Project 4 FYS4150

Vilde Mari Reinertsen

November 18, 2017

Abstract

Contents

1 Introduction

The aim of this project is to use a model of a magnetic material to explore different new aspects of programming. Among them are random number generators, the Monte Carlo method, the Metropolis algorithm and parallelizing. These are all important tools when programming physical systems.

We want to simulate a two dimensional ferromagnetic material using the Ising model. We are using Monte Carlo sampling with the Metropolis algorithm to find the steady state of the system at different temperatures and calculate the mean values of important properties of a magnetic material, the energy, the magnetic moment, the susceptibility and the heat capacity. Finally, we will use the results at different temperatures to find the critical temperature where the material has a phase transition from ferromagnetic with a spontaneous magnetic moment, to a paramagnet with zero magnetic moment.

2 Theory

INCLUDE? Markhow chain - convergance

The theory and method sections are based on chapter 12 and 13 in Jensen, [?].

2.1 Ising model

The two dimensional Ising model is a statistical model that allows us to investigate the temperature dependence of different properties of a magnet. Our model consists of a two dimensional lattice of spins that can be in two different states, spin up, $\uparrow$, or spin down, $\downarrow$ [1]. The number of spins in each dimension is L and the lattice is assumed to be quadratic. In our system we do not have an applied field, $B_a = 0$. The energy of our system is then given by Equation 1.

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

where in our system $J > 0$, J is the coupling constant, giving a ferromagnetic ordering, and s_k, $s_l = \pm 1$, s_k is the k-th spin of N number of spins.

2.1.1 Periodic boundary conditions

2.2 Scaling

In this project there is only one expression that needs to be scaled, namely $e^{-\beta E}$. By introducing $T' = \frac{k_B T}{J}$ and $E' = JE_{kl}$, with $E_{kl} = \sum_{<k,l>} s_k s_l$ this expression can be written as $e^{-E'/T'}$.

2.3 Introduction to statistics

The methods in this project require some statistics. Equation 2 shows the probability density and in the Ising model, the probability distribution is Boltzmann distribution (Equation 3) where Z is the partition function that normalizes the probability (Equation 4). To calculate Z, we need to know all the different states,i, and their energy, E_i. It can be calculated for small systems (2.5 Analytical solutions for L=2), but it would be very time consuming to include it in Monte Carlo Sampling. A method that does not include it is necessary.

$$P(a \leq X \leq b) = \int_b^a p(x)\, dx \tag{2}$$

$$P_i(\beta) = \frac{e^{-\beta E_i}}{Z} \tag{3}$$

$$Z = \sum_{i=1}^{M} e^{-\beta E_i} \tag{4}$$

In this project we are looking at different properties of a magnet. That is energy, magnetic moment, heat capacity and susceptibility. To get out this data form our Ising model, we need some statistical terms. We will use Monte Carlo cycles and the Metropolis algorithm to calculate the second order moment (Equation 5) and the first order moment also called the mean value (Equation 6) of the energy and the magnetic moment. These values, we can use to get the heat capacity and the susceptibility.

$$\langle x^2 \rangle = \int x^2 \, P(x) \, dx \tag{5}$$

$$\langle x \rangle = \int x \, P(x) \, dx \tag{6}$$

2.4 Magnetic properties

The energy of the system is found by using Equation 1. The total magnetic moment is found by adding the magnetic moments of all spins in the lattice. The susceptibility represents how a material responds to an applied magnetic field. The heat capacity is the ration of heat added to a material and the resulting temperature change.

derivative of F (with respect to T/external magnetic field): E, M

second derivative of F (_„_): X, Cv

We are looking at the temperature dependence of these properties and finding the critical temperature, T_C, where the phase transition between paramagnetic and ferromagnetic occurs. When $T < T_C$ the system exhibit spontaneous magnetization it is ferromagnetic, but when $T > T_C$ the net magnetization is zero it is paramagnetic. This phase transition is of second order. That means that the correlation length of the system diverges when $T \to T_C$, it spans the whole system. The correlation length represents the correlation between the spins. It is a spacial unit because it indicates how far away spins can be from each other and still correlate. We can find out that the Ising model exhibits a second-order phase transition since the heat capacity diverges.

$$T_C(L) - T_C(L = \infty) = aL^{-1/\nu} \tag{7}$$

The other equation also?

2.5 Analytical solutions for L=2

Here are the calculation of the analytical values for the system where L=2. These were used as benchmark calculations for the numerical calculations.

Table 2.1: This table lists the possible states and their accompanying energies when L=2.

# ↑	Degeneration	Energy	Magnetization
0	1	-8J	-4
1	4	0	-2
2	4	0	0
2	2	8J	0
3	4	0	2
4	1	-8J	4

First we calculated the partition function using Equation 1 and the numbers in Table 2.1:

$$Z = \sum_i^N e^{-\beta E_i} = e^{-\beta 8J} + e^{-\beta 8J} + e^{\beta 8J} + e^{\beta 8J} + 12$$

$$= 2e^{-\beta 8J} + 2e^{\beta 8J} + 12 = 4\left(\frac{e^{-\beta 8J} + e^{-\beta 8J}}{2}\right) + 12$$

$$= 4\cosh(\beta 8J) + 12$$

Using thermodynamical relations we calculate the expectation values of the energy and magnetic moment [?].

$$\langle E \rangle = k_B T^2 \left(\frac{\partial Z}{\partial T}\right)_{V,N}$$

$$= k_B T^2 \frac{\partial}{\partial T}\left[\ln\left(4\cosh\left(\frac{8J}{k_B T}\right) + 12\right)\right]$$

$$\frac{\partial \ln Z}{\partial T} = \frac{\partial Z}{\partial \beta}\frac{\partial \beta}{\partial T} = \frac{\partial \ln Z}{\partial \beta}\left(\frac{-1}{k_B T^2}\right)$$

$$\langle E \rangle = -\left(\frac{\partial Z}{\partial \beta}\right)_{V,N} = -\frac{\partial}{\partial \beta}\ln\left[4\cosh(8J\beta) + 12\right]$$

$$= \frac{-1}{4\cosh(8J\beta) + 12} 4\sinh(8J\beta) 8J\beta$$

$$= \frac{-8J\sinh(8J\beta)}{3\cosh((J\beta) + 4}$$

Following the same method, we found that:

$$\langle |M| \rangle = \frac{1}{Z}\sum_i^N M_i e^{\beta E_i} = \frac{(8J)^2\cosh(8J\beta)}{\cosh(8J\beta) + 3}$$

$$\langle M \rangle = 0$$

$$\langle E^2 \rangle = \frac{1}{Z}\left(\sum_i^N E_i^2 e^{\beta E_i}\right) = \frac{8\left(e^{8J\beta}+1\right)}{\cosh(8J\beta)+3}$$

$$\langle M^2 \rangle = \frac{1}{Z}\left(\sum_i^N M_i^2 e^{\beta E_i}\right) = \frac{2\left(e^{8J\beta}+2\right)}{\cosh(8J\beta)+3}$$

We can use these to calculate the rest, with these definitions:

$$C_V = k\beta^2\left(\langle E^2\rangle - \langle E\rangle^2\right)$$

$$\chi = \beta\left(\langle M^2\rangle - \langle M\rangle^2\right)$$

3 Method

In this project we tried out many new concepts in our algorithm.

3.1 Monte Carlo sampling

To reach the most likely state, the steady state, we use Monte Carlo sampling. Our Monte Carlo sampling function is given in Equation 3 and it is the Boltzmann distribution which is temperature dependent because $\beta = \frac{1}{k_b T}$. It is the probability of finding the system in state i. The sampling rule is explained in 3.2 Metropolis algorithm.

3.2 Metropolis Algorithm

We are looking at transition from one state to another. W_{ij} is the transformation probability of going from state j to state i. ω_i is our PDF, P_i. That means that a transition from i to j can be written:

$$\omega_i = W_{ij}\omega_j$$

The steady state of a system can be written as:

$$\omega_i = \sum_j W_{ij}\omega_j$$

or in matrix form:

$$\hat{\omega}(t+1) = \hat{W}\hat{\omega}(t)$$

In our case we do not know the transition probability, and we have to model it. We do this using the Metropolis algorithm. We separate W_{ij} into two parts, the probability of accepting the move from state j to state i, A_{ij}, and the probability of making the move to state i when in state j, T_{ij}. We define T_{ij} and A_{ij} to lead the system to the most likely state.

In our case the algorithm starts with suggesting a move from state j to state i, it does this by picking a random spin and flipping it. The picking of the spin is governed by the uniform distribution, that gives $T_{ij} = T_{ji}$, it is symmetric. The uniform distribution is baked into the random number generators.

After the move is suggested, it has to be accepted or not. This acceptance has to make the system go towards the most likely state, the steady state.

Our ratio between probabilities is:

$$\frac{A_{ij}}{A_{ji}} = \frac{\omega_i T_{ij}}{\omega_j T_{ji}} = \exp(-\beta(E_i - E_j)) = \exp(-\beta\Delta E_{ij})$$

with the Boltzmann distribution and the acceptance probability is:

$$A_{ij} = \begin{cases} \exp(-\beta\Delta E_{ij}) & \text{if } \Delta E_{ij} > 0 \\ 1 & \text{if } \Delta E_{ij} \leq 0 \end{cases}$$

We need to accept some moves where $\Delta E_{ij} > 0$ so that the algorithm is ergodic, that all possible states in the system can be reached, even though the probability of it happening is small.

In practice:

The metropolis algorithm is an algorithm that takes the system to the steady state. We want to find out what the real state of the system is when the outer parameters are what they are, for example temperature.

- Calculate total energy of initial lattice, E_{tot}

- Pick a random spin in the lattice

- Flip the spin

- Calculate the change in energy, ΔE (only five possibilities)

(flipping only one)

$$\Delta E = -J\sum_{\langle kl\rangle}^{M} s_k^2\left(s_l^2 - s_l^1\right)$$

$((s_l^2 - s_l^1) = -2$ if $s_l^1 = 1$ and $((s_l^2 - s_l^1) = 2$ if $s_l^1 = -1$:

$$\Delta E = 2Js_l^1 \sum_{\langle k \rangle}^{M} s_k^2$$

- If $\Delta E \leq 0$ - accept because we want to move to a state with the lowest energy

- If $\Delta E > 0$ - calculate $\omega = e^{-\beta \Delta E}$

- Compare ω with a random number r, if $r \leq \omega$ - accept new configuration

E += ΔE and M += $\Delta M = 2s_l^1$

- Update mean values

- Repeat

Should show how to find the five ΔEs.

3.3 Random number generators

The random number generator (RNG) used in this project is gain from a linear congruential relation (Equation 8). Which can, with good parameters, a and c give a list of quasi random numbers with a maximum periodicity M, that means that the numbers repeat themselves after M numbers. It is important that the periodicity is large compared with the number of Monte Carlo cycles, to the 'measurements' represent the truly random real world. The RNG gives a number $x_i \in [0, 1]$ because $x_i = N_i/M$. The PDF for generating random numbers in the interval $[0, 1]$ is the uniform distribution.

$$N_i = (aN_{i-1} + c)MOD(M) \tag{8}$$

3.4 The algorithm

```
for(i =0; i<temperature.size();i++){

    //make vector with all possible omega_i/
    omega_j
    //dependent on temperature and
   for(MC = 1; MC < MCcycles.size(); MC++){

       for(i=0; i< L*L; i++){
           ix, iy = random(i);
           // with random spin generator

           Matrix(ix, iy) *= -1;

           dE = dEs[i];

           MetropolisAlgorithm();
           // decide if the flip is accepted
           if(flip is accepted){
               Energy += dE;
               Magnetic += dM;
           }
       }
       //Add the new values to the sum of the
     values:
       mean_E += Energy;
       mean_E2 += Energy*Energy;
       mean_M += Magnetic;
       sum M\^2 += M\^2;
       sum |M| += |M|;
   }
//Before print, the values are divided by
    the number of Monte Carlo cycles to find
     the mean values.
}
```

3.5 Parallelizing

In this project we used parallelization. We used Message Passing Interface (MPI) to parallelize our code.Each process read and executes the whole program. We had to use different MPI functions to make sure that all processes sent their results to one main process that would write to file for example. Beneath is some samples of the program that shows the process.

```
int NProcesses, RankProcess;
//   MPI initializations
MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &NProcesses);
MPI_Comm_rank(MPI_COMM_WORLD, &RankProcess);

//Sending result to RankProcess = 0
for( int i =0; i < 5; i++){
    MPI_Reduce(&meanValues[i], &
    TotalMeanValues[i], 1, MPI_DOUBLE,
    MPI_SUM, 0, MPI_COMM_WORLD);
}

if (RankProcess == 0){
    writeToFile();
}
// MPI end
MPI_Finalize();
```

Table 4.1: This table compares the analytical values for L=2 with the numerical ones after 10^6 Monte Carlo cycles. The values are in units per spin and at T=1.0 $k_B T/J$.

	Numerical:	Analytical:
$\langle E \rangle$ $[E_{kl}]$	-1.9958	-1.9960
$\langle E^2 \rangle$ $[E^2_{kl}]$	15.9664	15.9679
$\langle M \rangle$	0.0451	0
$\langle M^2 \rangle$	3.9930	3.9933
$\langle \lvert M \rvert \rangle$	0.9986	0.9987
$\chi\,[J/k_B^T]$	3.9849	3.9933
$C_V\,[J^2/k_B^3T^2]$	0.0335	0.0321

3.6 Unit tests

In this project we did not write any unit tests, we only wrote out the results during the program development and saw if it was logical. We could have made a unit test that calculated the properties of the L=2-system and compared them to the analytical values calculated in the theory. This would have helped us make sure that when we made changes to the program, for example implementing parallelization and classes, the program was doing what it was supposed to do and not getting the wrong result.

Another unit test we could have included is just flipping one spin in a small matrix and checking is the change in energy is what we expect. We should use unit tests from the start in the next project.

4 Result and discussion

4.1 Matrix dimension L=2

The result form the two dimensional lattice with lattice size L=2 and at T= 1.0 $k_B T/J$ are in Figures 4.1, 4.2, 4.3 and 4.4. From the plots, we can see that the numerical result gives a good agreement with the analytical values calculated in 2.5 Analytical solutions for L=2 after approximately $5 \cdot 10^5$ Monte Carlo cycles. Table 4.1 lists the result after 10^6 Monte Carlo cycles, and it show a good agreement as well.

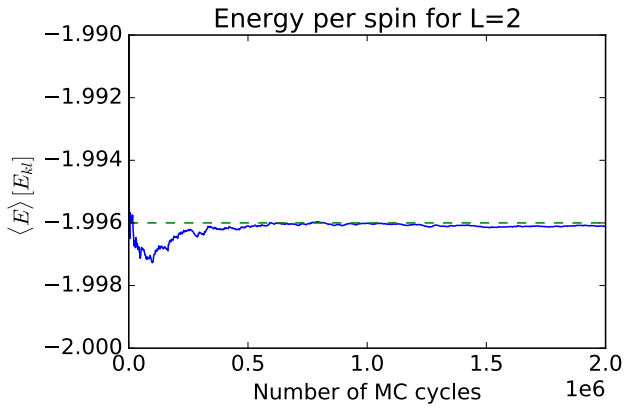

Figure 4.1: This is a plot of the expectation value of the energy per spin verus number of Monte Carlo cycles. The plot shows that we have a good agreement after $5 \cdot 10^5$ MC cycles.

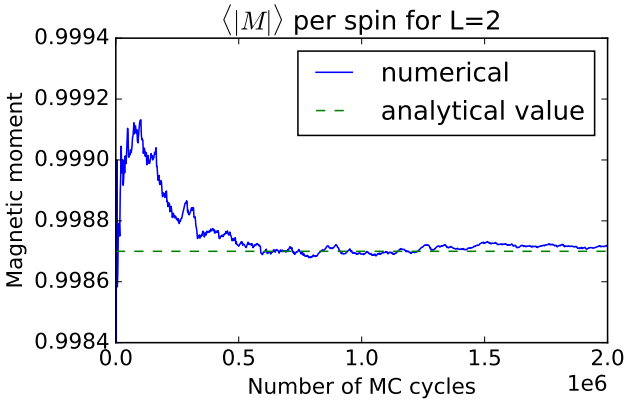

Figure 4.2: This is a plot of the expectation value of the mean absolute value of the magnetic moment per spin verus number of Monte Carlo cycles. The plot shows that we have a good agreement after $5 \cdot 10^5$ MC cycles.

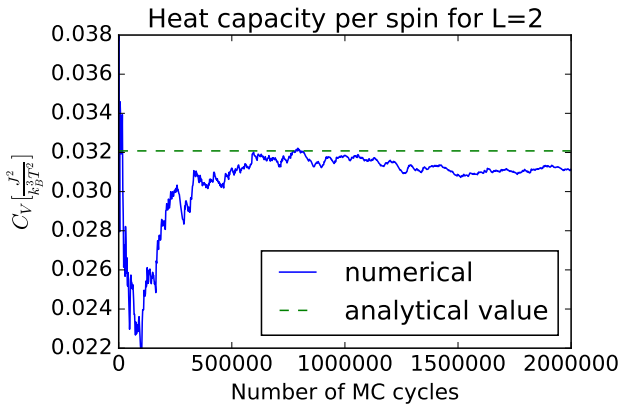

Figure 4.3: This is a plot of the heat capacity per spin verus number of Monte Carlo cycles. The plot shows that we have a good agreement after $5 \cdot 10^5$ MC cycles.

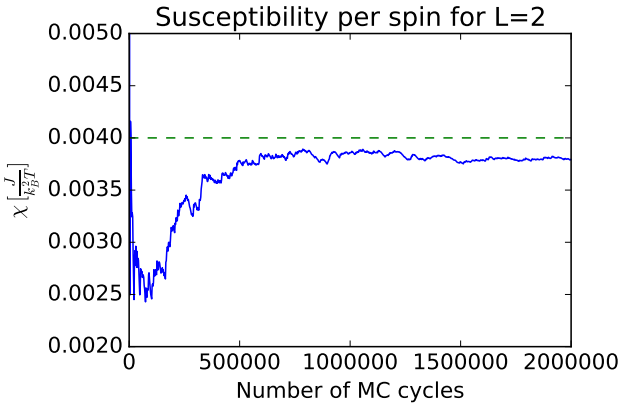

Figure 4.4: This is a plot of the susceptibility per spin verus number of Monte Carlo cycles. The plot shows that we have a good agreement after $5 \cdot 10^5$ MC cycles.

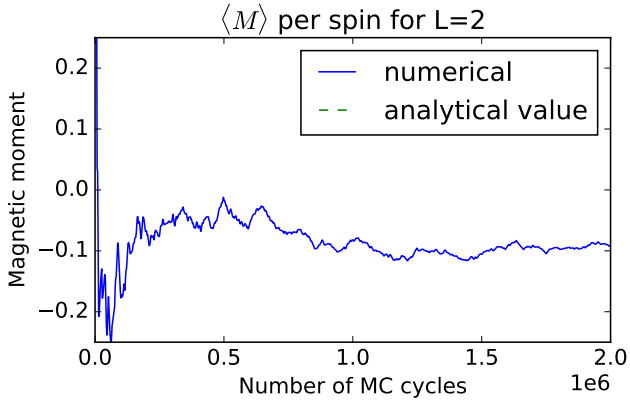

Figure 4.5: This is a plot of the expectation value of the mean value of the magnetic moment per spin verus number of Monte Carlo cycles. The plot shows that we would not have had a good agreement after $5 \cdot 10^5$ MC cycles, when not using the absolute value as in Figure 4.2.

We see that the energy converges fastest. The magnetic moment is not converging as fast but still fast, we are plotting the absolute value, and that makes it converge faster, because the oscillation between the same size, but different signs does not show. Figure 4.5 shows how the magnetic moment converges much slower. It can be shown however that the magnetic moment will converge, but slower and we would have needed more Monte Carlo cycles.

4.2 Matrix dimension $L = 20$

HMM: Should define an area that is enough for equilibrium!

OBS: Need the number of MC cycles to reach equilibrium!

OBS: Need equilibration time! (5 1e5?)

OBS: Comment accepted configs T dependency

4.2.1 Different temperatures

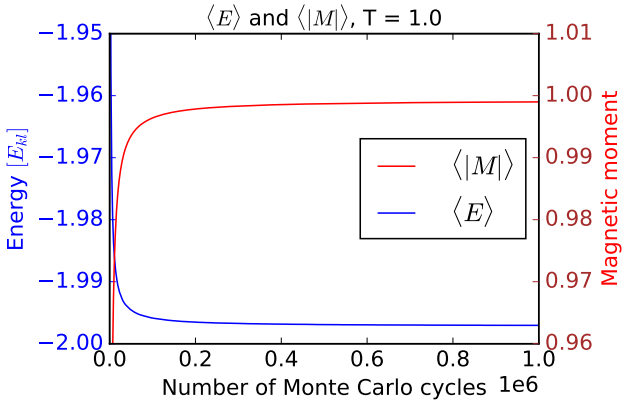

Figure 4.6: This is a plot of both the expectation value of the energy and absolute magnetic moment per spin verus number of Monte Carlo cycles at T = 1.0 K. The plot shows that an equilibrium is reached already at $2 \cdot 10^5$ MC cycles.

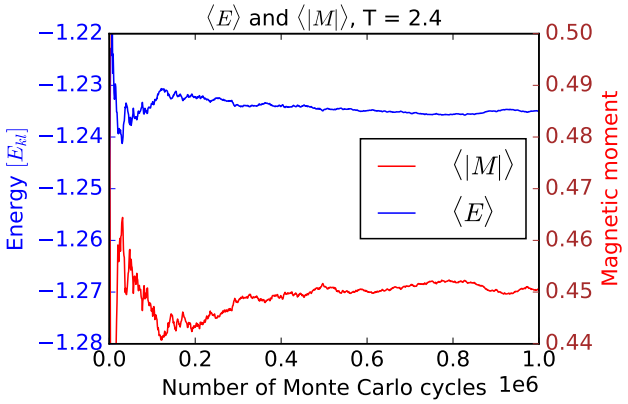

Figure 4.7: This is a plot of both the expectation value of the energy and absolute magnetic moment per spin verus number of Monte Carlo cycles at T = 2.4 K. The plot shows that an equilibrium is reached at around $5 \cdot 10^5$ MC cycles.

4.2.2 Initial state

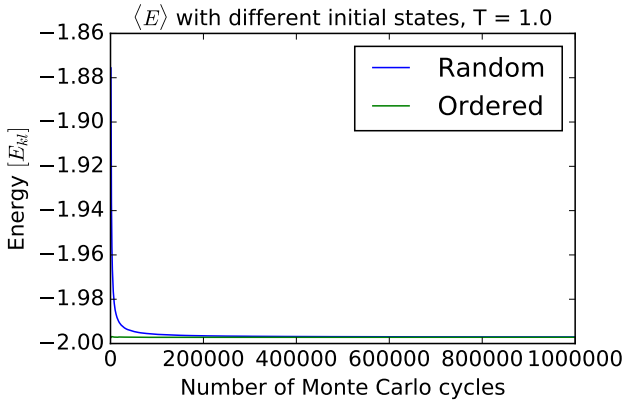

Figure 4.8: This is a plot of both the expectation value of the energy and absolute magnetic moment per spin verus number of Monte Carlo cycles at T = 1.0 K. The plot shows the difference in the behaviour of the ordered initial state and a random initial state.

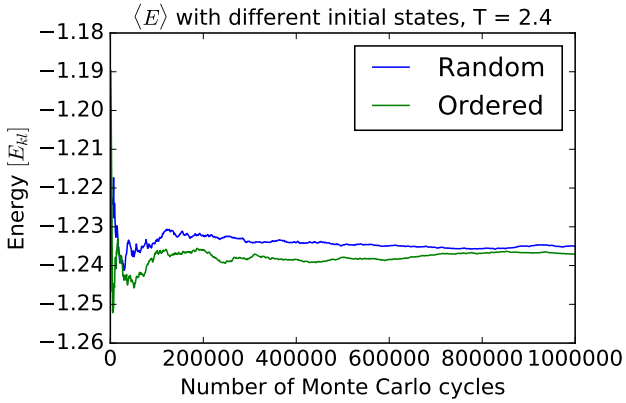

Figure 4.9: This is a plot of both the expectation value of the energy and absolute magnetic moment per spin verus number of Monte Carlo cycles at T = 2.4 K. The plot shows the difference in the behaviour of the ordered initial state and a random initial state.

4.2.3 Accepted configurations

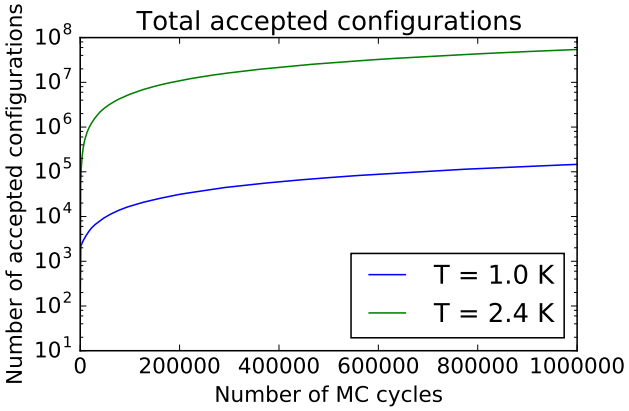

Figure 4.10: This is a plot of the total number of accepted configurations versus number of Monte Carlo cycles with random initial state.

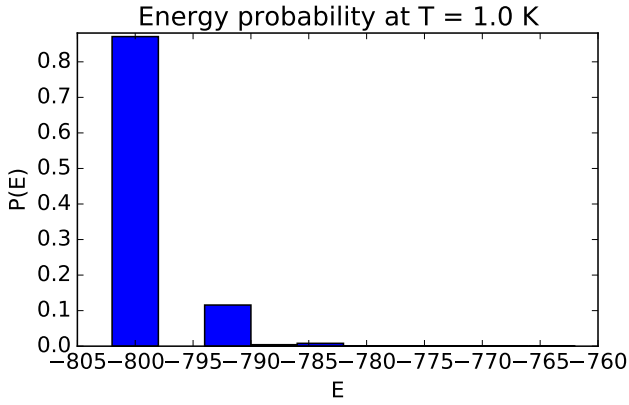

Figure 4.11: This is a plot of the energy probability when T = 1.0 K. The energy is the total energy of the 2D lattice with 20×20 spins.

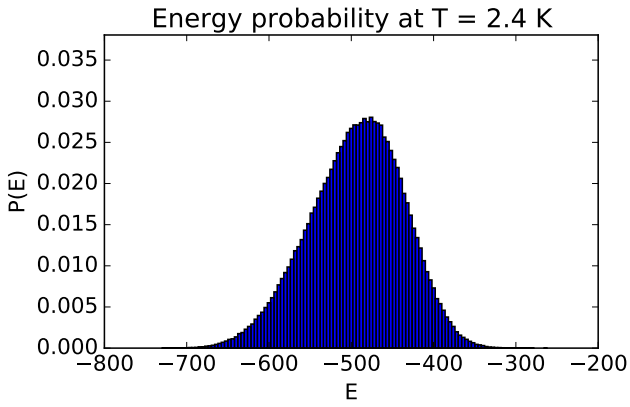

Figure 4.12: This is a plot of the energy probability when T = 2.4 K.The energy is the total energy of the 2D lattice with 20×20 spins.

4.3 Energy probability

OBS: Compare result with computed variance!

OBS: Discuss behavior (In Discussion - maybe just merge result and discussion?)

Computed variance (from same dataset?):

$$\sigma_E^2 = \langle E^2 \rangle - \langle E \rangle^2$$

$$\text{FWHM} = 2\sqrt{2ln2}\sigma \approx 2.355\sigma$$

T = 1.0 K:

$$\sigma_E^2 = 638181 - (-798.855)^2 = 11.69$$

$$\sigma = 3.42$$

$$\text{FWHM} \approx 2.355 \cdot 3.24 = 7.63$$

T = 2.4 K:

$$\sigma_E^2 = 247886 - (-494.628)^2 = 3229.14$$

$$\sigma = 56.8$$

$$\text{FWHM} \approx 2.355 \cdot 56.8 = 133.76$$

4.4 Increasing dimensionality / Critical temperature

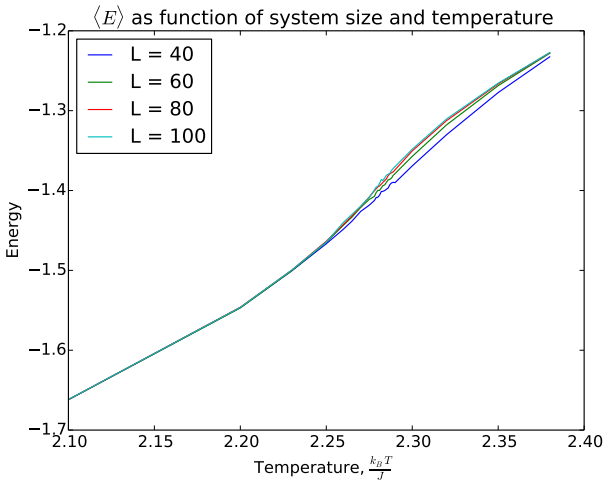

Figure 4.13: This is a plot if the energy versus temperature around the critical temperature for the different lattice sizes with $L = 40$, $L = 60$, $L = 80$ and $L = 100$.

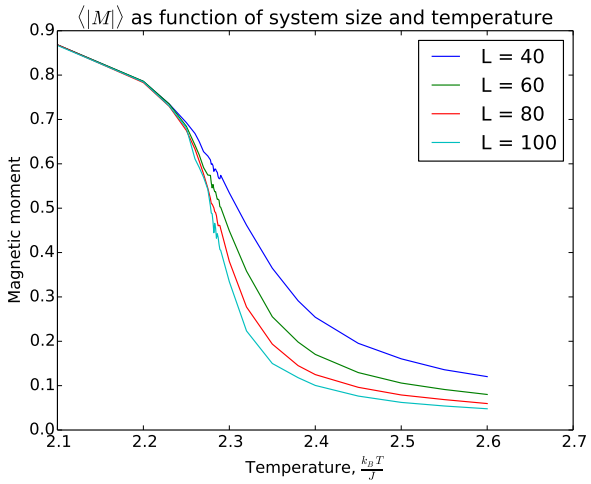

Figure 4.14: This is a plot if the absolute magnetic moment versus temperature around the critical temperature for the different lattice sizes with $L = 40$, $L = 60$, $L = 80$ and $L = 100$.

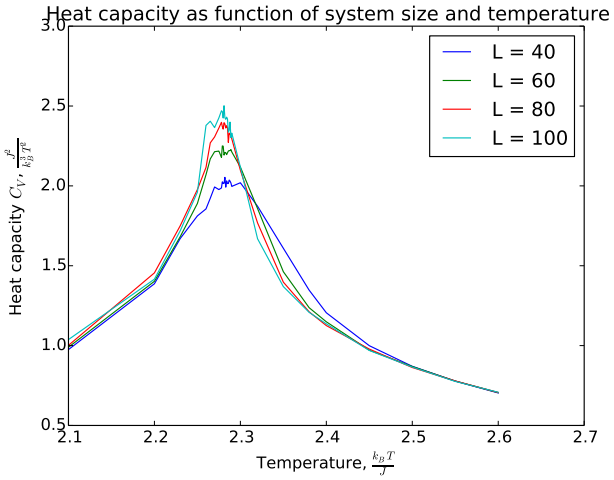

Figure 4.15: This is a plot if the heat capacity versus temperature around the critical temperature for the different lattice sizes with $L = 40$, $L = 60$, $L = 80$ and $L = 100$.

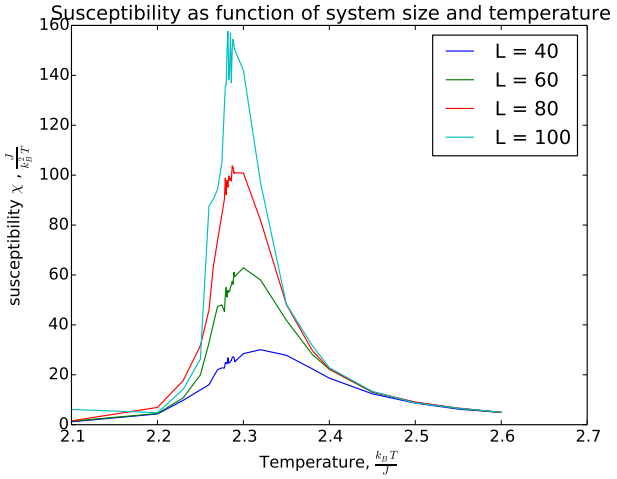

Figure 4.16: This is a plot if the susceptibility versus temperature around the critical temperature for the different lattice sizes with $L = 40$, $L = 60$, $L = 80$ and $L = 100$. The exact value is $T_C = kTC/J = 2/\ln(1+\sqrt{2}) \approx 2.269\,\mathrm{k_B\,K}$ [2].

OBS: Indication of phase transition? (Peak - at least for Cv and X)

OBS: Compare behaviour with equations?

OBS: Use Equation 7 to extract T_C.

Getting these equations from 7 where $\nu = 1$:

$$T_C(40) - T_C(\infty) = a \cdot 40^{-1}$$
$$T_C(60) - T_C(\infty) = a \cdot 60^{-1}$$
$$T_C(80) - T_C(\infty) = a \cdot 80^{-1}$$
$$T_C(100) - T_C(\infty) = a \cdot 100^{-1}$$

Table 4.2: This is a list of the CPU time with different numbers of processes for L=60 and number of Monte Carlo cycles were 10^6.

Number of processors:	CPU time [s]:
1	513.069
2	306.975

(Sett inn tall!)

$$T_C(\infty) = -a \cdot 40^{-1} + 2.28 \quad (9)$$
$$T_C(\infty) = -a \cdot 60^{-1} + 2.28 \quad (10)$$
$$T_C(\infty) = -a \cdot 80^{-1} + 2.28 \quad (11)$$
$$T_C(\infty) = -a \cdot 100^{-1} + 2.28 \quad (12)$$

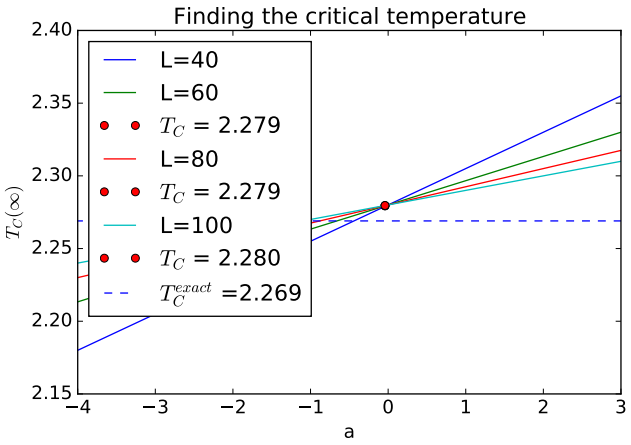

Figure 4.17: This is a plot of Equation 7 with different values of L (See Equations 9 - 12). The interssections represent the solution. They should have all had a cross section in the same place, and the y-value of the intersection would have been the critical temperature when $L \to \infty$.

Exact $T_C = kTC/J = 2/\ln(1+\sqrt{2}) \approx 2.269$ [2]

5 Conclusion

The aim of this project is to use a model of a magnetic material to explore different new aspects of programming. Among them are random number generators, the Monte Carlo method, the Metropolis algorithm and parallelizing. These are all important tools when programming physical systems.

We want to simulate a two dimensional ferromagnetic material using the Ising model. We are using Monte Carlo sampling with the Metropolis algorithm to find the steady state of the system at different temperatures and calculate the mean values of important properties of a magnetic material, the energy, the magnetic moment, the susceptibility and the heat capacity. Finally, we will use the results at different temperatures to find the critical temperature where the material has a phase transition from ferromagnetic with a spontaneous magnetic moment, to a paramagnet with zero magnetic moment.

References

[1] Ising model. https://en.wikipedia.org/wiki/Ising_model. Accessed: 2017-11-17.

[2] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65:117–149, Feb 1944.

Appendix

State	Spin	Energy	Magnetization
0	↓↓↓↓	−8J	−4
1	↓↓↓↑	0	−2
2	↓↓↑↓	0	−2
3	↓↑↓↓	0	−2
4	↑↓↓↓	0	−2
5	↓↓↑↑	0	0
6	↓↑↓↑	0	0
7	↓↑↑↓	8J	0
8	↑↓↓↑	8J	0
9	↑↓↑↓	0	0
10	↑↑↓↓	0	0
11	↓↑↑↑	0	2
12	↑↓↑↑	0	2
13	↑↑↓↑	0	2
14	↑↑↑↓	0	2
15	↑↑↑↑	−8J	4