FYS3150
Project 4 - Ising Model

Ethel Villeneuve

October 2017
University of Oslo
https://github.com/choukimono/Project_4.git

Abstract

In this project, we want to implement an Ising model, by using the object-oriented programming. The Metropolis algorithm will be run in a lattice 20×20 and then bigger. We will study the properties of the system : the mean energy E, the mean magnetization $|M|$, the specific heat and the susceptibility χ as function of the temperature.
The purpose is then to study the phase transitions. We will again study these properties near the critical temperature (Curie point) in order to detect the phase transition. To improve the performance of the program, we will use MPI to parallelize programs.

Unfortunately, I was not able to have good results for everything. But, from those I have, the equilibrium situation is reached after about 1×10^7 Monte Carlo cycles, which means that we need at least this number of cycles in order to compute the expectation values. We can also see that the number of accepted configurations is directly linked with the number of Monte Carlo cycles, and with the temperature. Moreover, for larger temperature, we have more possible values of the energy than for smaller temperature. I was not able to see the phase transitions with my simulations.

Contents

Introduction

This project is all about implementing the Ising model to study the phase transitions. We will be in the case of a two-dimensional squared lattice, with periodic boundary conditions. In this lattice, we have $L \times L$ particles characterized by their spin (up or down). The starting point of the program will be a random lattice or a lattice with all the spins-up. Then, the Monte Carlo method, using the Metropolis algorithm, will performe flips of spins and we will compute the expectation values of several properties. The different implementation of the Monte Carlo method will improve the performance by avoiding useless calculations.
After running simple simulations in the very simple case of a 2×2 lattice, we will run the model with bigger lattice to study phenomena, such as the time (modelized by the number of Monte Carlo cycles) to reach the most likely state, the difference between when we start the simulation with an ordered lattice or a random one, the probability distribution at different temperatures.

The programs have been done with Antoine Hugounet. We mainly work together for the results.

Chapter 1

Theory

1.1 The Ising model

1.1.1 The general model

The Ising model is a mathematical model used in statistical mechanics. It consists of discrete variables, which represent the magnetic moment of the spin, which can take only two values (here $+1$ or -1). The spins will only interact with their direct neighbors. With this model, we can study the phase transitions at finite temperature for magnetic systems. We can expressed the energy as

$$E = -J \sum_{\langle kl \rangle}^{N} s_k s_l - B \sum_{k}^{N} s_k \tag{1}$$

with J a constant expressing the strength of the interaction between the neighboring spins, $\langle kl \rangle$ indicating the fact that we only sum the nearest neighbors, N the number of spins, $s_{k,l} = \pm 1$ and B an external magnetic field interacting with the magnetic moment set up by the spins. This is the general expression of the Ising model. For our use, we will only focus on the case where $B = 0$.
Then, we will be able to calculate expectation values of the mean energy $\langle E \rangle$ and magnetization $\langle M \rangle$ at a given temperature. To do this, we will use a Boltzmann distribution

$$P_i(\beta) = \frac{e^{-\beta E_i}}{Z} \tag{2}$$

where P_i is the probability of finding the system in the state i, $\beta = \frac{1}{kT}$, T being the temperature and k the Boltzmann constant, E_i is the energy of a state i and Z is the partition function defined by $Z = \sum_{i=1}^{M} e^{-\beta E_i}$ with M the number of configurations. E_i, which is the energy in the state i, is given by, for k, l the spins of the state i :

$$E_i = -J \sum_{\langle kl \rangle}^{N} s_k s_l \tag{3}$$

A simple particular case of the Ising model, the two-dimensional square lattice model, allows us to have an analytical solution, which will be compared to the numerical results.

1.1.2 Two-dimensional square lattice model

This particular case is one of the simplest models to show a phase transition. It is defined by the following conditions : a null external magnetic field $B = 0$, this is a two-dimensional lattice with $N = L \times L$ sites with L the number of spins in each dimension, with periodic boundary conditions.
Let's take the case with $N = 2 \times 2$ spins. We have $2^4 = 16$ different possible states with those conditions. We reuse the equation (3) to find the energy of each configuration i with the spins-up $\uparrow$ taking the value $+1$ and the spins-down $\downarrow$ taking the value -1. Let's take for example the case where three spins are pointing up (and so one is pointing down) numbered from 1 to 4 : $\begin{matrix}\downarrow^{(1)} & \uparrow^{(2)} \\ \uparrow^{(3)} & \uparrow^{(4)}\end{matrix}$.

$$\begin{aligned} E &= -J \sum_{\langle kl \rangle}^{4} s_k s_l \\ &= -J(s_1 s_2 + s_2 s_1 + s_1 s_3 + s_3 s_1 + s_2 s_4 + s_4 s_2 + s_3 s_4 + s_4 s_3) \\ &= -J\left[(-1) + (-1) + (-1) + (-1) + 1 + 1 + 1 + 1\right] \\ &= -J \times 0 \\ \underline{E = 0} \end{aligned}$$

The magnetization formula is the simple sum of all spin of the state $M = \sum_{k}^{N} s_k$ with k each spin of the configuration. So in our example, we have

$$\begin{aligned} M &= s_1 + s_2 + s_3 + s_4 \\ &= (-1) + 1 + 1 + 1 \\ \underline{M = 2} \end{aligned}$$

The following table sums up the possible states of a two-dimensional square lattice model.

Number of spins-up	Possible configurations	Degeneracy	Energy	Magnetization
4	$\begin{matrix}\uparrow & \uparrow \\ \uparrow & \uparrow\end{matrix}$	1	$-8J$	4
3	$\begin{matrix}\downarrow & \uparrow \\ \uparrow & \uparrow\end{matrix}$ $\begin{matrix}\uparrow & \downarrow \\ \uparrow & \uparrow\end{matrix}$ $\begin{matrix}\uparrow & \uparrow \\ \downarrow & \uparrow\end{matrix}$ $\begin{matrix}\uparrow & \uparrow \\ \uparrow & \downarrow\end{matrix}$	4	0	2
2	$\begin{matrix}\uparrow & \uparrow \\ \downarrow & \downarrow\end{matrix}$ $\begin{matrix}\downarrow & \downarrow \\ \uparrow & \uparrow\end{matrix}$ $\begin{matrix}\uparrow & \downarrow \\ \uparrow & \downarrow\end{matrix}$ $\begin{matrix}\downarrow & \uparrow \\ \downarrow & \uparrow\end{matrix}$	4	0	0
2	$\begin{matrix}\uparrow & \downarrow \\ \downarrow & \uparrow\end{matrix}$ $\begin{matrix}\downarrow & \uparrow \\ \uparrow & \downarrow\end{matrix}$	2	$8J$	0
1	$\begin{matrix}\downarrow & \downarrow \\ \downarrow & \uparrow\end{matrix}$ $\begin{matrix}\downarrow & \downarrow \\ \uparrow & \downarrow\end{matrix}$ $\begin{matrix}\downarrow & \uparrow \\ \downarrow & \downarrow\end{matrix}$ $\begin{matrix}\uparrow & \downarrow \\ \downarrow & \downarrow\end{matrix}$	4	0	-2
0	$\begin{matrix}\downarrow & \downarrow \\ \downarrow & \downarrow\end{matrix}$	1	$-8J$	-4

Table 1.1: Energy and magnetization for a $N = 2 \times 2$ spins Ising model with periodic boundary conditions.

The 16 configurations known, we can calculate the partition function in its analytic form.

$$Z = \sum_{i=1}^{16} e^{-\beta E_i}$$
$$Z = (1 \times e^{-\beta \times (-8J)}) + (4 \times e^{-\beta \times 0}) + (4 \times e^{-\beta \times 0}) + (2 \times e^{-\beta \times 8J}) + (4 \times e^{-\beta \times 0}) + (1 \times e^{-\beta \times (-8J)})$$

$$\boxed{Z = 2(e^{8J\beta} + e^{-8J\beta} + 6)} \tag{4}$$
$$Z = 4[\cosh(8J\beta) + 3] \tag{4'}$$

Let's now compute the expectation value of the energy, defined by

$$\langle E \rangle = \sum_{i=1}^{M} E_i P_i(\beta)$$

From (2), we can write :

$$\begin{aligned} \langle E \rangle &= \frac{1}{Z} \sum_{i=1}^{M} E_i e^{-\beta E_i} \\ &= \frac{1}{Z} \left(2 \times (-8J) e^{8J\beta} + 2 \times 8J e^{-8J\beta} \right) \end{aligned}$$

$$\boxed{\langle E \rangle = -\frac{J}{Z} \left(16 e^{8J\beta} - 16 e^{-8J\beta} \right)} \tag{5}$$

$$\langle E \rangle = -8J \frac{e^{8J\beta} - e^{-8J\beta}}{e^{8J\beta} + e^{-8J\beta} + 6}$$
$$\langle E \rangle = -8J \frac{\sinh(8J\beta)}{\cosh(8J\beta) + 3} \tag{5'}$$

Similarly, we compute the mean value of the magnetic moment (or mean magnetization) in its absolute value $\langle |M| \rangle$:

$$\langle |M| \rangle = \sum_{i=1}^{M} |M_i| P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{M} |M_i| e^{-\beta E_i}$$
$$\langle |M| \rangle = \frac{1}{Z} \left(4 e^{8J\beta} + 4 \times 2 e^{0} + 4 \times 2 e^{0} + 4 e^{8J\beta} \right)$$

$$\boxed{\langle |M| \rangle = \frac{8 e^{8J\beta} + 16}{Z}} \tag{6}$$

$$\langle |M| \rangle = \frac{4 e^{8J\beta} + 8}{e^{8J\beta} + e^{-8J\beta} + 6}$$

which match with the fact that we have taken the external magnetic field $B = 0$.

We can use the expression of the expectation value of the energy to find the specific heat C_V which is defined by

$$C_V = \frac{\langle E^2 \rangle - \langle E \rangle^2}{kT^2}$$

The specific heat capacity represents the amount of energy requiered to raise the temperature of a unit of mass by a unit of temperature.

We need to compute $\langle E^2 \rangle$:

$$\begin{aligned}\langle E^2 \rangle &= \sum_{i=1}^{M} E_i^2 P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{M} E_i^2 e^{-\beta E_i} \\ &= \frac{1}{Z} \left(2 \times (-8J)^2 e^{8J\beta} + 2 \times (8J)^2 e^{-8J\beta} \right)\end{aligned}$$

$$\boxed{\langle E^2 \rangle = \frac{J^2}{Z} \left(128 e^{8J\beta} + 128 e^{-8J\beta} \right)} \tag{7}$$

$$\langle E^2 \rangle = 64 J^2 \frac{e^{8J\beta} + e^{-8J\beta}}{e^{8J\beta} + e^{-8J\beta} + 6}$$

$$\langle E^2 \rangle = 64 J^2 \frac{\cosh(8J\beta)}{\cosh(8J\beta) + 3} \tag{7'}$$

Then, with (5), we have

$$C_V = \frac{J^2}{ZkT^2} \left[\left[128 \left(e^{8J\beta} + e^{-8J\beta} \right) \right] - \frac{1}{Z} \left[16 \left(e^{8J\beta} - e^{-8J\beta} \right) \right]^2 \right]$$

$$\boxed{C_V = \frac{256 J^2}{ZkT^2} \left[\cosh(8J\beta) - \frac{4}{Z} \sinh^2(8J\beta) \right]} \tag{8}$$

Similarly, to have the susceptibility χ, which is its capacity to be attracted or not into a magnetic field, defined by

$$\chi = \frac{\langle |M|^2 \rangle - \langle |M| \rangle^2}{kT},$$

we start by computing $\langle |M|^2 \rangle$:

$$\begin{aligned}\langle |M|^2 \rangle &= \sum_{i=1}^{M} |M_i|^2 P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{M} |M_i|^2 e^{-\beta E_i} \\ &= \frac{1}{Z} \left(4^2 e^{8J\beta} + 4 \times 2^2 e^0 + 4 \times 2^2 e^0 + 4^2 e^{8J\beta} \right)\end{aligned}$$

$$\boxed{\langle |M|^2 \rangle = \frac{1}{Z} \left(32 e^{8J\beta} + 32 \right)} \tag{9}$$

$$\langle |M|^2 \rangle = 16 \frac{e^{8J\beta} + 1}{e^{8J\beta} + e^{-8J\beta} + 6}$$

Using (6), we can write

$$\boxed{\chi = \frac{1}{ZkT}\left[\left(32e^{8J\beta} + 32\right) - \frac{1}{Z}\left(8e^{8J\beta} + 16\right)^2\right]} \tag{10}$$

$$\chi = \frac{16}{kT}\left[\frac{3e^{8J\beta} + e^{-8J\beta} + 3}{(e^{8J\beta} + e^{-8J\beta} + 6)^2}\right]$$

1.2 Phase transition

A phase transition is a transformation of a system due to a variation of an external parameter (such as the temperature). It exists a critical point where the transition takes place.

1.2.1 The critical temperature

To study the phase transitions, we consider a critical temperature. The Ising model with our conditions (in two dimensions and with an null-external magnetic field $B = 0$) undergoes a phase transition of second-order, called like this because of the fact that there is a discontinuity in the second derivative of the free energy. That means that the new states grows continuously from the previous one and when $T \to T_C$, these two states are quantatively the same.

Near this temperature, a power-law applies to many physical quantities : one quantity varies as the power of another. In the Ising model, we can express the mean magnetization as

$$\langle M(T)\rangle \sim (T - T_C)^\beta$$

with $\beta = \frac{1}{8}$ one of the Ising critical exponents (in two dimensions).
We have similar expression for the heat capacity and the susceptibility :

$$C_V(T) \sim |T_C - T|^{-\alpha}$$

$$\chi(T) \sim |T_C - T|^{-\gamma}$$

with $\alpha = 0$ and $\gamma = \frac{7}{4}$ two other critical exponents.
We also defined the correlation length which is a quantity which represents how two spins are correlated. When the temperature T gets closer to T_C, the correlation length between two spins increases. This quantity is given by

$$\xi(T) \sim |T_C - T|^{-\nu},$$

defining the constant ν.

1.2.2 Around the critical temperature

When the correlation length spans the whole system, we say that we have a second-order (or continuous) phase transition. ξ is proportional to the size of the lattice L at the critical point ($\xi \propto L$), as it is always finite. With the finite size scaling relations, we can relate the temperature for a finite lattice and the one for an infinite lattice :

$$T_C(L) - T_C(L = \infty) = aL^{-\frac{1}{\nu}}$$

with a a constant.
With the last equation, we can write :

$$\langle M(T) \rangle \sim (T - T_C)^{\beta} \to L^{-\frac{\beta}{\nu}}$$

$$C_V(T) \sim |T_C - T|^{-\alpha} \to L^{\frac{\alpha}{\nu}}$$

$$\chi(T) \sim |T_C - T|^{-\gamma} \to L^{\frac{\gamma}{\nu}}$$

for T near T_C.

To find the critical temperature of our case, we will plot the values of the physical properties we study from the beginning (namely the expectation values for the energy E and the magnetic moment $|M|$, the specific heat C_V and the susceptibility ξ as function of the temperature T. The discontinuity in the graph would tell us an approximation of the critical temperature. We have a theorical value of the critical temperature which is

$$\frac{kT_C}{J} = \frac{2}{\ln(1 + \sqrt{2})} \approx 2.269 \tag{11}$$

according to the work of Lars Onsager exposed in the Physical Review 65, 116 in 1944.

1.3 The Monte Carlo method using the Metropolis algorithm

1.3.1 Introduction to the algorithm

The Metropolis algorithm is a Markov Chain Monte Carlo method (MCMC).
A Markov Chain is a Markov process, which is a stochastic process that satisfies the Markov property. This property says that to know the next state, we only need the current state. There is no need to remember the past ones. We obtain a sequence of random samples from a probability distribution, here the Boltzmann distribution.

Every Monte Carlo cycle is based on a Markov process to get a new random state. The point is to run it enough time to reach the most likely state of the system, starting with a random configuration. To reach this most likely state, and so an equilibrium distribution, we need either to accept or reject the new random state.

Our Monte Carlo sampling function will be the probability of finding the system in the state i given in (2) :

$$P_i = \frac{e^{-\beta E_i}}{Z}$$

The partition function Z is hard to compute as we need to have the energies of all states E_i : in two dimensions, we have 2^N configurations with $N = L \times L$ the number of spins. The Metropolis algorithm does not require us to compute this partition function. Indeed, we only need the difference between the energy of a configuration and the previous one.

1.3.2 The algorithm

Algorithm 1: Monte Carlo method using the Metropolis algorithm

```
Input: E_b the energy of the initial random configuration.
Input: E_t the energy of the configuration obtained by flipping one spin.
E_t − E_b → ΔE
E_t → E_b
init expectation values for the state with the energy E_b
for each new configuration do
    energy of the new configuration → E_t
    E_t − E_b → ΔE
    if ΔE ≤ 0 then
        accept the new configuration
        E_t → E_b
    else
        pick a random number r ∈ [0, 1]
        if r ≤ e^{−βΔE} then
            accept the new configuration
            E_t → E_b
        else
            keep the old configuration
        end
    end
    update expectation values
end
```

In this algorithm, we go through the lattice to calculate the energy of the actual configuration of the system. This is a Monte Carlo cycle. The choice of accepting or not a configuration is made by comparing the actual energy of the system with the energy of the previous one. If the energy of the actual configuration is lower than the previous one, we accept the configuration as we try to have the lowest energy at a given temperature.

The strength of this method is that we do not have to compute the total energy of each configuration but only the difference between the actual configuration and the previous one. Then we do not have to compute the exponential $e^{-\beta\Delta E}$ each time we go through the loop but we can pre-calculate its possible values. When we flip the spins only one by one we are

able to know all the ΔE which are possibles. For the Ising model in two dimensions, we have a total of five possible values of ΔE. We take a random spin in the lattice, surrounded by four other spins, which will be flipped. The direction of the four surronding spins determines the energy of the mini-system composed of the five spins. We only need to know how the energy is modified by the flip of the spin chosen to compute ΔE.
In the following array, we sum up all the different possible values of ΔE according to the number of spins-up around the chosen spin.

Number of surr. spins-up	Energy of the initial state	Flipped configurations	Energy of the resulting state	Difference of energy ΔE between the two states
$4:$ $\begin{matrix} & \uparrow & \\ \uparrow & \uparrow & \uparrow \\ & \uparrow & \end{matrix}$	$E_b = -4J$	$\begin{matrix} & \uparrow & \\ \uparrow & \downarrow & \uparrow \\ & \uparrow & \end{matrix}$	$E_t = 4J$	$\Delta E = 8J$
$3:$ $\begin{matrix} & \uparrow & \\ \downarrow & \uparrow & \uparrow \\ & \uparrow & \end{matrix}$	$E_b = -2J$	$\begin{matrix} & \uparrow & \\ \downarrow & \downarrow & \uparrow \\ & \uparrow & \end{matrix}$	$E_t = 2J$	$\Delta E = 4J$
$2:$ $\begin{matrix} & \downarrow & \\ \downarrow & \uparrow & \uparrow \\ & \uparrow & \end{matrix}$	$E_b = 0$	$\begin{matrix} & \downarrow & \\ \downarrow & \downarrow & \uparrow \\ & \uparrow & \end{matrix}$	$E_t = 0$	$\Delta E = 0$
$1:$ $\begin{matrix} & \downarrow & \\ \downarrow & \uparrow & \uparrow \\ & \downarrow & \end{matrix}$	$E_b = 2J$	$\begin{matrix} & \downarrow & \\ \downarrow & \downarrow & \uparrow \\ & \downarrow & \end{matrix}$	$E_t = -2J$	$\Delta E = -4J$
$0:$ $\begin{matrix} & \downarrow & \\ \downarrow & \uparrow & \downarrow \\ & \downarrow & \end{matrix}$	$E_b = 4J$	$\begin{matrix} & \downarrow & \\ \downarrow & \downarrow & \downarrow \\ & \downarrow & \end{matrix}$	$E_t = -4J$	$\Delta E = -8J$

Table 1.2: Possible values of ΔE for a two-dimensional Ising model

We will implement this algorithm in a little bit different way. We will not have to do the calculations of ΔE to save time and FLOPs.

Chapter 2

Implementation

For the programs of this project, we will use a object-oriented programing instead of standard programing with a lot of functions and variables. The program will then be closer to how we conceptualize the model and more programatic as well, which makes the development and the debugging easier because we rely on many small functions which are checked by unit tests. Most of these so-called small functions can be used by the user.

Those are quite simple to code but the goal is to make the algorithm fast because those functions will be called thousands times during a full Monte Carlo run. The first thing is to use inline methods, the second is to use static variables. However this can lead to strange behavior of the program, and we decided not to use this. The following table shows the performances of the function `lattice::change_random_spin(void)` if we add the `_emptyness_test()` of the lattice and how it can affect the performance.

`_emptyness_test()`	with	without	inline
Execution time (s)	0.767089	0.724986	0.727917
	0.820451	0.733801	0.724200
	0.818652	0.735491	0.731647
	0.764639	0.726894	0.740837
	0.803291	0.738665	0.788194
Total (s)	3.974122	3.659837	3.712795

Table 2.1: Comparison of time of execution with, without or inlining the function `_emptyness_text()` for a 20×20 lattice and 100000 iterations for the function `lattice::change_random_spin(void)`

.

with/without	7.9083%
inline/without	1.4470%
with/inline	6.5757%

Table 2.2: Ratios

We see that there is not much a difference by not using the function instead of writing it as an inline function.

2.1 Monte Carlo method using the Metropolis algorithm

2.1.1 A different way to implement the algorithm

We implement the Monte Carlo method in a different way than we did in the theory part (see Algorithm 1). This is way more efficient because we do not have to do useless calculations. We flip a random spin at each Monte-Carlo cycle[1], look for ΔE, and we accept the move by updating the calculations or reject it by not updating the calculations and re-flipping the spin.

Algorithm 2: Implementation of the Monte Carlo method using the Metropolis algorithm

Input: a random configuration
Input: the number of Monte Carlo cycle mc
init acc_conf the number of accepted configuration
for $n \leq mc$ **do**
  pick a random spin
  flip the chosen spin
  compute the sum of the neighbors spins of the chosen spin
  choose the corresponding ΔE ▷ see Table 1.2
  if $\Delta E \leq 0$ **then**
    accept the new configuration
    $E + \Delta E \to E$
    $M + 2 \times$ [the chosen spin] $\to M$
    acc_conf $+ 1 \to$ acc_conf
  else
    pick a random number $r \in [0, 1]$
    if $r \leq e^{-\beta \Delta E}$ **then**
      accept the new configuration
      $E + \Delta E \to E$
      $M + 2 \times$ [the chosen spin] $\to M$
      acc_conf $+ 1 \to$ acc_conf
    else
      reject the new configuration
      flip again the chosen spin
    end
  end
  $n + 1 \to n$
end

[1] We define a Monte-Carlo cycle as going through the lattice once and flipping only one spin. That is, for n Monte-Carlo cycles we flip n spins and accept m configurations with $m \leq n$.

There are two `montecarlo` methods in the class, one for a given temperature and one that increments automatically the temperature with the input temp-step. The precalculations of $e^{-\beta\Delta E}$ are doing after each Monte-Carlo run through a temperature because they change with the temperature. Those precalculations are member of the lattice and are initialized when we build the object.

Note that the random number generators are those from the cpp library proposed in this course and that they have a period over 10^9, which is our maximum number of Monte-Carlo cycles. It is also used with the high resolution clock from the standard C++ library.

2.1.2 Use of MPI and outputs

We use MPI to run two simulations simultaneously because the more data we have, the more reliable our results are. MPI allows to use more than one processor core. The data from the to distinct simulations can be merged in one set of data. Thus, when we output expected values, no matter if they are functions of the temperature or of the Monte Carlo cycles, that are the averages of the two simulations running simultaneously. This explains why we can have not-integer number of accepted configurations.

The program is then able to output $\langle E\rangle$, $\langle|M|\rangle$, C_V, σ_E^2 (the variance of E : $C_V = \dfrac{\sigma_E}{kT^2}$), χ and the number of accepted configurations at each Monte-Carlo cycle or at each temperature, depending on which `montecarlo` method we choose.

Chapter 3

Results

3.1 Two-dimensional squared lattice

3.1.1 2×2 case

This very simple case allows us to test the algorithm. We just have to compare the numerical results with the theoritical ones.

Using the equations (5), (6), (8) and (10), we compute respectively the mean energy, the mean absolute magnetization, the specific heat and the susceptibility. We take $T = 1.0$. Then,

$$\langle E \rangle = -7.98393 \tag{12}$$
$$\langle |M| \rangle = 3.99464 \tag{13}$$
$$C_V = 0.128315 \tag{14}$$
$$\chi = 0.0160394 \tag{15}$$

Let's see what we get with the simulations. We run the program four times.

The results starting with a random lattice are not exploitables as each simulation gives a different value. This is because of the size of the lattice, too small. When we begin with an ordered lattive however, the four simulations converges more or less to the same value which is nearly -7.983. This is totally coherent with our theorical value.
For the following plots, we will only take an ordered lattice to begin.

We see here that this is coherent with the theorical value. The four simulations give pretty much the same value which is $\langle |M| \rangle = 3.995$ and is very close to the theorical one.

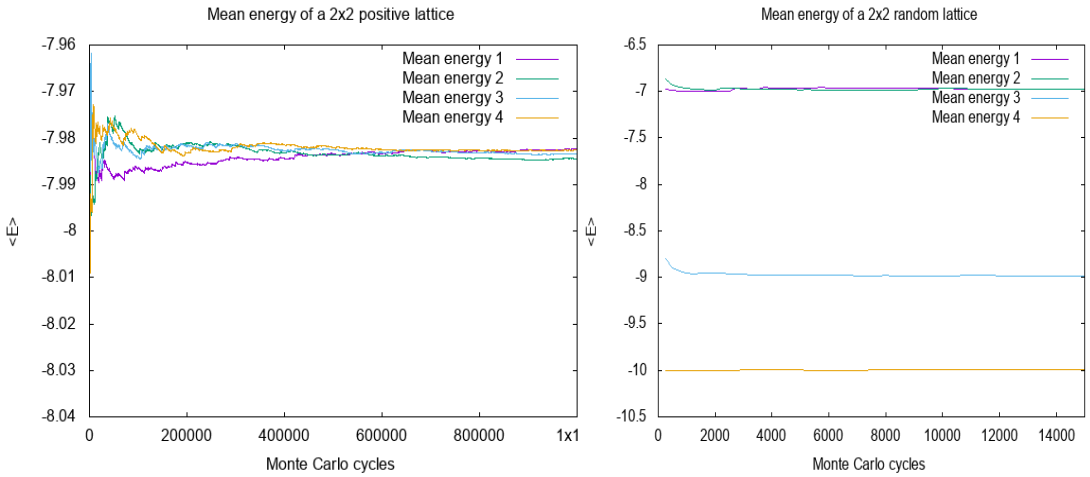

Figure 3.1: The mean energy as function of the number of Monte Carlo cycles, starting with an ordered lattice (all spins-up) (left) or a random lattice (right) at $T = 1$

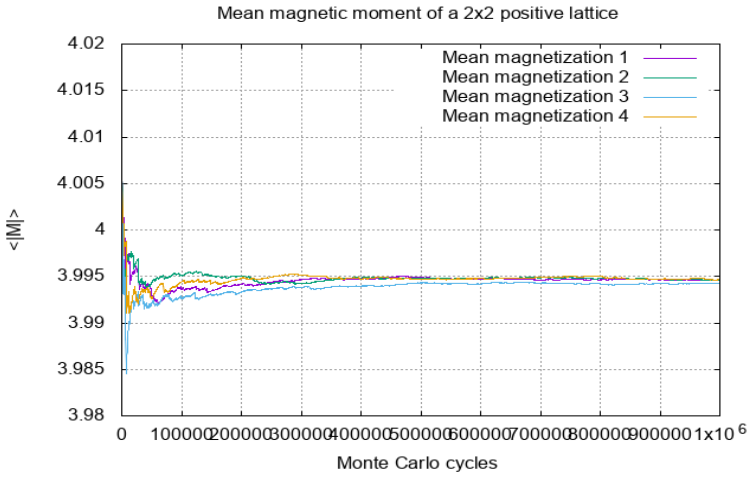

Figure 3.2: The mean magnetic moment as function of the number of Monte Carlo cycles, starting with all spins-up over four simulations at $T = 1$

We have a result here which is not at all what we expected. There is no point after which the specific heat stabilizes.

Except for the mean energy and the mean magnetic moment, the results we have for the case of a 2×2 lattice at $T = 1$ are obviously outliers. This may be because of the low temperature or because of the size of the lattice. We should take at least 200000 Monte Carlo cycles to reach an equilibrium situation.

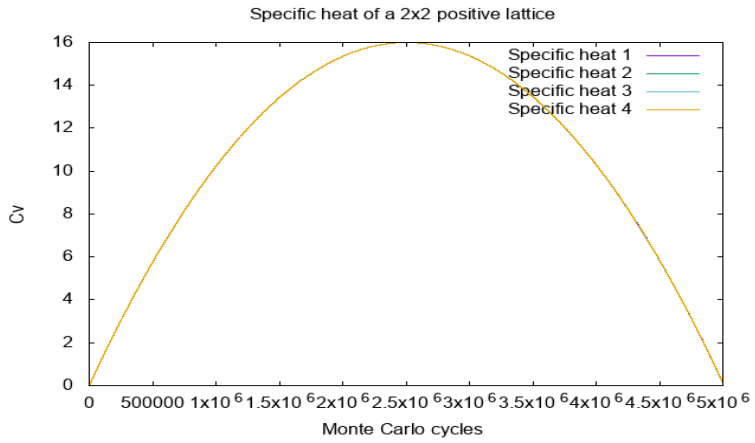

Figure 3.3: The specific heat as function of the number of Monte Carlo cycles at $T = 1$

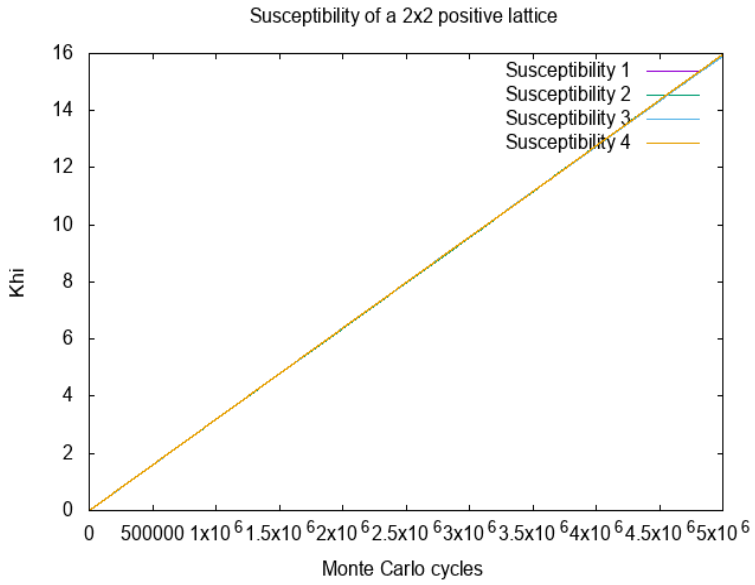

Figure 3.4: The susceptibility as function of the number of Monte Carlo cycles at $T = 1$

These results are not what I expected. The mean magnetic moment should fall at a given temperature (which is the critical temperature), the mean energy should not fall like this with the temperature.

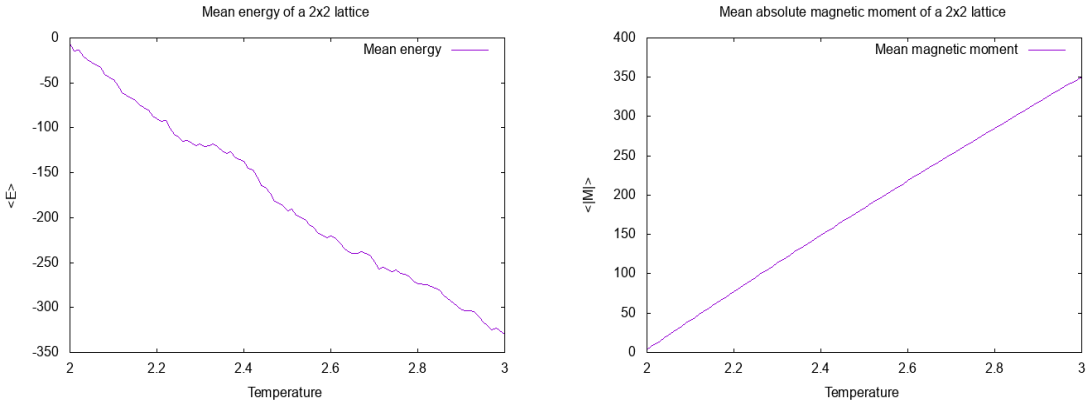

Figure 3.5: The mean energy and the mean magnetic moment as function of the temperature for 2×10^6 Monte Carlo cycles

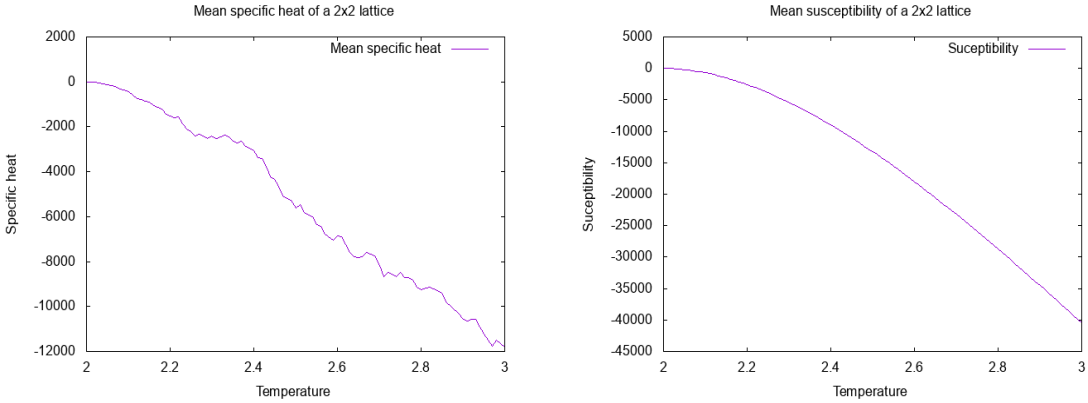

Figure 3.6: The specific heat and the susceptibility as function of the temperature for 2×10^6 Monte Carlo cycles

3.1.2 20×20 case

Reach of the most likely state

In this section we will study the time, which is here defined as the number of Monte Carlo cycles. At two different temperatures, we will run the program in order to know after how many cycles we reach an equilibrium situation. This equilibrium situation is the point at which we can begin to compute the expectation values. We will run a simulation at the temperature $T = 1$ and $T = 2.4$ starting with either a random lattice or an ordered one (taking all spins-up).

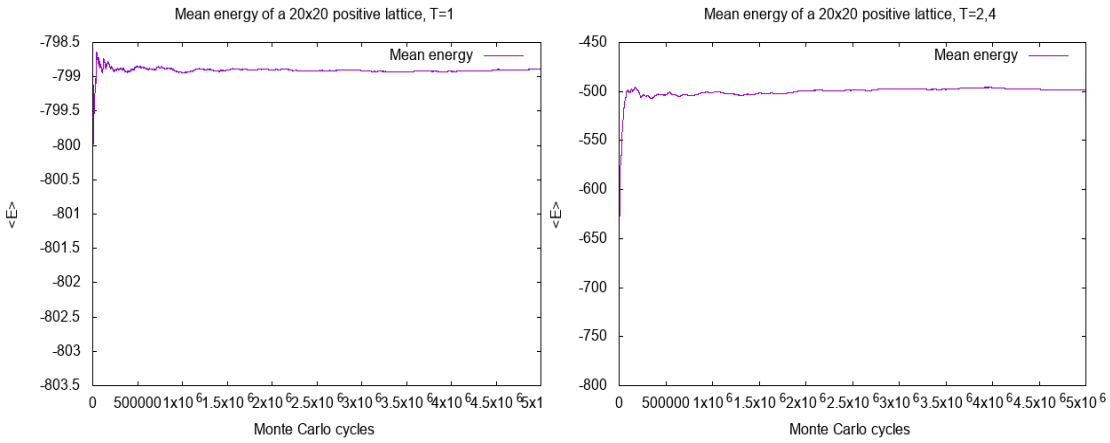

Figure 3.7: The expectation value of the energy as function of the number of Monte Carlo cycles, in the case of a starting ordered lattice (all spins-up), at $T = 1$ (left) and $T = 2.4$ (right)

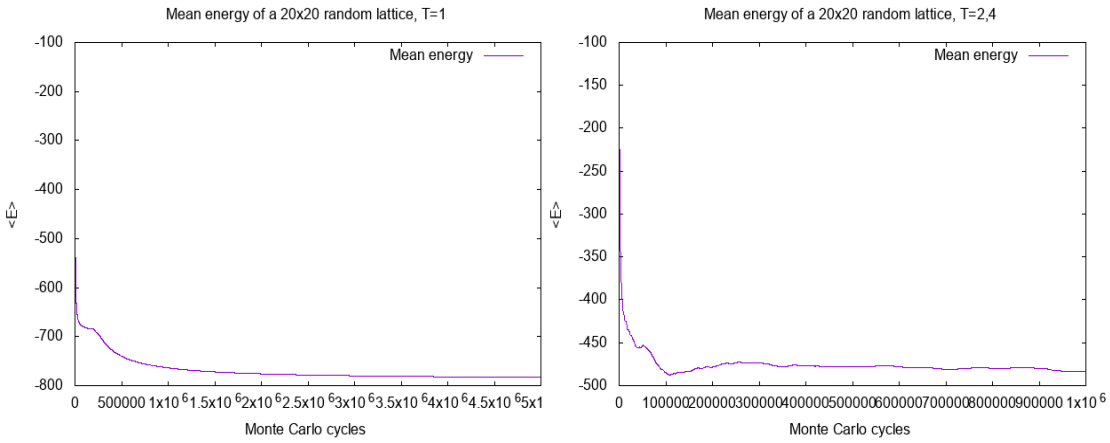

Figure 3.8: The expectation value of the energy as function of the number of Monte Carlo cycles, in the case of a starting random lattice, at $T = 1$ (left) and $T = 2.4$ (right)

We observe that the mean energy converges quickly to the most likely state. Approximatively, we need 500000 Monte Carlo cycles to reach an equilibrium state in the case of a low temperature ($T = 1$) and pretty much the same number of cycles needed for the temperature $T = 2.4$. We also observe that the mean energy is much lower for a low temperature : $\langle E \rangle_1 = 799$ against $\langle E \rangle_{2.4} = 500$ which is coherent with the fact that the thermal motion increases with the temperature.
Moreover, starting with an ordered lattice gives clearer results than a starting random lattice where it takes nearly 5×10^6 Monte Carlo cycles to be really stable.

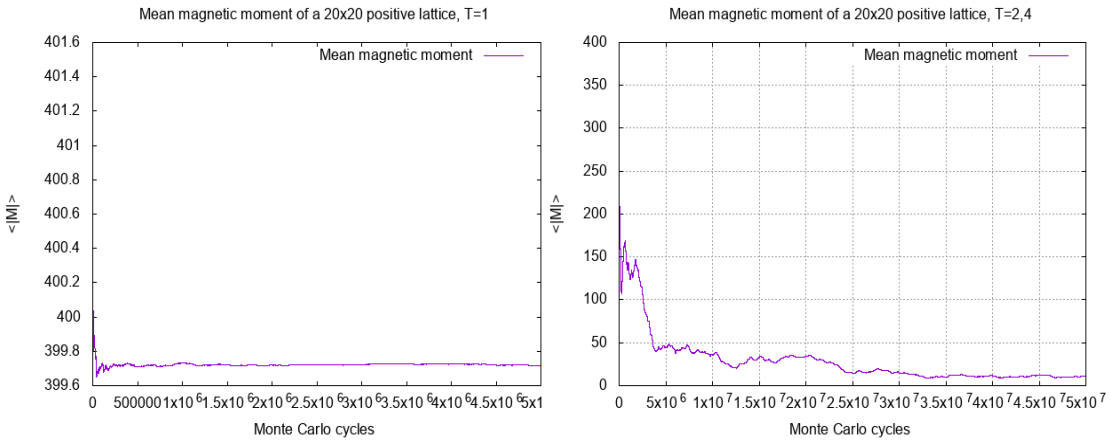

Figure 3.9: The expectation value of the magnetic moment as function of the number of Monte Carlo cycles, in the case of a starting ordered lattice (all spins-up), at $T = 1$ (left) and $T = 2.4$ (right)

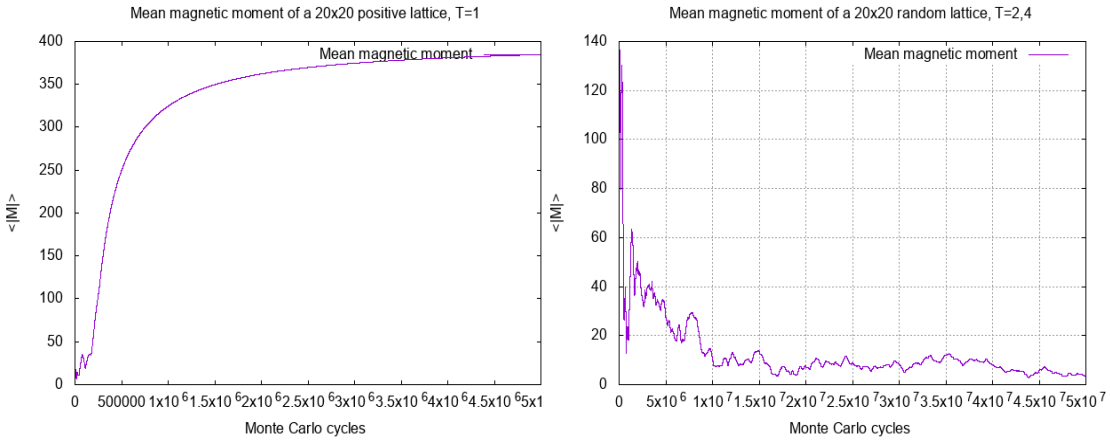

Figure 3.10: The expectation value of the magnetic moment as function of the number of Monte Carlo cycles, in the case of a starting random lattice, at $T = 1$ (left) and $T = 2.4$ (right)

The expectation value of the magnetic moment is quite stable after 250000 Monte Carlo cycles for a starting ordered lattice at $T = 1$. We can compare the results with C. N. Yang's exact result published in the Physical Review 85, 808-816 in 1952, "The spontaneous magnetization of a two-dimensional Ising model". The magnetic moment per spin is expressed as

$$m = \begin{cases} \left(1 - [\sinh(2\beta J)]^{-4}\right)^{\frac{1}{8}} & (T < T_C) \\ 0 & (T > T_C) \end{cases}$$

So, for a 20×20 lattice, that is 400 spins, and $T = 1 < T_C$ we should have

$$M_{400} = 400 \times \left(1 - [\sinh 2]^{-4}\right)^{\frac{1}{8}}$$
$$M_{400} = 399.710$$

For $T = 1$ the result we get are coherent with the value we should have. At $T = 2.4$ the mean magnetization tends toward zero which means that $T = 2.4 > T_C$.
For a random configuration as a starting configuration, the number of Monte Carlo cycles needed to be more or less stable is 1×10^7. So in order to start computations of the expectation values when an equilibrium situation has been reach, we have to wait 1×10^7 Monte Carlo cycles.

Let's see how the number of accepted configurations evolves as function of the number of Monte Carlo cycles.

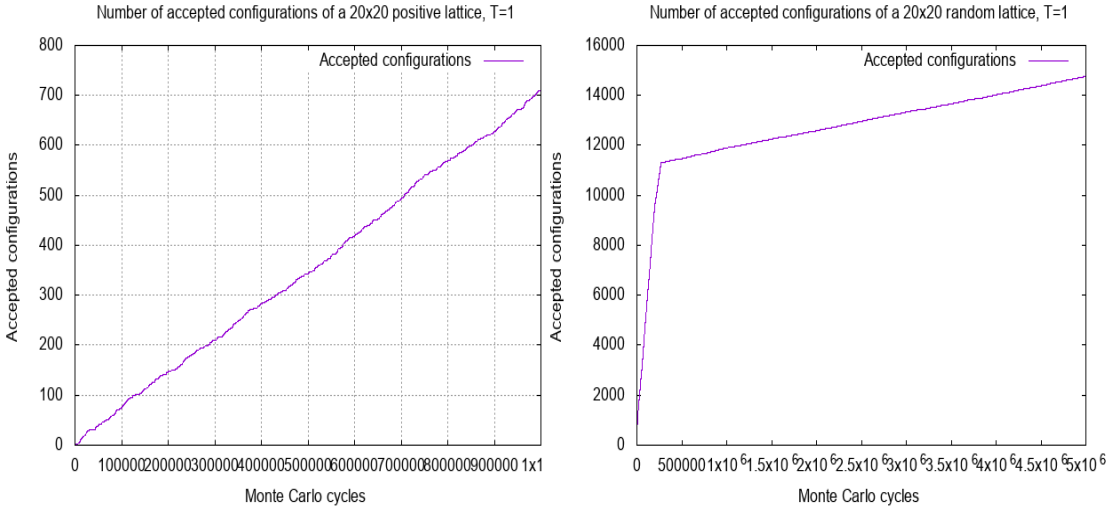

Figure 3.11: The number of accepted configurations as function of the number of Monte Carlo cycle, starting with a ordered lattice (left) or with a random one (right) at $T = 1$

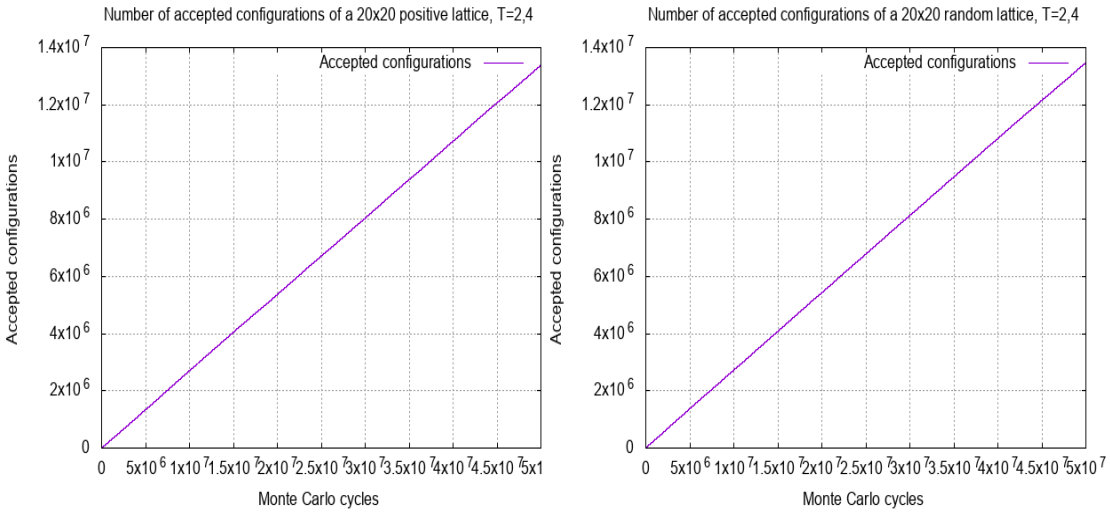

Figure 3.12: The number of accepted configurations as function of the number of Monte Carlo cycle, starting with a ordered lattice (left) or with a random one (right) at $T = 2.4$

The number of accepted configurations increase lineraly with the number of Monte Carlo cycles except for a starting random configuration at the temperature $T = 1$. It seems coherent as the probability of having the needed conditions to the acceptation increases with the number of Monte Carlo cycles.

The probability distribution of a given energy

In this section, we will looking for how many times a given energy appears in our computations. With these data, we would be able to analyze the probability distribution.

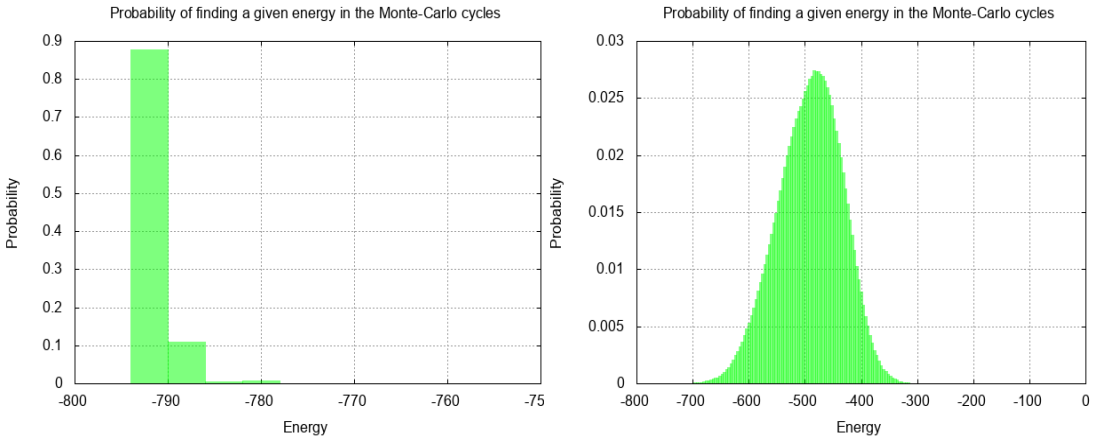

Figure 3.13: The probability distribution $P(E)$

At low temperature, we have very few possible values for the energy, the curve is centered near the expectation value of the energy. At temperature $T = 2.4$, the Gaussian distribution is way more extended. We have more possible values. The curve is centered at the expectation value we have determined above.
The variance in energy for $T = 1$ are really big whereas we expected low variances because the curve is not extended. The variance at $T = 2.4$ are smaller but still very big.

3.2 Phase transitions

3.2.1 The Ising model near the critical temperature

We will focus on a smaller interval around a supposed critical temperature : $T \in [2.0, 2.3]$. We take different size for the lattice : $L = 40$, $L = 60$, $L = 80$ and $L = 100$ (do not pay attention to the titre above the plot, the size of the lattice is wrong).

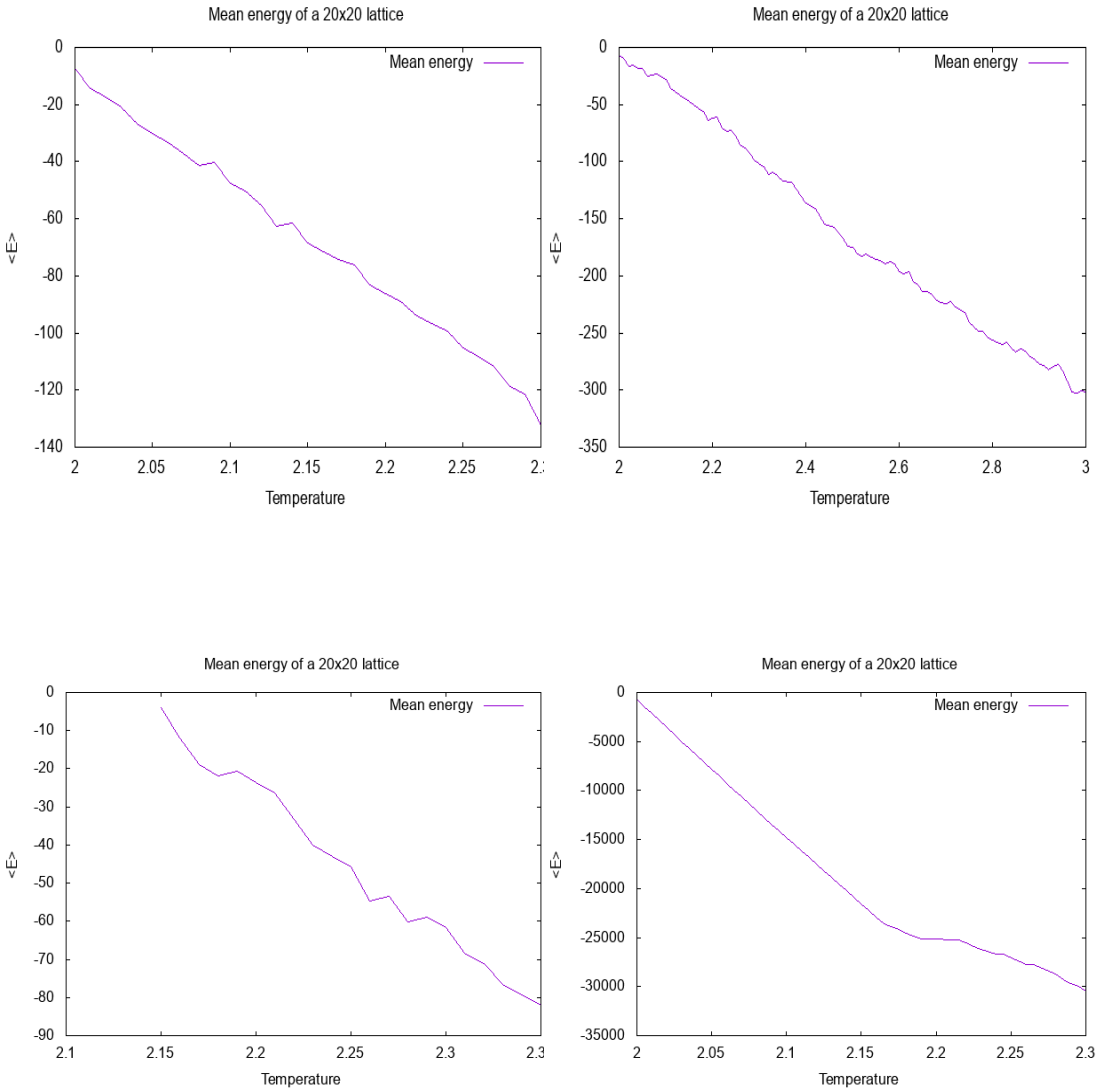

Figure 3.14: The mean energy as function of the temperature near the critical temperature for $L = 40, 60, 80, 100$

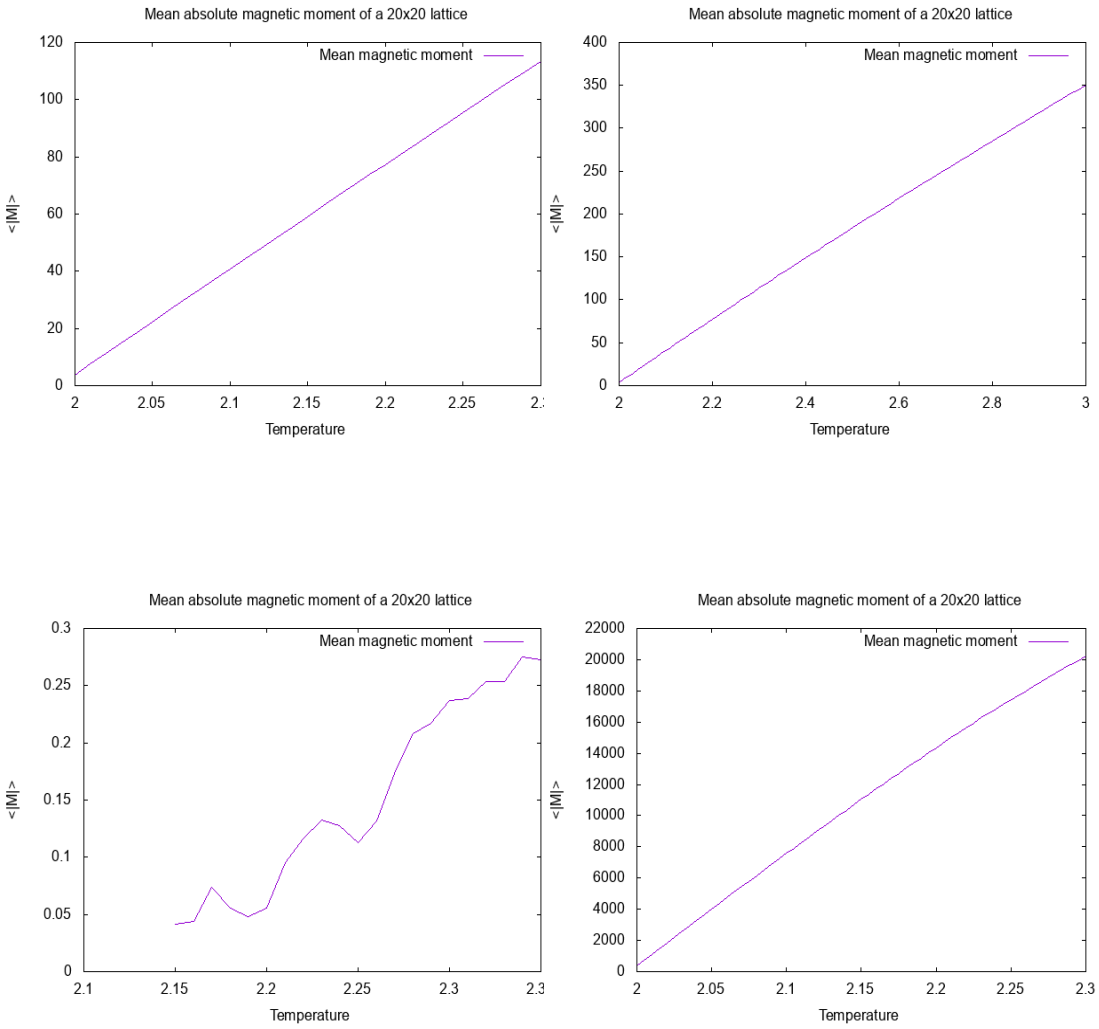

Figure 3.15: The mean magnetic moment as function of the temperature near the critical temperature for $L = 40, 60, 80, 100$

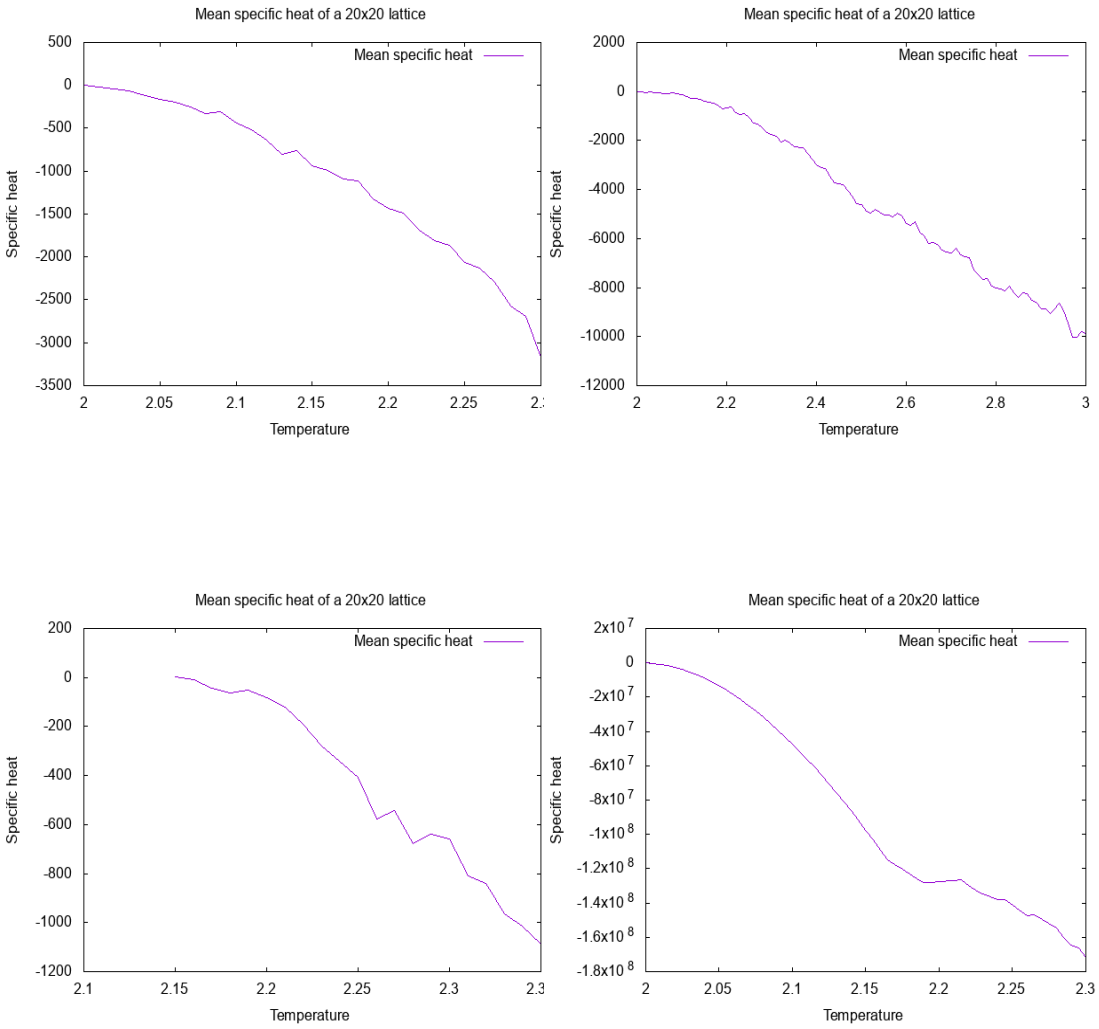

Figure 3.16: The specific heat as function of the temperature near the critical temperature for $L = 40, 60, 80, 100$

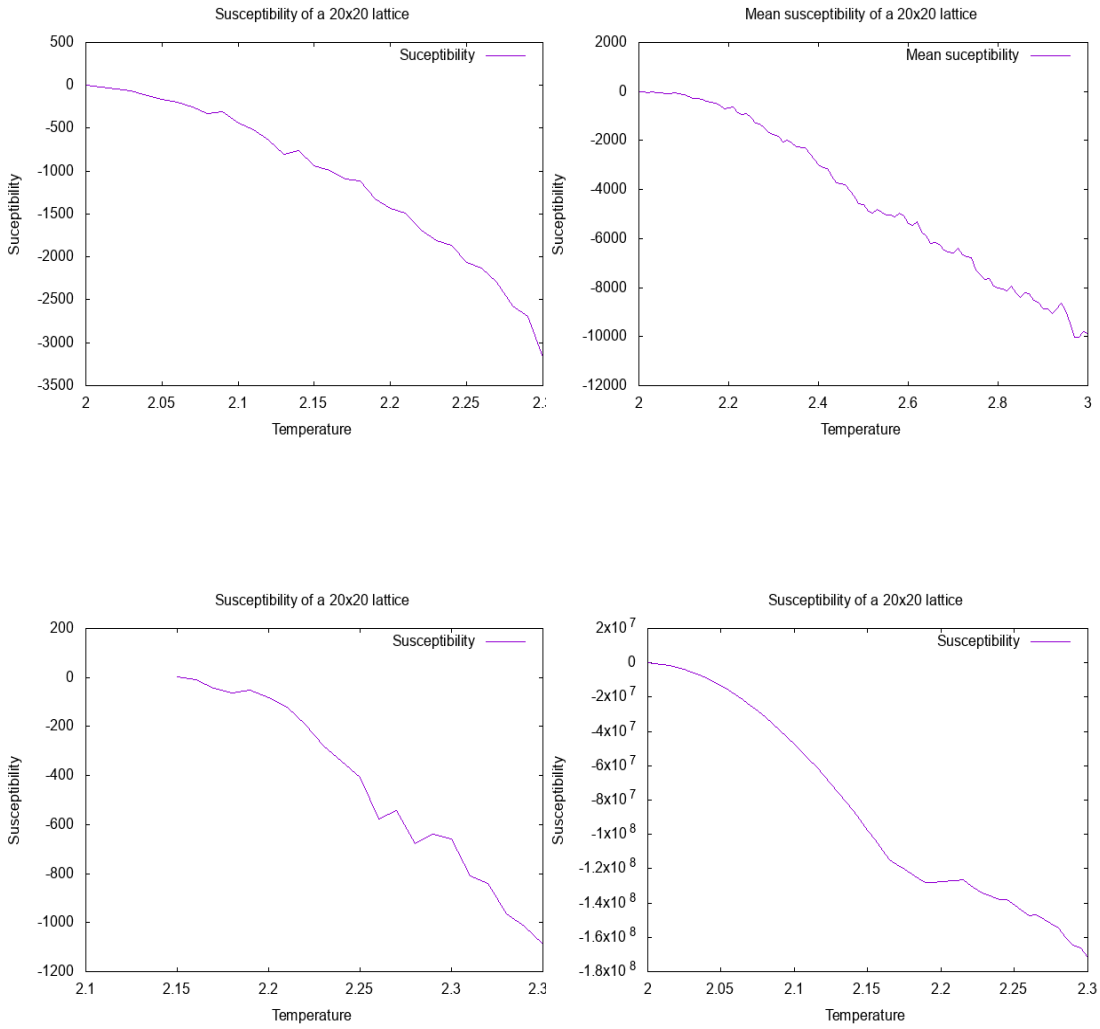

Figure 3.17: The susceptibility as function of the temperature near the critical temperature for $L = 40, 60, 80, 100$

We, again, do not have what we expected. We should see the phase transition on these plots. For example, the susceptibility should diverge at T_C. We are not able to find the critical temperature with these simulations.

Going from a debug mode to a release mode allows to go from a over 7mn computation time to an average 1mn30 computation time. That's at least one good result.

3.2.2 The critical temperature

As we do not have a value of $T_C(L)$, we are not able to use the equation. I will trust Lars Onsager and his value which is $\frac{kT_C}{J} \approx 2.269$.

Conclusion

Using the object oriented programing, we were able to have a clear code. Unfortunately, even with the clearest code, the results obtained are absolutely not what we expected. The program was however very efficient, and secured by several unit tests. We also have some proof that it may work, maybe with some corrections, as we had some good results.

Bibliography

- Lars Onsager (1944), Phys. Rev. 65, 117
- C. N. Yang (1952), *"The spontaneous magnetization of a two-dimensional Ising model"* Phys. Rev.85, 808–816