A numeric approximation of phase transitions in a ferromagnetic system

A classical approach

Sakarias Frette
Anders Vestengen
William Hirst

November 24, 2020

In the following rapport we will model a ferromagnetic system using the Ising model in the classical case. We will use our model to examine the accuracy of our model as a function of Monte Carlo cycles and the effect temperature has on the evolution of the system. We also used our model to numerically calculate the critical temperature T_c and to explore the phase transition of ferromagnetic materials when exposed to increased temperature. We found that T_c was 2.265, a deviation of -0.004 from Onsager's analytical work and found that when exposing a ferromagnetic material to a temperature increase from 2.1 to 2.4, it undergoes a phase transition from a ferromagnetic material to a para-magnetic material, loosing the ability to self magnetize.

Contents

1 Introduction

In this paper we aim to study a phase transition in a magnetic lattice through numerical computation of the classical case, without quantum mechanics. To do this we are using the widely popular Ising model, because when set up for our use-case it exhibits a sudden phase-transition at a critical temperature. This is ideal as we are looking to simulate the sudden change of magnetization that happens when a magnetic structure is subjected to rising temperatures, causing the structural magnetic lattice to become unstable, and the unison magnetic field to be disrupted. Specifically we are looking to numerically replicate the work of Lars Onsager in 1944[4], when he discovered the critical temperature of a two dimensional lattice, using the Ising model.

2 Theory

2.1 Monte Carlo simulation and metropolis.

A Monte Carlo method is used to simulate a system when a variable is unknown (for example) due to a random interference, making a calculation on the system impossible or very hard. The method is to use many random values to replace the unknown value, and use the average of all the calculations as an estimate. Given enough samples (random numbers) the estimate improves. If we wish to integrate a function $f(x)$, we do as follows

$$I = \int_a^b f(x)\,dx \approx \sum_i^N \omega_i f(x_i) \tag{1}$$

where x_i is a random number, ω_i is a weight and N is the number of samples. Now, for cumulative probability distribution we have that the probability for a stochastic variable X is less than x can be written as

$$P(X \leq x) = \int_{-\infty}^{x} p(x')\,dx' \tag{2}$$

where $p(x')$ is a probability distribution function. This PDF has to be normalized, and so we require them to follow this relation:

$$\sum_{x_i \in} p(x_i) = 1 \tag{3}$$

From this we can introduce the expectation value for a given function as

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i) p(x_i) \tag{4}$$

where $x \in [0,1]$. We can then approximate the integral in equation 1 to the equation above, i.e

$$I = \int_a^b f(x)\,dx \approx \langle f \rangle \tag{5}$$

We also have the variance, defined as

$$\sigma_f^2 = \langle f^2 \rangle - \langle f \rangle^2 \tag{6}$$

which is the amount f deviates from its mean. We are however mostly interested in the final mean value and the associated standard deviation, i.e square root of the associated variance, given as

$$\sigma_N^2 = \frac{\sigma_f^2}{N} \tag{7}$$

With these equations we can use the Monte Carlo algorithm[2]:

- Choose the number of Monte Carlo samples N.
- Perform a loop over N and for each step generate a a random number xi in the interval [0,1] through a call to a random number generator.
- Use this number to evaluate f(xi).
- Evaluate the contributions to the mean value and the standard deviation for each loop.
- After N samples calculate the final mean value and the standard deviation.

The Metropolis algorithm is a type of Monte Carlo simulation. The algorithm is used when the probability distribution, P for a state of a system, S (which depends on many elements), is unknown. The idea is to find the states that would be generated if P was known (S_P) by using a Markov process. The process is to randomly chose an element in S and decide to change the state of that element dependent on a rate of acceptance, the previous state(hence Markov process) and an element of randomness. Given that certain requirements of A (like being proportional to P) and P are satisfied, if enough changes are made to the original state S, the state should converge towards S_P.

One such requirement for P that is critical to our calculations, is the fact that any change from a state s_i to a state s_j given by P is reversible. This gives us the following equation:

$$\frac{P_{s_i \to s_j}}{P_{s_j \to s_i}} = \frac{A_{s_i \to s_j}}{A_{s_j \to s_i}} = \frac{P(s_i)}{P(s_j)} = e^{-\beta(E_i - E_j)} \tag{8}$$

,where $P_{s_i \to s_j}$ is the probability for a change from s_i to s_j and is unknown, $A_{s_i \to s_j}$ is the rate of acceptance from s_i to s_j and E_i and E_j are the energy's of s_i and s_j. The first

equality comes from the fact that A is proportional to P, and the final equality comes from statistical physics (2.4). From the equation above we can see that:

$$\frac{A_{s_i \to s_j}}{A_{s_j \to s_i}} = e^{-\beta(E_i - E_j)} \tag{9}$$

which tells us that the rate of acceptance for a change from one state to another, is given by the difference in the energy of the states. Therefore, even though we cannot know the probability distribution for a change in the state of the magnet, we can use the metropolis algorithm to simulate and find a realistic evolution of states (or at least good approximations of them) for the magnet as a hole given enough cycles.

2.2 The Ising Model

Is a mathematical model for studying ferromagnetic properties in statistical mechanics[5]. We will be focusing on the two dimensional Ising model in this rapport. Consider a structured lattice of units like in figure 1. For our use of the model we pretend these units are tiny spins of magnetic moment which can have the value $+1$ or -1. Consider that at any moment in time one of these units may flip its spin, depending on several factors. In this case the energy of the surrounding spins also changes and this may cause other spins to flip as well.

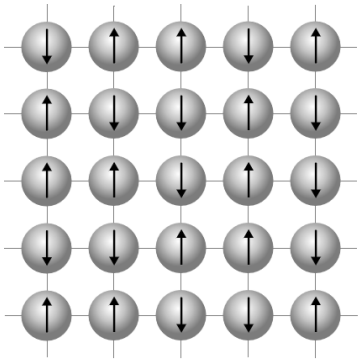


Figure 1: Two dimensional Ising model Source

In the Ising model the energy of the lattice is given by the spins of each unit and how it interacts with its neighbors and a inter-spin exchange coupling constant J. The energy is given as:

$$E = -J \sum_{<kl>}^{N} s_k s_l \tag{10}$$

, where s_k and s_l are two spin-values and the "kl" indicates we only sum over nearest neighbours to avoid duplicating energy. We then sum for number of spins (N). This

also highlights a few things which we will be discussing later on when commenting on the code, such as the fact that a two dimensional model only has a limited number of energies, which can be pre-calculated. That's also were we're going to discuss the boundary conditions of our model, an important part of simulation because the number of micro-states in a system like this grows exponentially when we increase the lattice size, and for our computers sake it must be limited. The consequences of this is that our lattice suddenly stops and we must deal with the boundaries that arise. [1]

In our model we set the inter-spin exchange coupling [J] (eq:10) to 1. This determines how much a spin is likely to shift based on the spins of its neighbours, and also determines the critical temperature[5]. If left undisturbed, this lattice of spins tends towards an ordered symmetry, because it will then have lower energy. However, this behaviour is altered with changing temperatures and is the basis for how the Ising model can predict phase-changes in systems like our magnetic structure.

$$M = \sum_{j=1}^{N} s_j \tag{11}$$

For Magnetization we simply iterate over the total number of spins. We don't have to take duplicates into account like we did for energy, because magnetization is an absolute value, and opposites will cancel each others contributions.

In this model, a phase transition happens when the temperature approaches the critical temperature T_c. In the Ising model, the critical temperature for a lattice of given size L^2 follows the following relation

$$T_c^{(L)} - T_c^{(L=\infty)} = aL^{-1/\nu} \tag{12}$$

, where a is a unknown constant, ν is another constant and the indexing on T_c represents how the critical temperature varies with the size of the lattice[3].

2.3 Random numbers in numerical calculations

A very important part of studying the Ising model using Monte Carlo methods, is the use of a PRNG or pseudo random number generator. We do this intentionally because PRNG's have several characteristics which are useful to us. First of all they are deterministic. This seems counter-intuitive, but it's actually very useful because it makes our results reproducible even though they employ random numbers. The other major factor of an PRNG is speed. Since they come from an algorithm and not a hardware implementation, it is much faster. And this is key when we have up to hundreds of millions of Metropolis tests, all of whom use several function calls to our PRNG every cycle.

[1] see Here for a more thorough discussion of the energy configurations. p.422

Specifically, our model uses the mt19937.64-bit PRNG[2], which is an implementation of the 'mersienne twister', an algorithm which has over $2^{19937} - 1$ different combinations and 'randomly' generates a number between 0 and 1. It also has very good uniformity, something which is important to the validity of our Monte Carlo strategy. We will come back to this later in the paper when discussing our results.

2.4 Statistical physics

In the canonical ensemble[3] we can calculate the expectation value for energy, $\langle E \rangle$, but we need a probability distribution in order to do so. This is given by the Boltzmann distribution as

$$P_i(\beta) = \frac{1}{Z} e^{-\beta E_i} \tag{13}$$

where $\beta = \frac{1}{k_B T}$ is the inverse temperature in units of joules, E_i is a given energy for a microstate and Z is the partition function, given as

$$Z = \sum_{i=1}^{M} e^{-\beta E_i} \tag{14}$$

From the thermodynamic identities we have Heimholtz free energy given as

$$F = -k_B T \ln Z = \langle E \rangle - T \left(k_B \ln Z + k_B T \left(\frac{\partial \ln Z}{\partial T} \right)_{N,V} \right) \tag{15}$$

This gives us the expectation value $< E >$ as

$$\langle E \rangle = \sum_{i=1}^{M} E_i P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{M} E_i e^{-\beta E_i} \tag{16}$$

This allows us to calculate the variance of the energy as

$$\sigma_E^2 = \langle E^2 \rangle - \langle E \rangle^2 = \frac{1}{Z} \sum_{i=1}^{M} E_i^2 e^{-\beta E_i} - \left(\frac{1}{Z} \sum_{i=1}^{M} E_i e^{-\beta E_i} \right)^2 \tag{17}$$

By the same token we find that the expectation value for the absolute value of the magnetization is given by

$$\langle |M_i| \rangle = \frac{1}{Z} \sum_{i=1}^{M} |M_i| e^{-\beta E_i} \tag{18}$$

and the variance of magnetization as

$$\sigma_M^2 = \langle M^2 \rangle - \langle M \rangle^2 = \frac{1}{Z} \sum_{i=1}^{M} M_i^2 e^{-\beta E_i} - \left(\frac{1}{Z} \sum_{i=1}^{M} |M_i| e^{-\beta E_i} \right)^2 \tag{19}$$

[2] C++ documentation

This gives us the following quantities heat capacity C_v and susceptibility χ. They are given as

$$C_v = \frac{\sigma_E^2}{k_B T^2} \tag{20}$$

and

$$\chi = \frac{\sigma_M^2}{k_B T^2} \tag{21}$$

2.5 Analytical solution for 2X2 lattice

From (2.4) we found analytical expressions for the mean energy and the absolute value of the magnetization, the heat capacity and the susceptibility of the lattice. For a 2x2 lattice this expressions can be found analytically. To do so we have to look at the different combinations of spin to find the different energy levels we can achieve. Since there are 4 units and two possible states the amount of micro states are $4^2 = 16$. Lets use an example to illustrate how we find the analytical expressions.

Lets imagine all we know is that three of the units are spin up. That means there are 4 possible ways the lattice could be organised. That means the degeneracy of the system is 4. If there are three up (s=1) and one down (-1) the magnetization can be calculated by (11) $\rightarrow$ M =1+1+1-1=2. Similarly we can use (10) to calculate the energy $\rightarrow$ E = -2+(1-1)+(1-1)+2=0. By repeating this process for all different combinations of spin we find the table:

Number spins up	Degeneracy	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

Figure 2: Table of possible Energy and magnetization values for 2x2 lattice with corresponding degeneracy [1].

The table shows all possible energy's and magnetization values possible for a 2x2 lattice. Using the table and equations presented in (2.4) we can find the values we are looking for. Lets start with the partition function.
From the table we can see that the lattice can have three different energy's; 8J,0 and

-8J. This results in the following partition function:

$$Z = 12 + 2e^{-\beta 8J} + 2e^{\beta 8J} \tag{22}$$

, where the coefficients (12,2 and 2) are given by the degeneracy's.

Now that Z is known, we can calculate $\langle E \rangle$. (16) and the table gives the following expression for the mean value of E:

$$\langle E \rangle = \frac{16e^{-\beta 8J} - 16e^{\beta 8J}}{Z} \tag{23}$$

To find the heat capacity, C_v we need to use (20), our expression for $\langle E \rangle$ and the table:

$$C_v = \frac{2(8 \cdot 8)^{-\beta 8J} + 2(8 \cdot 8)e^{\beta 8J}}{Z} - \left(\frac{16e^{-\beta 8J} - 16e^{\beta 8J}}{Z} \right)^2 \tag{24}$$

Finally we can repeat the process for the absolute value of the magnetization and susceptibility using the table, (18) and (21):

$$|M_i| = \frac{16e^{\beta 8J} + 16}{Z} \tag{25}$$

and

$$\chi = \frac{2(4 \cdot 4)e^{\beta 8J} + 8(2 \cdot 2)}{Z} - \left(\frac{16e^{\beta 8J} + 16}{Z} \right)^2 \tag{26}$$

2.6 Energy levels

In the general two-dimensional Ising model there are only five different energy levels. This is a small but significant part of the theory because it allows us to pre-calculate our energy levels so we don't have to do so in the Metropolis test. This saves computation time and as we shall see in our section on the results of the critical temperature, any savings in compute-time is warranted. Now let's get to why the energy levels are finite.

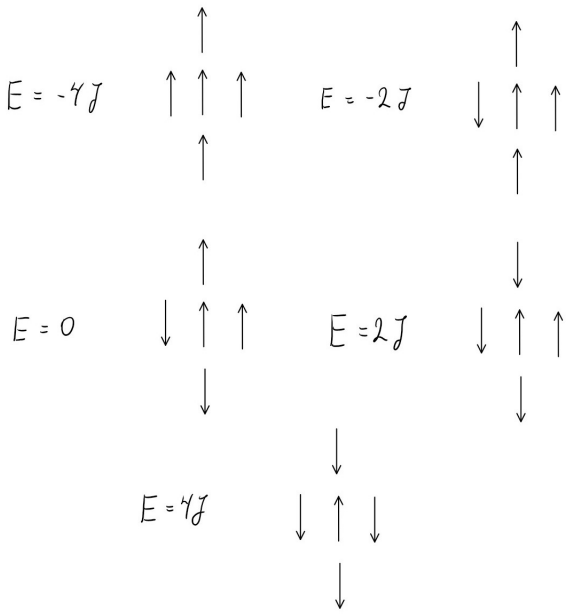


Figure 3: Possible energy levels in the two-dimensional Ising model

Initially you may assume the energy levels to be quite diverse but the way the Ising model is set up (figure 3) one spin can only have four adjacent neighbours. And since the energy is computed based on the integer value of the single neighbouring spins (atleast in our case). The model only reaches the five different values listed above. While there are four more configurations in a stricter sense, the energy levels of these are the same, with reverse polarisation.

3 Implementation

The programs used in this research article can be found on this Github address.

3.1 The code

The code is structured to have one large implementation, to which there are several smaller configurations able to run for specific use-cases. We find this kind of development advantageous because it let's us develop faster towards a general goal. We also acknowledge this leads to some rewrites when finalizing, but feel the early speed of implementation makes up for it. We also found that parallelizing the code halved our computation time. We discuss this more in the README file.

3.2 Boundary conditions

As discussed in the section on the Ising model, our computational limitations means that our lattice has a discrete size, and when iterating over this lattice with our Metropo-

lis test, we need to make sure that we don't go outside the bounds of our lattice. We will explain our algorithm for boundary detection, but first we have to introduce what we like to call 'ghosting'. Since the lattice is supposed to be infinitely adjacent. Like in a continuous substance, the spins always have a neighbour. We replicate this by making it look like the ends come around again. Like if we're able to take our flat sheet of lattice and fold it around itself to where the ends meet each-other. This way every time our Metropolis test picks a spin that has less than four adjacent neighbours, it will pick the spin opposite in the lattice to the one in question. This holds for all spins at the edge, and makes the lattice more like playing 'snake' where the player could exit the right side of the screen and appear on the left.
Our Boundary algorithm looks like this:

Algorithm 1: Boundary condition

input: Index i, Spin Matrix length L, Next or previous neighbour -1 or 1
procedure BOUNDARY CONDITION(i, L, Neighbour);
return: (i + L + neighbour) MOD (L);

The algorithm is used to inspect if the index we are about to use (i+neighbour) is inside the lattice, and if not it returns the index on the opposite pole of the lattice(1 or -1). The algorithm works because if the index is smaller than the limit, the Mod L will divide fully once and return the index. But, if the index is larger than the limit, the Mod (L) vill divide twice returning only the neighbour value (either 0 or -1). This is in our opinion an elegant solution to the problem, and means we don't have to employ if-tests, which would add to total FLOPS per cycle and increase our already lengthy computation time.

3.3 Implementing the Metropolis Algorithm

We know from empirical testing that the magnet will evolve towards a steady state, we call this state the MLS (Most likely state). In most cases that will be a state with lowest energy, at least if the original state was a stimulated one. Therefore our algorithm needs to be defined in such a way that if we change the spin of one (randomly chosen) particle so the energy decreases, the spin flips. But, since we know that not all systems evolve to the lowest energy there needs also to be a probability of the spin flipping even if the change in energy is positive.

If the spin flips or not is dependent on the rate of acceptance and a random number generated by our RNG. If the the random number is smaller than our A, the particle flips. From A we can see that if the difference in energy from the flips is negative (energy decreases), A is always bigger than our random number and the spin always flips. If the change in energy is positive, it is dependent on the random number, the change in energy and of T. We can now see that the Metropolis algorithm with our chosen A, well models the situation described above.

3.4 The algorithm and Monte Carlo cycles

The main idea behind the code is explained in (3.3). This section explains how we use the metropolis algorithm to evolve the state of the matrix from one time step to another. In our case one time-step is described as a full cycle of the metropolis algorithm N amount of times, where N is the amount of elements in the matrix. We call the cycle for Monte Carlo cycles (MCc). Since most of the theory's behind our calculations are based on statistics, we know that the validity of our results are dependent of the amount of Monte Carlo cycles.

One Monte Carlo cycles can therefore be explained with the following algorithm:

Algorithm 2: Metropolis algorithm

input: Spin Matrix of size $L * L$

special treatment of the first line;
for $spin = 0$ **to** $TotalSpins$ **do**
    Find random element in spin matrix;
    Calculate energy difference in case where spin is flipped;
    Find rate of acceptance;
    Check if random number satisfies rate of acceptance;
    If yes, flip spin;
    If yes, update Energy and Magnetization of spin matrix;
    If no, leave matrix unchanged;

Since the MCc represents time, it can be interesting to study our matrix development as a function of time. Therefore we measure the energy and magnetization of the matrix after each cycle. This manifests as the following

Algorithm 3: Monte Carlo Strategy

input: Spin Matrix of size $L * L$

Set up array for average values;
for $cycle = 1$ **to** $Maxcycle$ **do**
    Update Matrix using Metropolis;
    Find $\langle E \rangle$ *average*;
    Find $\langle E \rangle$ *average*;
    Find $\langle M \rangle$ *average*;
    Find $\langle M^2 \rangle$ *average*;
Update total energy average;
Update total energy standard deviation;
Update total magnetization average;
Update total magnetization standard deviation;

3.5 Unit tests

In order to verify that our code works properly, we run unit tests. We chose three tests to ensure the validity of our results. Firstly we check if the spins are randomly ordered when we chose them to be, meaning that the pseudorandom algorithm actually randomized the spins in the matrix. Because the distribution of spin up and spin down evens out more and more as we increase the amount of spins, we check if the average sum for a given lattice of 10 by 10 is larger than the average sum for a given lattice of 20x20. We are to expect the average sum to go zero as L^2, the amount of spins in a lattice, goes to infinity. We also test this by checking if the average spin in the randomly ordered 20x20 lattice is smaller than a certain threshold. We chose this threshold to be 0.0001.

The second test checked if the energy variance increased with an increased temperature. From equation 17 we have that β decreases with an increase in temperature, which in turn increases the overall expectation value. This in turn increases the variance. The physical interpretation of this is quite simple. By increasing the temperature, we are adding energy to the system, which in turn increases the amount of microstates, and so we would then expect that the probability for a given energy would get more wide, as the standard deviation, the square root of the variance, gets larger.

The third and final test compares the numerical and analytical values for the expectation value of the energy and magnetization up to a certain threshold. The analytical expressions are given by equation 16 and 18, and we put a threshold of one thousandth of the analytical value.

4 Results

4.1 Analytical vs Numerical for a 2x2 lattice

Using our analytical expression found in (2.5), setting T=1kJ/T we found the following analytical results:

	$\langle E \rangle$	$\langle \lvert M \rvert \rangle$	C_v	χ
Analytical values	-7.98	3.99	0.13	0.016

Table 1: The table shows the analytical values for the mean energy and magnetization, the heat capacity and the susceptibility. The values are calculated for a 2x2 lattice and T =1kT/J .

Using our numerical model we found, using the same parameters, the following

results:

Monte Carlo cycles	$\langle E \rangle$	$\langle \lvert M \rvert \rangle$	C_v	χ
10^4	-7.98	3.99	0.14	0.014
10^5	-7.98	3.99	0.12	0.013
10^6	-7.98	3.99	0.12	0.014
10^7	-7.98	3.99	0.13	0.016

Table 2: The table shows the numerical values for the mean energy and magnetization, the heat capacity and the susceptibility. The values are calculated for T =1kT/J and L=2 and number of Monte Carlo cylces ranging from $10^4 \rightarrow 10^7$.

The table shows the mean energy and magnetization, the heat capacity and the susceptibility of the matrix per cycle. The values were calculated with the lattice starting in a random configuration of spins. After 10^7 MCc the numerical values match all the analytical up to at least two leading digits.

4.2 Numerical analyses of 20x20 lattice

For all plots in the following section there were done two calculations; one where the spins were ordered with all spins pointing up and one where spins were ordered randomly.

4.2.1 T=1 [kT/J]

When running 10000 MCc on a 20x20 lattice with T=1kT/J, the resulting plot of the mean energy is the follwoing:

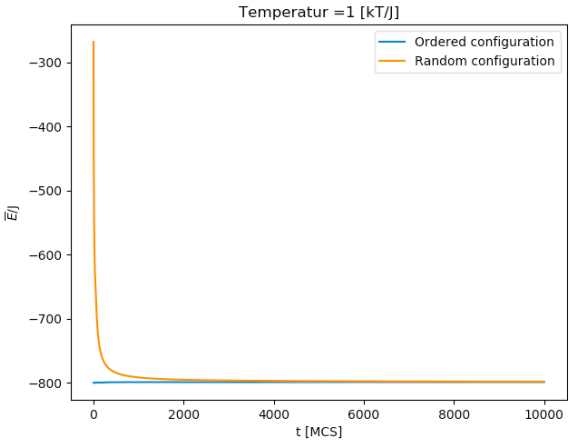


Figure 4: The figure shows the mean Energy as a function of time/Monte Carlo cycles. The curves were generated with L=20 and kT/J=1.The blue curve shows ordered orientation(all up) as starting configuration and orange shows random.

The plot shows the mean energy for the lattice as a function of time (MCc). The plot shows that for a ordered configuration the mean energy starts very close to the MLS, and even though the random configuration starts far away from the MLS it quickly converges to a stable state. The energy seems to reach equilibrium after 2000 Mcc.

For the same system as described above we found the mean of the absolute magnetization as a function of time (MCc):

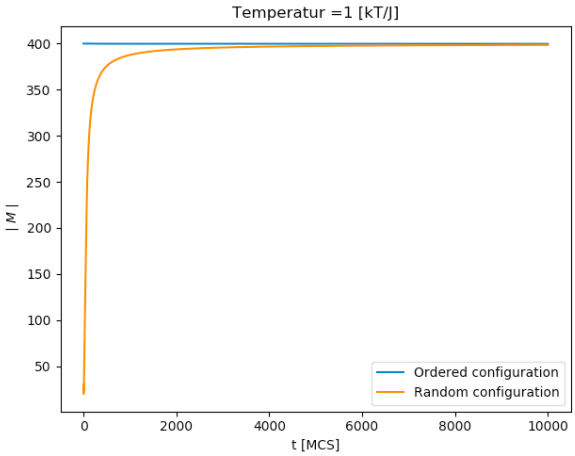


Figure 5: The figure shows the absolute value of the mean magnetization as a function of time/Monte Carlo cycles. The curves were generated with L=20 and kT/J=1. The blue curve shows ordered orientation(all up) as starting configuration and orange shows random.

The plot shows above shows the same trend as the plot for the mean Energy.
Finally we also calculated the total number of accepted configurations(changes in spin) as a function of MCc. The result is visualized in the figure bellow.

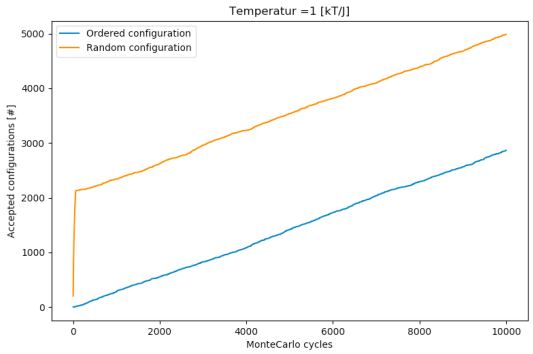


Figure 6: The figure shows the number of accepted configurations as a function of Monte Carlo cyles. Both of the curves were made with T=1 and L=20. The blue curve shows ordered orientation(all up) as starting configuration and orange shows random.

The plot above shows that the ordered configuration immediately reaches a steady increase in accepted configurations, where as the random configuration spends some cycles reaching the same steady increase.

4.2.2 T=2.4 [kT/J]

Again we run 10000 MCc on a 20x20 lattice, but this time we increase the temperature to 2.4kT/J. The resulting plot of the mean energy as a function of MCc is:

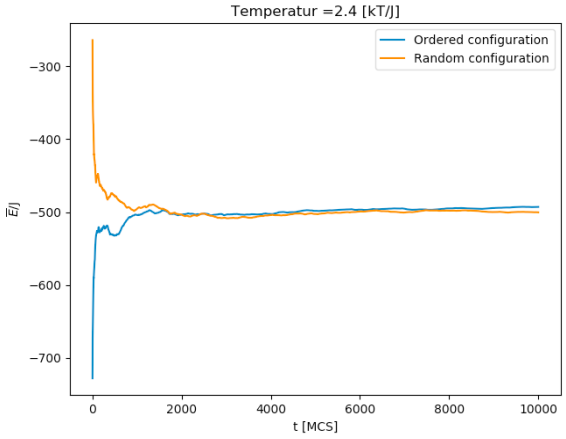


Figure 7: The figure shows the mean Energy as a function of time/Monte Carlo cycles. The curves were generated with L=20 and kT/J=2.4.The blue curve shows ordered orientation(all up) as starting configuration and orange shows random.

This time it is clear that both the ordered and the random configuration uses a significant time reaching a steady state. It is also a much larger fluctuation in both curves. Again the energy seems to reach equilibrium after 2000 Mcc.
Using the same parameters we plot the absolute value of the magnetization as a function of MCc:

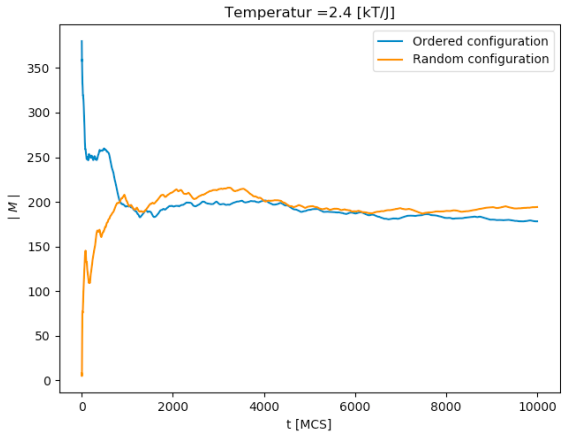


Figure 8: The figure shows the absolute value of the mean magnetization as a function of time/Monte Carlo cycles. The curves were generated with L=20 and kT/J=1. The blue curve shows ordered orientation(all up) as starting configuration and orange shows random.

Also here there both configuration spends many cycles stabilizing towards a steady state. Fluctuations are also much larger leading to none of the curves reaching complete stability.
Lastly we calculate the total number of accepted configurations as a function of MCCc:

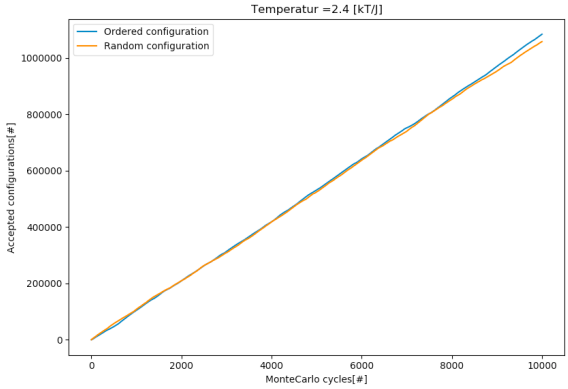


Figure 9: The figure shows the number of accepted configurations as a function of Monte Carlo cyles. Both of the curves were made with T=2.4 and L=20. The blue curve shows ordered orientation(all up) as starting configuration and orange shows random.

This time none of the curves show any significant time spent stabilizing. Both of the curves seem to reach a steady increase in accepted configurations after very few cycles.

4.3 Finding the probability distribution for the energy

We are interested in finding the probability distribution for the energy model for 10^5 MCc on a 20x20 lattice. We do this by counting the different times a energy level appears in our calculation. This can be represented in a histogram. For T = 2.4 kT/J the distribution is the following:

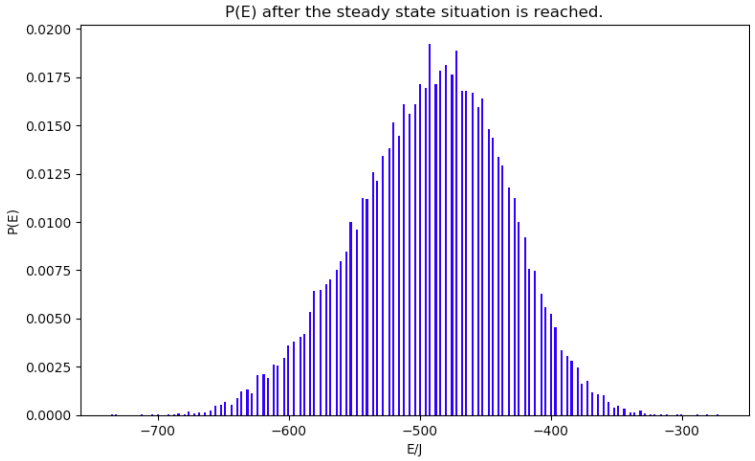


Figure 10: The figure shows a histogram of the Energy change per cycle with 300 bins between the lowest and the largest E value T =2.4 kT/J.

The y-axis is normalized so that the sum of all the bins equals to 1. By calculating the average distance from the average of the distribution squared, we found that the variance of the histogram is equal to 3269E/J. This makes the standard deviation equal to ≈ 57.
We then repeat the calculation but for T =1 kT/J. The histograms produced is:

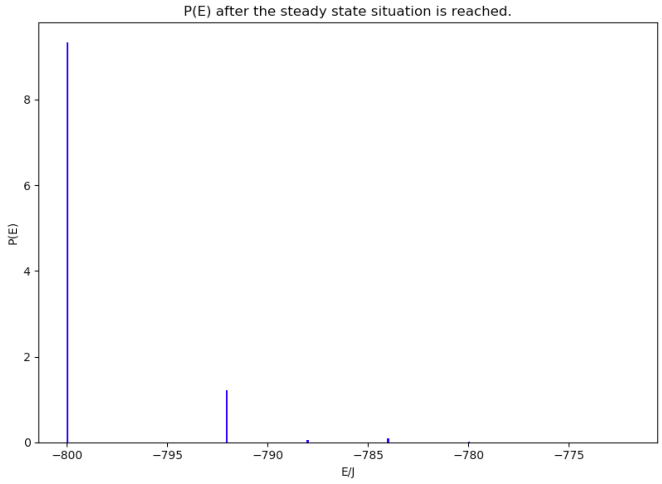
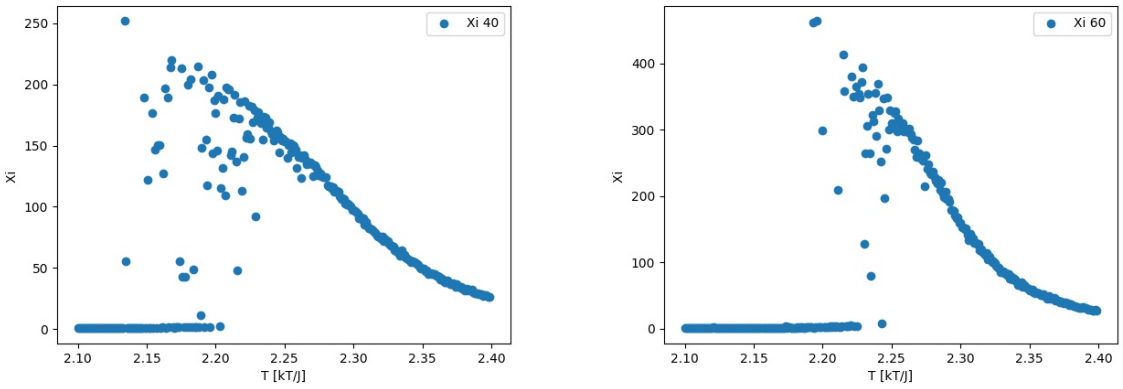


Figure 11: The figure shows a histogram of the Energy change per with 300 bins between the lowest and the largest E value for T =1 kT/J.

This histogram is also normalized, but this time the bins with significance seem fare more fragmented and random. Calculating the variance we found it to be 9.449 with a standard deviation of 3.074. This is much smaller than the standard deviation and variance of the previous histogram.

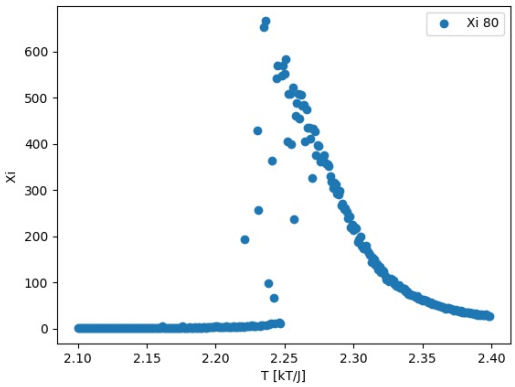
4.4 The critical temperature

When calculating the critical temperature we have to look at the heat capacity, the susceptibility, the average energy and magnetization.

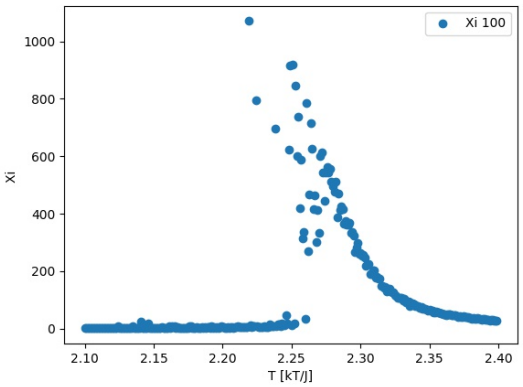


(a) $\chi^{(40)}(T)$

(b) $\chi^{(60)}(T)$



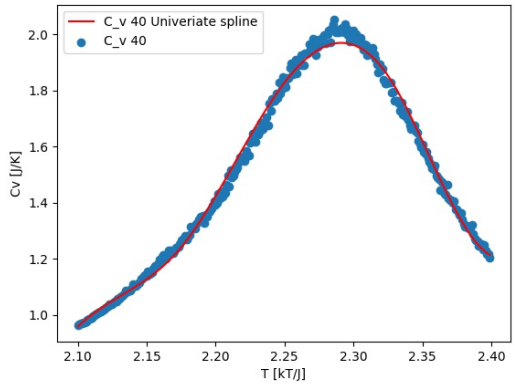
(c) $\chi^{(80)}(T)$



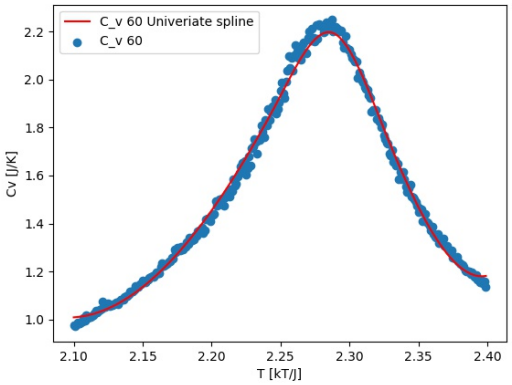
(d) $\chi^{(100)}(T)$

Figure 12: χ as a function of T in energy units, showing a move towards the critical temperature as L increases. Note that the unit for temperature in the images above is actually kT/J, and not J/Kelvin

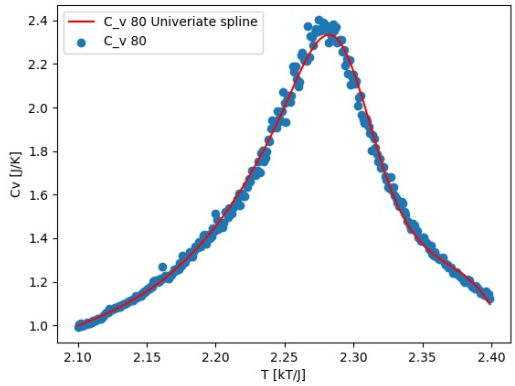
We see here that the ability for the spins to align itself to an external magnetic field decreases when passing the critical temperature.



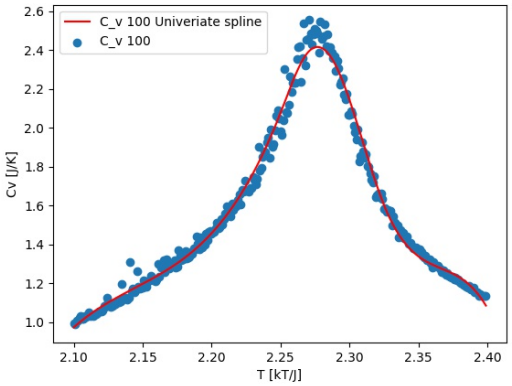
(a) $C_v^{40}(T)$



(b) $C_v^{60}(T)$



(c) $C_v^{80}(T)$



(d) $C_v^{100}(T)$

Figure 13: The figure shows C_v as a function of T in energy units with univeriate spline interpolation for lattices of size $40 * 40$, $60 * 60$, $80 * 80$ and $100 * 100$ spinns. Note that the unit for temperature in the images above is actually kT/J, and not J/Kelvin

We see also here that the distribution narrows as the lattice size increases. We see also that the amount of data around the peak heat capacity reduces as the lattice increases.

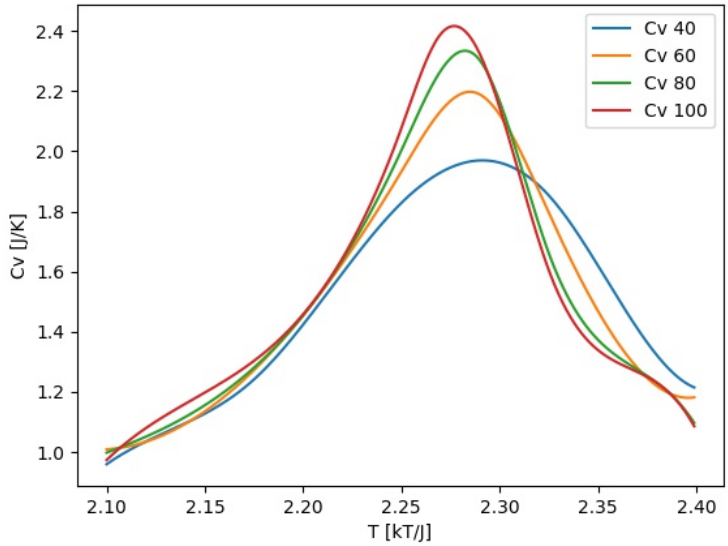
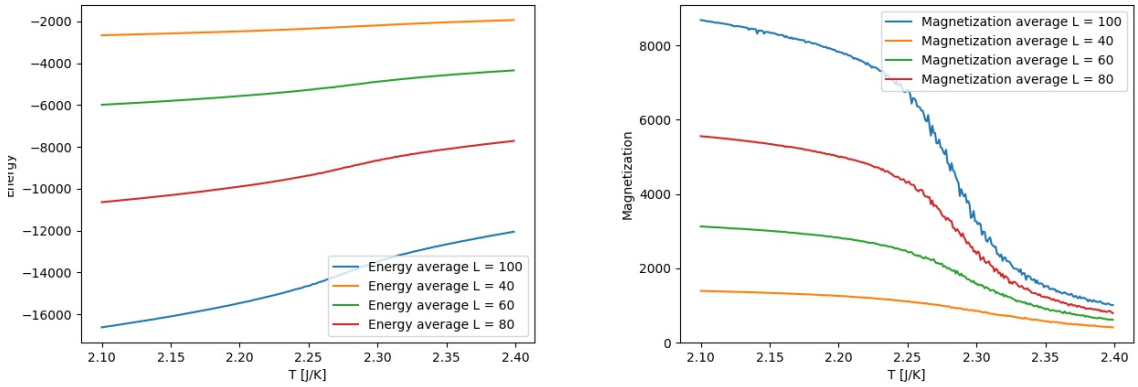


Figure 14: Univariate spline interpolation for the four latices as a function of T in energy units. Note that the unit for temperature in the image above is actually kT/J, and not J/Kelvin

We see here that the distribution narrows around the critical temperature for the given lattice.



(a) $\langle E \rangle(T)$ (b) $\langle M \rangle(T)$

Figure 15: The figure shows the average energy and magnetization of the lattices of size 40^2, 60^2, 80^2 and 100^2 as a function of temperature in energy unit joules. Note that the unit for temperature in the images above is actually kT/J, and not J/Kelvin

We see here that the magnetization converges towards zeros for the increase in

temperature, and that the energy increases. These data required a enormous amount of computation power, but was reduced due to parallelization of the code.

5 Discussion and analysis

5.1 Accuracy of model

In table (1) and (2) we compared our numerical result to analytical for a 2x2 matrix. All ready for 10^4 MCc all our numerical values match those of the analytical up to 1 leading digits. But in order to achieve a an accuracy of two leading digits we have to increase the MCc to 10^7. This shows that the numerical model works very well all ready after only 10^4 MCc, but in order to increases the accuracy (at least of the heat capacity and the susceptibility) we have to increase the number drastically. This could be due to the fact that the mean values are simple to calculate, but the susceptibility and heat capacity are dependent on several values.

Another reason to increase the number of Monte Carlo cycles is the fact that we wish to minimize the total standard deviation of the total average expectation value. As the strategy in section 3.4 explains, we find the expectation value for each cycle and sum them up to find the total average expectation value. To minimize the total standard deviation we increase the number of cycles, as seen in equation 7.

Another way of commenting on the validity of our model, is commenting on the level of difficulty of the unit-test our code passes. All three tests check crucial aspects of our code, either by checking if the implementation is correct or if the physical model it creates is realistic. But, all three tests are also based on calculations on only a couple systems. Meaning that it could be that code only works for systems with complexity up to a certain level. Therefore an alternative method could be to test the first two test (explained in (3.5)) with several system. For example by checking if the the average spin decreases withe increasing lattice. Or if the variance in the distribution of energy increases with larger temperatures.

5.2 20x20 lattice

The most interesting aspect of plots produced in our analyses of the 20x20 lattice, is the drastic difference when changing the temperature from 1 to 2.4. One of the biggest differences in the two cases is the smoothness of the curves. The curves for the larger temperature is much more rough. In other words the energy fluctuates much more even though all of the curves converge towards stability. This is due to the fact that the acceptance rate is dependent on the temperature. When the temperature is large, so is the acceptance rate. This means that even when the change in energy is positive, many of the changes will still be accepted. This leads to the seemingly random and rough curves.

Another interesting find in the analyses is the sudden jump in the number of accepted configurations for the random ordered system with T=1kT/J. This can be explained by looking at the plot for the mean energy in the same system. The plot shows a sudden drop in energy, meaning that many of changes in states are accepted to decrease the energy. This is also due to our definition of the acceptance rate. As explained in our (3.3), when the change in energy is negative, we flip. But, if the change in energy is positive, it is up to the size of the acceptance rate which is dependent on T. So, when the acceptance rate is small (T is small), most of the changes will be when the change in energy is negative. This mean the energy will quickly reach stability, which is what we're seeing in the spike.
Finally it is interesting to note that although both systems seem to reach (at least the energy) stability after the same amount of MCc, the stability in the system with the smaller temperature is far better. Again this is also due to the Acceptance rate. This all highlights the power of the Metropolis algorithm and the importance of the acceptance rate.

5.3 Probability distribution

The distribution of energy is given by the Boltzmann distribution. For T =2.4 kT/J the distribution of the energy (10) is clearly also given by a function very similar to Boltzmanns. This is a great indication for the validity of the code.
The distribution for T = 1kT/J is fare less "Boltzmann-like". Instead it seems like most values are spread out in only three or four bins. The resulting standard deviation is also far smaller, almost 364 times smaller. From the Boltzmann distribution we can see that when the temperature decreases, so should the width of the "curve". This is another way of saying that the variance decreases. Another way of explaining the difference in variance is the size of the acceptance rate. When T is very small, most of the changes in energy are changes which decrease the energy. This means that there wont be a large range of different energy's. Instead the energy quickly converges to the stable energy (hence the one large peak), and then hardly deviates from it.

5.4 Critical Temperature

From our results in figure 13 and figure 12 we got a mean value for the critical temperature equal to 2.265, which is 0.004 off from what Lars Onsager[4] found. The plots showing the susceptibility show the phase transition at the critical temperature as a drastic increase followed by a rough reciprocal inverse function form. This is shown also in figure 13 and figure 14 where the peak heat capacity marks the phase transition at the critical temperature.

Using equation 12, we found the mean value for a, and used it to find the mean value for the critical temperature at $L = \infty$. Now, we chose to use 10^6 Monte Carlo cycles with a Temperature $T \in [2.1, 2.4]$ with 300 steps. Had we increased the number of

Monte Carlo cycles and temperature steps, we could assume better results. Having said that, we ran the calculations on a desktop with a Intel i7 4770k processor, and they took approximately 16,5 hours, so we can only speculate to the degree of which the data would be more precise. Another point to note is that we used Scipy's Univariate spline interpolation, which interpolated the curves with a given smoothness. In order to fit the data more accurately we tuned this parameter, but one can surely tune it even more than we did. This method has a residual function, given in the table below. We see here that the spline method gets less and less accurate with the increase

L	Residual
40	1.29
60	5.04
80	9.27
100	12.97

in spins, which is expected. From picture 13 we see that the data spreads more out when we increase L. As mentioned earlier we could minimize this by increasing the number of cycles, narrowing the temperature domain and increasing the amount of steps.
As mentioned above we see a phase transition at the critical temperature, and this is more visual in figure 15, where we see that the magnetization converges towards zero after the critical temperature is passed, and the energy increases over all. The energy increases due to the fact that the temperature in the system increases, which then increases the chance that the metropolis algorithm accepts the positive energy more often. What we get is a lattice where roughly half of the lattice spins are oriented down and the other half is oriented up. We have thus turned the ferromagnetic material into a paramagnetic material. This is also supported from figure 12

5.5 Uniform Distribution

Aside from the results themselves, a key component to discussing the validity of our findings was in figure 9, where we can see that our probability function tends towards a normalized distribution. This is important because it confirms that our PRNG generator does not repeat, nor is it otherwise flawed (or skewed) towards a certain outcome. This would put all our other results into question, because our Monte Carlo strategy would be compromised by the fact that it isn't sampling from random variables. In our case the distribution is fairly even, and although there are a few peaks we believe these would even out at larger sample rates because the Mersienne twister we use to sample is fairly popular and well established. Our findings are then also in line with central theoretical pillars of Monte Carlo integration like the Central limit theorem. Although, that's more closely related to the theory of the PRNG and not so much the focus of our paper.

6 Conclusion

We have found that by using a Monte Carlo strategy we can approximate Lars Onsager's analytical solution to within a deviation of -0.004, finding the critical temperature to be $T_c^{(L=\infty)} = 2.265$. We consider this to be a pretty good approximation, and because of the way the Monte Carlo strategy works, we can assume the difference will be diminished by increasing our Monte Carlo count. The other aspects of our results worked to back up the validity of how our model behaves. When compared to the analytical values we fin that the accuracy increased with the number of MCc, as the theory of Monte Carlo implies. We found that the mean energy and magnetization of a 20x20 system converges towards a steady state, as it should. For a 2x2 lattice we found that approaching MCc cycles of 10^7 achieved an accuracy of two leading digits compared to the analytical results.

Although the Ising model seems to work well, it neglects many aspects of a real ferromagnetic system. For example it neglects quantum mechanical effects. Alternatively we could have used an expanded Hamiltonian to include certain properties. We could also have implemented a more sophisticated Metropolis test, which features more realistic ways of sampling. To improve the accuracy of the critical temperature, we could have checked for a smaller temperature domain with more steps, so that we be get smaller jumps in temperature. Combining this with an increase in Monte Carlo cycles would indeed improve our approximation. This comes however at a cost, and we would have had to use a super computer of a cluster of some sort.

We also found that given a large enough temperature (at least larger than 1 kT/J) the fluctuation in the mean energy and magnetization increases which agrees with the Boltzmann distribution. In our case we found that when increasing the temperature from 1 to 2.4 kT/J the variance increased by 364 times. We also found that when the system passed the critical temperature described by Onsager, is underwent a phase transition, where it started loosing its magnetization and increased its total energy, which is expected for heating of ferromagnetic materials.

References

[1] Morten Hjort Jensen. *Computational Physics Lectures: Statistical physics and the Ising Model*. URL: http://compphysics.github.io/ComputationalPhysics/doc/pub/statphys/html/statphys.html. (accessed: 23.11.2020).

[2] Morten Hjort Jensen. *Lecture Notes Fall 2015*. URL: https://raw.githubusercontent.com/CompPhysics/ComputationalPhysics/master/doc/Lectures/lectures2015.pdf. (accessed: 22.11.2020).

[3] Morten Hjort Jensen. *Statistical physics*. URL: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/HandWrittenNotes/NotesSeptember3.pdf. (accessed: 22.11.2020).

[4] Lars Onsager. "Crystal statistics. I. A two-dimensional model with an order-disorder transition". In: *Physical Review* Series II.65 (3–4) (1944), pp. 117–149. DOI: https://ui.adsabs.harvard.edu/abs/1944PhRv...65..117O/abstract.

[5] Satya Pal Singh. *The Ising Model: Brief Introduction and Its Application*. 1944. DOI: 10.5772/intechopen.90875. (accessed: 23.11.2020).