Project 4

Petter A. Langstrand, Eirik V. Winness & Andreas Bjurstedt

November 2019

Abstract

In this project we have used the Ising model to study a magnetic system in two dimensions which isn't exposed to an external magnetic field. The Metropolis algorithm is used to study the model for different lattice sizes, and finding the critical temperature by incriminating over a predetermined temperature interval. We used these critical temperatures to estimate the infinite lattice critical temperature, for which we obtained a reasonable value compared to the analytical calculation given by Lars Onsager. We find at these critical temperatures the system turns from ferromagnetic into paramagnetic, and at higher temperatures the system will no longer be magnetized.

1 Introduction

In this project, we use the Ising model to study a magnetic system in two dimensions. We assume that the system isn't exposed to an external magnetic field. The Ising model in two dimensions consists of a two dimensional lattice of discrete points (or point particles). Each point has the value -1 or 1, where -1 represents *spin down* ($\downarrow$) and +1 represents *spin up* ($\uparrow$). This means that the model can be represented by a matrix, where each matrix element has the value -1 or +1.

Given an initial state, the Metropolis algorithm is used in order to simulate the behaviour of the magnetic system. The Metropolis algorithm is a Markov Chain Monte Carlo method which allows us to simulate random walks, or a random process of spin flipping, which eventually brings the system towards its most likely state for a given temperature T. By changing the temperature systematically, we can find an estimation of

the system's critical temperature T_c. At T_c, the system makes a transition from a magnetic system ($T < T_c$), to a system with zero magnetization ($T > T_c$).

A thorough explanation of the Ising model and the Metropolis algorithm are given under the *Statistical physics* and the *Brownian motion and random walks* link respectievly in [1]

2 Methods

A 2 dimensional magnetic system modelled with the Ising model
The Ising model for a 2 dimensional magnetic system is thoroughly explained under the *Statistical physics* link in [1]. Without an external magnetic field, the energy E_i in a two dimensional magnetic system for a given spin configuration is

$$E_i = -J \sum_{<kl>}^{N} s_k s_l \tag{1}$$

according to the Ising model. N is the total number of spins in the system. s_k and s_l are neighbour spins in the system. Each spin can have the value -1 (*spin down*, $\downarrow$) or +1 (*spin up*, $\uparrow$). The notation $< kl >$ means that the value of two spins are multiplied if (and ony if) they are neighbours in the lattice. In a two dimensional lattice, each spin has four neighbours. So when the lattice is represented by a matrix, the energy contribution from a matrix element s_{kl} an its neighbours is:

$$E(< kl >) = -Js_{kl}(s_{k-1,l} + s_{k+1,l} + s_{k,k-1} + s_{k,l+1})$$

J is a coupling constant expressing the strength of the interaction between neighbouring spins. In our model, we set $J = 1$ for a ferromagnetic material.

With N spins in the lattice (N elements in the spin matrix), the lattice has 2^N possible configurations of spins.

In order to make a model with a finite number of spins, which represents an enormously large lattice, periodic boundary conditions are used. Then each spin in the lattice always has four neighbours. In a $N = L \times L$ matrix representation of the lattice, this means for example that the neighbours of spin element s_{11} is $s_{1n}, s_{12}, s_{n1}, s_{21}$. So if a spin is positioned on an edge or

in a corner of the lattice, one or two of its four neighbours are positioned on the opposite edge or in another corner of the lattice.

The expression for the magnetization of the lattice for a given spin configuration i, is the sum of all the spins in the lattice

$$M_i = \sum_{j=1}^{N} s_j \tag{2}$$

A 2×2 lattice has N = 4 spins and $2^N = 2^4 = 16$ spin configurations. As an example, one of these spin configurations is

$$\begin{bmatrix} s_1 & s_2 \\ s_3 & s_4 \end{bmatrix} = \begin{bmatrix} \downarrow & \uparrow \\ \uparrow & \downarrow \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

For the 2×2 lattice (1) with periodic boundary conditions becomes

$$E_i = -J(s_1 s_2 + s_2 s_1 + s_3 s_4 + s_4 s_3 + s_1 s_3 + s_3 s_1 + s_2 s_4 + s_4 s_2) \tag{3}$$

For our particular configuration example, $E_i = 8J$. The magnetization for the configuration follows directly from (2), and is $M_i = 0$. Table (1) below shows energy and magnetisation for the 16 spin configurations. *Degeneracy* just means the number of spin configurations with a particular number of spins up.

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	4	-8J	-4

Table 1: Energy and magnetization in the 16 spin configurations for the 2×2 lattice.

For the magnetic system with a 2×2 lattice and only 16 spin configurations, we can find analytical expressions for the mean energy <E>, the mean value of the magnetization <|M|> , the specific heat C_v

and the susceptibility $\mathcal{X}$ for a given temperature T. Then the results given by the Ising model with the metropolis algorithm can be controlled against these values for the 2×2 lattice.

In order to find the expressions for <E>, $<|M|>$, C_v and $\mathcal{X}$ for the 2×2 lattice with a given temperature T, we need the Boltzmann probability distribution

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

P_i is the probability for the energy state E_i in the system when the temperature is T. $\beta = 1/(K_B T)$, where K_B is the Boltzmann constant. Z is the partition function. For the 2×2 lattice with the N = 16 energy states from table (1), Z becomes

$$\begin{aligned} Z = \sum_{i=1}^{16} e^{-\beta E_i} &= 2e^{8\beta J} + 2e^{-8\beta J} + 12 = 2\left(e^{8\beta J} + e^{-8\beta J} + 6\right) \\ &= 2\left(2cosh(8 + \beta J) + 6\right) = 4\left(cosh(8\beta J) + 3\right) \end{aligned}$$

The expectation value for the energy becomes

$$\begin{aligned} < E > &= \sum_{i=1}^{16} E_i P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{16} E_i e^{-\beta E_i} = \frac{1}{Z}\left(-16Je^{8\beta J} + 16Je^{-8\beta J}\right) \\ &= \frac{-16J}{Z}\left(e^{8\beta J} - e^{-8\beta J}\right) = \frac{-16J}{4(cosh(8\beta J) + 3)} 2sinh(8\beta J) \\ &= -\frac{8Jsinh(8\beta J)}{(cosh(8\beta J) + 3)} \end{aligned}$$

In order to calculate the specific heat C_v, we also need the expectation value of the energy squared

$$\begin{aligned} < E^2 > &= \sum_{i=1}^{16} E_i^2 P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{16} E_i^2 e^{-\beta E_i} = \frac{1}{Z}\left(128J^2 e^{8\beta J} + 128J^2 e^{-8\beta J}\right) \\ &= \frac{128J^2}{Z}\left(e^{8\beta J} + e^{-8\beta J}\right) = \frac{128J^2}{4(cosh(8\beta J) + 3)} 2cosh(8\beta J) \\ &= \frac{64J^2 cosh(8\beta J)}{(cosh(8\beta J) + 3)} \end{aligned}$$

With the analytical expression for $< E >$ and $< E^2 >$, we can find the analytical expression for the specific heat for the 2×2 lattice

$$C_v = \frac{1}{K_B T^2} \left(< E^2 > - < E >^2 \right)$$

The expectation value for the absolute value of the magnetization for the 2×2 lattice becomes

$$\begin{aligned} < |M| > &= \sum_{i=1}^{16} |M_i| P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{16} |M_i| e^{-\beta E_i} = \frac{1}{Z} \left(8e^{8\beta J} + 16 \right) \\ &= \frac{8}{Z} \left(e^{8\beta J} + 2 \right) = \frac{2 \left(e^{8\beta J} + 2 \right)}{(cosh(8\beta J) + 3)} \end{aligned}$$

In order to calculate the susceptibility $\mathcal{X}$, we also need the expectation value of the magnetization squared

$$\begin{aligned} < M^2 > &= \sum_{i=1}^{16} M_i^2 P_i(\beta) = \frac{1}{Z} \sum_{i=1}^{16} M_i^2 e^{-\beta E_i} = \frac{1}{Z} \left(32e^{8\beta J} + 32 \right) \\ &= \frac{32}{Z} \left(e^{8\beta J} + 1 \right) = \frac{8 \left(e^{8\beta J} + 1 \right)}{(cosh(8\beta J) + 3)} \end{aligned}$$

With the analytical expression for $< |M| >$ and $< M^2 >$, we can find the analytical expression for the susceptibility for the 2×2 lattice

$$\mathcal{X} = \frac{1}{K_B T} \left(< M^2 > - < |M| >^2 \right)$$

As already stated, the coupling constant $J = 1$ in our model. Like for E_i its unit is *energy* (Joule or eV). We let $\tilde{E}_i$ and $\tilde{J}$ be dimensionless variables so that

$$J = \tilde{J} \ \ energy$$

and

$$E_i = \tilde{E}_i \ \ energy$$

We also choose dimensionless temperature $\tilde{T}$ so that

$$T = \tilde{T} \ \frac{energy}{K_B}$$

As a check, we see that the exponent in the Boltzmann distribution and partition function still is dimensionless

$$-\beta E_i = -\frac{E_i}{K_B T} = -\frac{\tilde{E}_i}{\tilde{T}} \frac{energy}{K_B \frac{energy}{K_B}} = -\frac{\tilde{E}_i}{\tilde{T}} \quad \text{(ok)}$$

We choose $K_B = 1$ The *tilde* notation is not used in the rest of the report.

Although it is valuable in itself to find the analytical expressions for the mean energy <E>, the mean value of the magnetization <|M|> , the specific heat C_v and the susceptibility $\mathcal{X}$, we find it simpler to find the values when T = 1 just by performing the summation over the 16 states in a Python script instead of using the analytical expressions. A comparison between the analytical values from the $N = 2 \times 2$ Ising model and the values calculated with the Metropolis algorithm are shown in the result section.

The Ising model and the Metropolis algorithm
The Metropolis algorithm is thoroughly explained under the *Brownian motion and random walks* link in [1], and a summary of the algorithm steps for the two dimensional magnetic system with a C++ program example included are found under the *Statistical physics* link (in [1]). We use this C++ program example as the foundation for the programs made in this project.

The Ising model simulated by the Metropolis algorithm can be described in the following way:

1. An $n \times n$ spin matrix with $N = n \times n$ number of spins is initialized. The starting configuration of the matrix can be ordered (all spins in the matrix have the spin value equal to +1 initially) or random (each spin value is chosen randomly to -1 or +1) The enery and magnetization for this starting configuration is then calculated

2. The Metropolis algorithm finds a random element in the spin matrix. If the element value is -1 (spin down), its new value will be +1 (spin up) if the spin is flipped. If the value is +1, its new value will be -1 if the spin is flipped. In both cases the energy difference due to the spin flip can be expressed as (when $J = 1$)

$$\triangle E = 2 s_l \sum_{<k>} s_k$$

s_l is the value of the randomly chosen spin we may flip, and the sum is over its four neighbour spins s_k. With four neighbour spins to the spin we may flip, the possible values of $\triangle E$ is limited to 5. These values are $\triangle E = -8$, $\triangle E = -4$, $\triangle E = 0$, $\triangle E = 4$ and $\triangle E = 8$.

3. The detailed balance (the ratio of two probabilities) w of the Boltzmann distribution is calculated.

$$w = e^{-(\beta \triangle E)} = e^{-(\triangle E/T)}$$

If $r \leq w$, where r is a random number in the interval $r \in [0, 1]$, the randomly chosen spin is flipped and the new spin configuration is accepted. Observe that if $\triangle E \leq 0$, $r \leq w$ is always fulfilled.

In order to bring the system towards its most likely state for a given temperature T, we want $\triangle E \leq 0$ and the system to move to a state with lower energy. If $0 < \triangle E$, $r \leq w$ has to be fulfilled in order to flip the randomly chosen spin and the system to move to a higher energy state. If $r \leq w$ is not fulfilled, the chosen spin is not flipped and the system remains in the same energy state.

The energy of the lattice is updated by adding $\triangle E$. The magnetization of the lattice is updated by adding the flipped spin value multiplied by 2.

4. Point 2. and 3 is performed N times, where N is the number of spins in the spin matrix.

5. Expectation values <E>, <$|M|$>, <E^2> and <M^2> are updated and one Monte Carlo cycle is completed

6. Point 2. to 5. are repeated in a chosen number of monte carlo cycles in order for the system to move towards its most likely state and to obtain a sufficiently good representation of the expectation values.

Estimating the critical temperature T_c based on computed T_c for two different lattice sizes

At the critical temperature T_c, the magnetic system represented by the two dimensional lattice makes a transition from a magnetic system ($T < T_c$), to a system with zero magnetization ($T > T_c$). As a part of the project we want to find the T_c for increasing lattice sizes: $N = 40 \times 40$, $N = 60 \times 60$, $N = 80 \times 80$ and $N = 100 \times 100$. This is done in our

numerical computations by increasing the temperature T successively in small steps $\triangle T$ in the temperature interval $T \in [2.0, 2.5]$ where we know T_c is, and then extracting the temperature which gives the maximum heat capacity C_v. Based on T_c for two different lattice sizes, we can estimate T_c for an infinitely large lattice, see the *Statistichal physics* link in [1]. Through so called finite size scaling relation it is possible to relate T_c of a finite lattice size $L \times L$ with T_c of an infinitely large lattice through the equation

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

where $\nu = 1$. For two different lattice sizes $L = L_1$ and $L = L_2$, where $L_2 > L - 1$, we then have

$$T_c(L_2) - T_c(L_1) = a(L_2^{-1} - L_1^{-1})$$

which gives

$$a = \frac{T_c(L_2) - T_c(L_1)}{L_2^{-1} - L_1^{-1}} \tag{4}$$

Then the estimation of the critical temperature T_c for an infinitely large lattice becomes

$$T_c(L = \infty) = T_c(L) - aL^{-1} \tag{5}$$

Object-Orientation
In order to structure our program we used Object-Orientation, constructing a class called IsingModel. This class has methods setup() and initializeOrder() and intitializeRand(). Which sets up the system's ground state. The two latter sets up the spins in an Orderly or pseudo-Random manner. For preforming the computations the class has the methods compute() which preforms the metropolis method and adds the computed energies and magnetisation for a number of cycles. The metropolis method preforms Metropolis sampling. The results are stored within the IsingModel object and can be written to file using the writeToFile() method. For each temperature we create a new IsingModel Object.

Parallization

To parallize our code we used the "mpi.h" package. We implemented parallization for our temperature interval. Such that each thread gets its

own interval of temperature and creates it's own IsingModel object for each iteration in its interval. This implementation does however not result in parallization within each temperature. But it should be faster than implementing parallization of the metropolis computation interval. Our reason for that is that the threads will be less dependent of each other and only wait for the other threads when the thread has completed its temperature interval. The programs are available at github [2]

3 Results

The 2 $\times$ 2 lattice
In the methods section we showed the analytical expressions for the mean energy <E>, the mean value of the absolute magnetization <|M|>, the specific heat C_v and the susceptibility $\mathcal{X}$ for the 2 $\times$ 2 spin lattice. For the temperature T = 1, we have made a Python script which calculates these values by performing the summation over the 16 spin states. Table (2) shows these values and the similar values obtained by the Metropolis algorithm when using 10^4 and 10^5 Monte Carlo cycles respectively. Increasing the number of cycles to 10^5 improves the numerical results, and our interpretation of them is that the Metropolis algorithm works properly

	Analytic	10^4 MCs	10^5 MCs
<E>	-7.98393	-7.9888	-7.9824
C_v	0.12833	0.08947	0.14049
<M>	3.99464	3.997	3.99422
$\mathcal{X}$	0.01604	0.00679	0.01705

Table 2: Analytical and numerical values of <E>, C_v, <M> and $\mathcal{X}$ for a 2$\times$2 spin lattice with temperature T = 1. The number of Monte Carlo cycles used in the two numerical analysis is 10^4 and 10^5 cycles respectively.

A 20 $\times$ 20 lattice and the most likely state
A series of Monte Carlo cycles can be looked upon as time as it goes by. The size of the lattice is increased to $N = 20 \times 20$ spins, and we study graphically the behavior of the mean energy <E> and the absolute value of the magnetization <|M|> as a function of Monte Carlo cycles, which

represents time. We do this for both an ordered starting spin configuration (all the spins have the value +1) and for a random spin configuration. The analysis are performed for the temperature T = 1.0 and repeated for T = 2.4.

For T =1.0 figure (1) compares <E> with ordered starting configuration and <E> with random starting configuration. Figure (2) takes a close up look at <E> when the starting configuration is ordered. Similar plots are shown for <|M|> in figure (3) and (4). The figures show that <E> and <|M|> quickly becomes stable when the starting configuration is ordered, so quite few Monte Carlo are necessary in order for the system to reach its most likely state. When the starting configuration is random <E> and <|M|> converges towards the most likely state as the number of Monte Carlo cycles (or the number of time steps) increases.

For T = 2.4, figure (5) take a close up look and compare <E> with ordered starting configuration and <E> with random starting configuration. A similar plot are shown for <|M|> in figure (6). The difference between ordered and random starting configuration is not so clear now as for temperature T = 1. The convergence towards the most likely state as the number of Monte Carlo cycles increases seems to be independent of whether the starting configuration is ordered or random. It also actually seems like using the random starting configuration gives convergence towards the most likely state faster when T = 2.4 than when T = 1.

So how many Monte Carlo cycles are needed before an equilibrium situation is reached? Based on plot (1) to (6) our estimation is 25000 Monte Carlo cycles when the starting configuration is ordered. When the starting configuration is random our estimation is also 25000 cycles when T = 2.4. When the starting configuration is random and T = 1 <|M|> does not seem to reach an equilibrium situation before the number of cycles are close to 200000.

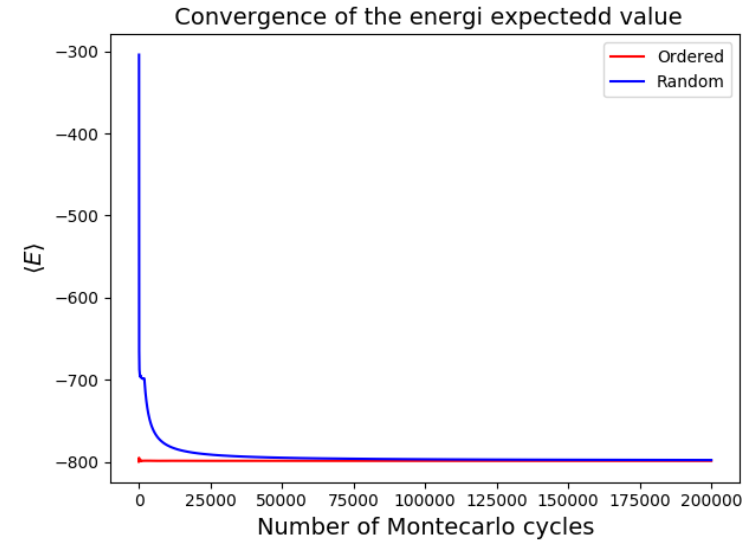

Figure 1: Energy expectation value <E> for ordered and random lattice starting configuration when the temperature T = 1.

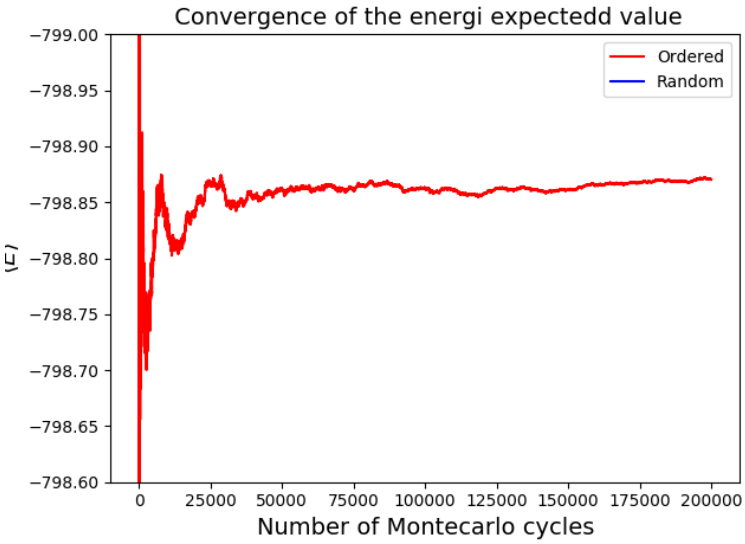

Figure 2: A close up look at <E> when the temperature T = 1 and the lattice starting configuration is ordered.

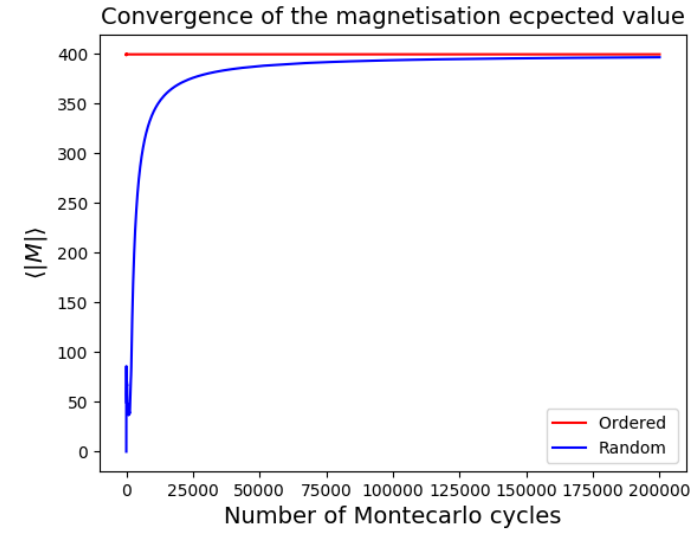

Figure 3: Expectation value of absolute magnetisation < | M | > for ordered and random lattice starting configuration when the temperature T = 1.

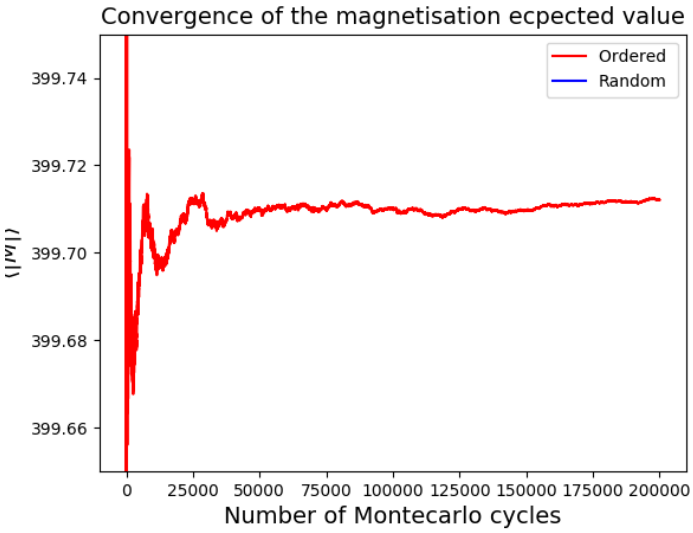

Figure 4: A close up at < | M | > when the temperature T = 1 and the lattice starting configuration is ordered.

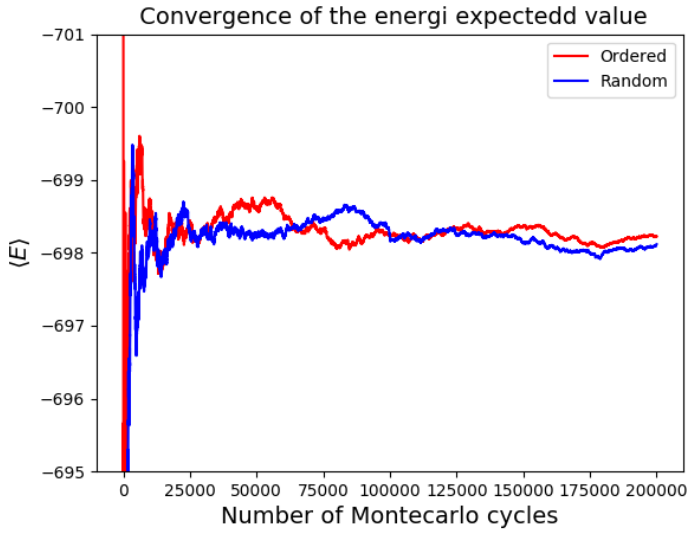

Figure 5: Close up look at energy expectation value <E> for ordered and random lattice starting configuration when the temperature T = 2.4.

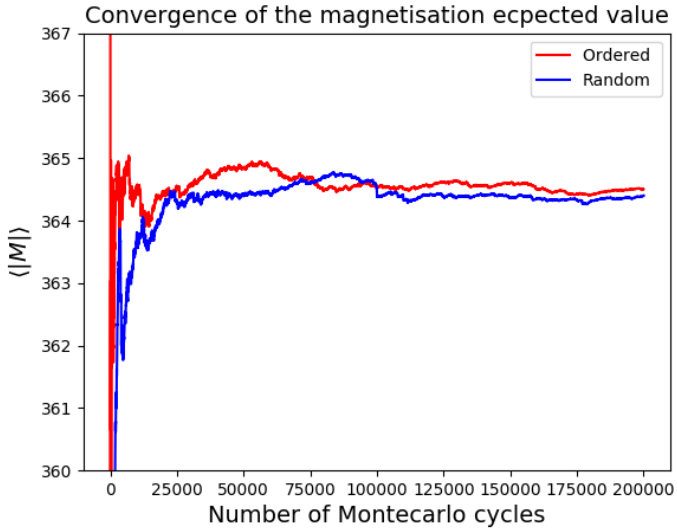

Figure 6: Close up look at expectation value of absolute magnetization <|M|> for ordered and random lattice starting configuration when the temperature T = 2.4.

The 20 $\times$ 20 lattice and the number of accepted spin configurations
In the Methods section, the Metropolis algorithm modelling the Ising

model numerically is described. In order for a spin flip and a new energy configuration to be accepted,

$$r \leq e^{-(\triangle E/T)} \tag{6}$$

where $\triangle E$ is the energy difference between the new and the old spin configuration and r is a random number in the interval $r \in [0,1]$. For a $N = 20 \times 20$ lattice, one Monte Carlo cycle includes 400 possible spin flips which leads to new energy configurations. Figure (7) and figure (8) show how many of these possible spin flips/ energy configurations which are accepted by equation (6) as a function of number of Monte Carlo cycles for temperature T = 1 and T = 2.4 respectively. The two figures show a linear connection between the number of Monte Carlo cycles and the number of accepted spin flips/energy configurations. The linear curve is steeper when T=2.4 though, indicating that a larger number of of spin flips/ energy configurations are accepted when the temperature rises. For T = 1, more of the spin flips/ energy configurations are accepted when the starting spin configuration is random instead of ordered. For T = 2.4 the starting spin configuration does not affect the result.

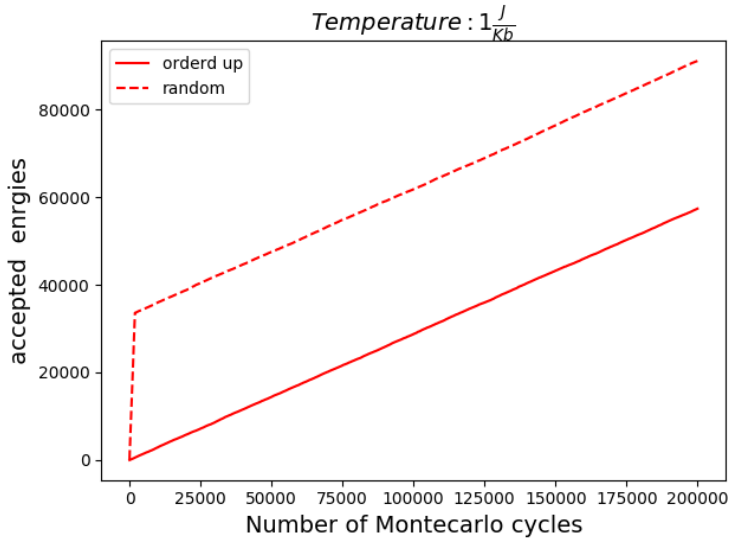

Figure 7: Number of accepted spin flips/energy configurations as a function of number of Monte Carlo cycles for ordered and random starting configuration and temperature T = 1.

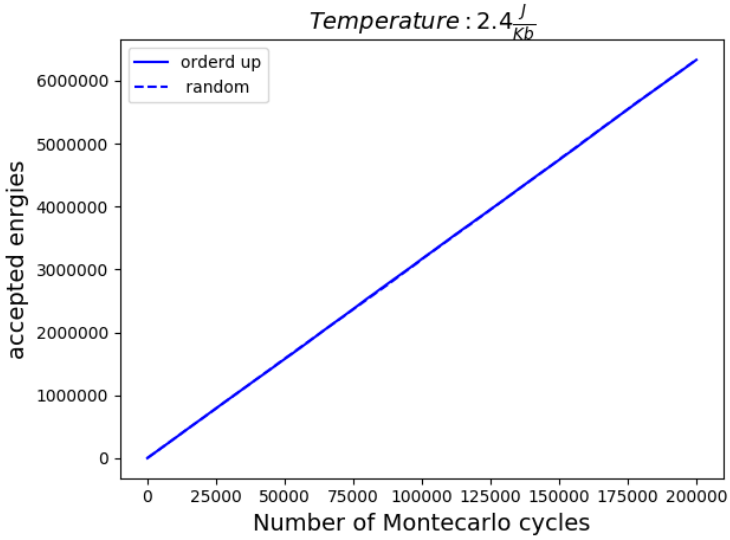

Figure 8: Number of accepted spin flips/energy configurations as a function of number of Monte Carlo cycles for ordered and random starting configuration and temperature T = 2.4.

Probability distribution of Monte Carlo cycle energy P(E)
The lattice size is still $N = 20 \times 20$ spins. When using a large number of Monte Carlo cycles in a numerical computation, P(E) is the probability that one Monte Carlo cycle calculates the lattice energy E. P(E) is calculated by counting the number of times each energy E appears in the computation and then divide by the total number of Monte Carlo cycle in the computation. P(E) is calculated for both T = 1.0 and T = 2.4, with both ordered and random starting configuration of the lattice. Only cycles after the steady state situation has been reached are included in the calculation of P(E). A total number of 200000 cycle are used in the calculations. The first 50000 of them are not included in the calculation of P(E) when T = 1 and the first 100000 of them are not included when T = 2.4. The number 50000 and 100000 are chosen based on the graphs in figure (2), figure (4) and figure (5).

Figure (9) and figure (11) show the distribution of 150000 Monte Carlo cycles with respect to which lattice energy each of them compute, when the temperature T = 1. In figure (9) the starting configuration is ordered and in figure (11) the starting configuration is random. P(E)=(histogram value)/ 150000. The random starting configuration gives a slightly wider distribution than the ordered starting configuration.

Figure (10) and figure (12) show the distribution of 100000 Monte Carlo cycles with respect to which lattice energy each of them compute, when the temperature T = 2.4. In figure (10) the starting configuration is ordered and in figure (12) the starting configuration is random. P(E)=(histogram value)/ 100000. For T = 2.4, the starting configuration hardly has an influence on the distribution.

The big difference is between T = 1 and T = 2.4. T = 2.4 gives a much wider distribution of energies than T= 1. This corresponds with the computed variance $Var(E) = \sigma_E^2$ in energy, which can be interpreted at the expected spread of E around <E>. When the starting configuration is random and the temperature T = 1, $Var(E) = 9.3$. When the temperature is increased to T = 2.4, $Var(E) = 292$. For the ordered starting configuration of the lattice, $Var(E)$ is not computed

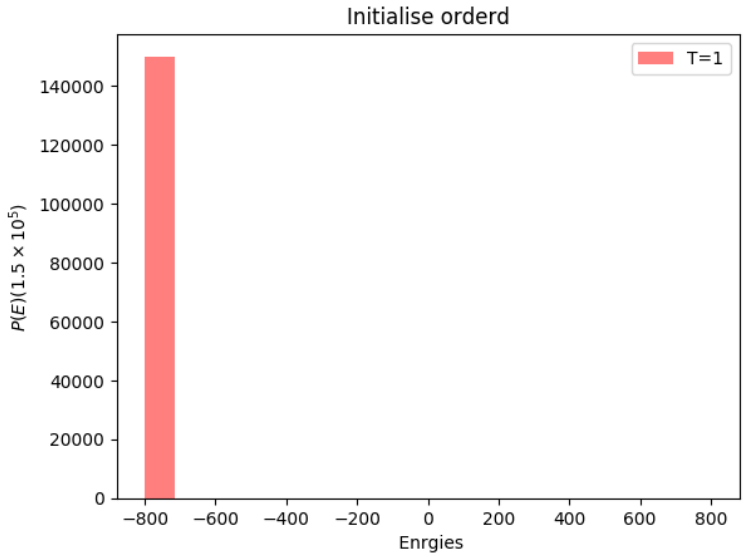

Figure 9: Distribution of the number of Monte Carlo cycles computing energies in particular energy intervals. The temperature T = 1 and the starting configuration is ordered. The total number of Monte Carlo cycles are 150000. P(E) = (histogram value)/ 150000.

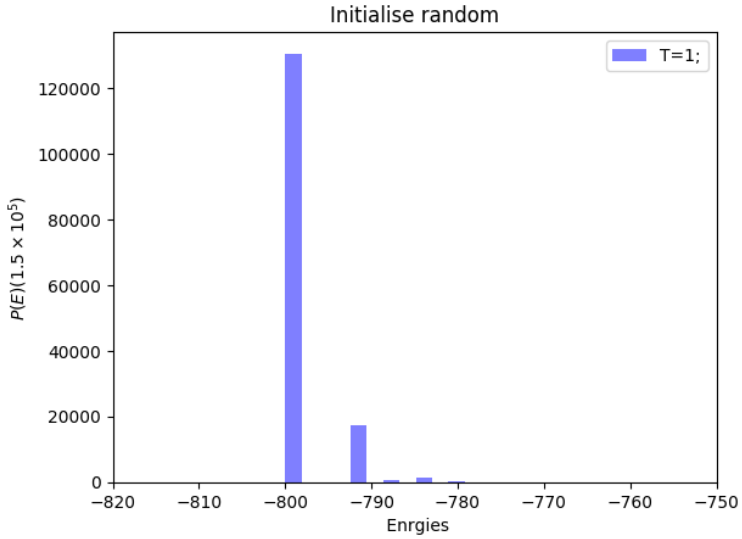

Figure 10: Distribution of the number of Monte Carlo cycles computing energies in particular energy intervals. The temperature T = 1 and the starting configuration is random. The total number of Monte Carlo cycles are 150000. P(E) = (histogram value)/ 150000.

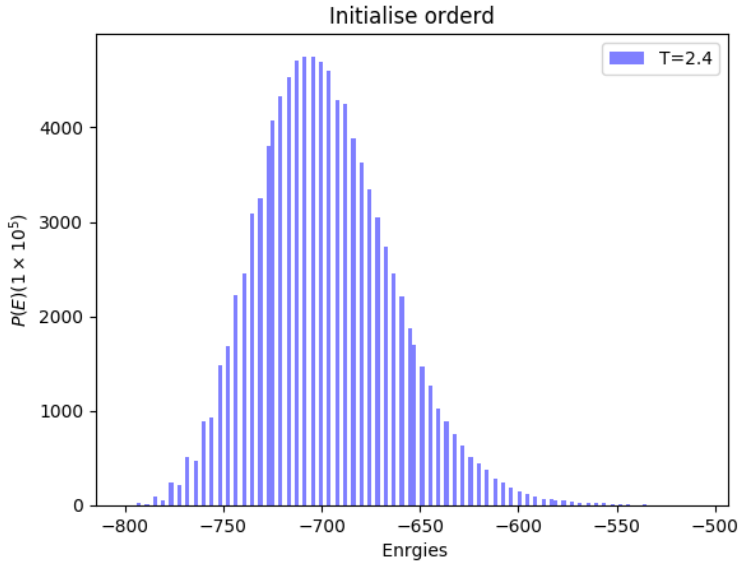

Figure 11: Distribution of the number of Monte Carlo cycles computing energies in particular energy intervals. The temperature T = 2.4 and the starting configuration is ordered. The total number of Monte Carlo cycles are 100000. P(E) = (histogram value)/ 100000.

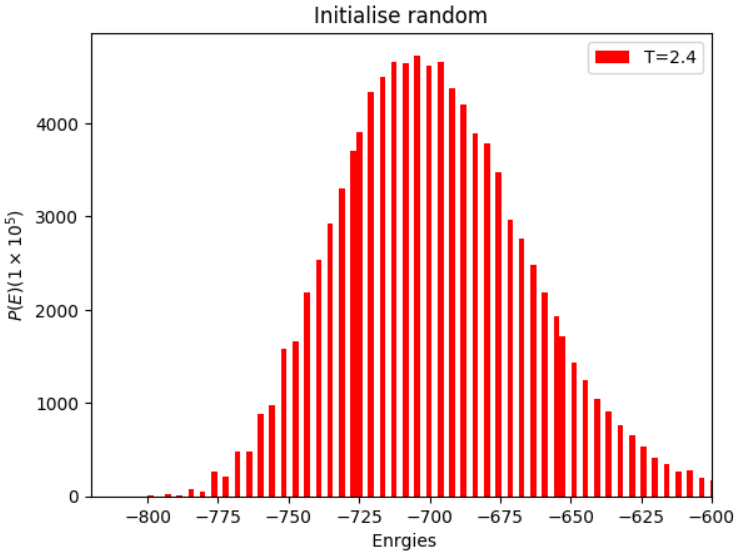

Figure 12: Distribution of the number of Monte Carlo cycles computing energies in particular energy intervals. The temperature T = 2.4 and the starting configuration is random. The total number of Monte Carlo cycles are 100000. P(E) = (histogram value)/ 100000.

Phase transition of 2D system

In figure 13 we can see the results for heat capacity C_v, expected energy $< E >$, susceptibility $\mathcal{X}$ and expected absolute magnetisation per spin $<|M|>$ for different temperatures in the interval $T \in [2.0, 2.5]$. In order to be able to compare these four variables for different quadratic lattice sizes, we have divided them with number of spins in the lattice

In the figure we can see for C_v that we have different critical temperatures for the different lattice sizes, while <E> is almost linear and roughly the same for all the lattice sizes. For $\mathcal{X}$ we can see that susceptibility turns positive and we thus obtain distinct critical temperatures. These critical temperatures are also shown in table 3. The plot of $<|M|>$ shows that a larger lattice size corresponds to a sharper fall in magnetisation.

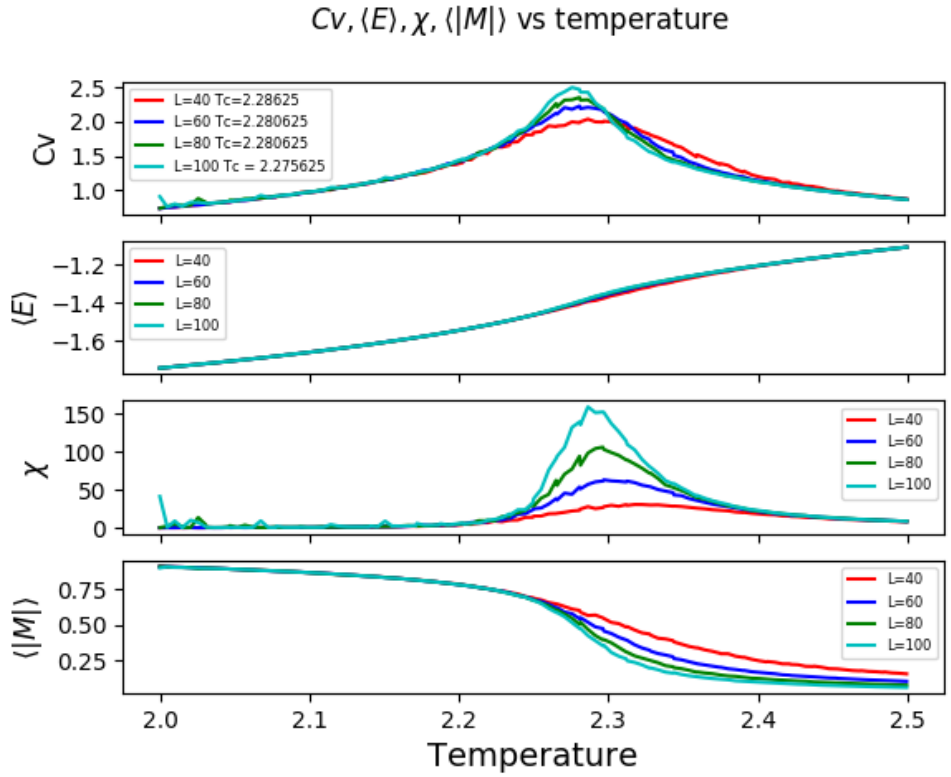

Figure 13: Shows the heat capacity Cv, expected energy $\langle E \rangle$, susceptibility χ, absolute magnetisation $\langle |M| \rangle$ per spin vs temperature plotted for the different lattice sizes of 40, 60, 80 and 100.

Table 3: shows the critical temperature for heat capacity and susceptibility for different lattice sizes

Lattice size	**Critical Temperature, Cv**	**Critical Temperature, χ**
40	2.28625	2.29925
60	2.280625	2.29925
80	2.280625	2.29325
100	2.275625	2.168

Critical temperatures

For each lattice size we find the critical temperature from the data set by first finding the largest C_v value. Its corresponding temperature value is then the critical temperature for this particular lattice. T_c. We used these data points points of critical temperature vs lattice size to fit a line to the

obtained data points using least square polynomial fit provided by numpy.polyfit(). And using this line interpolate from 0 to 500 as shown in figure 14. We can see that our interpolation line does not converge towards the infinite critical temperature ≈ 2.269. By using the equations 4 5 with $L = 100$ we where able to calculate the critical temperature more accurately, $T_c(L = \infty) = 2.2681$.

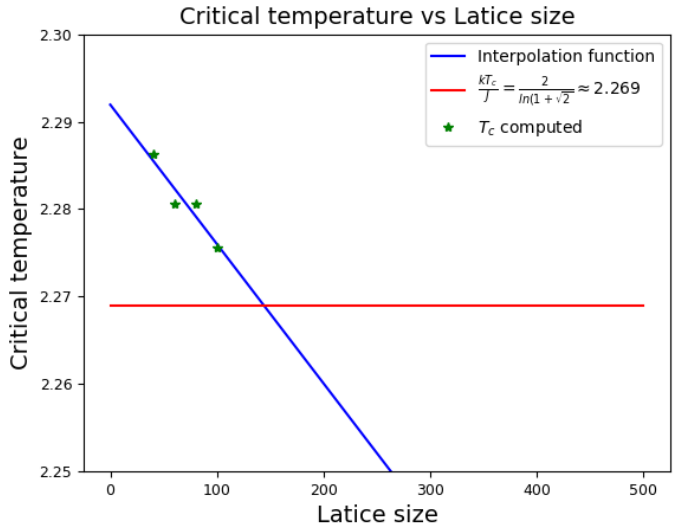

Figure 14: Shows the computed critical temperature for different lattice dimensions, and an interpolation based on these points, together with the critical temperature by Lars Onsager

Compiler flag Optimization

By running our program after using different compiler flags we got the results as shown in figure 15. Here we see a clear distinction in computation time between running the program after using optimization flags and no optimization flag. Running with no optimization takes close to the double amount of time, so we can see that there is a lot of time to be saved here. The difference between the optimization flags is smaller however. With the $-O3$ being the fastest then the $-O2$, the $-O$.

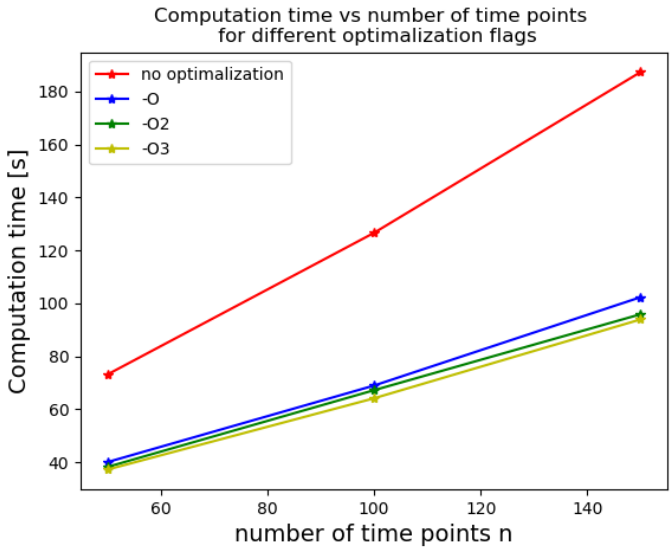

Figure 15: Shows computation time of our program for different number of points in the time interval using different compiler flags. Using a 20×20 lattice and 200,000 cycle computations and 16 threads.

4 Discussion & Conclusion

We've simulated various system sizes by using the Ising model.
we used the 2×2-lattice to confirm our implementation of the metropolis algorithm on the Ising model. By comparing its result towards the analytical solution we verified its correctness.

For the 20×20-lattice after a certain number of Monte carlo cycles the results stabilized on a certain value. Furthermore we saw that a temperatures closer to the critical temperature results in a larger spread or variance of possible energies. It would be interesting to see the possible energy distribution closer to the critical temperatures found to either support or discard this observation.

In our phase transition 2D results we can see that we obtain critical temperatures for heat capacity and susceptibility. However, in order to determine the critical temperature more accurately we should have more temperature points to make the tops more defined. This would result in a significant increase in computation time, though we could have reduced the temperature interval to $T \in [2.2, 2.4]$ perhaps. The results does show

that when the magnetisation begins to fall the susceptibility increases correspondingly. We can also see that larger systems or lattices have a sharper fall in magnetisation and an increasing rise in susceptibility. The increase in susceptibility indicates a change from a ferromagnetic material into a paramagnetic material. These results indicate a phase transition.

We could see that our interpolation based on our computed critical temperatures does not converge towards the infinite critical temperature. This is probably due to the small amount of temperature points. In order for infinite lattice critical temperature the interpolation line should be nearly parallel to the infinite lattice critical temperature. However by using the analytical phase transition equations we obtained a much more accurate result for infinite lattice critical temperature.

Our efforts to optimize the program using compiler flags seems successful as we saw in our results we obtained nearly half the computing time when using compiler flags.

References

[1] Morten Hjort-Jensen: *Course material in Computational Physics.*
http://compphysics.github.io/ComputationalPhysics/doc/web/course

[2] Github: *Programs developed for this project*
https://github.uio.no/petteala/FYS4150/blob/master/projects/project4