FYS3150
Project 4

Egil S. Køller

November 22, 2018

Abstract

In this project, we have used a Markov Chain Monte Carlo method with the Metropolis algorithm to study the Ising model in two dimensions. This model undergoes a phase transition at a critical temperature $kT_C/J = \frac{2}{\ln(1+\sqrt{2})} \approx 2.269$. Our estimate, through Monte Carlo calculations, matches this exact result ($kT_C/J = 2.276 \pm 0.027$).

We have also looked at how the system evolves towards equilibrium, which is an important aspect of Markov Chain Monte Carlo methods. To get enough data, we have depended on parallelisation and powerful computing servers.

1 Introduction

Monte Carlo methods are one of the most widely used algorithms in science, and have also found its way into fields like economics and finance [2]. The Metropolis algorithm, one of the central algorithms in Monte Carlo computations, has been ranked as the most influential algorithm in science and engineering of the 20th century [1]. In this project, we will apply Monte Carlo computations with the Metropolis algorithm on the Ising model in two dimensions with periodic boundary conditions.

The Ising model was introduced as a statistical mechanical model for magnetic materials, trying to explain the magnetic properties in terms of the interactions between the spins of the electrons. In a more general form, the model can be applied to statistical simulations in other fields of both natural and social science [4]. Unlike in the one-dimensional case (which was solved by Ising in 1924), in the two-dimensional case a phase transition from ferro- to paramagnetic occur at a critical temperature (showed by Onsager in 1944) [4]. In this project, we will study this phase transition by looking at how thermodynamic properties like the magnetic moment and the magnetic susceptibility changes with temperature.

For the one- and two-dimensional Ising model, there are analytical solutions to expectation values of several thermodynamic properties. We will use the analytical results for a 2x2 lattice to benchmark our numerical results.

Given Monte Carlo methods' hunger for data, the computations are quite demanding. To speed up the process, we will parallelise our code, which is effortlessly done when running Monte Carlo simulations.

2 Theory

2.1 Ising Model with 2x2 lattice

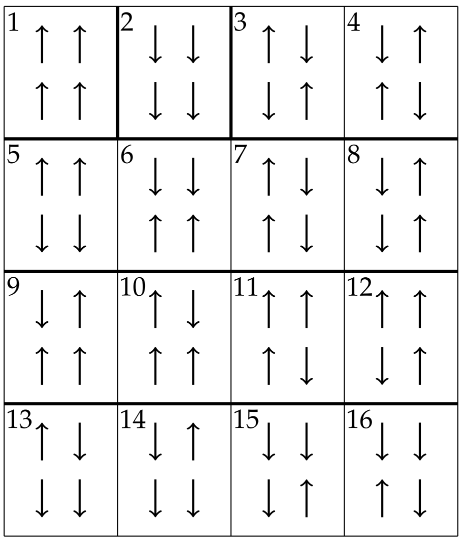

Figure 1: The different microstates for a 2x2 lattice

In the simplest form of the Ising model, we have a lattice where the objects at each lattice site can take only one of two values. In addition, each object interacts with its nearest neighbours only.

The energy of this system (without an external magnetic field) is given by

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

where $J > 0$ is the coupling constant (with ferromagnetic ordering), N is the number of spins, $s_i = \pm 1$ is the value of the spin i, and $< kl >$ indicates that we sum over nearest neighbours only. In the 2x2 case, we get that:

$$E = -J\left(2s_{11}s_{12} + 2s_{11}s_{21} + 2s_{12}s_{22} + 2s_{21}s_{22}\right) \tag{1}$$

The magnetisation, or magnetic moment, is given by

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

Applying equation 1 and 2 to the different microstates in Figure 1 (with periodic boundary conditions), we get the results given in Table 1.

We want to calculate the expectation values of the energy and the magnetisation, and the heat capacity and the magnetic susceptibility. We start of by calculating the partition function, which is given by:

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

Fig. 1	# of spins up	Degeneracy	Energy	Magnetisation
1	4	1	$-8J$	4
2	0	1	$-8J$	-4
3-4	2	2	$8J$	0
5-8	2	4	0	0
9-12	3	4	0	2
13-16	1	4	0	-2

Table 1: Energy and magnetisation for the microstates in Figure 1

where N now represents the number of microstates, E_i is the energy of the microstate i, and $\beta = \frac{1}{kT}$.

In the 2x2 case, we get that:

$$Z = 2e^{8J\beta} + 12e^{0} + 2e^{-8J\beta} = 2e^{8J\beta} + 2e^{-8J\beta} + 12$$

The expectation values of the energy and the magnetisation are given by:

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

Through thermodynamical considerations, it can be shown that the heat capacity and the magnetic susceptibility is related to the variance of the energy and magnetisation in the following way:

$$C_V = \frac{\sigma_E^2}{kT^2} = \frac{1}{kT^2}\left(\langle E^2 \rangle - \langle E \rangle^2\right)$$
$$\chi = \frac{\sigma_M^2}{kT} = \frac{1}{kT}\left(\langle M^2 \rangle - \langle M \rangle^2\right)$$

In this project, we will use the absolute value of M in the calculations of χ.

Omitting the (somewhat boring) algebra, we end up with the following expressions:

$$\begin{aligned}
\langle E\rangle &= 8J\frac{(e^{-8J\beta}-e^{8J\beta})}{e^{8J\beta}+e^{-8J\beta}+6}\\
\langle M\rangle &= 0\\
\langle |M|\rangle &= 4\frac{e^{8J\beta}+2}{e^{8J\beta}+e^{-8J\beta}+6}\\
C_V &= \frac{128J^2}{kT^2}\frac{3e^{8J\beta}+3e^{-8J\beta}+2}{\left(e^{8J\beta}+e^{-8J\beta}+6\right)^2}\\
\chi &= \frac{16}{kT}\frac{3e^{8J\beta}+e^{-8J\beta}+3}{\left(e^{8J\beta}+e^{-8J\beta}+6\right)^2}
\end{aligned}$$

2.2 Phase transitions and critical temperature

Near the critical temperature T_C, the behaviour of the thermodynamic quantities we are looking at can be characterised by different power laws. For the magnetisation, we have that (below T_C):

$$\langle M(T)\rangle \sim (T_C - T)^{\beta}$$

where $\beta = \frac{1}{8}$ is the so-called critical exponent. This behaviour corresponds to a loss of the spontaneous magnetisation above T_C.

We have similar expressions for the heat capacity and the magnetic susceptibility:

$$\begin{aligned}
\chi(T) &\sim |T_C - T|^{\gamma}\\
C_V(T) &\sim |T_C - T|^{\alpha}
\end{aligned}$$

where $\gamma = -\frac{7}{4}$ and $\alpha = 0$. We see that χ diverges near T_C.

The Ising model undergoes a second-order phase transition, which is characterised by a correlation length that spans the whole system at the critical point. The correlation length is the length at which the spins are correlated, and is of the order of the lattice spacing for $T >> T_C$. As the temperature approaches T_C the correlation will increase, and the behaviour of the correlation length near T_C can be characterised by the power law

$$\xi \sim |T_C - T|^{\nu}$$

where $\nu = -1$.

Since we are limited to a finite lattice in our numerical calculations, and ξ is proportional to the lattice size at the critical point, T_C will depend on the lattice size. Through finite size scaling relations, it is possible to relate the critical temperature of a infinite lattice to the results for finite lattices:

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

This can be rewritten as

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

We see that if we plot $T_C(L)$ versus L^{-1} and do a linear fitting, $T_C(\infty)$ will be given by the y-intercept. We will use the divergence of χ near T_C to estimate $T_C(L)$ for different values of L.

2.3 Monte Carlo with Metropolis

Monte Carlo methods are a wide family of statistical simulation methods utilising random sampling. In physics, these methods are used to simulate physical processes without the need to know the differential equations describing the system. However, the methods depend on a probability distribution function describing the system, as well as random numbers and a sampling rule. For the two-dimensional Ising model, our Monte Carlo procedure goes as follows:

1. Initialise the lattice in a configuration with energy E_{init}, either with all spins in the same direction or with a random orientation of the spins.
2. Select a random position in the lattice, and change the previous configuration by flipping the spin at this position.
3. Calculate the energy difference ΔE (see 2.4).
4. Compare $w = e^{-\beta \Delta E}$ with a random number $r \in [0, 1]$. If $r \leq w$, accept the new configuration. If not, keep the old configuration.
5. Update the various expectation values by adding the contribution by the current configuration.
6. Repeat step 2-5 for a given number of Monte Carlo cycles. Each sweep through the lattice constitutes a Monte Carlo cycle.
7. Divide the expectation values by the number of Monte Carlo cycles and the number of spins.

Step 4. is perhaps the most crucial one, and requires some further discussion. This step is the Metropolis algorithm, which defines our sampling rule.

The Metropolis algorithm follows from the theory of Markov Chains, and is used when we want to look at physical systems that evolves towards equilibrium. The changes in the configuration of the lattice can be described as time-steps in a Markov Chain. The time development of the systems PDF is then given by

$$w_i(t+1) = \sum_j W_{ij} w_j(t)$$

or

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

where $w_i(t)$ is the time dependent PDF of the microstate i, W_{ij} is the transition probability of going from the microstate j to i, and $\hat{w}$ and $\hat{W}$ is the vector and matrix representation of these probabilities. If $|\hat{w}(t+1) - \hat{w}(t)| \to 0$, we have reached the steady state, or equilibrium.

Next, we model the transition probability in the following way:

$$W_{ij} = T_{ij} A_{ij}$$

where T_{ij} is the probability of making a move from j to i (suggesting moves), and A_{ij} is the probability of accepting the move (accepting or rejecting moves). We can now express the time development of w_i as

$$w_i(t+1) = \sum_j \left[w_j(t) T_{ij} A_{ij} + w_i(t) T_{ji} (1 - A_{ji}) \right]$$

By using the fact that the probabilities are normalised, this can be rewritten as

$$w_i(t+1) - w_i(t) = \sum_j \left[w_j(t) T_{ij} A_{ij} - w_i(t) T_{ji} A_{ji} \right]$$

It can be shown that $w_i(t+1) = w_i(t) = w_i$ in the limit $t \to \infty$. We then have that

$$\sum_j w_j(t) T_{ij} A_{ij} = \sum_j w_i(t) T_{ji} A_{ji}$$

To avoid cyclic solutions, we apply the condition of detailed balance:

$$W_{ij} w_j = W_{ji} w_i \quad \Rightarrow \quad w_j(t) T_{ij} A_{ij} = w_i(t) T_{ji} A_{ji} \quad \Rightarrow \quad \frac{T_{ij} A_{ij}}{T_{ji} A_{ji}} = \frac{w_i}{w_j}$$

Now, it's time to introduce the PDF of our system, which is the Boltzmann distribution:

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

With this distribution, the condition of detailed balance results in

$$\frac{T_{ij} A_{ij}}{T_{ji} A_{ji}} = e^{-\beta(E_i - E_j)} = e^{-\beta \Delta E}$$

We see that the partition function disappears, which means that we avoid some heavy calculations.

If we assume that the probability T is symmetric, that is $T_{ij} = T_{ji}$, we get that

$$\frac{A_{ij}}{A_{ji}} = e^{-\beta \Delta E} \tag{3}$$

The next step is to model the acceptance probability A. From the Boltzmann distribution, we see that the states with the lower energies are the most probable. But we can't accept moves to lower energies only, because this would violate the condition of ergodicity, which states that it should be possible to reach every possible state of the system from any given starting point during a Markov process.

This can be solved in the following way: If $E_i - E_j = \Delta E < 0$, we set $A_{ij} = 1$. From 3, we then get that $A_{ji} = e^{\beta \Delta E}$. In other words:

$$A_{ij} = \begin{cases} e^{-\beta \Delta E}, & \text{if } \Delta E > 0 \\ 1, & \text{else} \end{cases}$$

The moves to higher energies are implemented by comparing the factor $w = e^{-\beta \Delta E}$ to a random number $r \in [0, 1]$, as seen in the layout of our Monte Carlo procedure. When $\Delta E < 0$, we have that $w > r$.

2.4 Changes in Energy and Magnetisation

One of the crucial steps of our algorithm is the efficient calculation of the changes in energy and magnetisation. As a result of the fact that the change from one configuration to another is given by simply flipping one spin, there is a limited set of possible energy differences. The neighbouring spins of the spin we flip can be arranged in five different ways (all in the same direction, 3 in the same direction, etc.), and the corresponding energy differences can easily be calculated in a similar fashion to what we did in section 2.1. This enables us to precalculate $e^{-\beta\Delta E}$, and limit the calls to the exponential function.

We still have to calculate the energy and magnetisation differences after each flip. Again, we exploit the fact that we only flip one spin at the time. The energy difference is given by

$$\Delta E = E_2 - E_1 = J \sum_{<kl>}^{N} s_k^1 s_l^1 - J \sum_{<kl>}^{N} s_k^2 s_l^2 = -J \left(\sum_{<kl>}^{N} \left(s_k^2 s_l^2 - s_k^1 s_l^1 \right) \right)$$

Since we only flip s_l (i.e. $s_k^1 = s_k^2$), this can be rewritten as

$$\Delta E = -J \sum_{<kl>}^{N} s_k \left(s_l^2 - s_l^1 \right)$$

where we only sum over the nearest neighbours of s_l. Since $s_l = \pm 1$, we have the following possibilities:

$$\begin{aligned} s_l^1 = 1 \rightarrow s_l^2 = -1 \quad &\Rightarrow \quad s_l^2 - s_l^1 = -2 \\ s_l^1 = -1 \rightarrow s_l^2 = 1 \quad &\Rightarrow \quad s_l^2 - s_l^1 = 2 \end{aligned}$$

This gives us the following expression for ΔE:

$$\Delta E = 2J s_l^1 \sum_{<kl>}^{N} s_k$$

Since we only flip s_l, the change in magnetisation is simply given by

$$\Delta M = s_l^2 - s_l^1 = \pm 2$$

which can be rewritten as

$$\Delta M = 2 s_l^2$$

3 Implementation

The programs used in this project can be found on `https://github.com/egilsk/fys3150x/tree/master/project4`.

The C++-programs are based on the program "IsingModel.cpp" in the course github repository [3]. We have utilised the C++ library Armadillo [5] [6] for vectors and matrices. We have also used the Numpy functions *polyfit* and *polyval* to perform linear fittings.

4 Results

(All expectation values are given per spin!)

# of MC cycles	E [J]	M [1]	C_V [J^2/kT^2]	χ [1/kT]
1000	-2.00 ± 0.08	1.00 ± 0.02	0.03 ± 0.04	0.001 ± 0.006
10 000	-2.00 ± 0.09	1.00 ± 0.03	0.03 ± 0.01	0.004 ± 0.002
100 000	-2.00 ± 0.09	1.00 ± 0.03	0.033 ± 0.004	0.0041 ± 0.0006
1 000 000	-2.00 ± 0.09	1.00 ± 0.03	0.032 ± 0.001	0.0040 ± 0.0002
10 000 000	-2.00 ± 0.09	1.00 ± 0.03	0.0321 ± 0.0003	0.00401 ± 0.00005
Analytic results	-1.996	0.9987	0.03208	0.004011

Table 2: Expectation values after 100 runs for a 2x2 lattice at T = 1.0 (in units of kT/J) for different numbers of MC cycles.

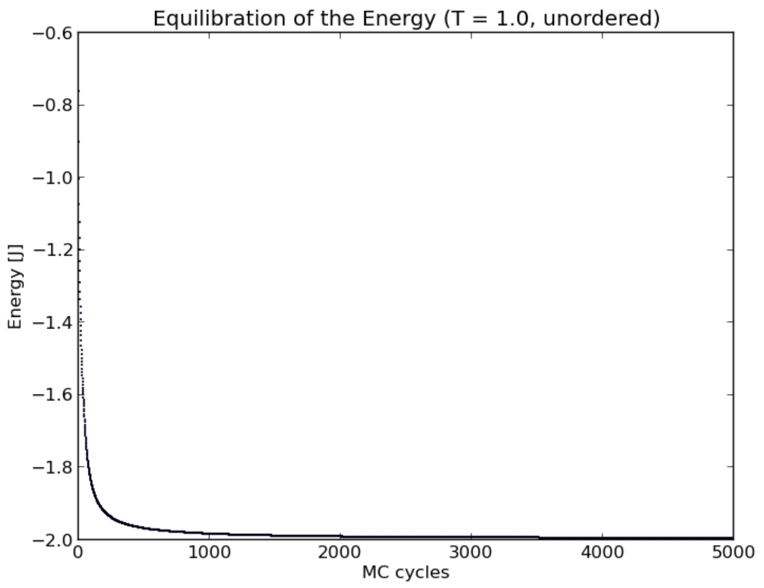

Figure 2: Mean energy as a function of the number of MC cycles, with T = 1.0 (in units of kT/J) and a disordered initial configuration (L = 20).

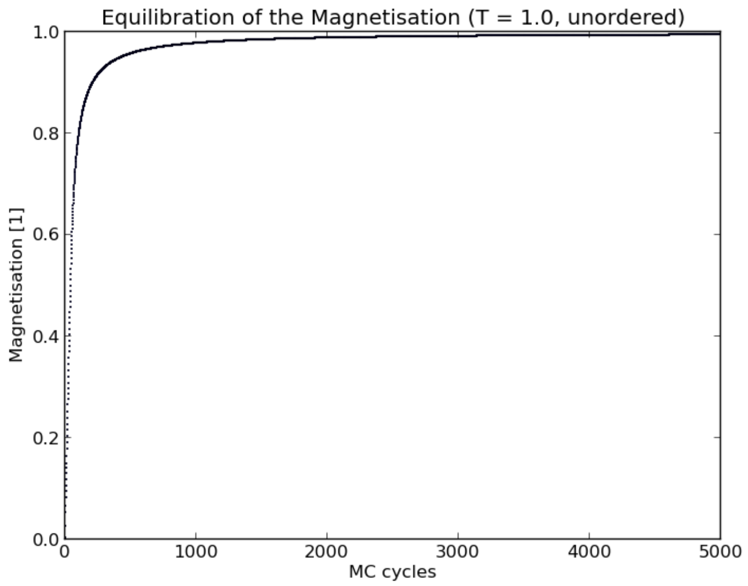

Figure 3: Mean magnetisation as a function of the number of MC cycles, with T = 1.0 (in units of kT/J) and a disordered initial configuration (L = 20).

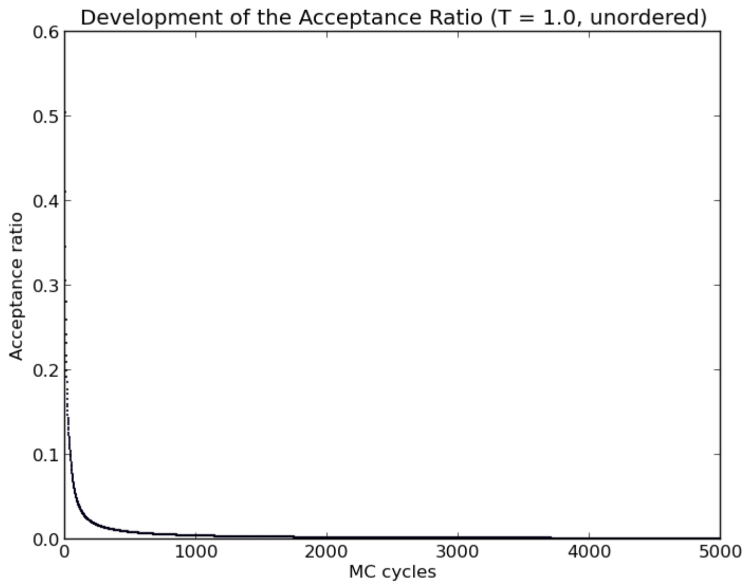

Figure 4: Acceptance ratio as a function of the number of MC cycles, with T = 1.0 (in units of kT/J) and a disordered initial configuration (L = 20).

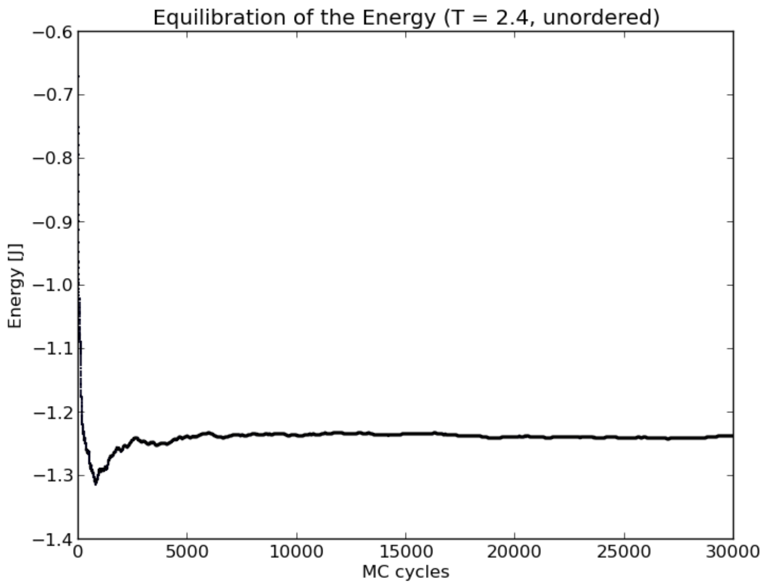

Figure 5: Mean energy as a function of the number of MC cycles, with T = 2.4 (in units of kT/J) and a disordered initial configuration (L = 20).

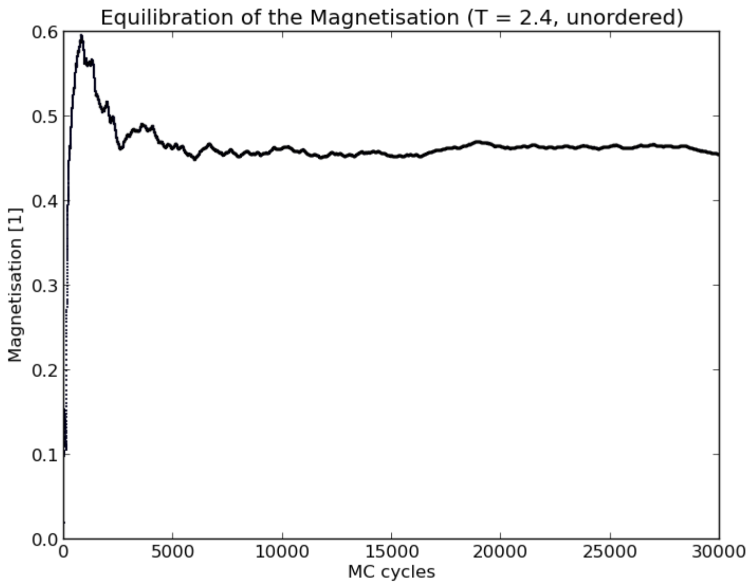

Figure 6: Mean magnetisation as a function of the number of MC cycles, with T = 2.4 (in units of kT/J) and a disordered initial configuration (L = 20).

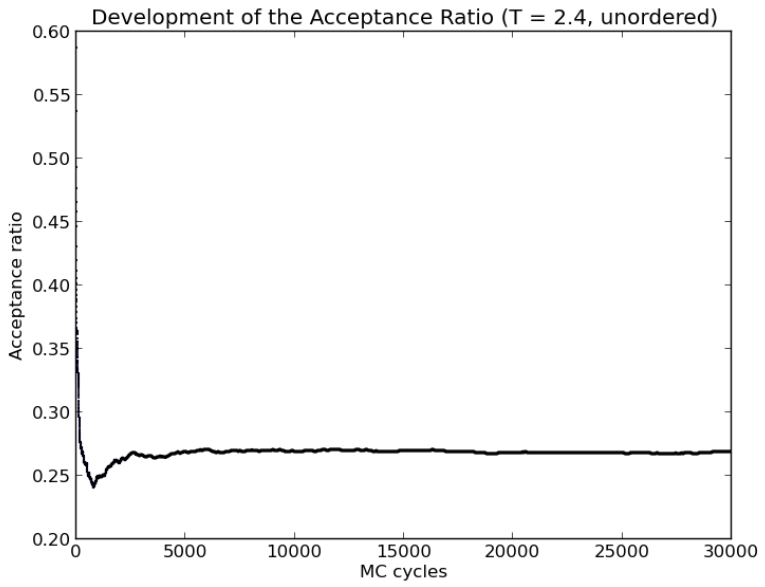

Figure 7: Acceptance ratio as a function of the number of MC cycles, with T = 2.4 (in units of kT/J) and a disordered initial configuration (L = 20).

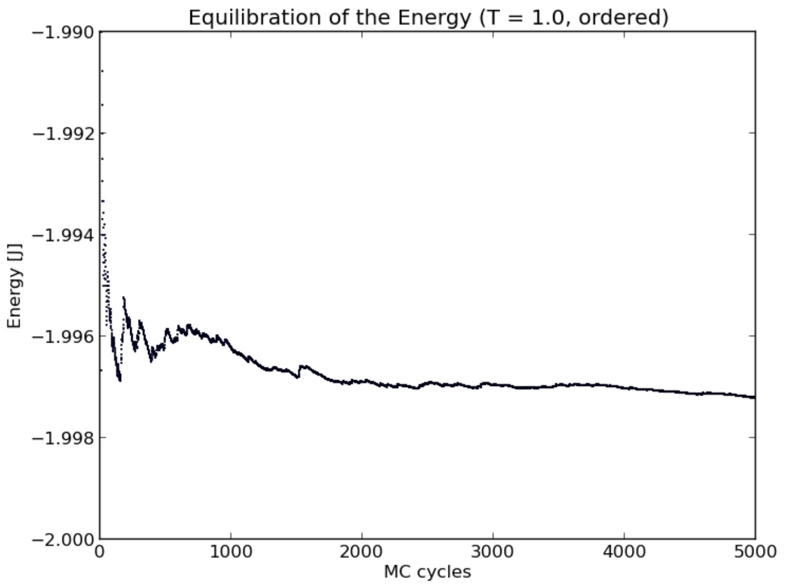

Figure 8: Mean energy as a function of the number of MC cycles, with T = 1.0 (in units of kT/J) and an ordered initial configuration (L = 20).

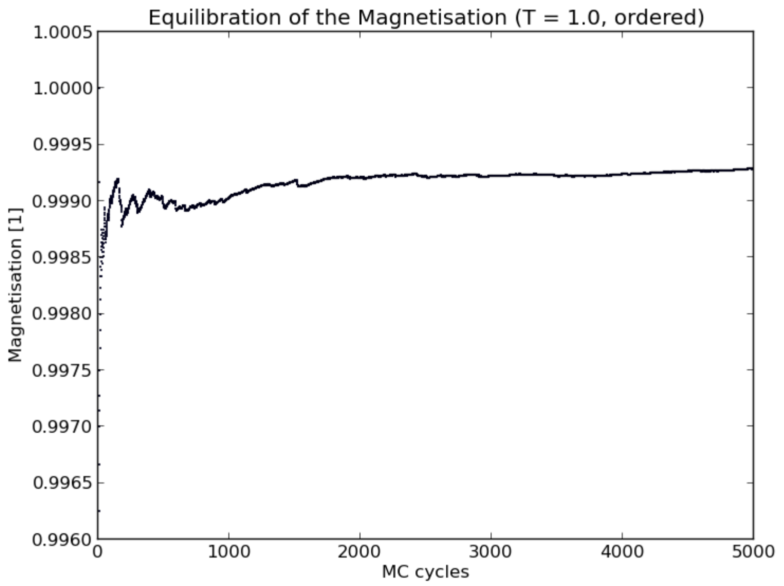

Figure 9: Mean magnetisation as a function of the number of MC cycles, with T = 1.0 (in units of kT/J) and an ordered initial configuration (L = 20).

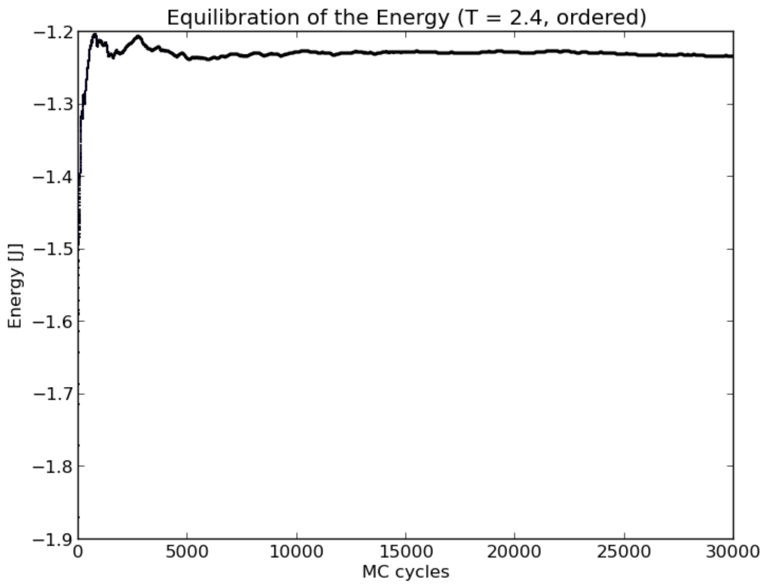

Figure 10: Mean energy as a function of the number of MC cycles, with T = 2.4 (in units of kT/J) and an ordered initial configuration (L = 20).

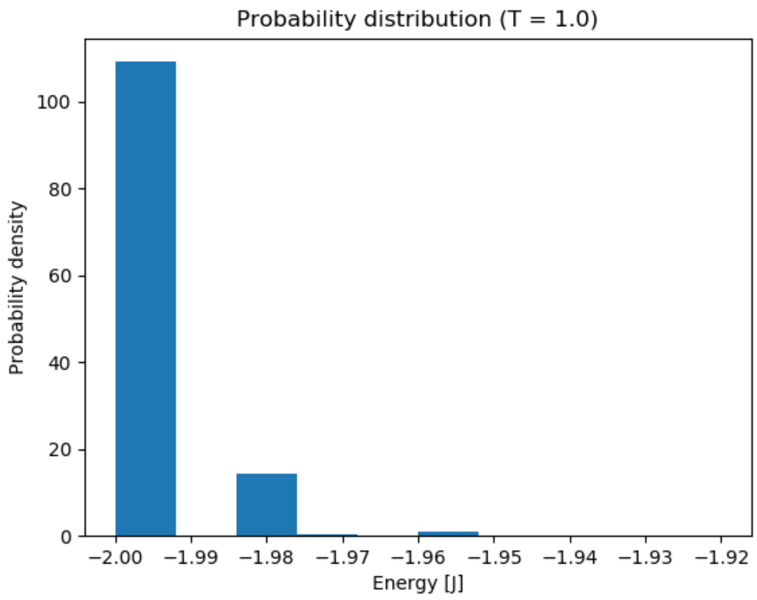

Figure 11: Probability distribution P(E), with T = 1.0 (in units of kT/J) (100 000 MC cycles, 10 000 equilibration cycles, L = 20). $\sigma_E = 0.008$.

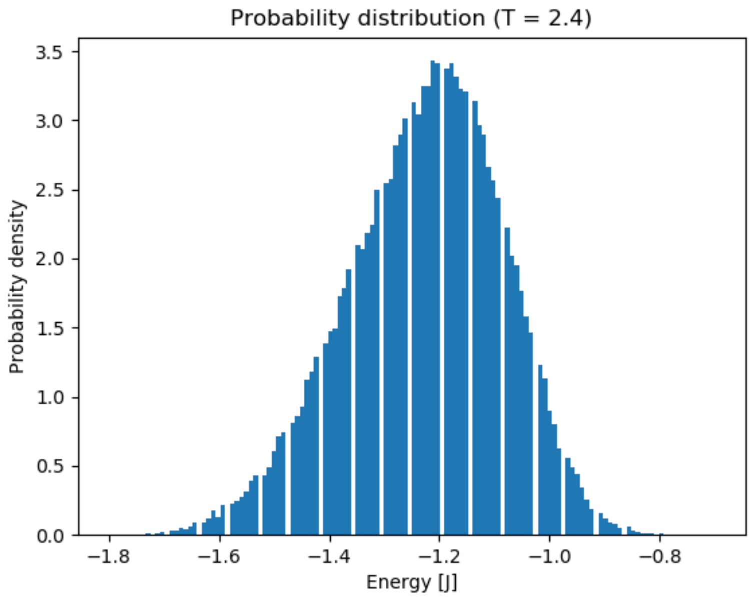

Figure 12: Probability distribution P(E), with T = 2.4 (in units of kT/J) (100 000 MC cycles, 10 000 equilibration cycles, L = 20). $\sigma_E = 0.14$.

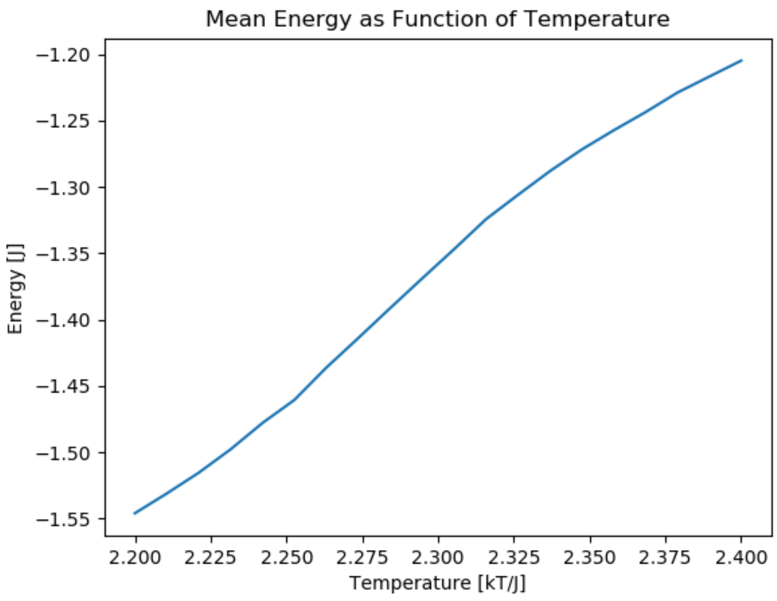

Figure 13: Mean energy as a function of temperature (1 000 000 MC cycles, 10 000 equilibration cycles, 20 temperatures, L = 60).

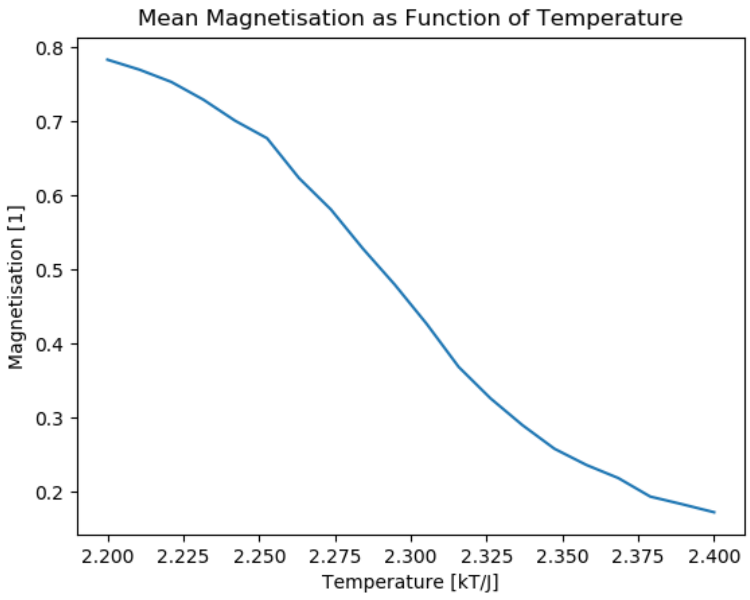

Figure 14: Mean magnetisation as a function of temperature (1 000 000 MC cycles, 10 000 equilibration cycles, 20 temperatures, L = 60).

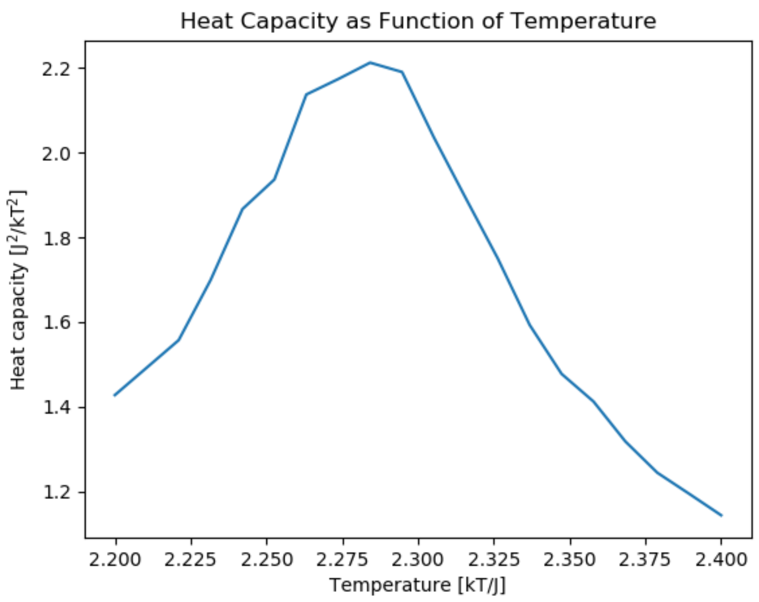

Figure 15: Heat capacity as a function of temperature (1 000 000 MC cycles, 10 000 equilibration cycles, 20 temperatures, L = 60).

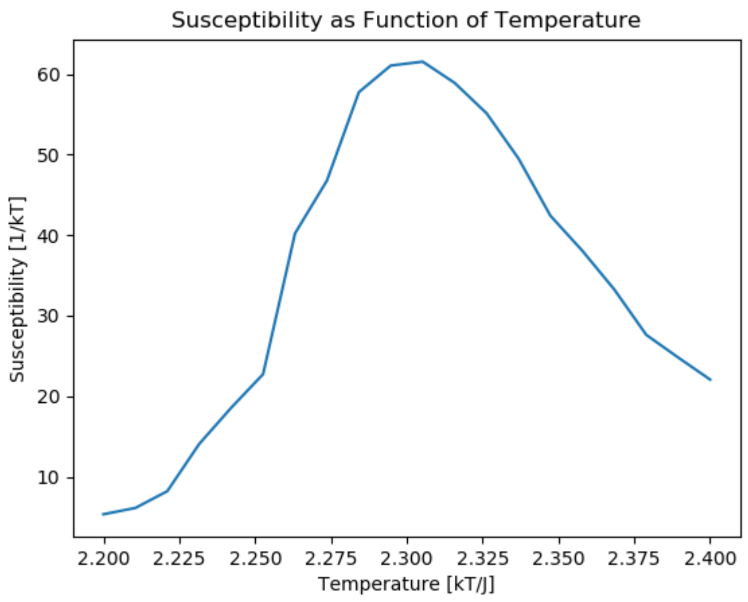

Figure 16: Magnetic susceptibility as a function of temperature (1 000 000 MC cycles, 10 000 equilibration cycles, 20 temperatures, L = 60).

Parallelised	# of MC cycles	# of temperatures	Time used [s]
Yes	10 000	1	0.299
No	10 000	1	0.268
Yes	10 000	16	0.431
No	10 000	16	4.16
Yes	10 000	32	0.591
No	10 000	32	8.39
Yes	100 000	1	2.76
No	100 000	1	2.68
Yes	100 000	16	3.81
No	100 000	16	41.57
Yes	100 000	32	4.92
No	100 000	32	82.91

Table 3: Timing of selected runs ($L = 20$, $T \in [2.0, 2.5]$)

L	T_C [kT/J]
40	2.3269
50	2.3167
60	2.3013
70	2.3038
80	2.3038
90	2.2705
100	2.3192
∞ (estimated)	2.276 ± 0.027
∞ (exact)	$\frac{2}{\ln(1+\sqrt{2})} \approx 2.269$

Table 4: Estimation of the critical temperature in the thermodynamic limit $L \to \infty$ (1 000 000 MC cycles, 20 000 equilibration cycles, 40 temperatures $\in [2.25, 2.35]$)

5 Discussion

We started of by looking at the simple 2x2 lattice, where we have found the expectation values of the energy, magnetisation, heat capacity and magnetic susceptibility analytically. From Table 2, we see that the obtained expectation values match the analytical ones. We also see that the expectation values for E and M don't vary with the number of Monte Carlo cycles, while the values for C_V and χ improves significantly. This can be explained by the fact that with such a tiny lattice at quite a low temperature, there will be almost no thermal fluctuations away from the equilibrium state. Since the calculations of C_V and χ depends on such fluctuations, we need quite a lot of MC cycles to get good results. For 1 000 000 MC cycles, we see that the error in C_V and χ is around 3 and 5 %, respectively.

The statistical physics in this project is valid for systems at equilibrium, and we therefore need to look at how our Markov Chain Monte Carlo method relaxes to the steady state. In Figure 2 - 10, we see how the equilibration time depends on both temperature and the initial configuration of the lattice. With an initial configuration of spins with random orientation, both the mean energy and the mean magnetisation stabilises nicely after only 2000 - 3000 MC cycles at $T = 1.0$. At $T = 2.4$, thermal fluctuations become more prominent, and the system needs around 10 000 MC cycles to reach its steady state.

When we look at the acceptance ratio, we see a clear dependence on temperature. At $T = 1.0$ the acceptance ratio goes to zero, since the thermal energy is too low to cause any significant number of spins to change their orientation. At $T = 2.4$, however, the thermal energy is sufficiently high and the acceptance ratio stabilises at around 0.27.

With an initial configuration of spins pointing in the same direction, we see that the system equilibrates faster in both temperature cases. This indicates that the ordered phase is closer to the equilibrium state than the disordered one for these temperatures.

In Figure 11 and 12, we have the probability distribution with respect to energy at $T = 1.0$ and $T = 2.4$, respectively. At the lower temperature, we see that most of the spins are in the lowest energy state, and the spread in energy is quite small ($\sigma_E = 0.008$) . For the higher temperature, we see that most of the spins occupy the higher energy states, and the spread in energy is increased by a factor of almost 20. We also see that unlike for the lower temperature, the probability distribution for this higher temperature resembles the Gaussian distribution.

We then look at the behaviour of our system near the critical temperature T_C. In figure 13 - 16, we see the temperature dependence of the four physical quantities we have been looking at (for a finite lattice $L = 60$). We see that the magnetisation drops at higher temperatures, which is what we would expect from the power law in section 2.2. We also see that the magnetic susceptibility has a distinct top at a particular temperature, namely the critical temperature (See section 2.2).

In Table 3, we have the timings of some selected runs of our program. We see that for higher numbers of temperatures, the parallelised code (where we have parallelised the loop over the temperatures) is between 10 and 16 times faster than the sequential code. This corresponds well with the fact that Freebee at Abel, the server the code was run on, has 16 CPUs.

Finally, we estimate the critical temperature in the thermodynamic limit $L \to \infty$. The results are given in Table 4. We see that the exact result are within the error of our estimate.

6 Conclusion

We have studied the famous Ising model for magnetic materials in two-dimension, with a focus on the phase transition from ferro- to paramagnetism. Through Monte Carlo calculations, we have obtained an estimate for the critical temperature in the thermodynamic limit, $kT_C/J = 2.276 \pm 0.027$. This matches the exact solution by Lars Onsager, $kT_C/J = \frac{2}{\ln(1+\sqrt{2})} \approx 2.269$.

Our Monte Carlo procedure includes the Metropolis algorithm, which ranks as one of the most influential algorithms in science. One of the main reasons for this is the algorithms simplicity and computational efficiency.

Our Monte Carlo method is really a Markov Chain Monte Carlo method, and we have looked at how our physical system moves towards equilibrium. This is important, since most of our statistical physical theories applies to systems at equilibrium. When we run our simulation, we therefore need to wait for the equilibration before we start calculating expectation values, probability distributions etc.

Another, more technical, aspect of this project is the Monte Carlo methods demand for data. To get accurate results for the critical temperature (within reasonable time), we have parallelised our code and run it on a powerful computing server (Freebee at Abel).

References

[1] Jack Dongarra and Francis Sullivan. Guest editors' introduction: The top 10 algorithms. *Computing in Science & Engineering*, 2(1):22–23, 2000.

[2] Morten Hjorth-Jensen. Computational physics — lecture notes fall 2015.

[3] Morten Hjorth-Jensen. Github repository. https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Programs/ParallelizationMPI/IsingModel.cpp.

[4] Jacob Linder. Ising modellen. In *Store norske leksikon*: https://snl.no/Ising-modellen, 12. February 2018. Accessed: 19. November 2018.

[5] Conrad Sanderson and Ryan Curtin. Armadillo: a template-based c++ library for linear algebra. *Journal of Open Source Software*, 1:26, 2016.

[6] Conrad Sanderson and Ryan Curtin. A user-friendly hybrid sparse matrix class in c++. *Lecture Notes in Computer Science (LNCS)*, 10931:422–430, 2018.