University of Oslo

Computational Physics

FYS3150

PROJECT 4

Author:
Erik Leven

November 15, 2016

Contents

I Abstract

This project will study the phase transitions of a ferromagnetic material by means of the two dimensional Ising model to locate the critical temperature, also called the Curie temperature T_C, as described by Lars Onsager (1994)[1]. We will assume no external magnetic field, $\mathbf{B} = 0$, and a ferromagnetic ordering with $J = 1$. Our material consists of a lattice occupied by atoms with interacting dipole spins in two dimensions where we use periodic boundary conditions to simulate a infinite grid. The critical temperature (T_C) denotes the highest temperature for which a non-zero magnetization can occur, for temperatures higher than T_C there is no total magnetization. When extracting the mean from several lattice sizes by means of the Metropolis algorithm we found our estimate of T_C to be 2.266 $[kT/J]$ compared to the results of Lars Onsager of $T_C \approx 2.269$ $[kT/J]$.

II Introduction

Magnetic phase transitions has long been an area of great interest. At some given temperature ferromagnetic material changed magnetic behaviour in the absence of a magnetic field. Above this temperature (known as the Curie temperature T_C) the net magnetization is zero but below it the material exhibits spontaneous magnetization. It was clear that for some specific temperature there had to be some sort of phase transition where the spin alignment changed from being ordered to disordered, see figure 1. In 1924 Ernst Ising (for whom the

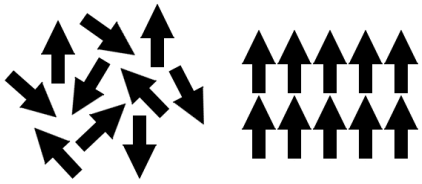


Figure 1: Example of spin alignments. To the left we see how the spin is in disarray creating a net magnetization of zero. To the right all spins are aligned in the same direction creating a net magnetization inside the material

Ising model is named), found a solution for ferromagnetic phase transition in one dimension. 20 years later, in 1944, Lars Onsager found an analytic solution to the two-dimensional Ising model, a result we are going to try to replicate by numerical methods. As a method we have chosen the Ising model with a Metropolis algorithm to achieve stable expectation values for. The Metropolis algorithm is a Monte Carlo algorithm, which need a big amount of data to produce precise enough results. We will therefore have to focus on making the program as fast as possible for practical reasons.

III Method

The thermodynamical properties we are interested in for this project are mainly

- The Expectation value of the total energy of the system, <E>
- The Expectation value of the total magnetization of the system, <M>
- The heat capacity of the system, Cv
- The magnetic susceptibility of the system, χ

As mentioned, one of the main problems in simulating thermodynamical systems is that all these properties are dependent of the partition function, Z. Z is mathematically defined as

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

and as we can see this sum gets really large even for medium size lattices. For the case of a $100x100$ grid we have 2^{10000} terms to sum over. We therefore need to find a model which can calculate the thermodynamical values without having to calculate the partition function.

The Ising model

The Ising model is a model for ferromagnetism which is used to identify phase transitions. Ferromagnetic materials can exhibit a total magnetization inside the material even without the presence of an external magnetic field for sufficiently low temperatures. The spins align themselves in the same direction since this configuration results in the lowest possible energy state. At high enough temperatures the thermodynamic energy is sufficiently high to create a disarray in the spin alignments resulting in a net magnetization of zero. The Ising model creates a setup where we can simulate this behaviour. The model consists of dipole moments of spins which can take the value -1 or 1 and is usually constructed as a lattice. The following shows a possible spin configuration for the $2x2$ lattice.

$$\begin{array}{cc} \downarrow & \uparrow \\ \uparrow & \downarrow \end{array}$$

The Ising model for two dimensions consists of a $N = LxL$ lattice where N is the total number of spins in the lattice. Since the force between two magnets is proportional to $\frac{1}{r^2}$ where r is the distance between 2 atoms, we can conclude that each atom is only affected by the interactions from its closest neighbours. For the simplified Ising model, with periodic boundary conditions and without an external mag-

netic field, we can therefore define the total energy of the lattice as

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

Each atom interacts with it's neighbours inside the central lattice. If the spin is located at the edge of the lattice, it also communicates with it's neighbours on the opposite side. To make a setup usable in $C++$ I chose to use an outer shell surrounding the lattice so that each atom inside the central $2x2$ grid has four neighbours. The neighbours in the outer shell corresponds the spins on the opposite side. The following setup shows the $2x2$ lattice from above with a surrounding shell for the periodic boundary conditions.

$$\begin{array}{cccc} \downarrow & \uparrow & \downarrow & \uparrow \\ \uparrow & \downarrow & \uparrow & \downarrow \\ \downarrow & \uparrow & \downarrow & \uparrow \\ \uparrow & \downarrow & \uparrow & \downarrow \end{array}$$

This is the model we will use for further simulations in the Metropolis algorithm.

The Metropolis algorithm

Since the goal is to estimate the thermal equilibrium state of the system we start with an initial spin setup. We can flip a spin inside the lattice one at a time and check if this new microstate is more probable than the last one. The Metropolis algorithm, as any other Monte Carlo method, demands a probability distribution function, a uniform random number generator within the interval $(0, 1)$ and a sampling rule to decide which steps to accept and which do reject. In order to find the sampling rules needed we need to consider the time development of a PDF in regards to Markov chains and random walks. This time development after one time step can be described as

$$w_i(t = \epsilon) = W(i \to j) w_j(t = 0)$$

which can be simplified to

$$w_i(t = \epsilon) = W_{ij} w_j(t = 0)$$

where W is a transition matrix for a random walk given by

$$W_{ij}(\epsilon) = W(il - jl, \epsilon) = \begin{cases} \frac{1}{2} |i - j| = 1 \\ 0 \qquad \text{else} \end{cases}$$

W can be viewed as the the transition probability represented by a matrix. Since both W and w represent probabilities for each time step we have:

$$\sum_i w_i(t) = 1$$

, and

$$\sum_i W(j \to i) = 1$$

and the fact that $0 \leq W_{ij} \leq 1$ and $0 \leq w_j \leq 1$. This leads us to a new way to define $w_i(t+1)$:

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

The main problem is that $W_{j \to i}$ is unknown. However, the Metropolis algorithm solves this problem by defining $W_{j \to i} = T(j \to i) A(j \to i)$ where $T(j \to i)$ is the probability of accepting a move from position j to i and $A(j \to i)$ is the probability of accepting a move from i to j. In other words:

- Suggest a move from j to i with probability $T_{j \to i}$
- We accept this move with a acceptance probability $A_{j \to i}$. This new state i is then used as our new starting point for the next move. Similarly we reject the proposed move with a rejection probability $1 - A_{j \to i}$.

What we need now is a way of establishing the properties of the probabilities T and A such as $w_i^{t \to \infty} \to w_i$. In other words no matter what starting point we have, we always end up in the correct distribution. The next step is to derive the dynamical process towards equilibrium. Assuming that T and A are time-independent we can rewrite the equation for $w_i(t + 1)$ as:

$$w_i(t + 1 =)$$

$$\sum_j [w_j(t) T_{j \to i} A_{j \to i} + w_i(t) T_{i \to j} (1 - A_{i \to j})]$$

which when normalizing $\sum_j T_{i\to j} = 1$ can be written as

$$= w_i(t) + \sum_j [w_j(t) T_{j\to i} A_{j\to i} - w_i(t) T_{i\to j} A_{i\to j}]$$

If moving the $w_i(t)$ term to the right hand side this equation is similar to the Master equation which relates the temporal dependence of a PDF $w_i(t)$ to various transition rates. The Master equation states that the rate of which a system moves from state j to a final state i is balanced by the rate at which the system undergoes transitions from the state i to a state j. At a steady state (equilibrium) is therefore defined as $\frac{dw_i(t)}{dt} = 0$. This means that at equilibrium

$$\sum_j w_j T_{j\to i} A_{j\to i} = \sum_j w_j T_{i\to j} A_{i\to j}$$

which can be rewritten as

$$w_i = \sum_j w_j T_{j\to i} A_{j\to i} = \sum_j w_j W_{j\to i}$$

which is simply the equation for a steady state in a Markov chain. To avoid so-called cyclic solution we introduce an additional condition, detailed balance:

$$W(j \to i) w_j = W(i \to j) w_i$$

. At equilibrium detailed balance gives us

$$\frac{W(j \to i) w_j}{W(i \to j) w_i} = \frac{w_i}{w_j}$$

.

In our specific case we are working with a Boltzmann distribution which can be expressed with

$$w_i = \frac{e^{-\beta E_i}}{Z} \tag{3}$$

where $\beta = \frac{1}{kT}$ and Z is the partition function. This is a typical PDF for thermodynamical properties where the Boltzmann factor

$$w = \frac{W_i}{W_{i+1}} = e^{-\beta \Delta E_i} \tag{4}$$

is the ratio of the Boltzmann distribution between two states. When performing the Boltzmann factor the partition function cancel out, giving us a way to calculate whether or not a suggested change in spin configuration is more probable without having to sum over all the terms in Z. If we would construct a sampling rule where we accept all new spin configurations where $\Delta E < 0$ and the configurations where $\Delta E > 0$ and $w \geq r$, where r is a uniform probability distribution from 0 to 1 we will force system towards an equilibrium state as explained above.

The Metropolis algorithm gives us the means to perform this method in an efficient way.

1. Establish an initial state and compute the total energy E_b of this system
2. Flip one spin and compute the new energy E_t of the new configuration
3. Compute the change in energy ΔE between these configurations
4. If $\Delta E < 0$ we accept this new configuration and can continue to point 6
5. If $\Delta E > 0$ but $r \leq w$ we accept the new configuration
6. Update expectation values
7. Repeat step 2-6 until a stable equilibrium state is achieved
8. Divide the expectation values with the number of Monte Carlo cycles and if preferable the number of total spins.

Finding the critical temperature

Even though we use a finite grid when performing our numerical methods there is a way to relate finite lattice sizes with those of infinite size through finite size scaling relations. The critical temperature scales as

$$T_C(L) - T_C(L = \infty) = a^{-1/v} \tag{5}$$

where v is defined through

$$\xi(T) \sim |T_C - T|^{-v} \tag{6}$$

and a is a constant. To find an approximation to $T_C(L = \infty)$ we start with finding an approximation for all our lattice combinations and divide by the number of combinations.

Once the constant a is found we simply iterate over all lattice sizes and divide by the number of lattice sizes. In this project the heat capacity development was chosen to estimate T_C. The lines of code looks as follows:

```
a = 0
count = 0
Cv = [Cv20, Cv40, Cv60, Cv80, Cv100, Cv120, Cv140]
for i in xrange(len(Cv)):
    for j in xrange(len(Cv)):
        if i != j:
            a += (T[np.argmax(Cv[i])] -
            T[np.argmax(Cv[j])])/(1.0/((i+1)*20) -
            1.0/((j+1)*20))
            count += 1
a/=count

Tc_limit = 0
count = 0
Cv = [Cv20, Cv40, Cv60, Cv80, Cv100, Cv120, Cv140]
for i in xrange(len(Cv)):
    Tc_limit += T[np.argmax(Cv[i])] - a/((i+1)*20)
    count += 1
Tc_limit/=count
print Tc_limit
```

As Lars Onsager proved the Curie temperature for our situation lies in the region of T = 2.269 $[kT/J]$. For this reason we will only study temperatures in this region to speed up the analysis process as other temperatures are not needed for this assignment. The temperature interval we have chose is $T\epsilon[2.2, 2.3]$ with a step size of $dT = 0.01$. We therefore have 11 different temperatures to analyze all the thermodynamical properties for. As shown in table 2 the values converge towards the analytical at around 1000000 Monte Carlo cycles, and this is the number of cycles we therefore will use when simulating larger lattice sizes.

Making the code more efficient

A common problem when performing Monte Carlo simulations is the amount of data needed for precise calculations. For this purpose we need to make the code as efficient as possible.

1. The first step to make the code more time efficient has already been taken. The fact that we constructed a shell around the inner lattice where we establish the means of performing the periodic boundary conditions seems to run faster than for example an inline function as described by Hjorth-Jensen, Morten, (Lecture notes 2015)[2].

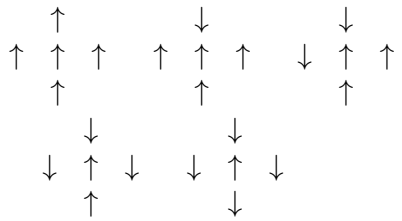
2. When calculating the change in energy ΔE when performing a spin flip we do not have to run over all spin inside the lattice every time. Since $\Delta E = E_2 - E_1$ we can write this as:

$$\Delta E = J\sum_{<kl>}^{N} s_k^1 s_l^1 - J\sum < kl >^N s_k^2 s_l^2$$
$$\Delta E = -J\sum_{<kl>}^{N} s_k^2(s_l^2 - s_l^1)$$
$$\Delta E = 2Js_l^1\sum_{<k>}^{N} s_k \tag{7}$$

The last step can be done by realizing that the spins can only take values ± 1, ie if $s_l^1 = 1$ then $s_l^2 = -1$. The other spins keep their values such as $s_k^1 = s_k^2$. If $s_l^1 = 1$ then we must have $s_l^1 - s_l^2 = 2$ and if $s_l^1 = -1$ then we must have $s_l^1 - s_l^2 = -2$.

3. We can use the same argumentation as in point 2 to show that $M_2 = M_1 + 2s_l^2$ saving us the computational time of running through the whole lattice here as well.

4. When performing the sampling rule $w \geq r$ we do not have to perform the exponential in w for every calculation. To understand this we can go back to equation (5) in point 2 in our calculations for ΔE. This equation states that each total change in energy only depends on 4 spins surrounding a randomly chosen spin in the current configuration. The total 5 spins can only take 5 forms



which corresponds to $\Delta E = 8, 4, 0, -4, -8$ respectively. We therefore only have five different values ΔE can take and therefore only five values of w, which we can precalculate. This saves us the time consuming part of calculating the exponential in w N times for every cycle.

5. Since we have many operations that are independent of each other parallelization of the program is advisable. Parallelization of the program means that instead of running the program once, where a large amount of the CPU just stands by waiting for things to do, we construct several different processes in which the computer can work in parallel. As an example, if your computer has 4 processors, you could run 4 processes at once without sacrificing more CPU than the parallelization software would need.

Testing the code

As previously stated, the partition function makes analytical expectation values hard to derive. We can however calculate this for small lattices. For the $2x2$ lattice with periodic boundary conditions we have the following configurations:

Table 1: List of degeneracy, energies and magnetization for different macrostates in the 2x2 lattice

spins up	states	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

For the analytic solutions of <E>, <M>, <|M|>, Cv and χ see calculations appendix[3]. Given the values $\beta = 1$ ($\frac{1}{kT} = 1$) and J = 1 we get the following values for the expectation values, Cv and χ as seen in table 2.

Table 2: Numeric and analytical calculations of energy and magnetic values for the 2x2 lattice for spin ±1a and T [TK/J] = 1. We notice that we need up to 1.000.000 Monte Carlo cycles before our numerical values come close to the analytical values. <E> is given as energy pr. spin.

Cycles	<E>	<|M|>	Cv	χ
Analytic	-1.996	4.000	0.128	15.956
100	-1.800	3.960	0.634	0.356
10000	-1.980	3.993	0.102	14.964
1000000	-1.998	3.995	0.126	15.966

Note that even though <E> and <|M|> converges towards the analytical value after relative few cycles, the heat capacity and magnetic susceptibility converges after around 1 000 000 cycles. We can however conclude that the program works for a large number of cycles.

IV Results

Analysis of equilibrium

The following plots shows the behaviour of our Ising model properties as we close in our the equilibrium state.

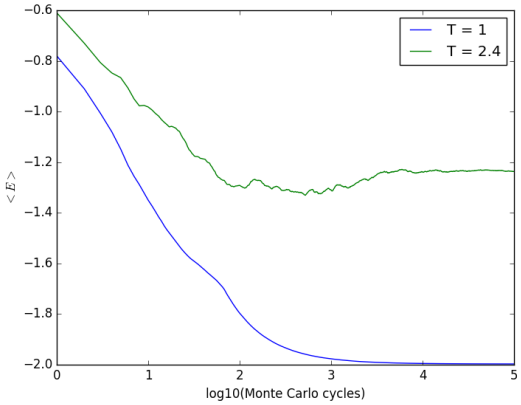


Figure 2: Plot of the development of <E> pr. spin as a function of Monte Carlo cycles for $T = 1.0$ and $T = 2.4$. X-axis is scaled with log10. For both temperatures we achieve equilibrium after around 10 000 Monte Carlo cycles

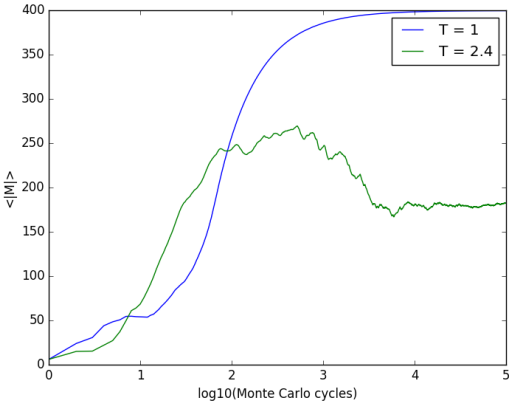


Figure 3: Plot of the development of total <|M|> as a function of Monte Carlo cycles for $T = 1.0$ and $T = 2.4$. X-axis is scaled with log10. Here as well equilibrium for both temperatures i achieved after around 10 000 Monte Carlo cycles

Analysis of Curie temperature

The following plots shows the behaviour of our Ising model properties as we increase the lattice size.

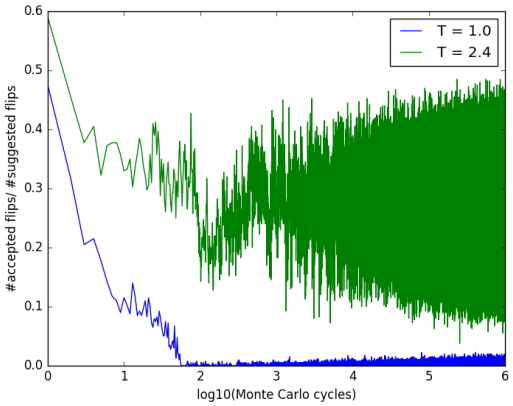


Figure 4: Plot of the probability of accepting a new state as a function of Monte Carlo cycles for $T = 1.0$ and $T = 2.4$. X-axis is scaled with log10. The state with $T = 2.4$ accepts many more suggested spin flips than the $T = 1.0$ model does, even after equilibrium is achieved.

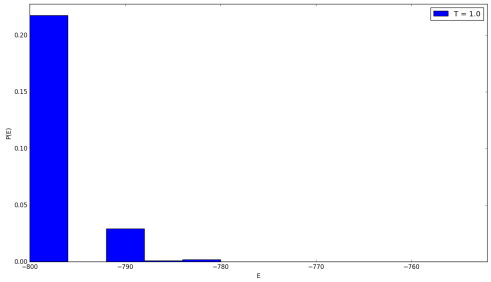


Figure 5: Histogram over P(E) after equilibrium is achieved for $T = 1.0$ with a 20x20 lattice.

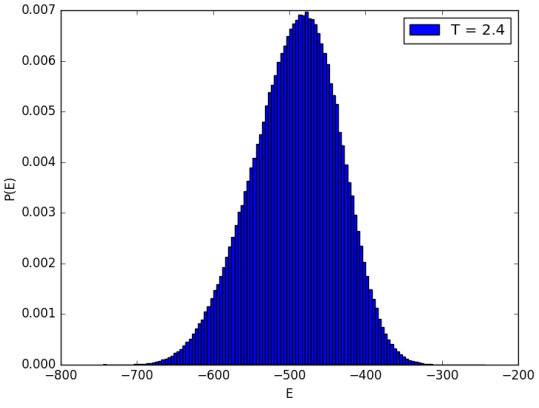


Figure 6: Histogram over P(E) after equilibrium is achieved for $T = 2.4$ with a 20x20 lattice.

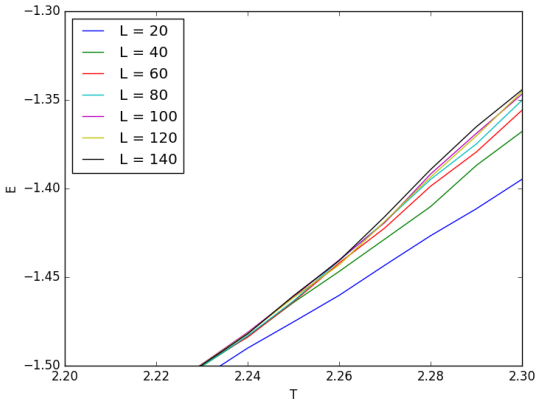


Figure 7: Analysis of <E> pr. spin as a function of lattice size.

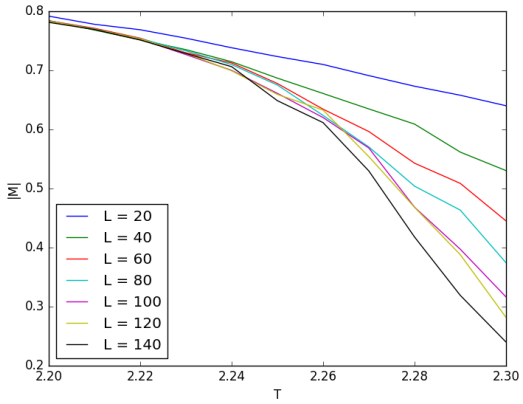


Figure 8: Analysis of </M/> pr. spin as a function of lattice size.

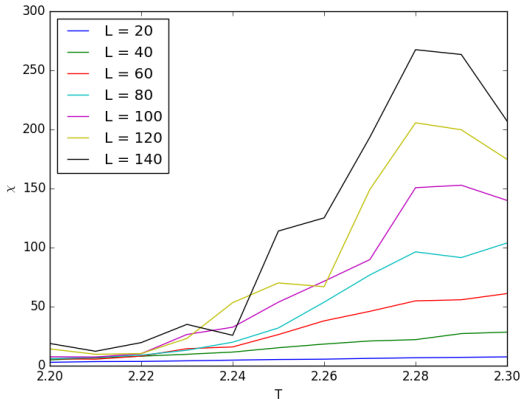


Figure 10: Analysis of magnetic susceptibility pr. spin as a function of lattice size.

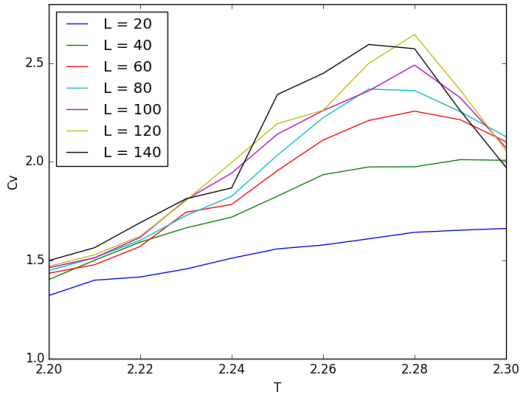


Figure 9: Analysis of heat capacity pr. spin as a function of lattice size.

V Conclusion

To be able to establish if the Ising model, and whether or not our implication of the Ising model, works we look to table 2 and figures 2-6. Table 2 shows a clear convergence towards the analytical values as the number of Monte Carlo cycles increase. Figures 2 and 3 even shows that the Metropolis algorithm only after a couple of thousand cycles have reached an estimate in the equilibrium region. We can also by comparing figure 6 and the computed σ_E^2 argue for the validity of the Ising model. The computed energy variance σ_E^2 was 3244.3 $[J^2]$ for the whole $20x20$ lattice. This corresponds to a standard deviation, σ_E of around 57 [J], which seems to match figure 6. As shown in figures 2 and 3, the values of <E> and <|M|> stabilize after around 100.000 Monte Carlo cycles. However, table 2 shows stable values for Cv and χ only after 1.000.000 and arguably even higher. The reason for choosing 1.000.000 cycles in our lattice size analysis was a combination of how precise the values became and how long the program takes to run. Bearing this in mind we should expect some deviations in our calculations of the Curie temperature T_C due to this compromise. The analytic results of Lars Onsager made it possible to narrow the number of temperature values to the realm of T $\epsilon[2.3, 2.4]$ with a dT of 0.1 $[kT/J]$. When studying figures 7-10 we see a clear trend specially in the T-region of $2.22 - 2.30$. We see a spike in both heat capacity and magnetic susceptibility in the region around T = 2.28 as we increase the lattice size. We also see a trend of steeper slopes for the expectation values for E and |M|, which is expected in the region of a phase transition. When studying the <|M|> plot in figure 8 we see how the total magnetization goes towards zero. This is to be expected from a ferromagnetic material which undergoes a phase transition, since ferromagnetic materials have zero magnetization in the absence of an external magnetic field for temperatures above T_C. Even though the Metropolis algorithm is not very efficient in the regions close to the critical temperature we can still calculate an estimate to the Curie temperature. When applying the code snippet displayed in the method-section we find our estimate of the constant $a = 0.839$ and a corresponding $T_C \approx 2.266$ $[kT/J]$ compared to the results of Lars Onsager of $T_C \approx 2.269$ $[kT/J]$. The results seems to match up to 2 leading digits after the decimal point. For a more precise numerical estimation more Monte Carlo cycles is advised as table 2 clearly showed increased accuracy when increasing the number of cycles. A smaller dT would also be advisable as the final estimation of T_C is only a mean based on results from relatively few temperature values. In conclusion, even though we have finite grids with only 1.000.000 Monte Carlo cycles we were able to achieve a fair approximation to the Curie temperature by means of the Metropolis algorithm applied to the 2D Ising model.

VI Appendix

Git-hub address to project including scripts and pictures.

https://github.com/erikalev/FYS3150/tree/master/Project%204

References

[1] Onsager, Lars (1944). http://journals.aps.org/pr/abstract/10.1103/PhysRev.65.117

[2] Hjorth-Jensen, Morten (2015). https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf

[3] Leven, Erik (2016) https://github.com/erikalev/FYS3150/tree/master/Project%204