

---

---

# Studying second order phase transitions in the Ising model in two dimensions with the metropolis algorithm

---

PROJECT 4, FYS-3150

INA K. B. KULLMANN

---

## **Abstract**

The aim of this project is to numerically find the critical temperature for the two dimensional Ising model by using the metropolis algorithm. We will first test the implementation of the algorithm carefully, first by comparing with theoretical values calculated for a small system. Then we will see if the algorithm behaves as expected according to our physical intuition for a larger system.

When we have found a estimate for the critical temperature we will compare it to Lars Onsagers analytical result.

All source codes can be found at: [https://github.com/inakbk/Project\\_4](https://github.com/inakbk/Project_4).

---

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Theory</b>	<b>3</b>
2.1	The Ising model in two dimensions . . . . .	3
2.2	Obtaining analytical values for the simple $L = 2$ system . . . . .	3
2.3	Units and scaled parameters . . . . .	7
2.4	Second order phase transitions . . . . .	7
<b>3</b>	<b>Numerical methods</b>	<b>8</b>
3.1	Metropolis algorithm . . . . .	8
3.2	Coding $\Delta E$ , $\Delta \mathcal{M}$ and $w$ efficiently . . . . .	10
<b>4</b>	<b>"Experiment"</b>	<b>10</b>
4.1	Testing the implementation of the model and algorithm . . . . .	10
4.2	Equilibrium time . . . . .	13
4.3	Probability . . . . .	18
4.4	Calculations to find $T_c$ . . . . .	19
4.5	Estimating the critical temperature . . . . .	22
<b>5</b>	<b>Discussion and experiences</b>	<b>22</b>

# 1 Introduction

A second order phase transition at the critical temperature  $T_c$  is described by a macroscopic change to a system. At the critical temperature some thermodynamic values approaches zero with an infinite slope and 'disappears'. Other variables are discontinuous or diverge at the critical temperature in the thermodynamic limit.

In this paper we will study such a phase transition by using the Ising model and the metropolis algorithm to simulate results. The approach used in this paper is also applicable for other physical models and therefore very useful.

## 2 Theory

### 2.1 The Ising model in two dimensions

The Ising model is a simple model for ferromagnetism in statistical physics. The model consists of magnetic spins that are allowed to interact with its neighbours. The magnetic dipole moments are allowed to have two values, 1 and -1.

In this project we will study the ising model in two dimensions which allows the identification of a second order phase transition. We have chosen to set the external magnetic field to zero throughout this paper.

The Ising model gives that the energy for the whole system can be expressed as

$$E = -J \sum_{\langle kl \rangle}^N s_k s_l \quad (1)$$

where the value of the spins are  $s_k = \pm 1$  and  $N$  is the total number of spins. The variable  $J$  is a coupling constant expressing the strength of the interaction between neighboring spins. We will divide all variables by the number of spins to obtain all results per spin.

The symbol  $\langle kl \rangle$  indicates that we sum over nearest neighbors only. We will assume that we have a ferromagnetic ordering, viz  $J > 0$  and use periodic boundary conditions. We choose to use a square lattice of  $L$  spins in x and y direction. The total number of spins is then  $N = L^2$  and the total number of microstates, the different configurations the spins can have is given by  $M = 2^N$

The partition function for such a system is then given by:

$$Z = \sum_{i=1}^M e^{-\beta E_i}$$

### 2.2 Obtaining analytical values for the simple $L = 2$ system

First we assume that there is only two spins in each dimension (x and y), we set  $L = 2 \Rightarrow N = 2 \cdot 2 = 4 \Rightarrow M = 2^4 = 16$ . So there is 16 possible energies of the spin system when  $L = 2$ . So to calculate the partition function we have to find the energies:

$$Z = e^{-\beta E_1} + \dots e^{-\beta E_{16}}$$

The spin system can be visualized as 16 matrices on the form:  $\begin{bmatrix} s(0,0) & s(0,1) \\ s(1,0) & s(1,1) \end{bmatrix}$  with corresponding energies.

Table 1: All possible spin combinations and energies for the  $L = 2$  system.

$E_1 = 0$	$E_2 = 0$	$E_3 = 0$	$E_4 = 0$
$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$
$E_5 = 0$	$E_6 = 0$	$E_7 = 0$	$E_8 = 0$
$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$
$E_9 = 0$	$E_{10} = 0$	$E_{11} = 0$	$E_{12} = 0$
$\begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}$
$E_{13} = 8J$	$E_{14} = 8J$	$E_{15} = -8J$	$E_{16} = -8J$
$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

Periodic boundary conditions give that every spin has a neighbour. The neighbours of the spin  $s(1,1)$  is  $s(0,1)$  twice (above and below) and  $s(1,0)$  twice (above and below). To find the energy we have to sum over the nearest neighbours for all the spins in the system. The product  $s_l s_k$  of two neighbours should only be calculated once. This is solved if the index of  $s_l$  is fixed while  $s_k$  has one higher index in each direction at a time:

$$E = -J \left( s(0,0) \cdot [s(1,0) + s(0,1)] + s(0,1) \cdot [s(1,1) + s(0,0)] \right. \\ \left. + s(1,0) \cdot [s(0,0) + s(1,1)] + s(1,1) \cdot [s(0,1) + s(1,0)] \right) \quad (2)$$

The 16 states with corresponding energies calculated with equation 2 is given in table 1.

Looking at table 1 we see that there is only three possible values for the energies,  $E \in \{-8J, 0, 8J\}$  with corresponding multiplicity  $\{2, 12, 2\}$ . The partition function can then be calculated:

$$Z = 2e^{-\beta(-8J)} + 12e^{-\beta \cdot 0} + 2e^{-\beta \cdot 8J} = 2(e^{\beta \cdot 8J} + e^{-\beta \cdot 8J}) + 12 \\ = 4 \cosh(8J\beta) + 12$$

using that  $\cosh(x) = \frac{1}{2}(e^{-x} + e^x)$ .

Now that we have the partition function various expectation variables can be calculated. We start with the expectation value for the energy:

$$\langle E \rangle = \frac{1}{Z} \sum_{i=1}^M E_i e^{-\beta E_i} = \frac{1}{Z} [2 \cdot (-8J)e^{8J\beta} + 0 + 2 \cdot 8J e^{-8J\beta}] = \frac{1}{Z} [-16J e^{8J\beta} + 16J e^{-8J\beta}] \\ = -\frac{16J}{Z} [-e^{-8J\beta} + e^{8J\beta}] = -\frac{32J}{Z} \sinh(8J\beta) = -\frac{8J \sinh(8J\beta)}{\cosh(8J\beta) + 3}$$

using that  $\sinh(x) = \frac{1}{2}(-e^{-x} + e^x)$ . We will now calculate the heat capacity

$$C_v = \frac{1}{k_b T} \sigma_E^2 = \frac{1}{k_b T} (\langle E^2 \rangle - \langle E \rangle^2)$$

Table 2: Magnetization with the corresponding energy and multiplicity for the  $L = 2$  system.

# spins up	Multiplicity	Energy	Magnetization
4	1	-8J	4
3	4	0	2
2	4	0	0
2	2	8J	0
1	4	0	-2
0	1	-8J	-4

where

$$\begin{aligned}
\langle E^2 \rangle &= \frac{1}{Z} \sum_{i=1}^M E_i^2 e^{-\beta E_i} = \frac{1}{Z} [2 \cdot (-8J)^2 e^{8J\beta} + 0 + 2 \cdot (8J)^2 e^{-8J\beta}] = \frac{128J^2}{Z} [e^{8J\beta} + e^{-8J\beta}] \\
&= \frac{128J^2 \cdot 2 \cosh(8J\beta)}{4 \cosh(8J\beta) + 12} = \frac{64J^2 \cosh(8J\beta)}{\cosh(8J\beta) + 3}
\end{aligned}$$

so that

$$\begin{aligned}
C_v &= \frac{1}{k_b T} \left[ \frac{64J^2 \cosh(8J\beta)}{\cosh(8J\beta) + 3} - \left( -\frac{8J \sinh(8J\beta)}{\cosh(8J\beta) + 3} \right)^2 \right] = \frac{1}{k_b T} \left[ \frac{64J^2 \cosh(8J\beta)}{\cosh(8J\beta) + 3} - \frac{64J^2 \sinh^2(8J\beta)}{(\cosh(8J\beta) + 3)^2} \right] \\
&= \frac{64J^2}{k_b T} \left[ \frac{\cosh(8J\beta)(\cosh(8J\beta) + 3) - \sinh^2(8J\beta)}{(\cosh(8J\beta) + 3)^2} \right] = \frac{64J^2}{k_b T} \left[ \frac{\cosh^2(8J\beta) + 3 \cosh^2(8J\beta) - \sinh^2(8J\beta)}{(\cosh(8J\beta) + 3)^2} \right] \\
&= \frac{64J^2 \beta}{T} \left[ \frac{1 + 3 \cosh(8J\beta)}{(\cosh(8J\beta) + 3)^2} \right]
\end{aligned}$$

We define the magnetization  $\mathcal{M}$  as the sum of all the spins  $s$ . It is easy to see that for the  $2 \times 2$  system there are only 5 possible values for the magnetization,  $\mathcal{M} \in \{-4, -2, 0, 2, 4\}$ . The corresponding energy and multiplicity is given in table 2.

The expectation value of the magnetization and the magnetization squared can be calculated as

$$\begin{aligned}
\langle \mathcal{M} \rangle &= \frac{1}{Z} \sum_{i=1}^M \mathcal{M} e^{-\beta E_i} \\
\langle \mathcal{M}^2 \rangle &= \frac{1}{Z} \sum_{i=1}^M \mathcal{M}^2 e^{-\beta E_i}
\end{aligned} \tag{3}$$

which we can use to calculate the susceptibility, the parameter of special interest:

$$\chi = \frac{1}{k_b T} \sigma_{\mathcal{M}} = \frac{1}{k_b T} (\langle \mathcal{M}^2 \rangle - \langle \mathcal{M} \rangle^2).$$

When looking at equation 3 and table 2 we quickly see that the expectation value of the magnetization is zero for all temperatures. The microstates with opposite magnetization have the same multiplicity so they cancel each other out. When running numerical calculations on large systems, for large  $L$ , the numerical value of  $\langle \mathcal{M} \rangle$  will not reach zero unless the simulation is run for a (extremely) long time. This is because it takes a long time for the simulation to

Table 3: Absolute magnetization with the corresponding energy and multiplicity for the  $L = 2$  system.

Absolute Magnetization	Energy	Multiplicity
4	-8J	2
2	0	8
0	0	4
0	8J	2

go through all the possible microstates after the simulation have reached a very probable state. But we want to have a reasonable measure of the magnetization and suceptibility for large systems and at the same time minimize the excecution time of the numerical simulation. To do this we choose to use the absolute value of the magnetization in the definition of the suceptibility.

There are 3 possible values for the absolute value of the magnetization,  $|\mathcal{M}| \in \{0, 2, 4\}$ . The corresponding energy and multiplicity is given in table 3.

We are then ready to calculate the expectation value of the absolute value of the magnetization by using the multiplicity for each energy and corresponding magnetization:

$$\begin{aligned}
\langle |\mathcal{M}| \rangle &= \frac{1}{Z} \sum_{i=1}^M |\mathcal{M}| e^{-\beta E_i} = \frac{1}{Z} \left( 2 \cdot 4e^{-\beta(-8J)} + 8 \cdot 2e^0 + 0 \cdot 4e^0 + 0 \cdot 2e^{-\beta 8J} \right) \\
&= \frac{8}{Z} (e^{8J\beta} + 2) = \frac{8(e^{8J\beta} + 2)}{4 \cosh(8j\beta) + 12} = \frac{2(e^{8J\beta} + 2)}{\cosh(8j\beta) + 3}.
\end{aligned} \tag{4}$$

We also obtain the expectation value of the square of the magnetization:

$$\begin{aligned}
\langle \mathcal{M}^2 \rangle &= \frac{1}{Z} \sum_{i=1}^M \mathcal{M}^2 e^{-\beta E_i} = \frac{1}{Z} (2 \cdot 4^2 e^{-\beta(-8J)} + 8 \cdot 2^2 e^0 + 0 + 0) = \frac{32}{Z} (e^{8J\beta} + 1) \\
&= \frac{8(e^{8J\beta} + 1)}{\cosh(8J\beta) + 3}.
\end{aligned}$$

And finally the suceptibility is given by

$$\begin{aligned}
\chi &= \frac{1}{k_b T} (\langle \mathcal{M}^2 \rangle - \langle |\mathcal{M}| \rangle^2) = \frac{1}{k_b T} \left[ \frac{8(e^{8J\beta} + 1)}{\cosh(8J\beta) + 3} - \left( \frac{2(e^{8J\beta} + 2)}{\cosh(8j\beta) + 3} \right)^2 \right] \\
&= \frac{1}{k_b T} \left[ \frac{8(e^{8J\beta} + 1)(\cosh(8j\beta) + 3) - 4(e^{8J\beta} + 2)^2}{(\cosh(8j\beta) + 3)^2} \right] \\
&= 4\beta \left[ \frac{2(e^{8J\beta} + 1)(\cosh(8j\beta) + 3) - (e^{8J\beta} + 2)^2}{(\cosh(8j\beta) + 3)^2} \right]
\end{aligned}$$

### 2.3 Units and scaled parameters

We will now write the equations for the expectation values, the heat capacity and the susceptibility in terms of a scaled temperature

$$\begin{aligned} T' &= T \frac{k_b}{J} \\ \Rightarrow T &= T' \frac{J}{k_b} \\ \Rightarrow \beta &= \frac{1}{k_b T} = \frac{1}{k_b \cdot T' \frac{J}{k_b}} = \frac{1}{T' J} \end{aligned}$$

We then set  $J = 1$  so that:

$$\beta = \frac{1}{T'}$$

We will use this new temperature  $T'$  in the numerical calculations. We also scale the heat capacity as:

$$C'_v = \frac{C_v}{k_b} = \frac{64}{T'^2} \frac{1 + 3 \cosh(8/T')}{(\cosh(8/T') + 3)^2}$$

In terms of  $T'$  the expectation values and the susceptibility for the  $2 \times 2$  system can be written as:

$$\langle E \rangle = -\frac{8 \sinh(8/T')}{\cosh(8/T') + 3} \quad (5)$$

$$\langle E^2 \rangle = \frac{64 J^2 \cosh(8/T')}{\cosh(8/T') + 3} \quad (6)$$

$$\langle |\mathcal{M}| \rangle = \frac{2(e^{8/T'} + 2)}{\cosh(8/T') + 3} \quad (7)$$

$$\langle \mathcal{M}^2 \rangle = \frac{8(e^{8/T'} + 1)}{\cosh(8/T') + 3} \quad (8)$$

$$\chi = \frac{4}{T'} \frac{2(e^{8/T'} + 1)(\cosh(8/T') + 3) - (e^{8/T'} + 2)^2}{(\cosh(8/T') + 3)^2} \quad (9)$$

These theoretical values will be used to test the implementation of the numerical method for the  $L = 2$  system.

### 2.4 Second order phase transistions

Near  $T_C$ , where the phase transition happens, we can characterize the behavior of many physical quantities by a power law behavior. As an example the mean magnetization is given by

$$\langle \mathcal{M}(T) \rangle \sim (T - T_C)^\beta, \quad (10)$$

where  $\beta = 1/8$  is a so-called critical exponent. A similar relation applies to the heat capacity

$$C_V(T) \sim |T_C - T|^\alpha, \quad (11)$$

and the susceptibility

$$\chi(T) \sim |T_C - T|^\gamma, \quad (12)$$

with  $\alpha = 0$  and  $\gamma = 7/4$ . Another important quantity is the correlation length, which is expected to be of the order of the lattice spacing for  $T \gg T_C$ . Because the spins become more and more correlated as  $T$  approaches  $T_C$ , the correlation length increases as we get closer to the critical temperature. The divergent behavior of  $\xi$  near  $T_C$  is

$$\xi(T) \sim |T_C - T|^{-\nu}. \quad (13)$$

A second-order phase transition is characterized by a correlation length which spans the whole system. Since we are always limited to a finite lattice,  $\xi$  will be proportional with the size of the lattice. Through so-called finite size scaling relations it is possible to relate the behavior at finite lattices with the results for an infinitely large lattice. The critical temperature scales then as

$$T_C(L) - T_C(L = \infty) = aL^{-1/\nu}, \quad (14)$$

with  $a$  a constant and  $\nu = 1$  defined in Eq. (13). This is the equation we will use to estimate the critical temperature with the numerical results and compare with Lars Onsagers theoretical results  $T_c \approx 2.269$ .

Let us first rewrite the equation 14 and using  $T_C(L = \infty)x$  for the variable that we wish to estimate and using two different values of  $L$ . We then combine

$$T_C(L) - x = a/L$$

with

$$T_C(L^*) - x = a/L^*$$

and obtain

$$x = \frac{T_c(L^*)L^* - T_c(L)L}{L^* - L}$$

### 3 Numerical methods

#### 3.1 Metropolis algorithm

We imagine to start out with an initial spin state that is random or ordered. We will then use the following approach to simulate the system develop in time:

1. The system are in a spin state with energy  $E$
2. Create a trial state with the trial energy  $E_t$  by flipping one spin
3. Compute  $\Delta E = E_t - E$ 
  - If  $\Delta E \leq 0$ 
    - Accept the trial state as the new state
  - If  $\Delta E > 0$  calculate  $w = e^{-\beta\Delta E}$  and create a random number  $r$ 
    - If  $r \leq w$  (the metropolis test): Accept the trial state as the new state
    - Else discard the trial state (do nothing with the state)
4. Do 1.-3.  $L \times L$  times (one MC cycle)
5. Update mean values



6. Repeat 1.-6. until enough statistics is sampled, the desired number of MC cycles

One Monte Carlo cycle corresponds to one loop that have the possibility to flip all the spins in the lattice.

The algorithm can also be expressed as a Pseudocode:

```
for numberOfTemperatures:
{
    //create initial state here
    //calculate initial energy and magnetization here
    for numberOfMCcycles:
    {
        for numberOfspins:
        {
            //do one spin flip here
        }
        //update mean values here
        //print mean values to file here
    }
    //or print values to file here
}

//then read data from file and plot in python
```

This is the set up that is used in this project. Some of the versions of the algorithm the temperature loop is left to the python code, which also compiles and runs the cpp code. This is one way of parallellizing the code. In the final version of the program we have parallellized the code using `openMPI`.

We will also implement the algorithm so that we discard all contributions to the expectation values before the system have reached equilibrium. To do this we have to test the algorithm and study how many Monte Carlo cycles the system uses to reach the equilibrium state.

The folder structure are as follows:

- Folders used for testing:
  - `test_ising_metropolis` for the  $L = 2$  system
  - `equilibrium_ising_metropolis` to study the equilibrium time and the number of accepted cycles
  - `probability_ising_metropolist` to study the probability for a given energy and temperature
- Final folder to find the critical temperature  $T_c$ :
  - `ising_metropolis_MPI`
- Folder for this report:
  - `report4`

where all folders are found at [https://github.com/inakbk/Project\\_4](https://github.com/inakbk/Project_4).

### 3.2 Coding $\Delta E$ , $\Delta \mathcal{M}$ and $w$ efficiently

We wish to run the simulations for large numbers of Monte Carlo cycles  $10^5$ . It is therefore very important to code the calculation of  $\Delta E$  efficiently because it will be calculated  $N$  times each Monte Carlo cycle.

The difference in the energy can be expressed as:

$$\Delta E = E_2 - E_1 = -J \sum_{\langle kl \rangle}^N s_{k,2} s_{l,2} + J \sum_{\langle kl \rangle}^N s_{k,1} s_{l,1}$$

where the  $l$  index is the spin that was flipped and  $k$  the neighbours of  $l$ . We only flip one spin at a time, so the neighbours of  $s_l$  are unchanged meaning that  $s_{k,2} = s_{k,1} = s_k$ . The difference in energy can then be written as:

$$\Delta E = J \sum_{\langle kl \rangle}^N s_k s_{l,1} - J \sum_{\langle kl \rangle}^N s_k s_{l,2} = J \sum_{\langle kl \rangle}^N s_k (s_{l,1} - s_{l,2})$$

The spin can only take two values, so  $s_{l,1}$  is either 1 or -1. If  $s_{l,1} = 1$  then after the flip  $s_{l,2} = -1$  and  $s_{l,1} - s_{l,2} = 1 - (-1) = 2$ . If  $s_{l,1} = -1$  then after the flip  $s_{l,2} = 1$  and  $s_{l,1} - s_{l,2} = -1 - 1 = -2$ . We can therefore write  $s_{l,1} - s_{l,2} = 2s_{l,1}$  and then we obtain:

$$\Delta E = J \sum_{\langle kl \rangle}^N s_k \cdot 2s_{l,1} = 2Js_{l,1} \sum_{\langle k \rangle}^N s_k$$

Similarly for the magnetization:

$$\begin{aligned} \Delta \mathcal{M} &= \mathcal{M}_2 - \mathcal{M}_1 = \sum_i^N s_{i,2} - \sum_i^N s_{i,1} = \sum_i^N (s_{i,2} - s_{i,1}) = s_{l,2} - s_{l,1} = -2s_{l,1} \\ \Rightarrow \mathcal{M}_2 &= \mathcal{M}_1 - 2s_{l,1} \end{aligned}$$

This way of calculating the difference in energy and magnetization makes the algorithm for flipping one spin run faster than other solutions. We also want to avoid to calculate the exponential of  $\Delta E$ , the value of  $w = e^{-\beta \Delta E}$  for every time the trial energy is higher than the current energy. We can avoid this by noticing that there are only 5 possible values  $\Delta E$  can take when we only flip one spin at a time (in the two dimensional case)  $\Delta E \in \{-8, -4, 0, 4, 8\}$ . So the energy minimum have to change in steps of 4, and maximum in steps of 8 when we flip one spin at a time. Then there are also only 5 possible values of  $w$  for a given temperature. We can exploit this further by realizing that we only use  $w$  when  $\Delta E > 0 \Rightarrow \Delta E \in \{4, 8\}$ . For a given temperature we then calculate the two values of  $w$  once and pass it on to the metropolis test which uses the value corresponding to  $\Delta E$ .

With this methods implemented in the algorithms the execution time is much faster.

## 4 Implementation and results

### 4.1 Testing the implementation of the model and algorithm

First we will compare the results for the metropolis algorithm with the theoretical values for the  $L = 2$  case calculated in section 2.2. All calculations and results in this report is per spin, meaning that we have divided the values with the total number of spins  $N = L^2$ .

In figure 1 we see the expectation value of the energy (top), energy squared (middle) and the heat capacity (bottom) as a function of the temperature. The simulation was run for  $5 \cdot 10^5$  Monte Carlo Cycles. The simulation was started with an initial state with all spins up. In figure

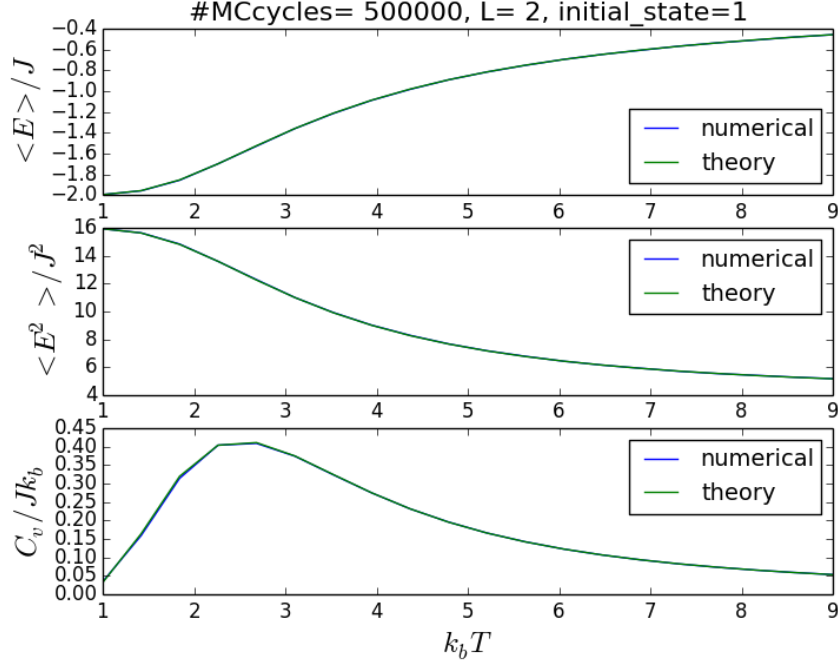


Figure 1: Expectation value of the energy (top), energy squared (middle) and the heat capacity (bottom) per spin as a function of the temperature. The numerical calculation was run for  $5 \cdot 10^5$  Monte Carlo Cycles with a initial state of all spins up (`initial_state=1`) with  $L = 2$ . The numerical (blue line) and the teoretical (green line) values are plotted together.

The numerical values agree very well with the theoretical values. In figure 3 we see the absolute error in the Expectation value of the energy (top), energy squared (middle) and the heat capacity (bottom) per spin as a function of Monte Carlo cycles. In figure 4 we see the absolute error in the expectation value of the absolute magnetization (top), magnetization squared (middle) and the susceptibility (bottom) as a function of Monte Carlo Cycles. The temperature in both plot 3 and 4 was held at  $T = 1$ .

The spin system with  $L = 2$  is very small so it takes very few MC cycles to reach the equilibrium state, which for low temperatures are close to the lowes energy state. I does therefore not matter which initial state the system start out with. The error in the numerical calculcations will not depend on the initial state, but on the amount of statistics, how many MC cycles. In the simulation run in figure 3 we have used a initial state of all spins up (`initial_state=1`) which is the state with the lowest energy.

In the figures 3 and 4 we see that the error falls quickly and stabilizes after  $2 \cdot 10^5$  MC cycles. The heat capacity has the largest error with absolute error below 0.01 after stabilizing. Most of the error comes from the expectation value of the energy squared while the error in the expectation value of the energy is much smaller, 0.002. The error in the suceptibility falls below 0.002, smallest error is for the the expectation value of the absolute magnetization of

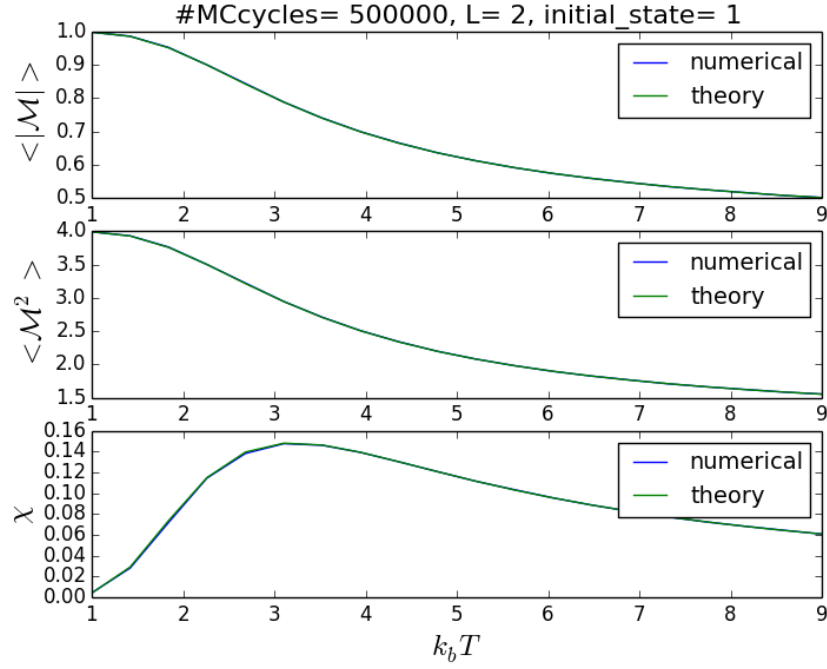


Figure 2: Expectation value of the absolute magnetization (top), magnetization squared (middle) and the susceptibility (bottom) per spin as a function of the temperature. The numerical calculation was run for  $5 \cdot 10^5$  Monte Carlo Cycles with a initial state of all spins up (`initial_state=1`) with  $L = 2$ . The numerical (blue line) and the teoretical (green line) values are plotted together.

less than 0.0005.

From looking at figure 3 and 4 it seems reasonable to choose the number of MC cycles to be more than  $2 \cdot 10^5$ .

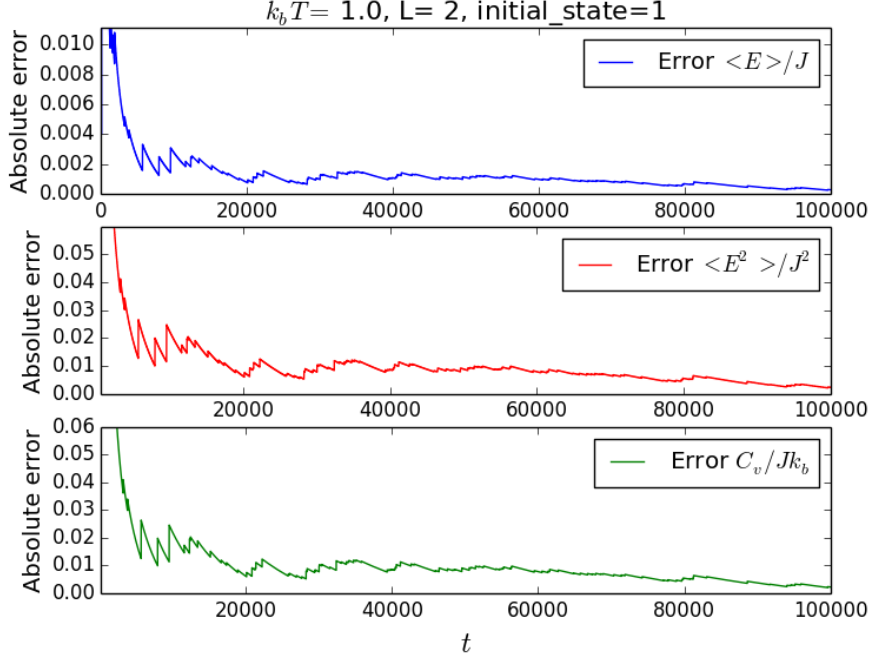


Figure 3: Absolute error in the expectation value of the energy (top), energy squared (middle) and the heat capacity (bottom) a function of Monte Carlo Cycles with a initial state of all spins up (`initial_state=1`) with  $L = 2$ . The temperature was held at  $T = 1$ .

## 4.2 Equilibrium time

We will now look at a larger system with  $L = 20$  and study the equilibrium time, how many Monte Carlo cycles the system uses to reach the stable equilibrium state. The calculations of mean values should start after the equilibrium state is reached so that the statistics are not 'ruined' by the path from the initial state to the equilibrium state. We will therefore try to decide the equilibrium time. We will look at two temperatures,  $T \in \{1.0, 2.4\}$  and start in both a random and ordered initial state. For the ordered state we will have all spins up.

In figure 5 we see the expectation value of the energy per spin as a function of Monte Carlo cycles at the top and the expectation value of the absolute magnetization per spin as a function of Monte Carlo cycles on the bottom. The plots to the left are with the initial state ordered and we can clearly see that the system stabilizes around a value both for the mean energy and mean absolute magnetization.

The plots to the right in figure 5 started with a random initial state and have a different unexpected behaviour. We see that the expectation values have a 'hump' before the stabilizing. The same strange behaviour can be observed in figure 6 which is the same plot, but for a higher temperature  $T = 2.4$ . We see that the plots with random initial state are very smooth and do not have any statistically behaviour.

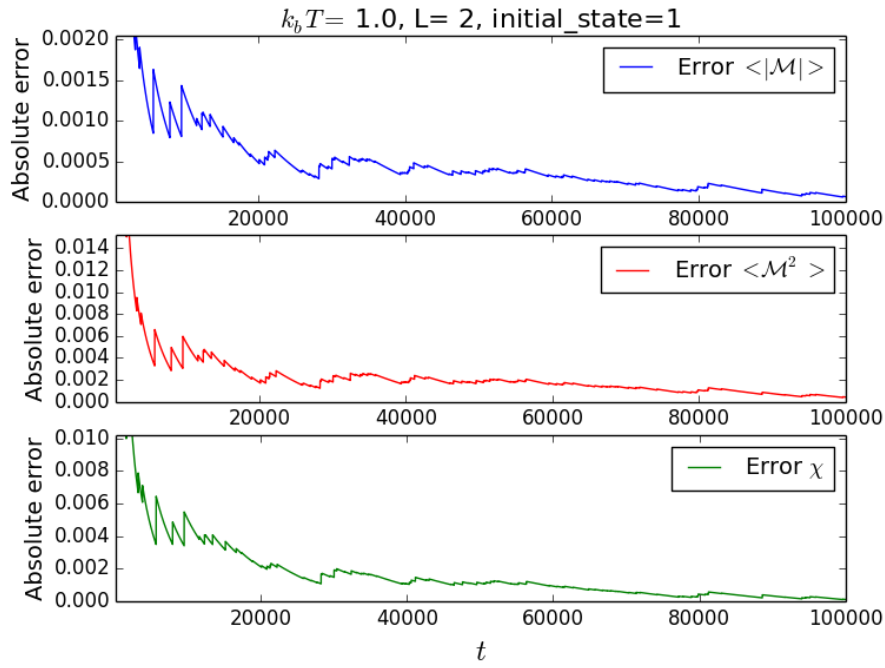


Figure 4: Absolute error in the expectation value of the absolute magnetization (top), magnetization squared (middle) and the susceptibility (bottom) as a function of Monte Carlo Cycles with a initial state of all spins up (`initial0.8_state=1`) with  $L = 2$ . The temperature was held at  $T = 1$ .

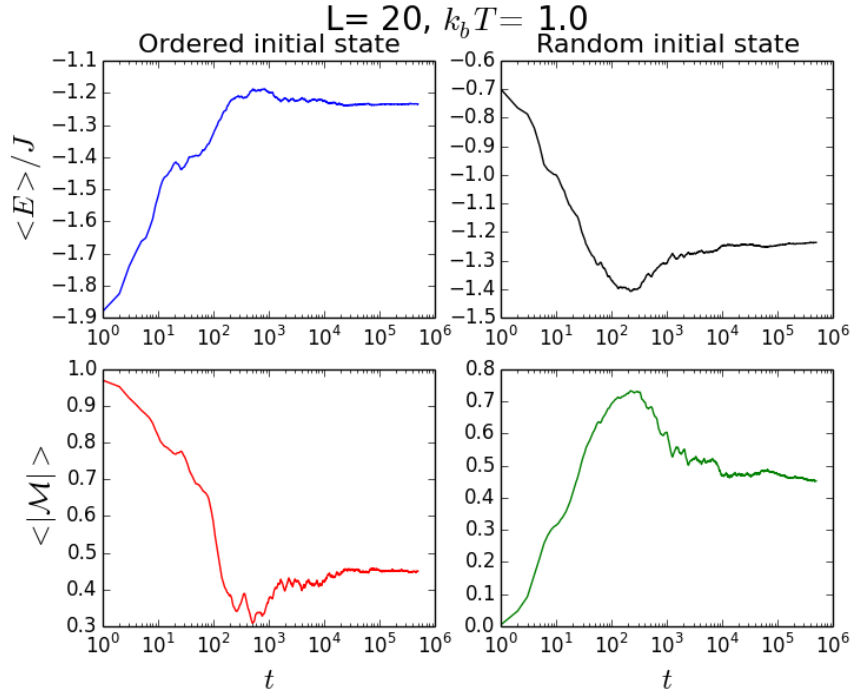


Figure 5: Top: Expectation value of the energy per spin as a function of Monte Carlo cycles. Bottom: Expectation value of the absolute magnetization per spin as a function of Monte Carlo cycles. To the left the initial state is ordered and to the right the initial state is random. The size of the lattice is  $L = 20$  and the temperature is  $T = 1$ .

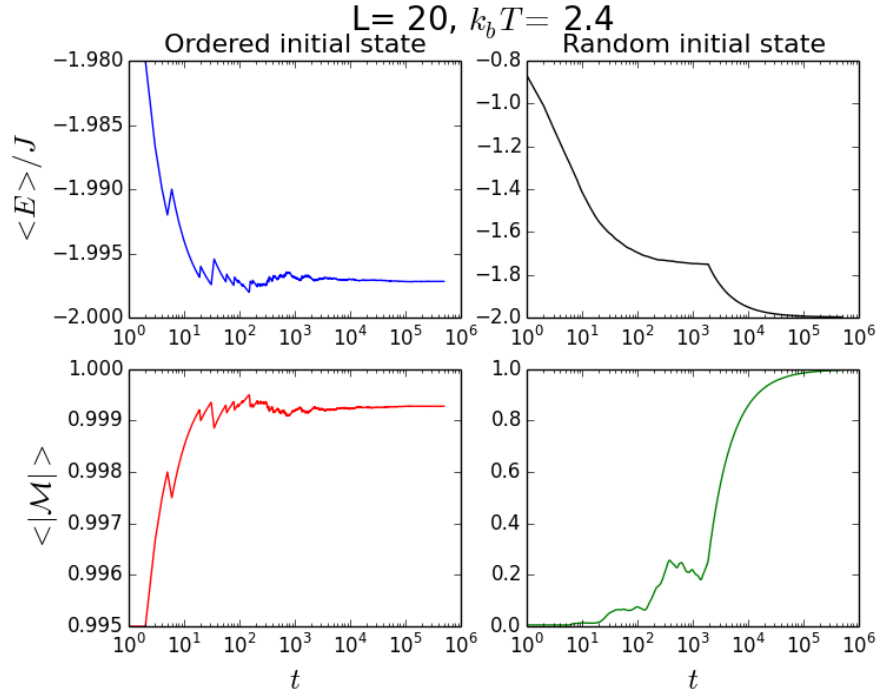


Figure 6: Top: Expectation value of the energy per spin as a function of Monte Carlo cycles. Bottom: Expectation value of the absolute magnetization per spin as a function of Monte Carlo cycles. To the left the initial state is ordered and to the right the initial state is random. The size of the lattice is  $L = 20$  and the temperature is  $T = 2.4$ .



In figure 7 we see the accepted configurations per spin as a function of MC cycles for both ordered and random initial state for the temperature  $T = 1$ . We see that the number of accepted configurations per spin grows as number of MC cycles increase and then flats out. For the random initial state we see the opposite behaviour, the number of accepted configurations decrease as a function of MC cycles. We would expect that for a low temperature as  $T = 1$  the equilibrium state are closer to the ordered state than the random state. It is natural that for the random initial state the number of accepted configurations are very large in the start as the system is 'trying' to reach the equilibrium state, and when the equilibrium state is found the number of accepted states flats out.

The corresponding plot for  $T = 2.4$  is shown in figure 8. We see that the number of accepted configurations flats out for the ordered and random initial state here too. But the Random initial state have a very smooth slope, which the low temperature case did not have.

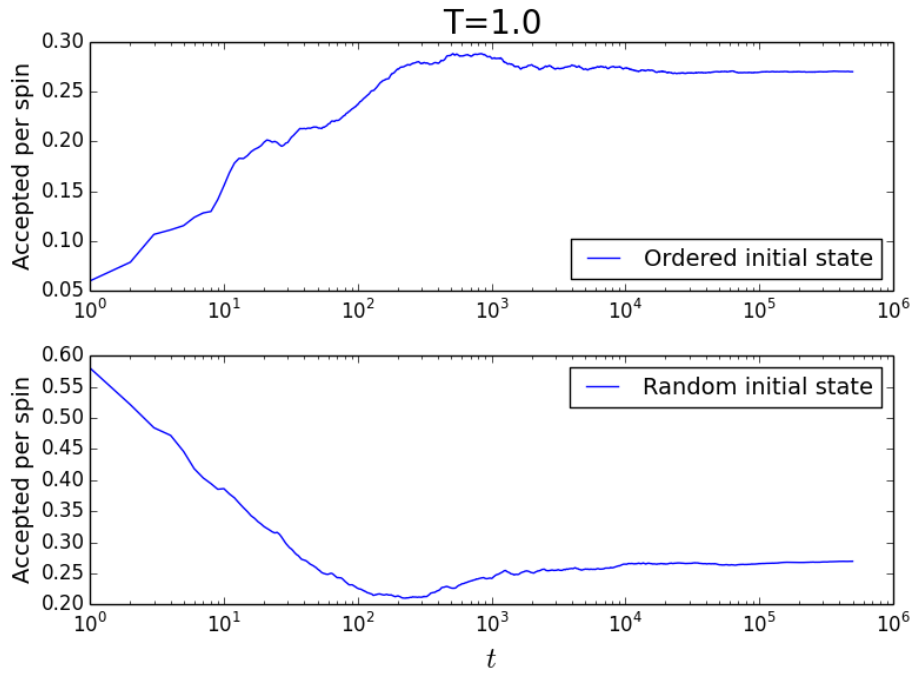


Figure 7: Accepted configurations per spin as a function of MC cycles. Top: Ordered initial state (all spinst up). Bottom: Random initial state. The temperature is  $T = 1$ .

In this subsection we have looked at the behaviour of the  $L = 20$  system as a function of Monte Carlo cycles. There are some unexpected non-statistical behaviour in the plot with the random initial state that we have no explanation for. We have therefore decided to always start with the ordered initial state in the further calculations. For larger temperatures and  $L$  this could mean that we would have to use unnecessary many MC cycles to reach the equilibrium state, but we choose to ignore this in this paper.

For both temperatures the system stabilized for  $10^5$  MC cycles so we will use a equilibrium time of  $t_{eq} = 2 \cdot 10^5$  MC cycles in this paper.

In figure 9 we see the number of accepted configurations per spin as a function temperature for the ordered initial state. We start the simulation in the ordered state. It is as expected that the number of accepted configurations increase with temperature when the total number

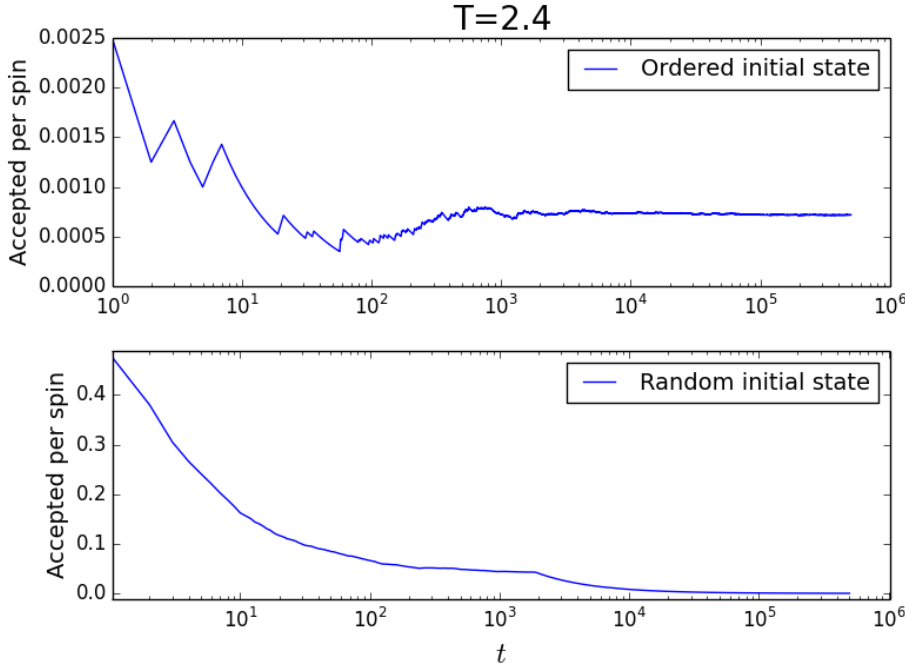


Figure 8: Accepted configurations per spin as a function of MC cycles. Top: Ordered initial state (all spinst up). Bottom: Random initial state. The temperature is  $T = 2.4$ .

of Monte Carlo cycles is much larger (500000) than the equilibrium time. This is because when the temperature is low it is very unlikely to flip a spin after the equilibrium state is reached because the state is very close to the ground state. The probability to flip a spin increases as the temperature increases because the spread in energy is larger,  $\sigma_E$  is larger. Thus the number of accepted cycles increase as a function of temperature.

### 4.3 Probability

In this subsection we will look at the probability for a given energy and see if this matches our physical intuition about the system. We will look at the case  $L = 20$  and the temperatures  $T \in \{1.0, 2.0, 1.4\}$ . We have used the values for the energy after the system have reached the equilibrium state. This is solved by letting the simulation run for the equilibrium time first and then start to store the energies as the time goes.

In figure 10 we see the probability plotted as a function of energy per spin for the temperatures  $T \in \{1.0, 2.0, 1.4\}$  with the corresponding standard deviations  $\sigma_E \in \{0.15, 1.71, 2.8\}$ . We see that for the lowest energy it is extremely likely to find the system in the ground state. This corresponds well to the very low standard deviation  $\sigma_E = 0.15$ . This means that the distance between the mean energy and the other energies are small. Almost every state is in the ground state at low temperatures, then the spread in energy must be small. We see that for the temperature  $T = 2.0$  the probability is more spread out. It is also more likely to find the system in a slightly higher energy than for the lowest temperature. For the highest temperature  $T = 2.4$  we see that the probability have shifted even further to higher temperature. The spread of the energies corresponds well with the standard deviation  $\sigma_E = 2.8$ . It seems

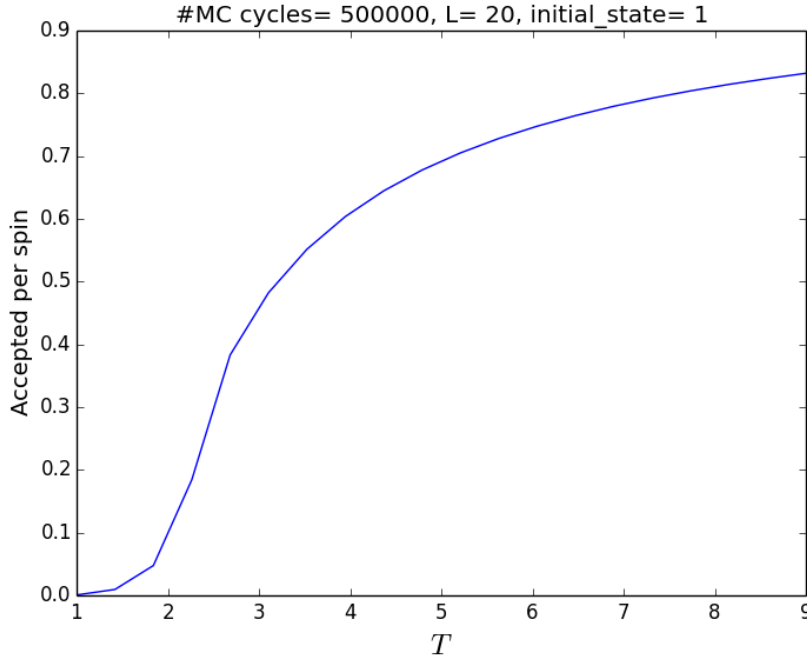


Figure 9: Accepted configurations per spin as a function temperature. Top: Ordered initial state (all spinst up). Bottom: Random initial state. The total number of monte carlo cycles is 500000 for the  $L = 20$  and ordered initial state.

reasonable that the system fluctuates around a higher energy when the temperature is higher, and that it closes up to the ground state when the temperature is low.

We have now seen that our system behaves as expected for the ordered initial state. The theoretical values correspond well with the numerical values, the system reaches the equilibrium state and fluctuates around it and the probability as a function of energy behaves as expected for high and low energies. Therefore it seems reasonable to assume that the numerical algorithm is implemented correctly.

#### 4.4 Calculations to find $T_c$

We will now use this algorithm to find the critical temperature for the two dimensional system. We will plot the expectation value for the energy and absolute magnetization, the heat capacity and the susceptibility as a function of temperature for four different values of the lattice,  $L \in \{20, 40, 60, 80\}$ . We will then try to see if we can see a phase transition and from this calculate the critical temperature for which the phase transition happens.

In figure 11 we see the expectation value for the energy and absolute magnetization, the heat capacity and the susceptibility as a function of temperature for  $L = 20$ . We see that the energy increases as a function of temperature. This is reasonable as seen in the probability plot, for a higher energy the spin state is more random. We also see that the absolute magnetisation is close to 1 for low temperatures. This is natural as then the system is close to the ground state with either spins up or down,  $|\mathcal{M}| = 1$ . For high temperatures the system is again in a more random state with as many spins up as down. So it is reasonable that the absolute

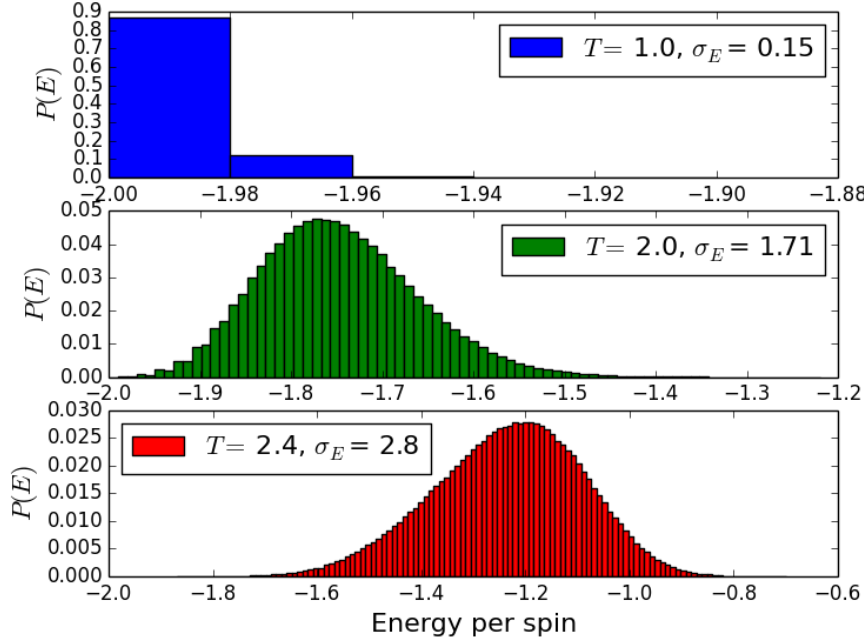


Figure 10: Probability as a function of energy per spin for the temperatures  $T \in \{1.0, 2.0, 1.4\}$  with the corresponding standard deviations  $\sigma_E \in \{0.15, 1.71, 2.8\}$ .

Table 4: Table of the values for the critical temperature read from figure 13. The values for  $T_c$  varies as a function of  $L$  and is read from both the plot of the heat capacity and the susceptibility.

$L$	$T_C(C_V)$	$T_c(\chi)$
20	2.32	2.34
40	2.29	2.31
60	2.28	2.30
80	2.27	2.3

magnetisation goes to zero for higher temperatures. This is also a sign for a phase transition, that a variable 'disappears'.

In figure 12 we also see the expectation value for the energy and absolute magnetization as a function of temperature for all lattice sizes,  $L \in \{20, 40, 60, 80\}$ . We see that the tendency is the same for all the different values of  $L$ .

In figure 11 we also see that both the heat capacity and the susceptibility have a maximum at about the same temperature. This is also a sign for a phase transition. In figure 13 we see that this maximum becomes sharper for increasing lattice size.

The values for the critical temperature read from figure 13 are listed in table 4. The values for  $T_c$  varies as a function of  $L$  and is read from both the plot of the heat capacity and the susceptibility.

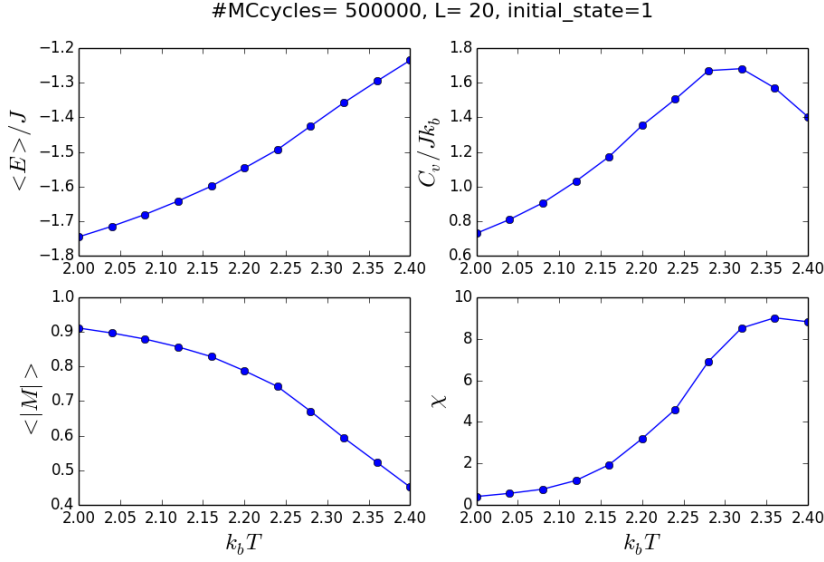


Figure 11: The expectation value for the energy and absolute magnetization, the heat capacity and the susceptibility as a function of temperature for  $L = 20$ . The total number of MC cycles is 500000 and the initial state is ordered.

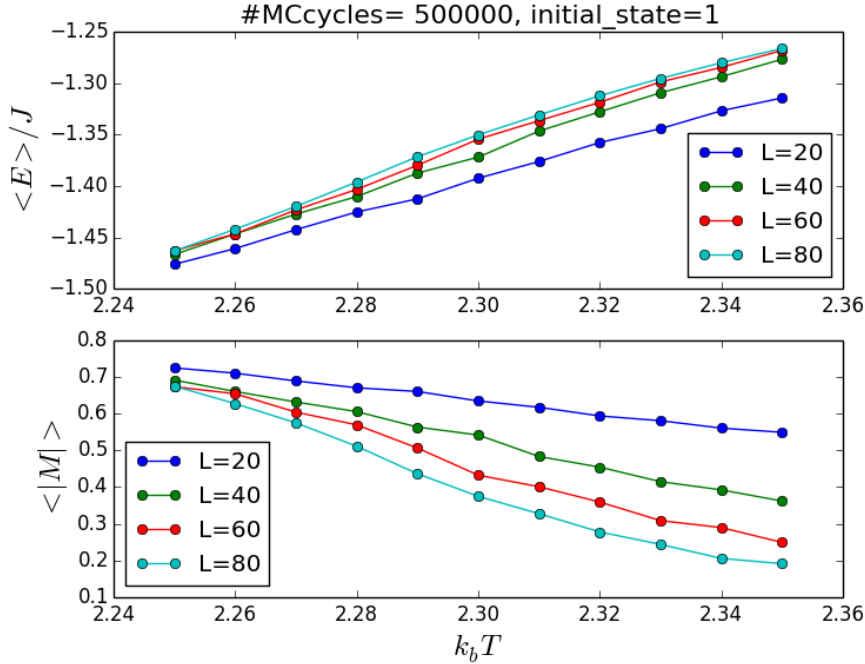


Figure 12: The expectation value for the energy and absolute magnetization as a function of temperature for  $L \in \{20, 40, 60, 80\}$ . The total number of MC cycles is 500000 and the initial state is ordered.

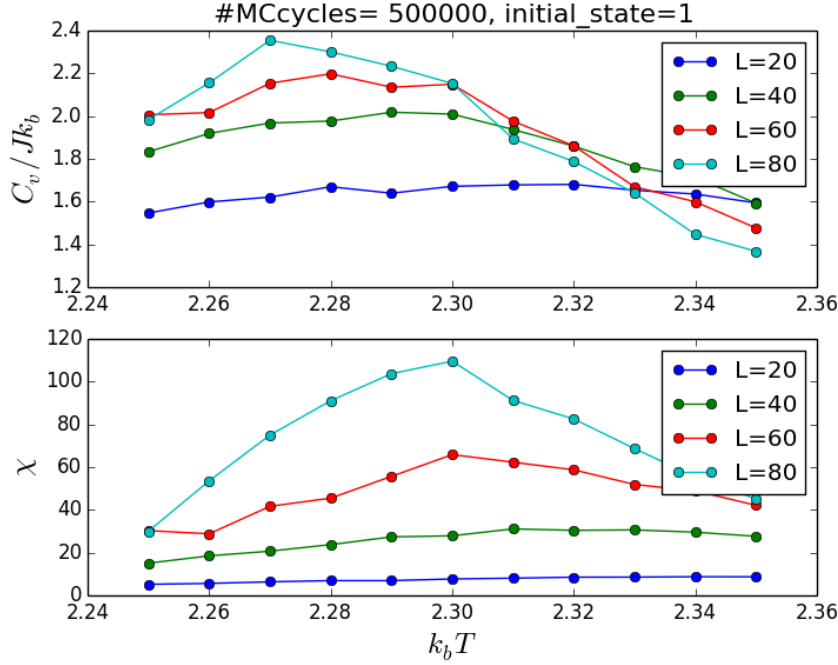


Figure 13: The heat capacity and the susceptibility as a function of temperature for  $L \in \{20, 40, 60, 80\}$ . The total number of MC cycles is 500000 and the initial state is ordered.

#### 4.5 Estimating the critical temperature

We will now use equation 14 and calculate the best estimate for the critical temperature  $T_c$  and compare with Lars Onsagers theoretical result  $T_c = 2.269$ . This is done in the python script `calculate_Tc.py`. We run this python script and obtain the output:

```
Numerical T_c = 2.27
Theoretical T_c = 2.269
Error in estimate = 0.001
```

We see that the absolute error in the estimate of the critical temperature is as small as 0.001!

### 5 Discussion and experiences

This method for obtaining the critical temperature have proved itself to be useful and quite accurate as the error in the estimate was very low!

But there is room for improvement. We could have run the methods for many times with different seeds in the random generator function for that we would obtain a even larger amount of statistics. It is possible to do this by letting the seed be the negative time in the core in seconds:

```
seed = -time(NULL);
```

when we which to produce large amounts of data.

We could also have let all the negative values of  $\Delta E$  go to the metropolis test to avoid the if test for  $\Delta E \leq 0$ . This is because  $e^{-\Delta E/T} > 1$  when  $\Delta E \leq 0$  and then the metropolis test  $e^{-\Delta E/T} > r$  will never be true.

We could also have regulated the equilibrium time so that it corresponded to the lattice size  $L$  and the temperature. As we could also have chosen the initial state according to the temperature. Both cases would have saved a lot of execution time. But that was not a priority in this project because the lattice sizes run are not so huge. Imagine running for lattice sizes of order a 1000...

Either way we could be happy about estimating the critical temperature with quite good precision.