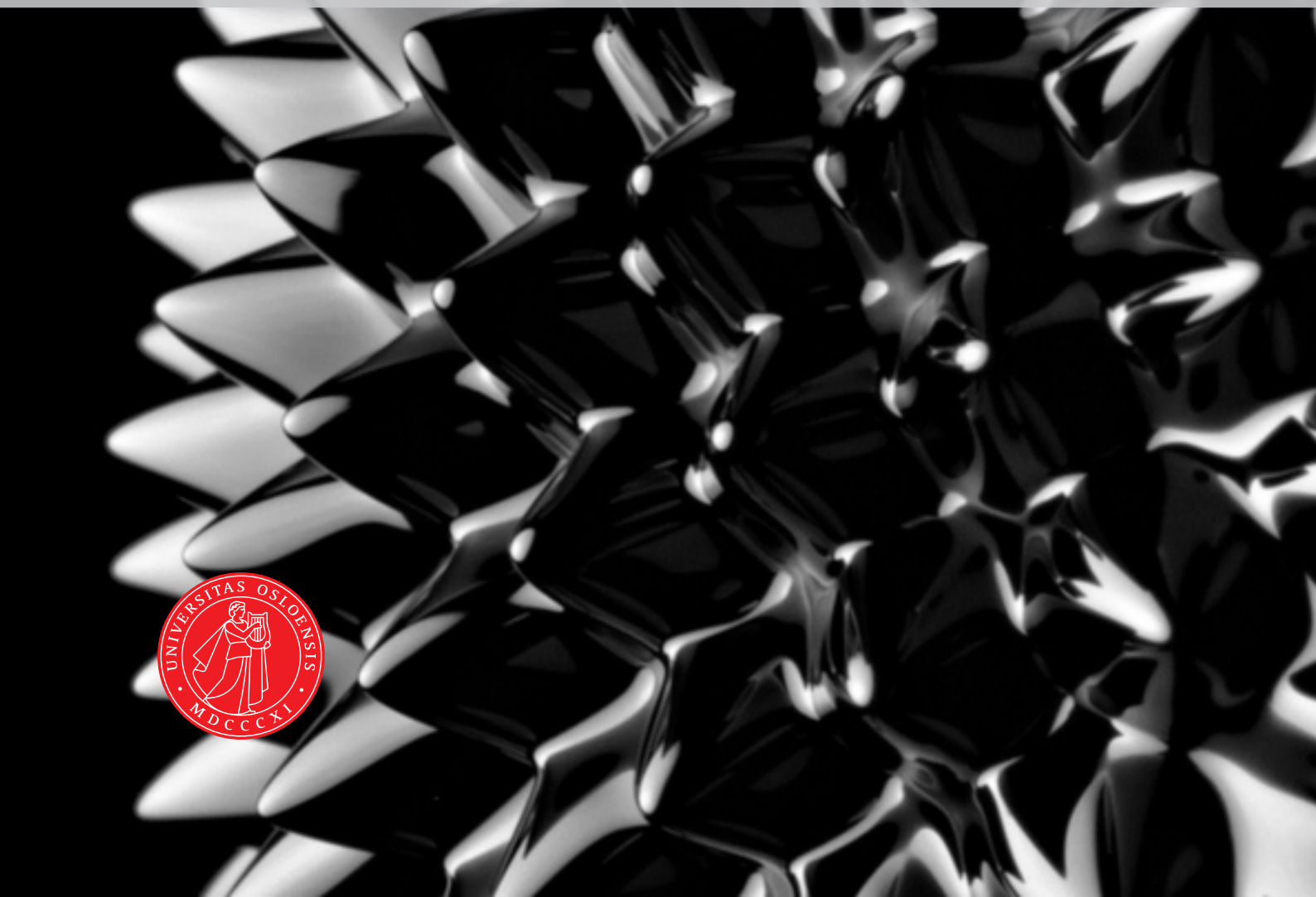
UiO : Department of Physics
University of Oslo

Studies of phase transitions in magnetic systems

Erik Skaar



Contents

Abstract

1 Introduction

These laws are not enough to solve the motion of the planets. From the laws one can derive differential equations for the motion, which are not trivial or even possible to solve analytically. This is where computational methods are useful. With the tools developed in computtational physics we can make a prediction to the motion of the planets in our solar system.[1] And because of our assignment we kind of have to do this to pass the course.[2]

[1] Semester page for FYS3150 - Autumn 2017.

2 Theory

2.1 the Ising model

The Ising model describes a coupled system. Where only the nearest neighbor affect each other. In this report the Ising model will be applied to a two dimensional magnetic system. This will be a grid of spins, where each spin s_i can either have 1 or 0 as value. The total energy is expressed as:

$$E = -\sum_{<i,j>} J_{i,j} s_i s_j$$

Where the symbol $<$ kl $>$ indicates that we sum over nearest neighbors only. If we assume that each coupling has the same magnitude J, then the energy is expressed as:

$$E = -J \sum_{<i,j>} s_i s_j \tag{1}$$

2.1.1 Periodic boundary conditions

When working with a finite matrix we run into a problem with the boundaries. They are missing neighbours. We solve this by introducing periodic boundary conditions. This means that the right neighbour for S_n is assumed to take the value of S_1.

2.2 Statistical physics

2.2.1 the partition function

Boltzmann distribution is used as the probability distribution. Boltzmann distribution states the probability for E_i is proportional to $e^{-\beta E_i}$, where β is $\frac{1}{k_B T}$. k is the Boltzmann constant. For this to be a probability distribution, it needs to be normalized. To normalize the distribution divide the sum of probabilities by a constant Z:

$$1 = \frac{\sum_i e^{-\beta E_i}}{Z}$$
$$Z = \sum_i e^{-\beta E_i}$$

Z is called the partition function.

2.2.2 Calculation of values

The partition function is very useful. In combination with the Boltzmann distribution we get a expression for the probability.

$$P(E_i) = \frac{e^{-\beta E_i}}{Z}$$

For finding a mean value, one can simply make a sum over $P(E_i)$ multiplied by the value of interest. For instance the mean energy is given by:

$$\langle E \rangle = \sum_i E_i P(E_i)$$

Expressions for important expectation values can be derived such for the energy E, magnetic moment |M|, specific heat capacity C_v and the susceptibility χ. The expressions used in this report are listed below[1]:[2]

$$\langle E \rangle = \sum_i E_i P(E_i) \tag{2}$$

$$\langle |M| \rangle = \sum_i M_i P(E_i) \tag{3}$$

$$\langle C_V \rangle = \frac{1}{kT^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right) \tag{4}$$

$$\langle \chi \rangle = \frac{1}{kT} \left(\langle M^2 \rangle - \langle |M| \rangle^2 \right) \tag{5}$$

2.3 Phase transition

The two dimensional Ising model is able to predict a phase transition in the material. At a critical temperature T_C the quantities for the material will start to behave differently. For C_V and for χ the phase transition is a sharp peak when plotted against Temperature. For $|M|$ and E it can be seen, but only as a slight change in value.

A second order phase transition is characterized by a correlation length. For finite lattice the correlation length is equal to the length of the system. T_C can be obtain through scaling of the results from a finite system with a infinite system:

$$T_C(L) - T_C(L = \infty) = aL^{\frac{-1}{v}} \tag{6}$$

a is an unknown constant and v = 1. For finding a we use eq. 6 with two different L. Substract the expression with L_i by the expression with L_j and we get:

$$T_C(L_i) - T_C(L_j) = a \left(L_i^{\frac{-1}{v}} - L_j^{\frac{-1}{v}} \right)$$

$$a = \frac{T_C(L_i) - T_C(L_j)}{L_i^{\frac{-1}{v}} - L_j^{\frac{-1}{v}}} \tag{7}$$

We combine this with eq. 6 and we get an expression for $T_C(\infty)$:

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

$$T_C(L = \infty) = T_C(L) - \frac{T_C(L_i) - T_C(L_j)}{L_i^{\frac{-1}{v}} - L_j^{\frac{-1}{v}}} L^{\frac{-1}{v}} \tag{8}$$

[2] lecture note page 420

3 Method

3.1 Metropolis algorithm

The metropolis algorithm only has a few steps. First, pick one site in the matrix of spins. This process need to be random. For that site, calculate the energy difference if the spin is flipped. Then the algorithm decide whether to flip the spin or not. This is decided based on the energy difference. If the difference is negative flip, then flip the spin. If not, then pick a random number between 0 and 1 and if this number is less then $e^{-\beta\Delta E}$ flip the spin. Else keep the spin. Finally update expectation values.

3.1.1 Precalculate

The energy difference is expressed as an exponential function. Exponential values are expensive to calculate. In two dimensions there is a finite number of energy differences. We can precalculated the exponentials. By calculating these in advance the program will run more efficient. It can be shown that the energy difference then is:

$$\Delta E = 2Js_j \sum_{<k>} s_k$$

Writ
more

4 Implementation

The metropolis algorithm was implemented as discussed in section 3.1 in the programs called main-"...".cpp. Their is a few different versions of the main.cpp. The only difference is basicly how they write to file. All of the programs discussed in this section can be found at github.

These calculations are expensive in terms of FLOPs. That is why a parallelized version has been made. Not all the code needs to be parallelized. Most of the code in the parallelized version is the same as for the non-parallelized. Except each thread open a specific file for that thread and runs the metropolis algorithm for $\frac{1}{\text{nr. of temperatures}}$. What temperature that each thread calculate is determined by the rank of the thread. Since the rank is a unique number for each thread, all the temperatures calculated is unique from the other temperatures.

MPI is used since it is easy to implement and it does not have shared memory. The parallelized version should be (nr. of threads - 1) times faster then the normal version.

Table 1: The grids ran for 50'000 Monte Carlo cycles. The test ran on a macbook pro 13. It has a dual core CPU. Expected difference is 2.

Size	Normal	MPI	Expected difference	Actual difference
40x40	15.251s	5.991 s	2.000	2.546
60x60	33.923s	13.351 s	2.000	2.541
100x100	92.584s	36.245 s	2.000	2.554

The difference was higher, then expected. This is due to the Hyper-Threading technology in this CPU [3].

For size scaling we expect a time increased proportional to the size increase squared.

Table 2: The table shows how we expect the time to develop and how it actually it develops. There is a minor difference from expected and calculated and that comes from the fact that the program does more then just the algorithm and the fact that the algorithm has not been perfectly implemented.

Size	Expected time	Actual time	$\frac{T_i}{T_{40}}$
40x40	x	8.490 s	1.000
60x60	2.25x	19.350 s	2.279
100x100	6.25x	54.068 s	6.368

[3] Intel Hyper-Threading Technology

5 Result & Discussion

5.1 Analytic 2x2

5.1.1 Microstates 2x2

Table 3: This shows the different microstates that is possible for a 2x2 spinmatrix. It also states the energy and magnetic moment for each microstate.

State	Energy	Magnetic moment	State	Energy	Magnetic moment
↑ ↑ ↑ ↑	-8J	4	↓ ↓ ↓ ↓	-8J	-4
↓ ↑ ↑ ↑	0J	2	↑ ↓ ↓ ↓	0J	-2
↑ ↓ ↑ ↑	0J	2	↓ ↑ ↓ ↓	0J	-2
↑ ↑ ↓ ↑	0J	2	↓ ↓ ↑ ↓	0J	-2
↑ ↑ ↑ ↓	0J	2	↓ ↓ ↓ ↑	0J	-2
↓ ↓ ↑ ↑	0J	0	↑ ↑ ↓ ↓	0J	0
↓ ↑ ↓ ↑	0J	0	↑ ↓ ↑ ↓	0J	0
↑ ↓ ↓ ↑	8J	0	↓ ↑ ↑ ↓	8J	0

Table 4: The table shows a summary from table 5.1.1.

Number of ↑	Multiplicity	Energy	Magnetic moment
4	1	-8J	4
3	4	0J	2
2	2	8J	0
2	4	0J	0
1	4	0J	-2
0	1	-8J	-4

5.1.2 Quantities

We will use the equations from section 2.2.2.

For energy the eq. 2 will result in:

$$
\begin{aligned}
Z &= \sum_i = e^{-\beta E_i} \\
\mathrm{T} &= \mathrm{kT/J} = 1 \\
Z &= \sum_i = e^{-\beta E_i} = 2e^8 + 2e^{-8} + 12
\end{aligned}
$$

For energy the eq. 2 will give the result:

$$
\begin{aligned}
\langle E \rangle &= \sum_i E_i P(E_i) \\
\mathrm{T} &= \mathrm{kT/J} = 1 \\
\langle E \rangle &= \frac{1}{Z} \sum_i E_i e^{-E_i} \\
\langle E \rangle &= \frac{1}{Z} \left(-16e^8 + 16e^{-8} \right) = -7.9839 \\
\langle E \rangle / N &= \frac{\langle E \rangle}{4} = -1.9959
\end{aligned}
$$

For energy the eq. 3 will give the result:

$$
\begin{aligned}
\langle |M| \rangle &= \sum_i M_i P(E_i) \\
\mathrm{T} &= \mathrm{kT/J} = 1 \\
\langle |M| \rangle &= \frac{1}{Z} \sum_i M_i e^{-E_i} \\
\langle |M| \rangle &= \frac{1}{Z} \left(4 \cdot 1e^8 + 2 \cdot 4e^0 + 0 \cdot 2e^{-8} + 0 \cdot 4e^0 + 2 \cdot 4e^0 + 4 \cdot 1e^8 \right) \\
\langle |M| \rangle &= \frac{1}{Z} \left(16 + 8e^8 \right) = 3.9946 \\
\langle |M| \rangle / N &= \frac{\langle M \rangle}{4} = 0.9986
\end{aligned}
$$

For C_V we need to calculate $\langle E^2 \rangle$:

$$\begin{aligned}
&\langle E^2 \rangle = \sum_i E_i P(E_i) \\
&\mathrm{T} = \mathrm{kT}/\mathrm{J} = 1 \\
&\langle E^2 \rangle = \frac{1}{Z} \sum_i E_i^2 e^{-E_i} \\
&\langle E^2 \rangle = \frac{1}{Z} \left(128e^8 + 128e^{-8} \right) \\
&C_V = \langle E^2 \rangle - \langle E \rangle^2 = 0.12832 \\
&C_V/N = 0.03208
\end{aligned}$$

For χ we need to calculate $\langle M^2 \rangle$:

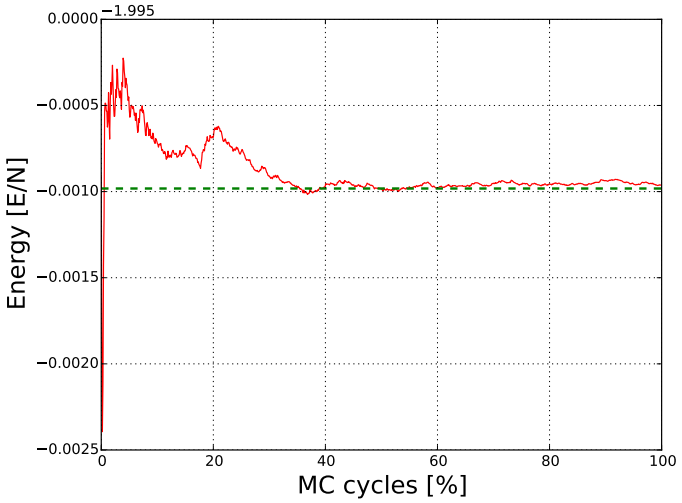
$$\begin{aligned}
&\langle M^2 \rangle = \sum_i M_i^2 P(E_i) \\
&\mathrm{T} = \mathrm{kT}/\mathrm{J} = 1 \\
&\langle M^2 \rangle = \frac{1}{Z} \sum_i M_i^2 e^{-E_i} \\
&\langle M^2 \rangle = \frac{1}{Z} \left(16 \cdot 1e^8 + 4 \cdot 4e^0 + 0 \cdot 2e^{-8} + 0 \cdot 4e^0 + 4 \cdot 4e^0 + 16 \cdot 1e^8 \right) \\
&\langle M^2 \rangle = \frac{1}{Z} \left(32 + 32e^8 \right) = 15.9732 \\
&\langle \chi \rangle = 0.01604 \\
&\langle \chi \rangle / N = 0.004010
\end{aligned}$$

Below you can see a summary for the quantities:

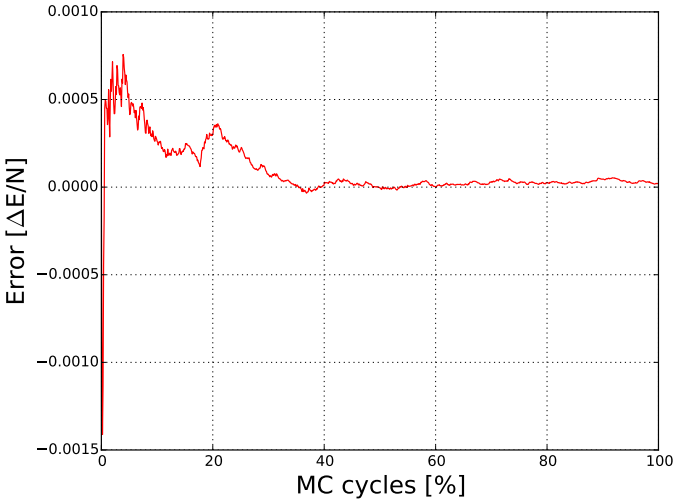
$$\begin{aligned}
\langle E \rangle / N &= -1.9959 & \langle |M| \rangle / N &= 0.9986 \\
C_V / N &= 0.03208 & \langle \chi \rangle / N &= 0.004010
\end{aligned}$$

5.2 Simulation 2x2

These simulations ran for 10^5 Monte Carlo cycles. All the simulations were done at T=1 and for a 2x2 grid. In general the left sub figure is the actual value plotted with the analytic answer and the right sub figure is the difference for analytic and simulated value. All the values converges at about 40% of the Monte Carlo cycles.

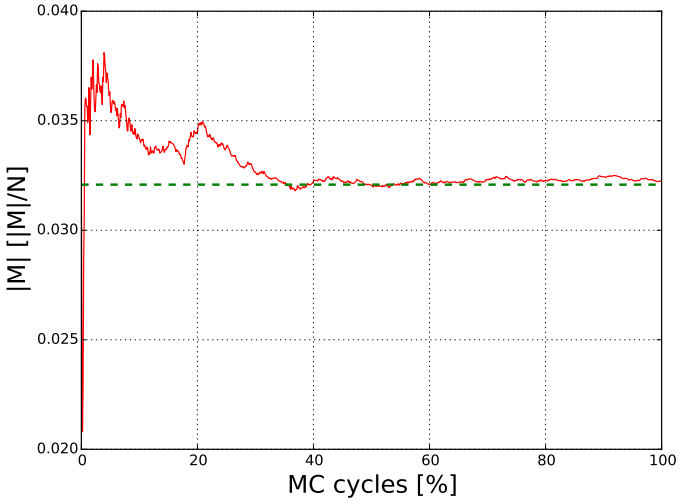


(a)

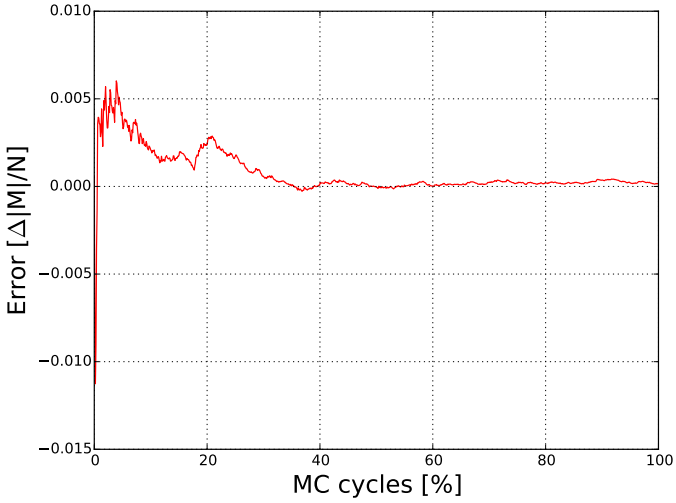


(b)

Figure 1: a) Shows how the expectation value for E varies versus Monte Carlo cycles. b) Shows how the error develops.

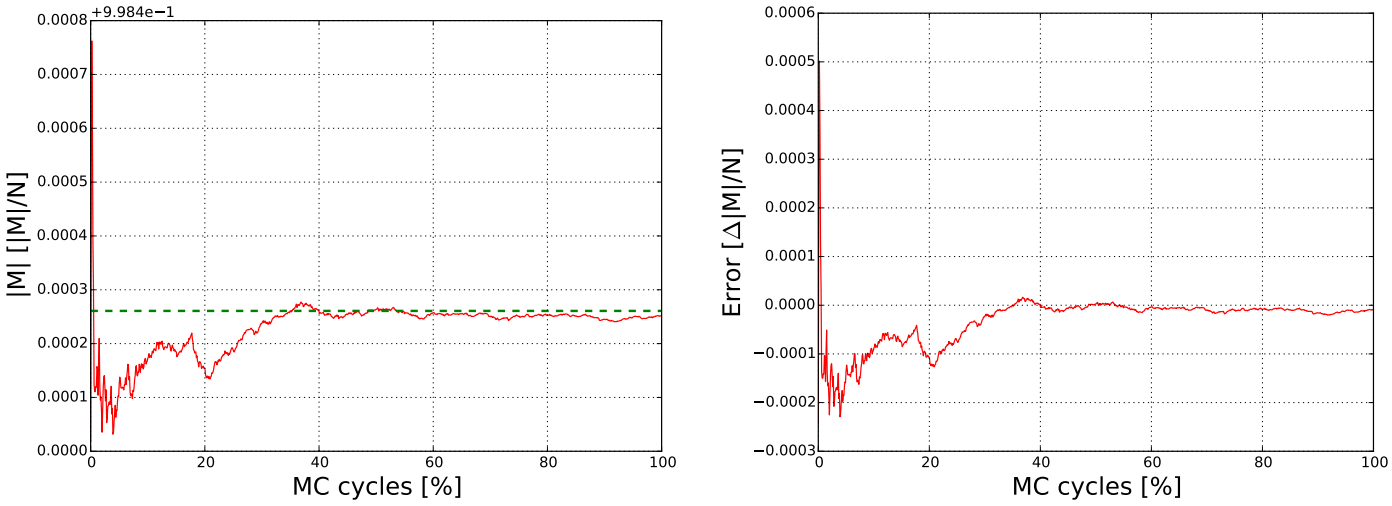


(a)



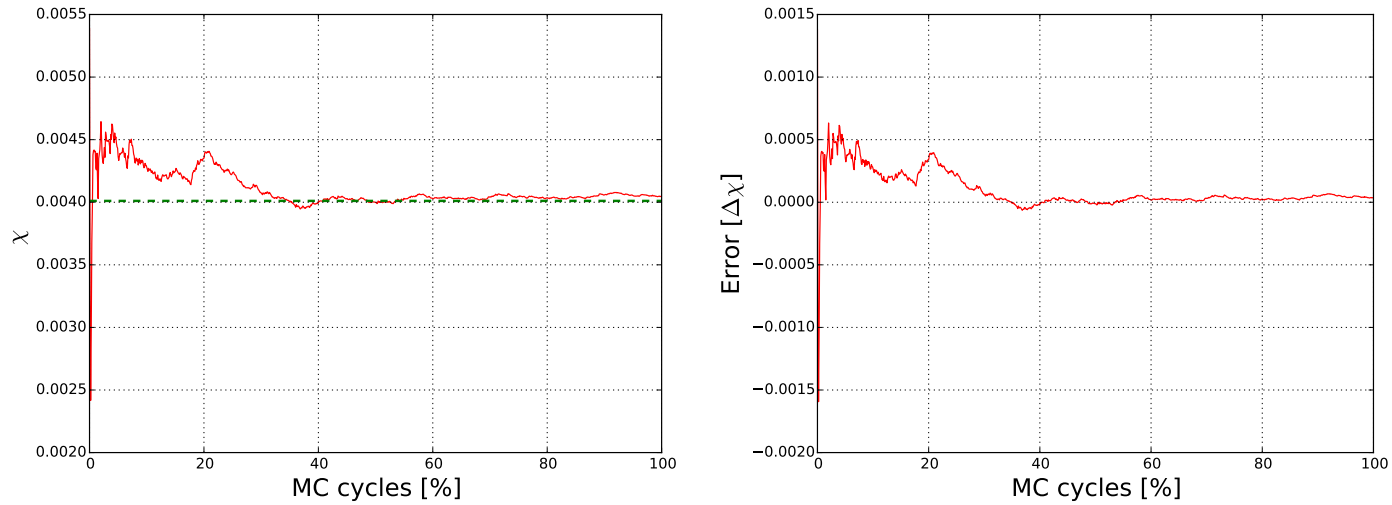
(b)

Figure 2: a) Shows how the expectation value for C_V varies versus Monte Carlo cycles. b) Shows how the error develops.



(a) (b)

Figure 3: a) Shows how the expectation value for |M| varies versus Monte Carlo cycles. b) Shows how the error develops.

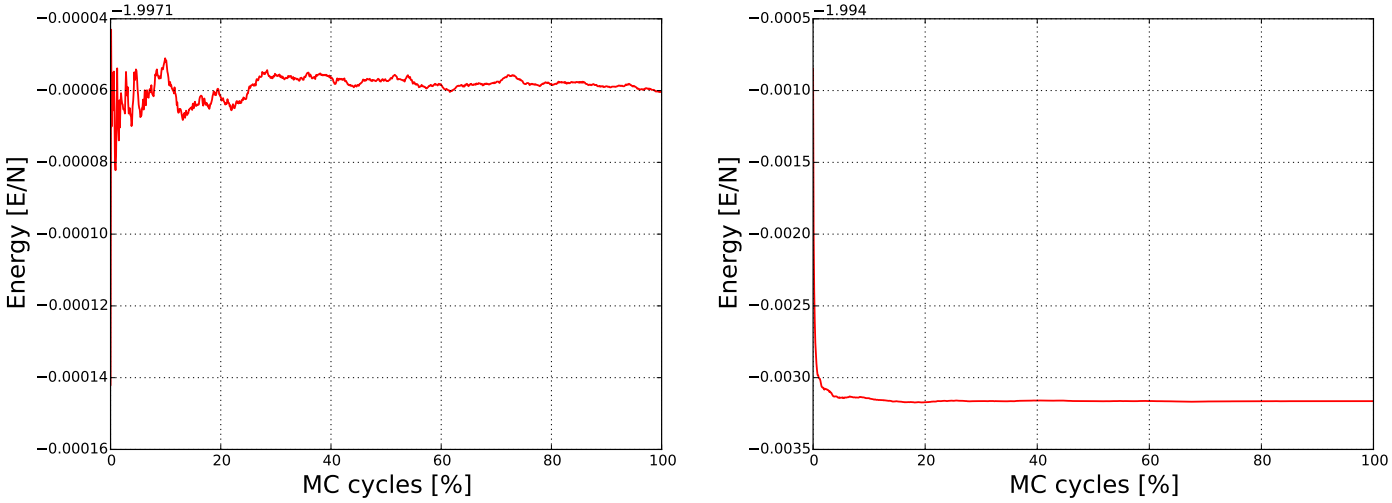


(a) (b)

Figure 4: a) Shows how the expectation value for χ varies versus Monte Carlo cycles. b) Shows how the error develops.

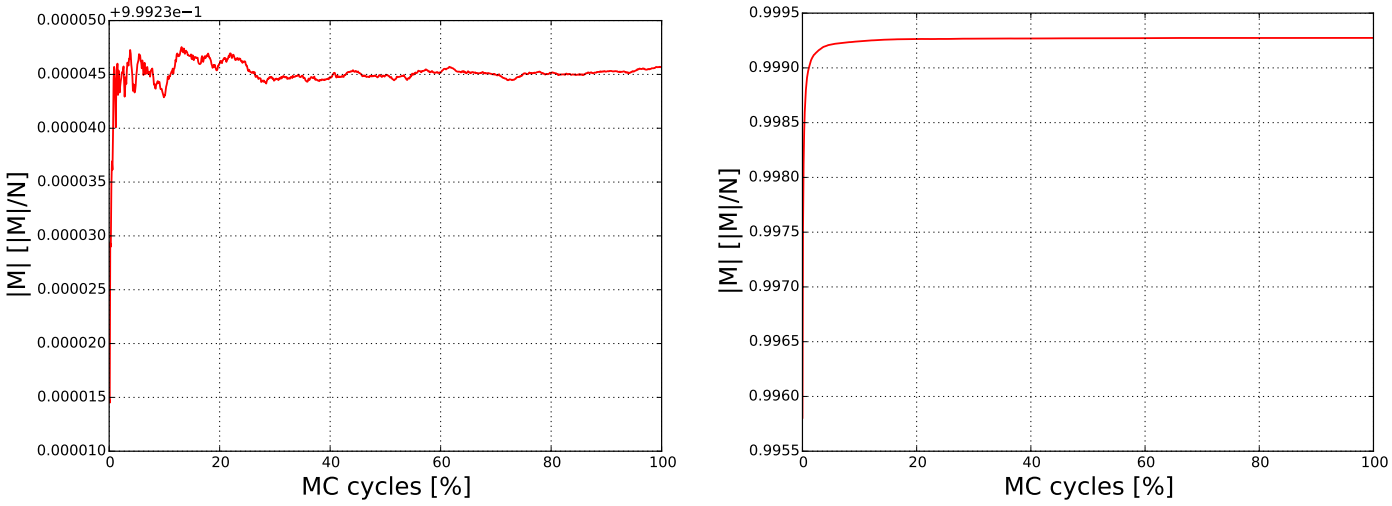
5.3 Convergence of 20x20

In this section convergence for a 20x20 grid is studied. All grids were simulated for 10 million Monte Carlo cycles. Unless something else is specified in the figure, the left figure is for a initial configuration of the spin in the same direction and the right figure is for a random configuration.



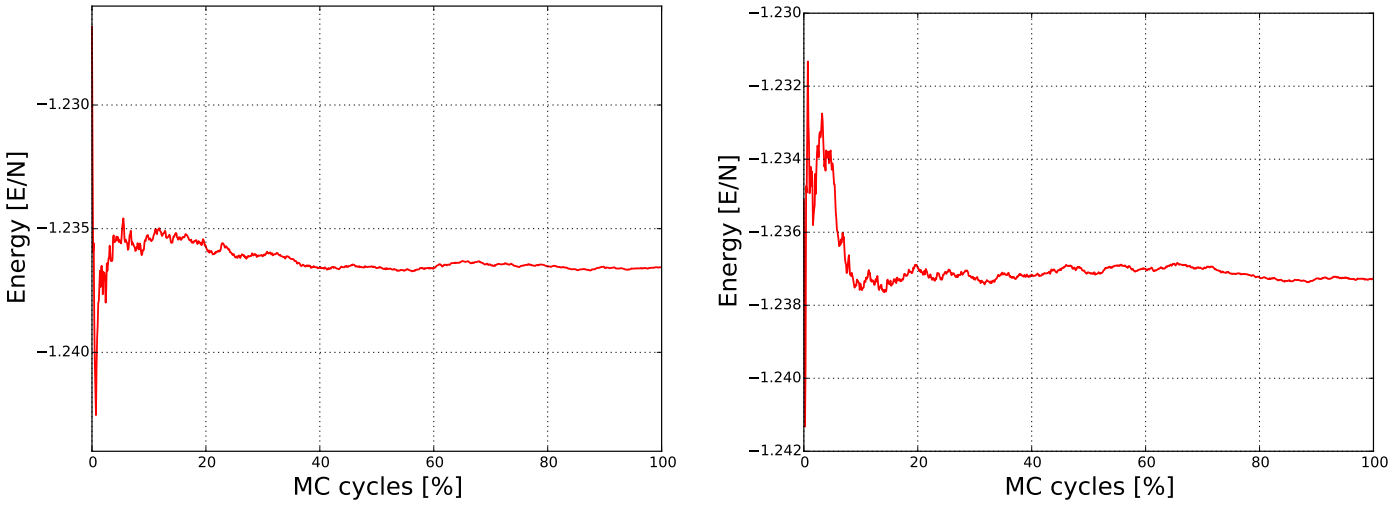
(a) (b)

Figure 5: The figure shows the development of E when T = 1.



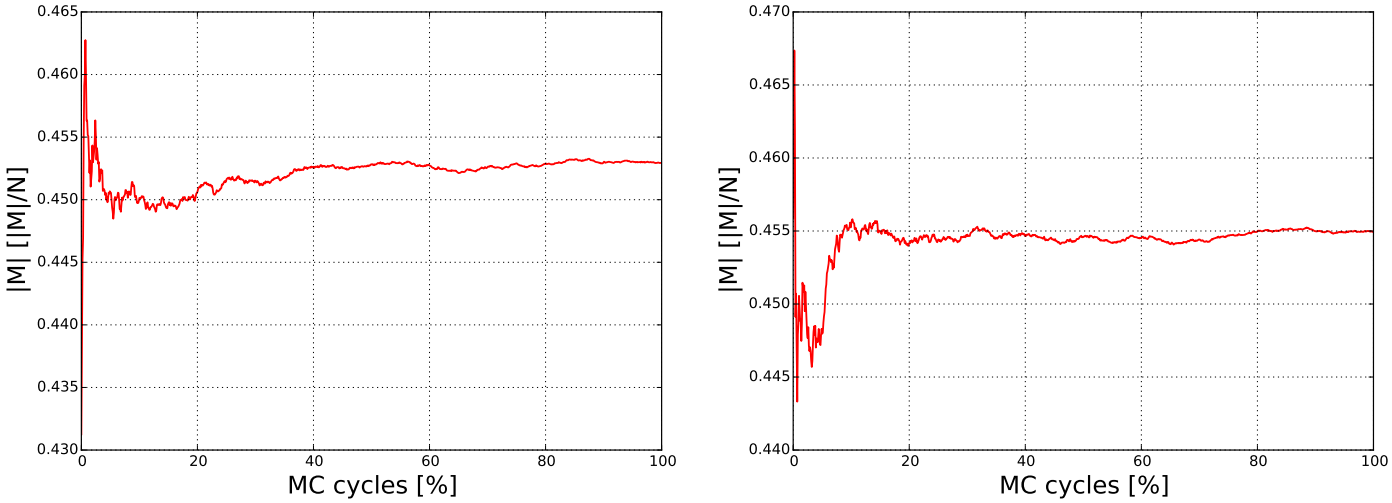
(a) (b)

Figure 6: The figure shows the development of |M| when T = 1.



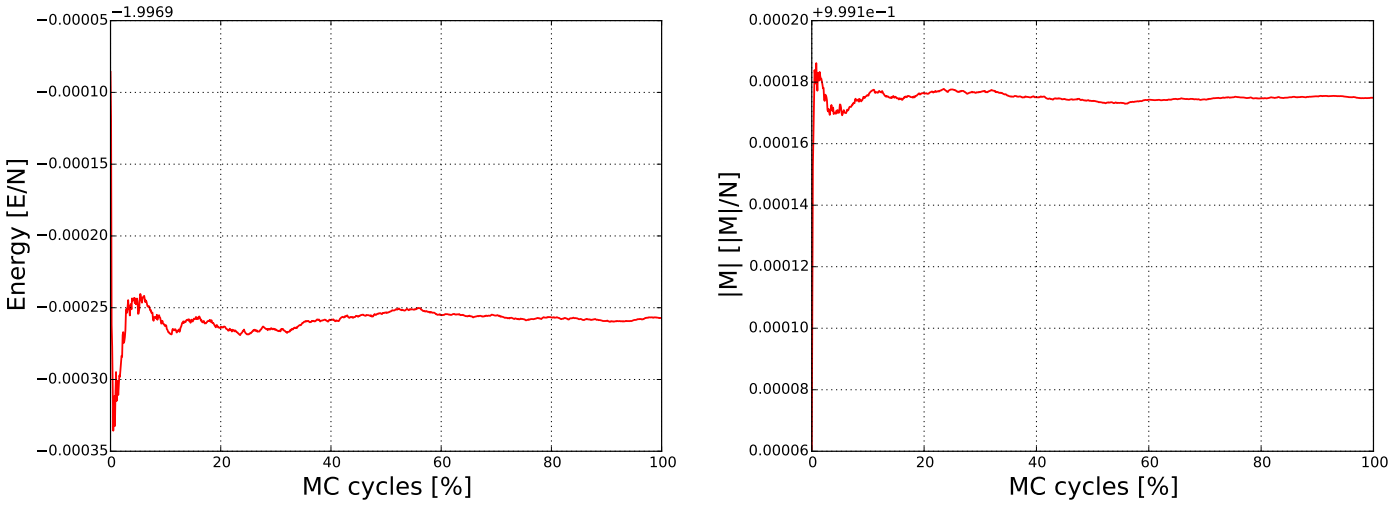
(a) (b)

Figure 7: The figure shows the development of E when T = 2.4 .



(a) (b)

Figure 8: The figure shows the development of |M| when T = 2.4.



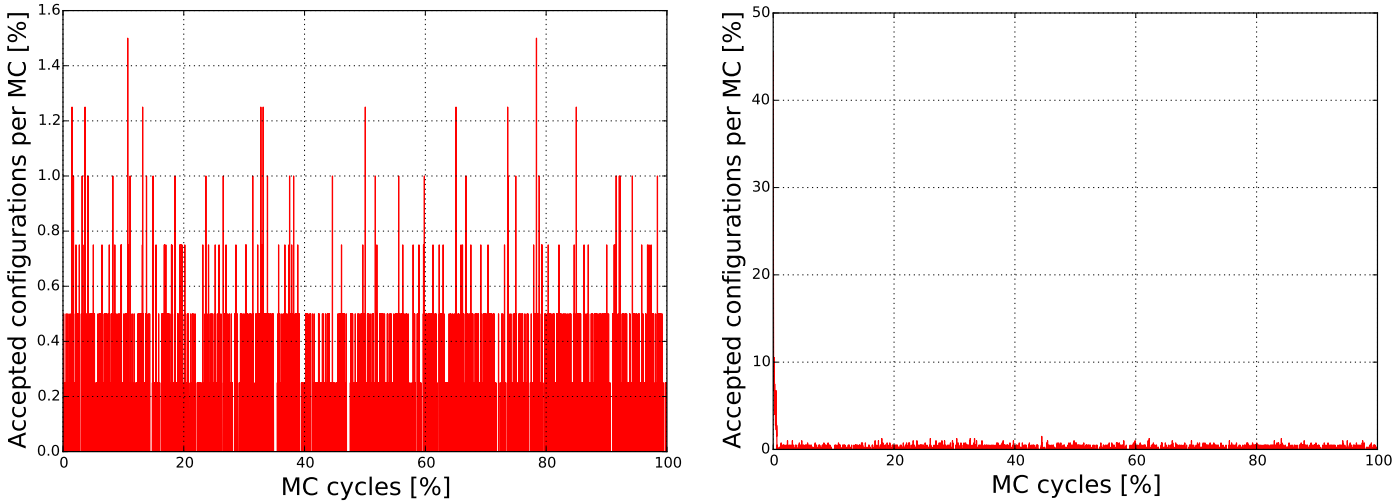
(a) (b)

Figure 9: The figure shows the development of E when T = 1 when we dont use the first 10^5 steps.

For most of the figures in this section, almost 40% of the MC seem to be used for obtaining equilibrium, but when removing one hundred thousand Monte Carlo cycles equilibrium seem to be obtained almost immediately. The first results are so bad that the system uses a longer time to achieve equilibrium, because the system needs to compensate for the first results. When removing this first part all the results that are used to calculate the expectation value is of good quality. For the rest of the report 10^5 Monte Carlo cycles will be used to obtain equilibrium. If you have a lot of computational power and time, one can use use 10^6 to obtain equilibrium.

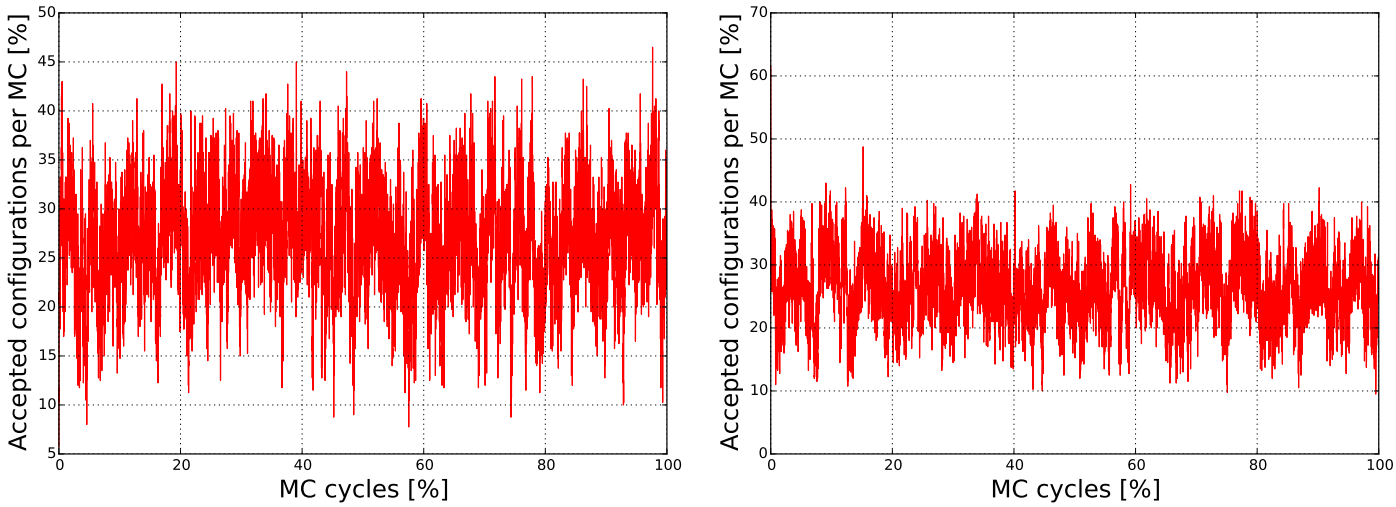
5.4 Accepted configurations

In this section acceptance of configurations for a 20x20 grid is studied. All grids were simulated for one million Monte Carlo cycles. Unless something else is specified in the figure, the left figure is for a initial configuration of the spin in the same direction and the right figure is for a random configuration.



(a) (b)

Figure 10: This figure shows how the acceptance of configurations develop over time. T = 1 and for N=10^6.



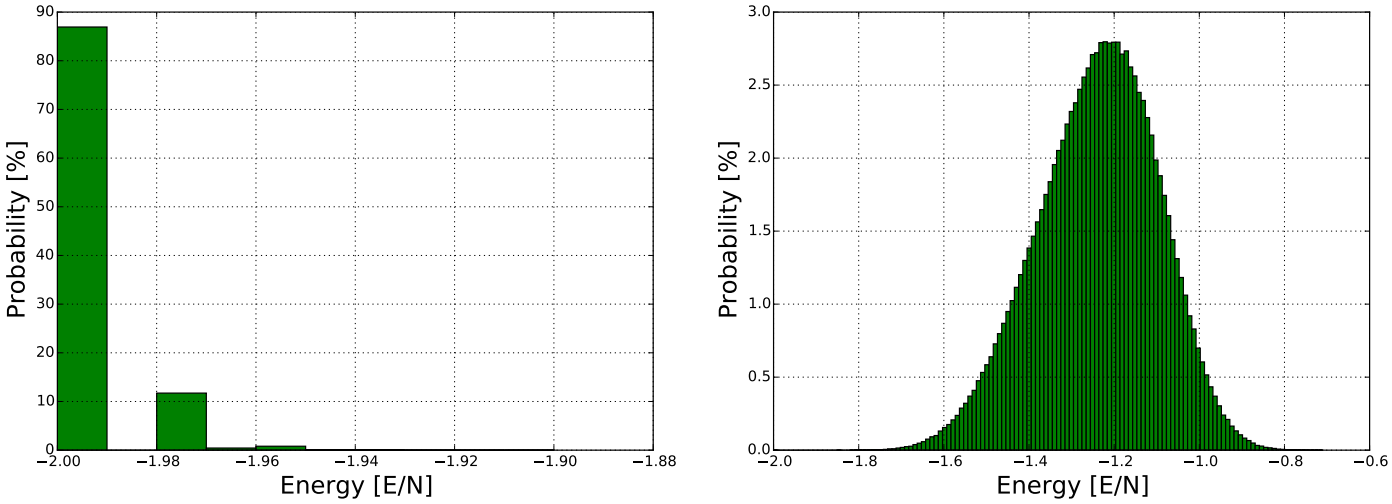
(a) (b)

Figure 11: This figure shows how the acceptance of configurations develop over time. T = 2.4 and for N=10^6.

There are two things worth noting. First the difference between is non-existent and second there is a huge increase in percentage of accepted configurations when we increase the temperature. This is because of the energy difference acceptance that was discussed in section 3.1. When we go higher in temperature, the probability for accepting a state becomes larger.

5.5 Probability distribution

This is not done



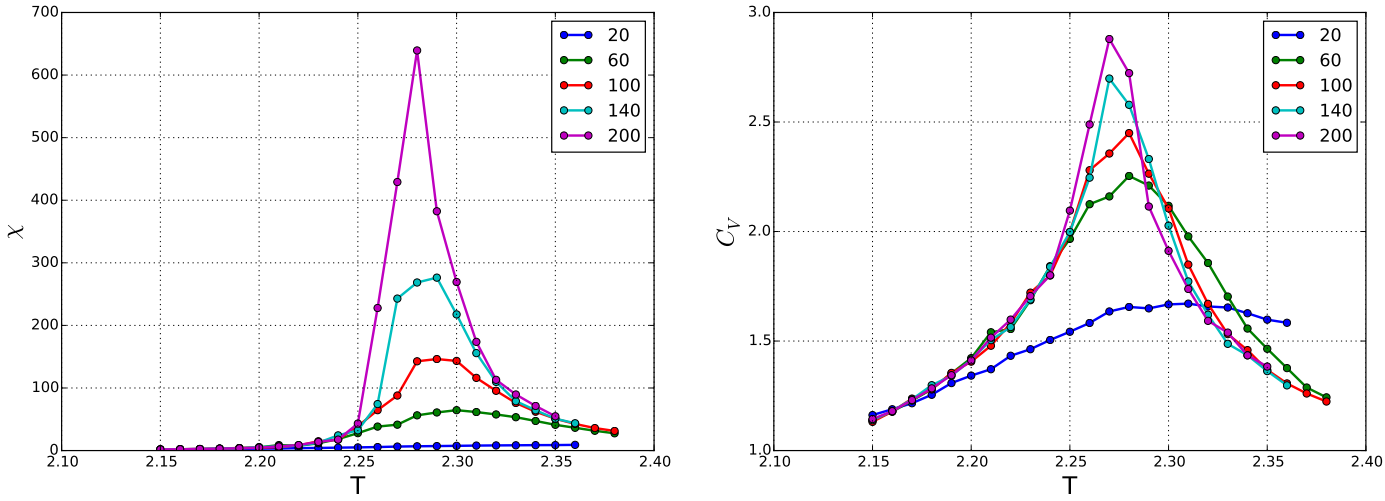
(a) (b)

Figure 12: a)

5.6 Phase transition

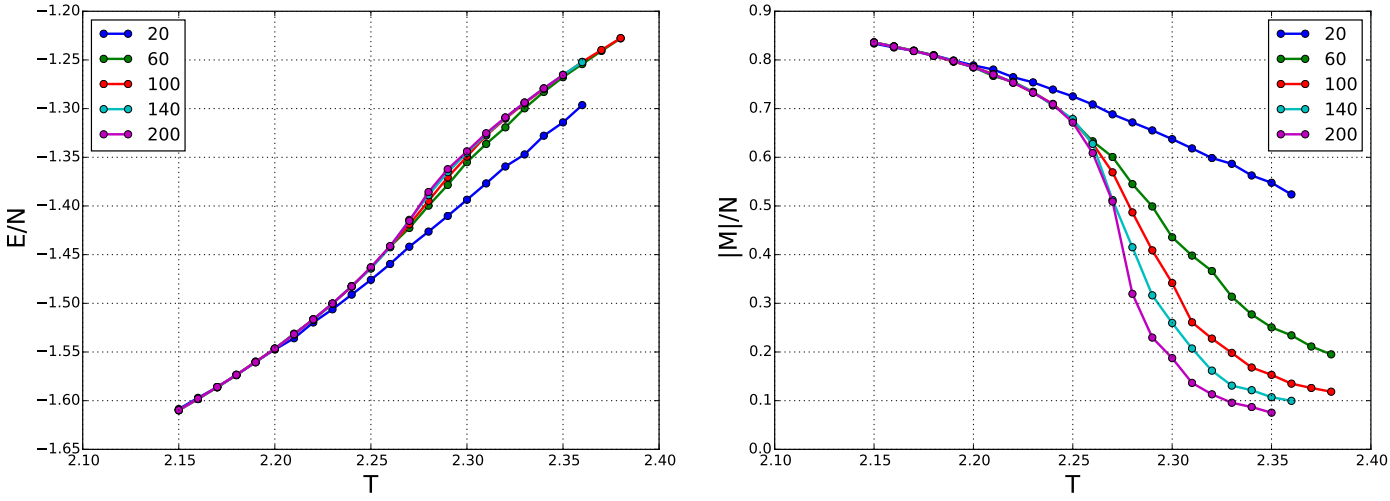
5.6.1 Numerical studies of phase transition

Several simulations were made with different L. Below a selected few of these simulations are shown. All the data and figures for this section are available at my github. The figures below were created with a python script.



(a) (b)

Figure 13: a)



(a) (b)

Figure 14: a)

5.6.2 Extracting the critical temperature

To extract a equation 7 is used. The result can be seen in table 5.6.2.

Table 5: The table shows how a differ for which L_i and L_j one uses. The values for T_C were picked from figure 13 a)

L_i	L_j	T_{C_i}	T_{C_j}	a
60	100	2.30	2.29	0.00025
60	140	2.30	2.28	0.00025
60	200	2.30	2.27	0.0002142
100	140	2.29	2.28	0.00025
100	200	2.29	2.27	0.0002
140	200	2.28	2.27	0.0001666

From the values for a above, we can extract the average value of a, $\overline{a}$. $\overline{a}$ is 0.0002218. $\overline{a}$ is used in equation 8. The results are in table 5.6.2.

Table 6: The table shows how a differ for which L_i and L_j one uses. The values for T_C were picked from figure 13 a)

L_i	$T_C(L_i)$	$T_C(\infty)$ with $\overline{a}$
60	2.30	2.2999
100	2.29	2.2899
140	2.28	2.2799
200	2.27	2.2699

An other way to calculate the critical temperature for an infinite lattice is to plot the finite T_C versus the 1/L. And then make a polynomial fit of first degree. The intersection with the y-axis is the critical temperature for a infinite lattice.

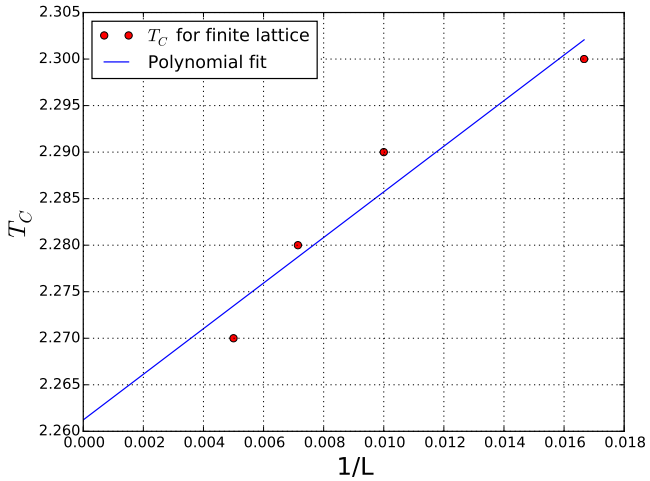


Figure 15: This figure shows the linear regression of the T_C from the finite lattices. From the graph one can read out the y-axis intersection, 2.261

6 Conclusion

test[3]

7 References

References

[1] Morten Hjorth-Jensen. *Computational Physics*. Lecture notes. 2015. URL: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf.

[2] Morten Hjorth-Jensen. *Computational Physics*. Project-4. 2017. URL: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects/2017/Project4/pdf/Project4.pdf.

[3] Lars Onsager. *Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition*. Phys. Rev. 65. 1944. URL: https://journals.aps.org/pr/abstract/10.1103/PhysRev.65.117.