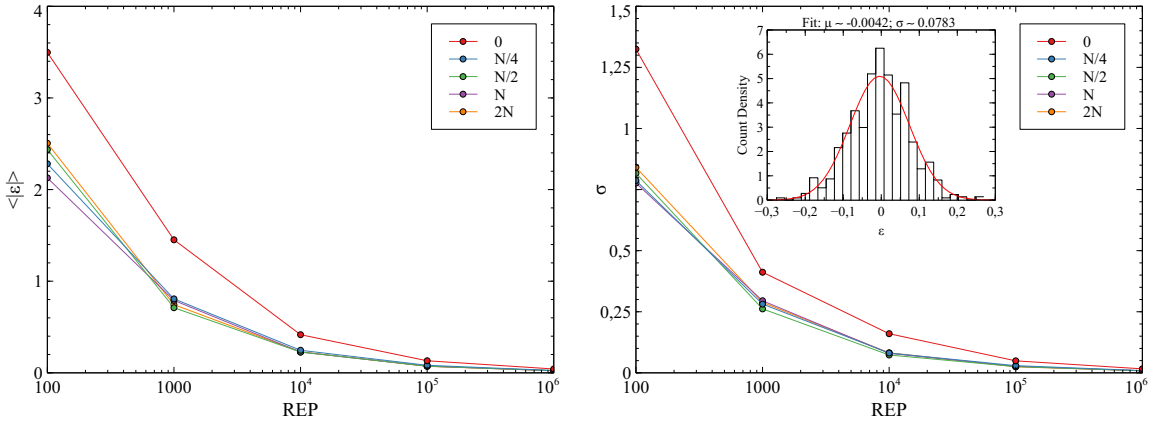


Figure 3.3: Joint density of states for an L4 Ising system in a simple square lattice computed by the Flat Scan Sampling method.

In Figure 3.3, we can see the JDoS computed by the FSS method for an L4 Ising square lattice. Comparing with the phase space of the exact solution for L4 SS Ising system, Figure 1.2, we can see that, in fact, the whole phase space was correctly explored meaning that the FSS method is able to correctly discover all of the macrostates available to the system.

Since the method can find all of the points in the phase space correctly, let us now study how exact and precise those estimations are. For the second part of the validation of the method, let us study how the mean absolute error, Equation 3.4, changes with different REP and skip values.



(a) (b)

Figure 3.4: (a) Mean value of the absolute error for the L4 SS JDoS, Equation 3.4, computed by the Flat Scan Sampling method. (b) Variance of the error of the configurations as a function of REP. The inset is the fit from which the variance was taken from. Both of the plots, (a) and (b), feature various skip values, 0, $N/4$, $N/2$, N and $2N$. The results were average between 1000 simulations to reduce statistical errors.

The mean absolute error from the JDoS for and L4 SS Ising system computed by FSS for different skip and REP values is represented in the Figure 3.4(a). As we increase REP, the absolute error of the JDoS converges to zero as we increase the value of REP. Thus we can have very accurate results from the method at the cost of computing time. Moreover, for different values of skip, the tendency is for the error to get smaller and for small REP values, as we increase skip, the the estimated JDoS becomes more exact while for larger REP values, this increase in accuracy is not as noticeable.

Another crucial aspect when performing a statistical analysis of a Monte Carlo is to study the convergence of the solution. We can study this by determining the mean deviation of the error. There are many ways to find the mean deviation. For this case, as the error follows a normal distribution, inset Figure 3.4(b), the mean deviation is the width of the normal distribution curve. In this case the mean deviation gives us the range of values for the error of a random FSS simulation. In Figure 3.4(b), we can see that as we increase the number of samples, the mean deviation of the error of Flat Scan Sampling computations diminish to a value that seams close to zero.

A typical behaviour of the mean deviation σ for a Monte Carlo method is to be linearly proportional to the inverse square-root of number of samples taken to construct the solution [22]. For the FSS method, the number of steps is proportional to the value of REP, since each point in the phase space has to be sampled REP times. So, for this method the relationship should follow

$$\sigma \propto \frac{1}{\sqrt{REP}}. \tag{3.5}$$

By the analysis of Figure 3.5, we can see a clearly linear relationship between the mean deviation σ and $1/\sqrt{REP}$, meaning that the convergence of FSS is linear with $1/\sqrt{REP}$. Moreover, the inset shows the extrapolation of the mean deviation for an infinite an infinite number of samples per point in the phase space. As we should expect by the analysis of Figure 3.4(b) the variance vanishes as we increase REP. Therefore we can make the simulations as precise as possible at the cost of more computing time by increasing REP.

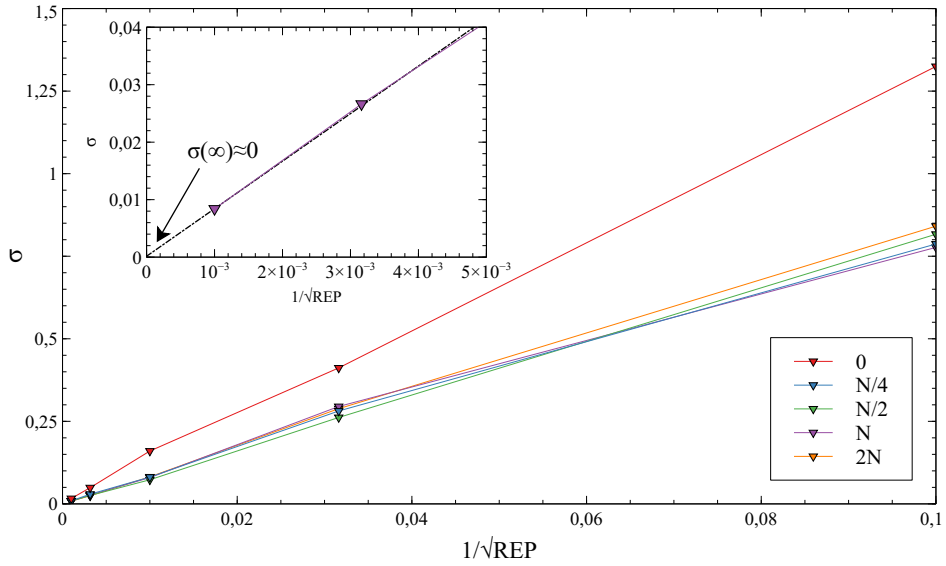


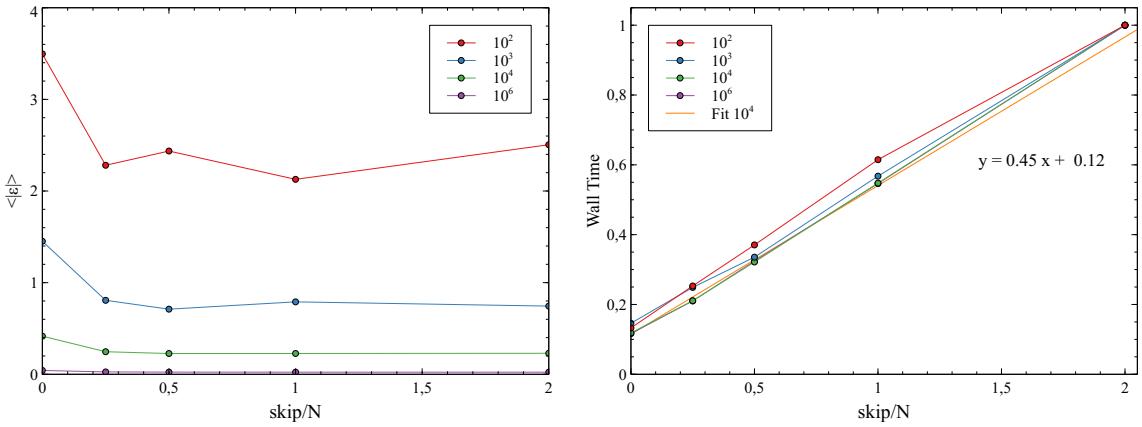
Figure 3.5: Mean deviation as a function of the inverse square-root of REP.

After this study there are three main conclusions to be taken. Flat Scan Sampling method is able to, correctly, find all of the macrostates available to a given system, the method is very accurate as we increase REP, i.e. the error in the estimated results vanishes, and that FSS converges to the exact solution with a mean deviation proportional to the inverse square-root of REP, in other words, it behaves like a standard Monte Carlo method and as we increase REP we expect the range of values for the error to vanish.

3.4.1 Effect of Skip

Figure 3.4(a) was hinting about the behaviour of the parameter skip, with the effect of skip being more noticeable for small values of REP and being phased out when REP is high. In this section a more in depth study of the skip parameter will be given.

The function of skip is to reduce correlation between sequential samples. This naturally leads to less statistical errors when the number of samples (REP in this case) is small. However when the number of samples is high, reducing the correlation won't yield significant results. Analysing Figure 3.6(a), we can see that the mean absolute error of the JDoS computed by FSS does not vanish when skip is increased, like it does when plotted against REP. Instead it plateaus for skip$\approx N/2$ when REP is low, and it does not have a significant effect when the REP value is high. When REP is low for the system, in this case 10^2, skip does it's function and reduces correlation between successive samples therefore reducing the mean absolute error. However, when we increase REP thus taking much more samples per point in the phase space, the correlation between them does not have much effect on the accuracy of the computation.



(a) (b)

Figure 3.6: (a) Mean absolute error of the L4 SS Ising JDoS computed by FSS method as a function of the skip parameter for various REP values. (b) Normalized wall time as a function of the skip parameter with a linear fit for the wall time of REP = 10^4. The results were averaged over 1000 simulations in order to reduce statistical errors.

In the next section a more in depth look at the performance of FSS will be given, but it is worth noting that the wall time is linear with the skip value, Figure 3.6(b). The linear relation can be approximated by $y = 0.45x + 0.12$.

With the insertion of the parameter skip in the FSS implementation, we can have less statistical errors while not increasing the wall time by a big margin. For skip$> N/2$ we will have diminishing returns, so the most advantageous value for skip is $N/2$ or even $N/4$.

3.5 Comparison with Wang-Landau - Accuracy

Having validated Flat Scan Sampling, studied its convergence and the effect of the skip parameter, let us now compare FSS to the widely accepted and studied Wang-Landau Sampling. In this section we will only compare accuracy and precision in the computations and in the next chapter after analysing the performance of FSS we will compare it to WL's performance. We will be comparing results from simulations of the Ising model on a L8 simple square lattice. As we do not have the exact solutions for this lattice size, we will resort to study the two known points in the phase space, the Checker Board and the Slice.

Knowing that the Checker Board has exactly 2 configurations and that the Slice has 16 (2×8) configurations we can compare the precision and accuracy of both methods. In Figures 3.7(a) and 3.7(b) we can see the medium value of the absolute error for the two configurations. The error of the Wang-Landau computations never vanishes, therefore it converges to an inexact solution, while with Flat Scan Sampling we can obtain a really accurate estimation of the JDoS.

Another important aspect worth noting is the difference between the FSS precision in the Checker Board and Slice configurations. The WL computations have roughly the same error in the two configurations, instead the FSS method is less accurate in the Slice configuration. This is due to the nature of the method. As FSS estimates the JDoS in a sequential manner, and the computed JDoS is used to estimate the new one, errors are more susceptible to propagate through each iteration.

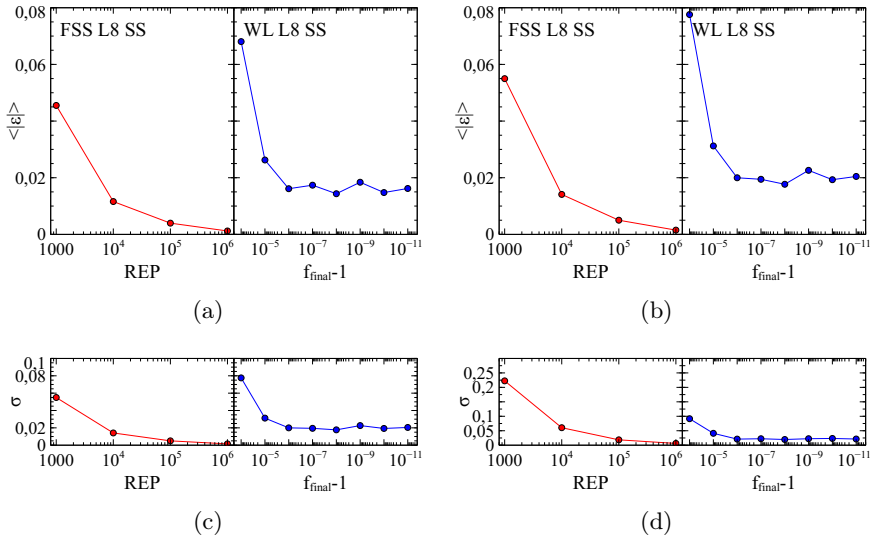


Figure 3.7: Mean absolute error and standard deviation as a function of the method parameter for the Checker Board configuration (a) and (c) and for the Slice configuration (b) and (d). Red lines represent the results for the Flat Scan Sampling and blue lines for the Wang-Landau sampling. Results were averaged through 1000 simulations to reduce statistical error.

As shown in the previous section the error follows a normal distribution so we can analyse its mean deviation. In Figures 3.7(c) and 3.7(d) we can see the sigma values of both methods for the Checker Board and Slice configurations, respectively.For the WL sampling, the mean deviation follows the same pattern as the mean absolute error, it never vanishes, instead it converges to a non-zero small positive value. This is illustrated in Figure 3.8, as we can clearly see, from an extrapolation, that the mean deviation of FSS at infinite samples is zero and, at infinite steps in the random walk, the sigma for WL does not vanish but converges to a small enough value that can be ignored.

In Figure 3.8 we can observe that the mean deviation Flat Scan Sampling results goes to zero while being linear with REP and the WL's does not. This indicates that WL does not behave like a standard Monte-Carlo method, Equation 3.5.

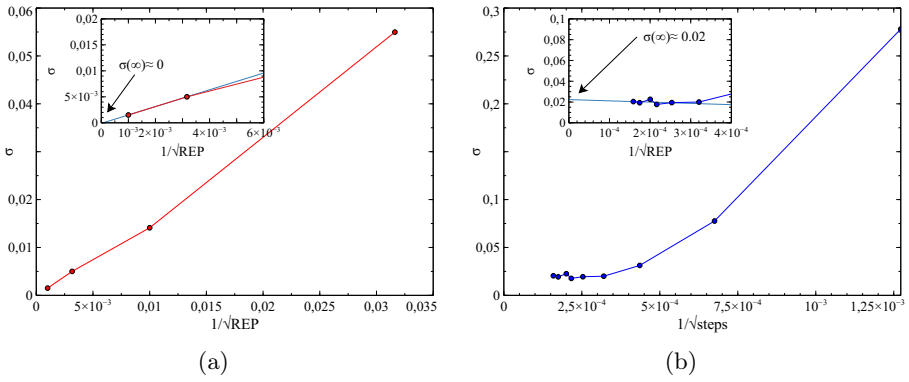


Figure 3.8: Mean deviation as a function of the inverse square-root of the number of steps taken in the random walk during the simulation. (a) FSS (b) WL

In Figure 3.8 we can also see that the mean deviation Flat Scan Sampling results converges to zero while being linear with REP and the WL's does not. This shows that WL does not behave like a standard Monte-Carlo method, Equation 3.5.

Chapter 4

Parallel Programming and Performance

A short introduction to parallel programming and scalability will be given,along with the writers parallel implementation of FSS, a performance and scaling overview and a short comparison with WL.

4.1 Parallel Programming

As the majority of modern CPUs have at least 4 physical processing cores, CPU code parallelization has become more important then ever to exploit all of the performance provided by modern CPUs. Not only that but as the problem that we are trying to solve becomes more complex or when we want to simulate bigger systems, single core computations may not be able to solve the problem in a reasonable amount of time. This way there is a need for parallelization. GPU computing is also an alternative but, generally, is a more complex approach.

In computer science there are two main paradigms for parallel applications [36]. One based on threads, shared memory parallel programming, and another based on processes, distributed memory parallel programming. A shared memory program is executed in multiple threads that coexist in the same memory space. Thus each one has access to the other memory and vice-versa. Therefore communication between concurrent executions of the code within the process is easy. However with multi-threading the code can only be executed in the same computer since it is only run in a single process. Multiple threads running within the same process can reduce parallel performance,. In C/C++, there are many libraries that implement this style of programming, such as OpenMP and PThreads. On the other hand, a distributed memory concurrent program is executed through various processes each in their own memory space. Communication between processes is harder than thread communication. Multi process communication is more performance taxing since they live in different memory spaces. Excessive communication can lead to performance downfalls This downside, however, comes with better parallel performance and the ability to execute computations in various computing nodes. The default C/C++ library for distributed memory parallel programming is the message passing interface (MPI) having many implementations, such as OpenMPI, MPICH or MVAPICH.

4.1.1 Parallel Scalability and Amdahl's Law

Consider a certain task that takes a time T to be solved sequentially by one core. Using N workers, ideally, it would only take T/N to complete the task. This is called a speedup of N. Ideally, this division assumes that the task can be divided into N pieces of equal complexity, that take the same amount of time to finish. In reality this is not true, the N pieces could have little differences in complexity resulting in some workers having to wait for the other to finish. This is known as load imbalance and induces serialization in the computations.

Let us now construct a model for scalability [36]. Considering the overall problem size is $s + p = 1$, where s is the nonparallelizable part and p is the perfectly parallelizable part.In theory, we would want $s = 0$ and $p = 1$. In reality, there are many reasons for a nonvanishing serial part. There can be algorithmic limitations meaning that there might be operations that can not be performed in parallel, bottlenecks in the computer system, such as shared paths to memory between different cores, and communications overhead meaning that in order for workers to inter-communicate there must be some serialization. This last point is often the cause for poor parallel performance.

The serial wall time can be written as

$$T_s = s + p, \tag{4.1}$$

while the wall time of solving the same problem with N workers comes to

$$T_p = s + \frac{p}{N}. \tag{4.2}$$

This way, application speedup or parallel scalability can be define as the quotient of parallel and serial performance. Serial performance is the serial work done over time

$$P_s = \frac{s+p}{T_s} = 1, \tag{4.3}$$

and parallel performance is

$$P_p = \frac{p+s}{T_p} = \frac{1}{1-p+\frac{p}{N}}. \tag{4.4}$$

The application speedup is now

$$S = \frac{P_p}{P_s} = \frac{1}{1-p+\frac{p}{N}}. \tag{4.5}$$

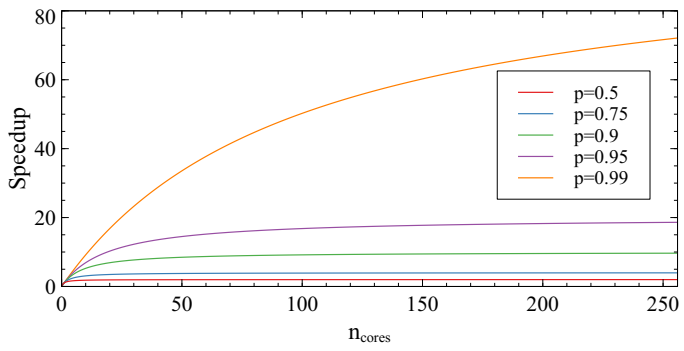


Figure 4.1: Speedup from Amdahl's Law as a function of the number of cores for five different values of the parallel portion of the code p.

Equation 4.5 is known as Amdahl's Law, derived by Gene Amdhal in 1967 [37]. This limits the speeup when $N \to \infty$ to $1/(1-p)$. This famous equations tells us how much faster can a program run when using N CPU cores. In Figure 4.1 the Amdahl's Law speedup is plotted against the number of cores for different values of p.

There are modifications to this law [38] to accommodate system bottlenecks, parallel synchronization and inter-core communications. The most noteworthy is the Hill and Marty's model [39], which introduces a new way of defining the work and performance. An asymmetric architecture has one manager that does not work and a symmetric architecture has a manager that also does work. The performance of the Hill and Marty's model depends on the type of architecture considered.

$$P(N,r) = \begin{cases} \frac{N-r}{r}\sqrt{r} & \text{if symmetric} \\ n-r & \text{if assymetric} \end{cases} \tag{4.6}$$

where r is the size of the sequential core used during the parallel execution.

4.2 Implementation

Due to the nature of FSS algorithm we can have various independent walkers sampling their own histogram. Each walker performs an independent random walk with access to the whole energy space and constructs a histogram for a magnetization M_q. All of the sampled histograms are combined to compute the DoS at M_{q+1}. The next iteration is performed using the computed DoS by joining all of the histogram contributions in the last iteration. This way, in one iteration there is only need for 2 communications, one is the broadcasting of the computed DoS in the last iteration to all of the walkers, the other is the collection of the sampled histograms by all of the walkers. There is a need for one walker to execute all of the serial computations such as the outputting to the terminal and to files, and joining all of the DoS at the end of an iteration and broadcasting it to all of the other walkers. Considering this approach, distributed memory parallelism was the better option for a high performance implementation and MPI was used to carry the process to process communications.

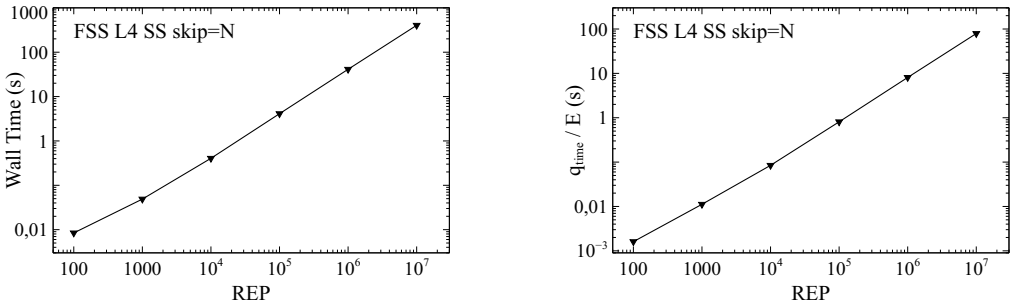
The MPI implementation starts by assigning each walker a different seed for the RNG, so the results come from different streams of random numbers. This is achieved by multiplying a defined seed by the number assigned to that walker. Then, the first iteration is computed by the manager process and broadcasted to all of the walkers. When all of the walkers have sampled the assigned number of configurations, they send the histogram to the manager and it computes the DoS at the third magnetization finally broadcasts it to the walkers. This is repeated until $q = q_{max}$. At the end, the manager writes the output files. As the only serial operations is the output to the terminal and the normalization of the JDoS at the end of an iteration we can expect a close to linear parallel scalability.

Two scenarios of this implementation were explored. One where the manager also performed is own random walk through the energy space, acting like a manager and a walker, and another were the manager only performed the single core operations.Here the advantage is on the side of the first scenario, since there is one more walker and having that walker also performing single core tasks gives an overall better performance than having one idle processes during 95% of the computation.

4.3 Method Performance

Now let us discuss the performance of FSS method of both implementations starting by the single core performance and moving on to scaling tests. The following single core benchmarks were performed in a computer equipped with a Ryzen 9 5950X at stock speeds.

In Figure 4.2(a), we can observe the wall time and the time spent sampling per energy value as a function of the parameter REP. We can see as we increase REP by 10x, the wall time is also increased by 10x.



(a) (b)

Figure 4.2: (a) LogLog plot of the single core wall time of FSS as a function of the number of samples per each energy point, REP. (b) LogLog plot of the time spent sampling for each energy value, q_{time}/E.These are computations for a system with 16 spins. Data was averaged over 1000 computations to reduce statistical error.

The time spent sampling configurations for each energy value, q_{time}/E, is a very important metric, Figure 4.2(b). This value is computed by dividing the wall time spent on that iteration by the number of sampled energy points. Each iteration of the simulation we should be attentive to the behaviour of q_{time}/E, since it can tell us if the computations are stable or the number of samples, REP, must be increased. For a stable calculation, this time must remain approximately the same throughout the computation. If q_{time}/E keeps increasing each iteration then the value of REP is insufficient to correctly explore the energy space for a given magnetization. This causes the JDoS estimate to be less accurate or in the worse cases to be wrong.The obvious solution would be to increase the value of REP, but increasing the skip value can be enough in some cases.

The FSS method scales linearly with the repetitions,Figure 4.2, so given the wall time of that system with a certain REP value we can easily estimate how much time the computations will take for a another REP value. When running the algorithm for a new system we can not have a precise estimation for the wall time. However, knowing only the q_{time}/E for the first iterations and assuming that the REP value is sufficient so that this time does not increase, we can estimate the full wall time to estimate the complete JDOS. Assume that the system has NE values of energy and NM values of magnetizations available.The JDoS will have $NE \times NM$ points, but since we only compute half of the JDoS, $(NE \times NM)/2$. Considering the typical shape of the (E,M) phase space of the Ising model, about 1/3 to 1/2 of those points will be different than 0, so the estimation for the wall time comes as

$$\text{Wall Time} \approx (q_{time}/E)\frac{NE \times NM}{2} \times \text{fraction of non-zero points.} \tag{4.7}$$

Table 4.1: Wall Time of some Ising systems with their respective wall time estimates by Equation 4.7 with two different values for the fraction of non-zero energy points.

Lattice	N	$NE \times NM$	q_{time}/E	Wall Time	Estimate 1/3	Estimate 1/2
SS	16	289	$0.11s$	$4s$	$5s$	$8s$
SS	64	4225	$0.85s$	$728s$	$600s$	$900s$
SS	256	66049	$1s$	$18942s$	$10898s$	$16512s$
SC	64	6305	$1.1s$	$1148s$	$1144s$	$1733s$
FCC	256	197633	$20s$	$871118s$	$652188s$	$988165s$

Table 4.1 presents different wall times for single core computations and their respective estimation through Equation 4.7 for two different fractions of non-zero energy points, 1/3 and 1/2. For larger Ising systems in 3D lattices, computation times can get very large. As we will see in the next Section, we can greatly reduce them through parallelism.

4.3.1 Parallel Scaling

The tests for parallel scaling were performed on the Cirrus-A computing cluster on INCD facilities under the Advanced Computing Projects grant, reference CPCA/A1/7252/2020 with support from FCT. In this project tests with a maximum of 320 cores were performed to better understand the parallel scalability of the Flat Scan Sampling method.

Having constructed a model for parallel scalability, Equation 4.1, we can fit Amdahl's Law to experimental wall time values as a function of the number of cores n. For this we need to solve Equation 4.1 for p since our interest is to gain an insight about the portion of the algorithm that is perfectly parallelized.

$$p_n = \frac{n}{n-1}\left(1 - \frac{1}{S(n)}\right), \tag{4.8}$$

where $S(n) = T_1/T_n$. The mean value of p is then

$$p = \frac{1}{N-1}\sum_{i=2}^{N} p_i. \tag{4.9}$$

This is the standard method to estimate p given the wall time per number of cores. Through this method we find $p = 0.9655 \approx 0.97$, blue line in Figure 4.3(b).

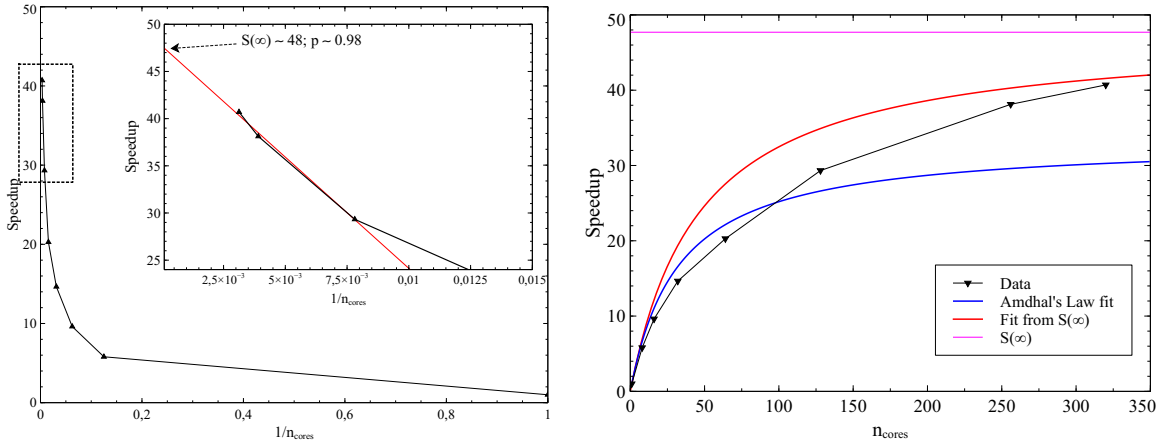
Another method to estimate p is to find the speedup for and infinite number of cores $S(\infty)$ and by using the relation

$$S(\infty) = \frac{1}{1-p}, \tag{4.10}$$

we can estimate p. The best way to find the speedup of infinite cores is to plot $S(1/n)$ and perform a linear extrapolation for $1/n = 0$, Figure 4.3(a).Here we obtain a limiting speedup of 48 and a value for p of 0.98, very close to the valued found by the standard method.

In Figure 4.3(b), the Amdahl's Law curves for the two values of p estimated are shown along with the experimental data from the cluster. Neither of the methods of estimating p give accurate results, since in our model we are not considering communications overhead nor architectural constrains. A more in depth and accurate study could be done, considering for example the Hill and Marty's model [39].

However some strong conclusions can still be drawn.The standard method for estimating p acts as a lower bound for the speedup while the $S(\infty)$ method acts as an upper bound, since the experimental data is bounded by these two fits. We have found that $0.97 < p < 0.98$.



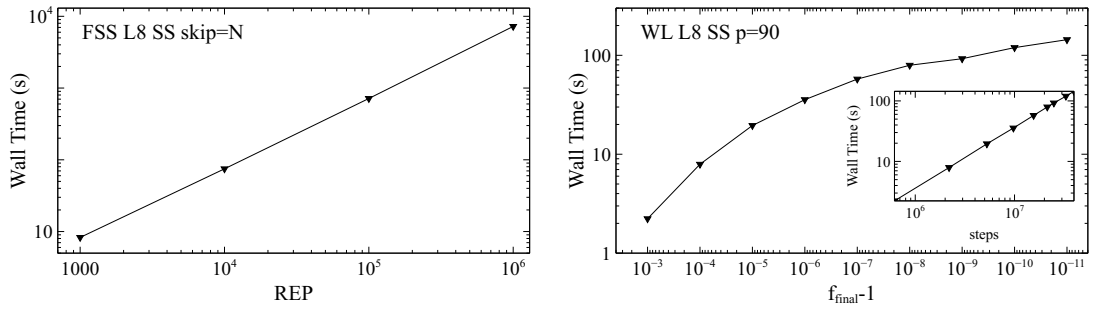
(a) (b)

Figure 4.3: (a) Extrapolation of the speedup for an infinite number of cores. We have a limiting speedup of 48 and an $p \approx 0.98$ (b) Amdahl's law fitted for the wall time data. The fit from Amdahl's Law and from the speedup at infinite cores create a lower and upper bound, respectively.

This is a remarkable result, since it means that 97 to 98% of the algorithm can be perfectly parallelized. In other words, if we split the between N walkers, 98% of the single core wall time will be divided by the N walkers. Thus we can conclude that Flat Scan Sampling is a highly parallelizable algorithm and can have performance gains upto 48 times the single core wall time.

4.4 Comparison with Wang-Landau - Performance

In the previous Chapter we have already look at the differences in precision and accuracy between Flat Scan Sampling and Wang Landau. Now a comparison of the performance will be given. In this Section will use the same L8 SS results from the last comparison Section.



(a) (b)

Figure 4.4: Wall time in seconds as a function of REP for FSS (a) and as a function of f_{final} and the steps taken during the simulation for WL sampling (b).

The single core wall time of FSS scales linearly with the parameter REP, Figure 4.4(a), however for the WL sampling, the wall time is not linear with the parameter f_{final}. Our interest in this Section is to study the performance of WL as a function of the steps in the simulation. In the inset of Figure 4.4(b) we can see that WL also scales linearly with the steps in the simulation.

One important point to highlight is that the wall time of FSS is very predictable as shown in Table 4.1. The wall time of WL sampling can not be that easily predicted. Due to the real time estimation of the JDoS, Wang-Landau's random walk is very unforeseeable and it can get stranded, for a while, between a few points in the phase space, wasting a lot of time, Figure 4 and 5 in [40]. Thus we can have faster than average simulations or very long simulations. On the other hand, we have seen if the value of REP is not high enough, Flat Scan Sample's random walk can get stuck and never finish. With Wang-Landau we are assured that the computation will always finish, even if it takes a very long time.

A critical advantage of Flat Scan Sampling is the power of parallelization. Since FSS is an embarrassingly parallel algorithm and, if well implemented and optimized, $0.97 < p < 0.98$, Figure 4.3(b), we can achieve parallelization with up to 48 times performance gains. This also means that a GPU implementation is easily in reach. Although it is possible to parallelize the Wang-Landau algorithm, it is much harder than FSS, since we have to divide the phase space between different walkers and have them communicate at the barriers [41].

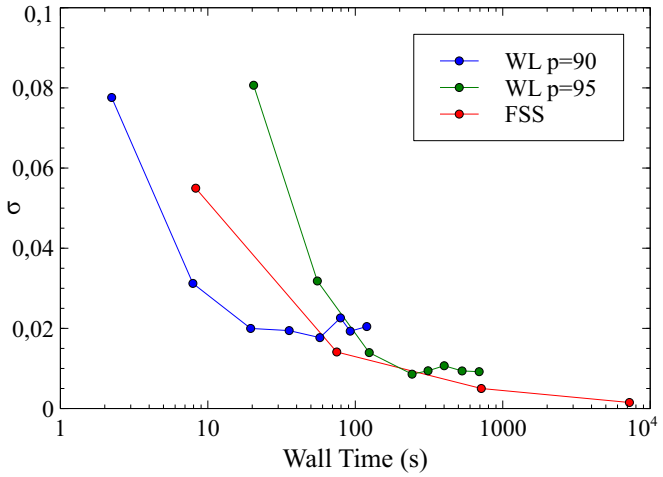


Figure 4.5: Standard deviation as a function of the wall time for the Checker Board configuration for an L8 SS Ising system.

To finalize this comparison, let us shift our focus to Figure 4.5. Here we can see the standard deviation as a function of wall time for the FSS and the WL method for two flatness conditions. For the Wang-Landau with $p = 90$, we can get a more precise solution with Flat Scan Sampling int the same or less computing time with REP $= 10^4$ (second point).In order to get a more refined solution with WL sampling, we need to increase the flatness condition, green line in Figure 4.5. Flat Scan Sampling still has more precision with less or equal computing time with a value of REP between 10^4 and 10^5. Therefore, we can conclude that FSS will have better precision in the calculations per computing time than the WL method.

Having done this comparison we can take a few important and string conclusions. Flat Scan Sampling and Wang-Landau method are two vastly different methods. FSS can be as accurate at the cost of computing time, while WL's accuracy and precision plateaus for all flatness criteria.WL's precision per simulation time is worse in most cases than FSS's. Thus, FSS is a better method to obtain fairly quick and very precise computations while with WL we can get an inaccurate solution very quickly.

Chapter 5

Thermodynamics and Finite Size Scaling

In this Chapter, a comparison between mean and exact thermodynamic variables is presented along with the estimation of the Curie temperature for the infinite lattice.

Firstly one important note on thermodynamic calculations. Extracting thermodynamic information from the JDoS can be a difficult process, specially when trying to study large systems. Although the formulas from Chapter 1 are easy to understand, the problem arrives when trying to actually compute the partition function and consequent Boltzmann factors for the ensemble average variables.

For large systems, the values from the JDoS are very large, for instance, for an L16 SS Ising system, some of the values of the JDoS are of the order 10^{74}. We can not keep summing and multiplying these values to compute Z, since we run into overflow errors. For low temperatures we run into the same error, since the exponential gets very large. Instead we have to work with the logarithms of these functions to extract the thermodynamics out of the JDoS. We can use the following observation,

$$\ln\left[\sum_i a_i\right] = \ln(a_0) + \ln\left[1 + \sum_{i\neq 0} \frac{a_i}{a_0}\right] \tag{5.1}$$

5.1 Mean and Exact Thermodynamic Variables

As previously mentioned in Chapter 1, we can calculate thermodynamic variables from the JDoS in two different ways. One by using the ensemble averages through the partition function and Boltzmann factor and another by using the minimum of the Helmholtz free energy. For three different square lattice system sizes, these thermodynamic quantities can be seen in Figure 5.1. For the C_{minF}, there is only one size since smaller sizes give very inaccurate results due to numerical differentiation, Equation 1.10.

We would expect that the thermodynamic quantities computed by these two equivalent ways would yield similar results. By a quick analysis of Figure 5.1(a) and 5.1(b) we can see that the magnetization curves computed by the two methods are not that similar. The $\langle|M|\rangle$ has a flatter curve while the M_{minF} has a more pronounced curve, where you can better see the phase transition. If we kept increasing the number of spin particles in our simulations, eventually these two ways would give us very similar results, and, for the infinite lattice, they would be equivalent and equal to the Onsager solution. Another noteworthy aspect is the Curie temperature T_C that we can extract from these two curves. The point in which a phase

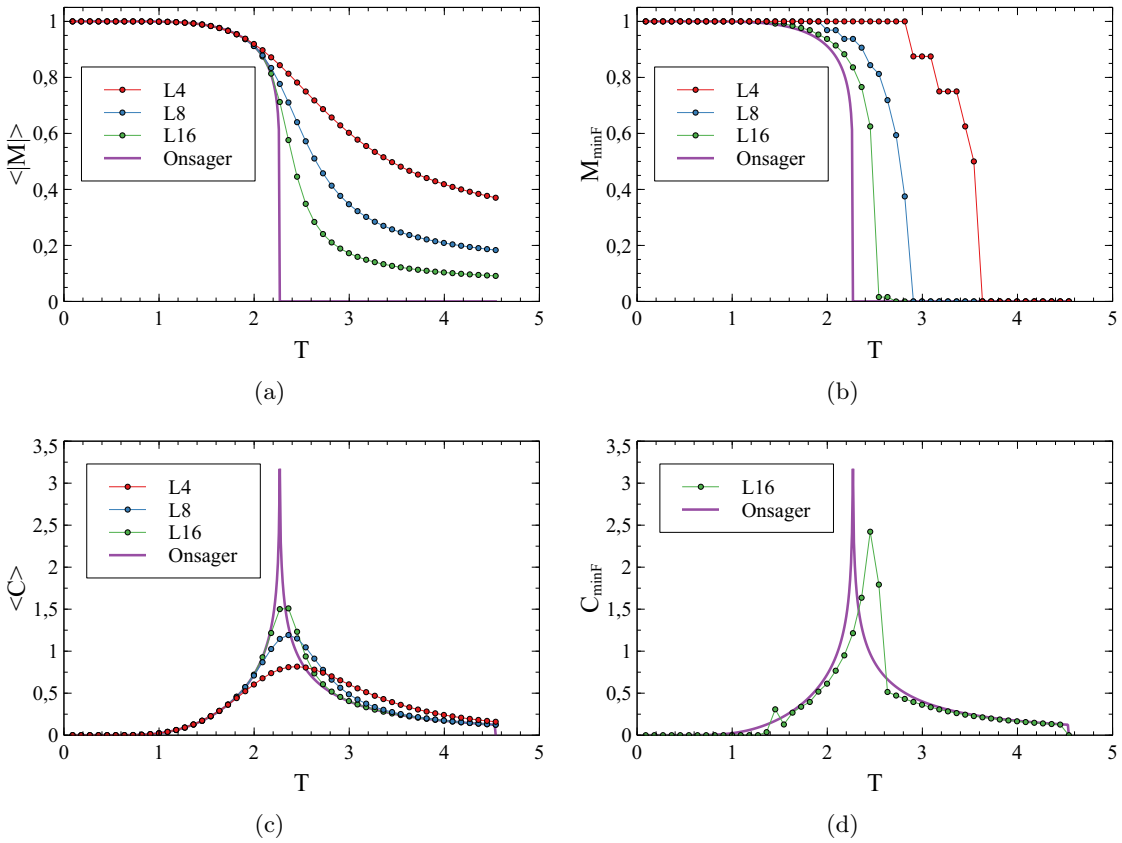


Figure 5.1: Thermodynamics for 3 different lattice sizes compared with the exact solution by Onsager. (a) $\langle|M|\rangle$. (b) M_{minF}. (c) $\langle C\rangle$. (d) C_{minF}.

transition occurs is the inflexion point, therefore the point where the first derivative has a relative extrema. In Table 5.1 there can be seen the Curie temperatures estimated through the two magnetization curves for the different system sizes.

Table 5.1: Curie temperature estimated by the $\langle|M|\rangle$ and M_{minF} for the three different lattice sizes and their respective error. The first derivative of the magnetization has a curve with a similar shape of a Gaussian function. Thus the error can be estimated considering the half height width.

L	T_C $\langle\|M\|\rangle$	$\Delta-$	$\Delta+$	T_C M_{minF}	$\Delta-$	$\Delta+$
4	2.542	0.726	0.908	3.541	0.091	0.091
8	2.452	0.454	0.363	2.815	0.182	0.091
16	2.361	0.272	0.182	2.452	0.091	0.091

The Curie temperatures taken from the average and exact thermodynamic variables are much different. The $T_C\langle|M|\rangle$ is closer to the T_C for the infinite lattice but T_C M_{minF} has much less statistical error thus it is a more precise value, since the magnetization curve is more pronounced.

The mean specific heat and the specific heat obtained from the minimum of F, Figure 5.1(c) and 5.1(d),respectively, have similar properties as the magnetization. The mean heat capacity has a smoother curve compared to the Onsager solution while the exact specific heat has a curve similar to the Onsager solution where the phase transition and Curie temperature are more pronounced.

5.2 Estimating T_C for the Infinite Lattice

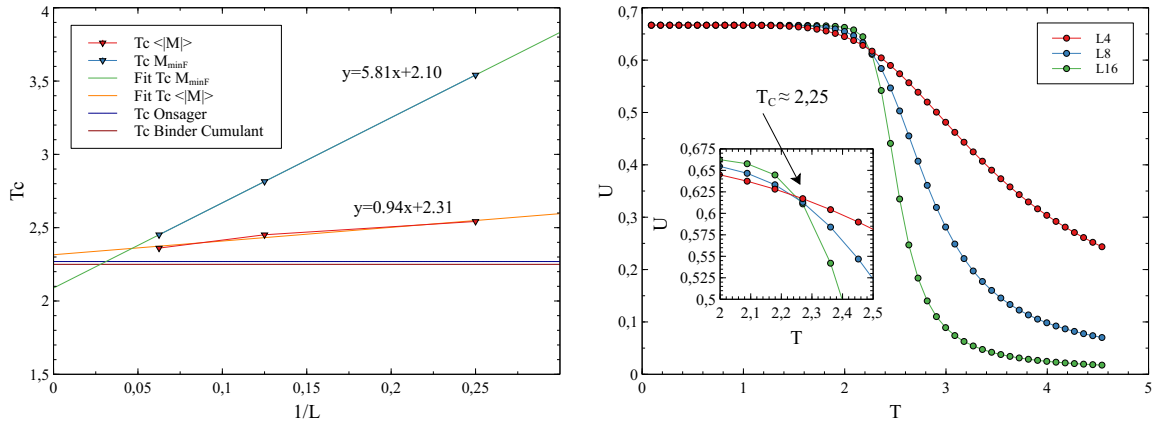
The estimation the Curie temperate for the infinite system can be obtained by two different methods [22]. The first method relies on knowing the Curie temperature for each L for example the peak on the heap capacity of by the differentiation of the magnetization. This $T_C(L)$ defines a phase transition in the finite lattice. Thus we can use the linear relation

$$T_C(L) = T_C(\infty) + cL^{-1/\nu}, \tag{5.2}$$

where ν is equal to unity to consider the Onsager solution, $T_C \approx 2.269$ [11]. Another way is to use the fourth order cumulant U_L, known as the Binder cumulant, defined as

$$U_L(T) = 1 - \frac{\langle M^4 \rangle}{3\langle M^2 \rangle^2}. \tag{5.3}$$

For a infinite system size, $L \to \infty$, the cumulant takes a value of 0 for $T > T_C$ and 2/3 for $T < T_C$. Plotting the Binder cumulant for different system sizes, the curves all cross a point. The temperature of this point is an estimation of the Curie temperature for the infinite lattice. The accuracy of these two methods greatly improves if larger system sizes are considered.



(a) (b)

Figure 5.2: (a) Linear regression of the $T_C(L)$ from $\langle|M|\rangle$ and M_{minF} as function of $1/L$. (b) Binder cumulant for different system sizes.

For the first method, a linear regression with the $T_C(L)$ values from Table 5.1 for the two magnetization curves is presented in Figure 5.2(a). As we are only interested in the $T_C(\infty)$ from Equation 5.2, for the the $\langle|M|\rangle$ curve, the estimated critical temperature for the infinite lattice is approximately 2.31 while for the M_{minF} is approximately 2.10. For the second method, the Binder cumulant for different L values is displayed in Figure 5.2(b). The critical point is the point in which the curves intersect each other. This yields an estimation $T_C \approx 2.25$.

For the available system sizes, the Binder cumulant gave the best results with $T_C \approx 2.25$. This is a great method of estimating the T_C for the infinite lattice when we do not have results from bigger lattices, L32 or L64. The critical temperatures estimated from linear fitting the T_C values form magnetizations is a good method to estimate the T_C for the infinite if we have a JDoS for a bigger system.All of these estimation could be improved by considering larger systems, for example the L32 simple square.

Chapter 6

Conclusion

In this work a novel Monte Carlo method, Flat Scan Sampling, to estimate the Joint Density of States of the Ising model was presented. It combines some aspectes from both RPS and WL sampling. A C++ implementation was proposed with two main parameters, REP and skip. FSS can correctly find all of the points in the phase space of an Ising system. The error and mean deviation of the computations vanish when REP is increased and FSS behaves like a standard Monte Carlo method, i.e. the standard deviation converges to zero with the inverse of the square-root of REP. The skip parameter reduces statistical correlation between samples, reducing also the error in the results. When the REP is lower, these effects are more pronounced. Moreover, the wall time of FSS scales linear with REP and skip and a formula to estimate single core wall time is proposed with success.

A parallel version of the Flat Scan Sampling C++ implementation is also presented and developed with success. With this implementation, studies of parallel scalability were preformed up to 320 computing cores. By fitting Amdahl's Law to experimental results, a value for $p = 0.98$ was obtained, meaning that 98% of the algorithm can be perfectly parallelized. This translates to gains up to 48 times of the performance of single core computations that can be achieved.

A comparison with Wang-Landau sampling was performed to test convergence, accuracy and performance of both methods. The error in Wang-Landau's computations is prominent and never vanishes, due to biased samples at the beginning of the simulation. With FSS, the error does vanish for high REP, costing computing time. As mentioned before, FSS behaves like a standard Monte Carlo method and in this study, it was confirmed that WL does not. It's mean deviation is not linear with the inverse of the square-root of the number of samples nor it converge to 0. Wang-Landau's wall time is linear with the random walk steps like FSS. Finally, FSS will have in average better precision per computing time than WL method. This way, Flat Scan Sampling is a better method to obtain fairly quick and very precise computations while with Wang-Landau we can get an inaccurate approximation very rapidly.

At the end, the differences between mean and exact thermodynamic variables and some methods to estimate Curie temperatures T_C for the infinite lattice were discussed.Mean thermodynamic variables, i.e., variables computed through ensemble averages, have a smoother curve thus a less defined phase transition region resulting in exact but inaccurate results. On the other hand, exact thermodynamic quantities, the phase transition is more visible meaning that critical temperatures are more precise. Estimating the critical temperature for the infinite lattice can be achieved through the linear fitting of T_C values from the magnetization plots or the Binder cumulant. The Binder cumulant method gave better results since it can estimate, with a high degree of accuracy, the T_C for the infinite lattice with results from smaller lattices. The linear fitting can be improved by having some results from the L32 or

L64 lattice.

This method could easily be extended to the SpinS Ising model where instead of spin-1/2 particles we would consider spin-S particles. In theory we can get better simulate real materials since not all atoms in materials have a total angular momentum of 1/2.

A GPU/CUDA implementation of Flat Scan Sampling would also be of interest since the method is highly parallelizable and using thousands of GPU computing cores solutions to complicated and larger systems could be achieved with ease.

Bibliography

[1] O. Gutfleisch, M. A. Willard, E. Brück, C. H. Chen, S. G. Sankar, and J. P. Liu, "Magnetic materials and devices for the 21st century: Stronger, lighter, and more energy efficient," *Advanced Materials*, vol. 23, no. 7, pp. 821–842, 2011.

[2] D. J. Griffiths, *Introduction to Eletrodynamics*. Cambridge University Press, 4th ed., 2013.

[3] J. C. S. Amaral, "Estudo de Manganites Modificadas com Iões de Terra Rara," 2005.

[4] Stephen Blundell, *Magnetism in Condensed Matter*. 2001.

[5] A. Fujita, S. Fujieda, Y. Hasegawa, and K. Fukamichi, "Itinerant-electron metamagnetic transition and large magnetocaloric effects in La(FexSi1-x)13 compounds and their hydrides," *Physical Review B - Condensed Matter and Materials Physics*, vol. 67, no. 10, pp. 1044161–10441612, 2003.

[6] O. Tegus, E. Brueck, K. H. J. Buschow, and F. R. de Boer, "Transition-Metal-Based Magnetic Refrigerants for Room-Temperature Applications.," *ChemInform*, vol. 33, no. 14, pp. no–no, 2010.

[7] S. Curtarolo, G. L. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy, "The high-throughput highway to computational materials design," *Nature Materials*, vol. 12, no. 3, pp. 191–201, 2013.

[8] W. Chen, J. George, J. B. Varley, G. M. Rignanese, and G. Hautier, "High-throughput computational discovery of In2Mn2O7 as a high Curie temperature ferromagnetic semiconductor for spintronics," *npj Computational Materials*, vol. 5, no. 1, 2019.

[9] S. Sanvito, C. Oses, J. Xue, A. Tiwari, M. Zic, T. Archer, P. Tozman, M. Venkatesan, M. Coey, and S. Curtarolo, "Accelerated discovery of new magnets in the Heusler alloy family," *Science Advances*, vol. 3, no. 4, pp. 1–10, 2017.

[10] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Zeitschrift für Physik*, vol. 31, no. 1, pp. 253–258, 1925.

[11] L. Onsager, "A Two-Dimensional Model with an Order-Disorder Transition," *Physical Review Letters*, vol. 65, no. 3 and 4, pp. 117–149, 1944.

[12] G. Bihlmayer, *Density Functional Theory for Magnetism and Magnetic Anisotropy*. 2020.

[13] N. Marzari, A. Ferretti, and C. Wolverton, "Electronic-structure methods for materials design," *Nature Materials*, vol. 20, no. 6, pp. 736–749, 2021.

[14] R. K. Pathria and P. D. Beale, *Statistical Mechanics*. 3rd ed.

[15] A. Pelissetto and E. Vicari, "Critical phenomena and renormalization-group theory," *Physics Report*, vol. 368, no. 6, pp. 549–727, 2002.

[16] D. Stauffer, "Social applications of two-dimensional Ising models," *American Journal of Physics*, vol. 76, no. 4, pp. 470–473, 2008.

[17] P. Abell and P. Abell, "The Journal of Mathematical Sociology Some Aspects of Narrative Method," no. July 2015, pp. 37–41, 2010.

[18] D. Nettle, "Is the rate of linguistic change constant?," *Lingua*, vol. 108, no. 2-3, pp. 119–136, 1999.

[19] L. Damodaran and K. M. Udayanandan, "Dynamics of stock market , using Ising model," pp. 71–78.

[20] P. dvorak, "Ising Model in Finance From Microscopic Rules to Macroscopic Phenomena," pp. 1–56, 2012.

[21] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[22] D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*. 2005.

[23] F. Wang and D. P. Landau, "Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States," *Physical Review Letters*, vol. 86, pp. 2050–2053, mar 2001.

[24] D. P. Landau, S.-H. Tsai, and M. Exler, "A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling," *American Journal of Physics*, vol. 72, no. 10, pp. 1294–1302, 2004.

[25] C. Zhou, T. C. Schulthess, S. Torbrügge, and D. P. Landau, "Wang-landau algorithm for continuous models and joint density of states," *Physical Review Letters*, vol. 96, no. 12, pp. 8–11, 2006.

[26] P. Poulain, F. Calvo, R. Antoine, M. Broyer, and P. Dugourd, "Performances of Wang-Landau algorithms for continuous systems," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 73, no. 5, pp. 1–11, 2006.

[27] C. Zhou and R. N. Bhatt, "Understanding and improving the Wang-Landau algorithm," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 72, no. 2, pp. 1–4, 2005.

[28] R. E. Belardinelli and V. D. Pereyra, "Fast algorithm to calculate density of states," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 75, no. 4, pp. 1–5, 2007.

[29] K. A. Maerzke, L. Gai, P. T. Cummings, and C. McCabe, "Simulating phase equilibria using wang-landau-transition matrix monte carlo," *Journal of Physics: Conference Series*, vol. 487, no. 1, 2014.

[30] Q. Yan, R. Faller, and J. J. De Pablo, "Density-of-states Monte Carlo method for simulation of fluids," *Journal of Chemical Physics*, vol. 116, no. 20, pp. 8745–8749, 2002.

[31] Q. Yan and J. J. de Pablo, "Fast Calculation of the Density of States of a Fluid by Monte Carlo Simulations," *Physical Review Letters*, vol. 90, no. 3, p. 4, 2003.

[32] J. S. Amaral, J. N. Gonçalves, and V. S. Amaral, "Thermodynamics of the 2-D Ising Model from a Random Path Sampling Method," *IEEE Transactions on Magnetics*, vol. 50, no. 11, 2014.

[33] N. Fortunato, "Random Path Sampling Applied to Vector Models of Magnetism," 2015.

[34] C. O. A. J. S. Amaral, N. M. Fortunato, "Giant magnetocaloric effect of compressible Ising and Heisenberg lattices. ," 2016.

[35] R. K. Korotana, G. Mallia, N. M. Fortunato, J. S. Amaral, Z. Gercsi, and N. M. Harrison, "A combined thermodynamics and first principles study of the electronic, lattice and magnetic contributions to the magnetocaloric effect in La0.75Ca0.25MnO3," *Journal of Physics D: Applied Physics*, vol. 49, no. 28, p. 285001, 2016.

[36] G. Hager and G. Wellein, *Introduction to High Performance Computing for Scientists and Engineers*. CRC Press, 2011.

[37] G. M. Amdahl, "Validity of the single pro cessor approach to achieving large scale computing capabilities," *AFIPS spring joint computer conference*, 1967.

[38] R. G. L. Yavits, A. Morad, "the Effect of Communication and in Multicore Systems," *Parallel Computing*, vol. 40, no. 1, pp. 1–16, 2014.

[39] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era," *Computer*, vol. 41, no. 7, pp. 33–38, 2008.

[40] P. H. Nguyen, E. Mittag, A. E. Torda, and G. Stock, "Improved Wang-Landau sampling through the use of smoothed potential-energy surfaces," *Journal of Chemical Physics*, vol. 124, no. 15, 2006.

[41] J. Yin and D. P. Landau, "Massively parallel Wang-Landau sampling on multiple GPUs," *Computer Physics Communications*, vol. 183, no. 8, pp. 1568–1573, 2012.