Project 4, Phase Transitions in Magnetic Systems

Computational Physics FYS3150

Jonas Thoen Faber and George Stanley Cowie

(Dated: November 19, 2019)

In this report we have studied the behaviour of a system of magnetic spins using the Ising model. We found that the energy distribution is heavily dependent on temperature and that increasing the temperature causes the system to become more disordered. As long as the model is not initially in an equilibrium state it needs to be run for a couple of thousand Monte Carlo cycles before it reaches equilibrium. At a critical temperature the model exhibits a phase transition. As we increased the lattice size the critical temperature moved towards $T_C = 2.260$. After the transition the magnetization of the system moves towards zero.

I. INTRODUCTION

In this project we will take a closer look at the Ising model. The model represents a magnet which contains many particles that have either spin up or spin down, a so called binary system. The spins will not only decide the strength of the magnetization, but also the energy in the system. We have modeled the system with both ordered and random initial states. Ordered means that all the spins in the lattice system are oriented in the same direction. The model is restricted to two dimensions, meaning that every particle has four neighbours, excluding the particles along the edge of the lattice. The systems we are modelling are imagined to be a part of a much larger system that can not be modeled in full due to computational restrictions. To mitigate this problem we will apply periodic boundary conditions to the particles at the edge of the lattice.

The focus of this project is to study the Ising model in a phase transition at the critical temperature. The model has analytical solutions in two dimensions for various expectation values that would be interesting to further investigate. We will use these analytical solutions to test our model. The algorithm of choice in this project is the Metropolis algorithm which is based on the statistical Markov Chain Monte Carlo method. The Metropolis algorithm uses a random number generator to decide if a given configuration should be accepted or not. The configuration consist of flipping a single spin in the lattice. Flipping one spin will change both the energy and magnetic moment in the system. What we should achieve is that the system would converge to a steady state after a number of Monte Carlo cycles. This means that the change of energy will be the factor that determines if a configuration may be accepted or not.
We will start by introducing the relevant theory, methods and algorithms used to solve the problems in this project. Thereafter we will present our results, followed by analysis and discussion of the results.

II. METHODS

This sections introduces the methods used for finding the analytical solution of a 2X2 lattice and the set up of the Metropolis algorithm in detail. The lattice size is then increased for further study of phase transitions that may occur within a temperature interval. The probability distribution of the average energy and average magnetization is also introduced in this section.

A. Setting up a simple model

The simplest form of the Ising model without any externally applied magnetic field is expressed as

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

with $s_k = \pm 1$, N represents the total number of spins and J is a coupling constant expressing the strength of the interaction between neighbouring spins. $\langle kl \rangle$ Is the sum of the nearest neighbours.

In order to benchmark our calculations we will find analytical expressions for a simple 2x2 lattice. Here we only have two spins in each dimension so L = 2.
The partition function for this lattice is

$$Z = \sum_{i=1}^{M} e^{-\beta E_i} \tag{2}$$

Where M is the total number of microstates and $\beta = 1/k_b T$ is the inverse temperature value. k_b is Boltzmann constant and T is the temperature. With a 2x2 lattice with spins ±1 we have M = 2^4 = 16 microstates. To calculate partition function we will first have to find E for each particle in the lattice. We will take into consideration that the particles at the boundary does not seem to have any neighbor particle. It is therefore convenient to use periodic boundary conditions where the first particle is the neighbour of the last particle. For a 2×2 lattice,

Neighbour spins (s)	Particle energy (E)	Counts
4 ↑	-4 J	1
3 ↑ 1↓	-2 J	4
2 ↑ 2 ↓	0	6
1 ↑ 3 ↓	2 J	4
4 ↓	4 J	1

Table I. Possible energies for a particle in different states. The energies of the particles are dependent on its neighbour. The possible energies may occur a number of times.

Spins (s)	Energy (E_i)	Counts
4 ↑	-8 J	1
3 ↑ 1↓	0	4
2 ↑ 2 ↓ (horizontal/vertical symmetry)	0	4
2 ↑ 2 ↓ (Diagonal symmetry)	8 J	2
1 ↑ 3 ↓	0	4
4 ↓	-8 J	1

Table II. The sum of possible energy states. Each sum of the spins may occur a number of times depending on the formation of the spins.

we get the energies as shown in Table I using Equation 1 and the sum of the energy of the particles for different states are given in Table II.

We notice from Table I that the energy from the middle particle is positive. The difference is that the absolute energy would be the same but that the sign changes. Also, there are only five possible energies for each particle.

The partition function depends on both β and E_i where the energy corresponds to the sum of the energies in an arbitrary state. Now that we know the sum of the energies for each state we may find the partition function. It goes as follows

$$Z = \sum_{i=1}^{16} e^{-\beta E_i} \tag{3}$$

$$= e^{8\beta J} + 4e^0 + 4e^0 + 2e^{-8\beta J} + 4e^0 + e^{8\beta J} \tag{4}$$

$$= 2e^{-8\beta J} + 2e^{8\beta J} + 12 \tag{5}$$

The probability for each of the states is given as follows:

$$P_i(\beta) = \frac{e^{-\beta E_i}}{Z} \tag{6}$$

The probability of any state is only β dependent since Z changes whenever the energies E_i of each state changes. The probability is therefore:

$$P_1(\beta) = \frac{e^{-8\beta J}}{Z} \tag{7}$$

,

$$P_2(\beta) = \frac{e^{8\beta J}}{Z} \tag{8}$$

or

$$P_3(\beta) = \frac{1}{Z} \tag{9}$$

We might as well check if our partition function and probability values are correct by summing over all probabilities (sum of all probabilities must be 1). We get

$$\sum_{i=1}^{16} P_i(\beta) = \frac{2e^{-8\beta J} + 2e^{8\beta J} + 12}{Z} = \frac{\cancel{Z}}{\cancel{Z}} = 1 \tag{10}$$

Now that we know both the probability values and the partition function, we can find $\langle E \rangle$ and $\langle E^2 \rangle$ by using the definition of expectation value like so:

$$\langle E \rangle = \sum_{i}^{M} E_i \cdot P_i(\beta) \tag{11}$$

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

$$= -\frac{16J}{Z}\left(e^{8\beta J} - e^{-8\beta J}\right) \tag{13}$$

And then finds $\langle E^2 \rangle$:

$$\langle E^2 \rangle = \sum_{i}^{M} E_i^2 \cdot P_i(\beta) \tag{14}$$

$$= (-8J)^2\frac{e^{8\beta J}}{Z} + 2(8J)^2\frac{e^{-8\beta J}}{Z} + (-8J)^2\frac{e^{8\beta J}}{Z} \tag{15}$$

$$= \frac{128J^2}{Z}\left(e^{8\beta J} + e^{-8\beta J}\right) \tag{16}$$

Now that we have both $\langle E \rangle$ and $\langle E^2 \rangle$, we may find the specific heat capacity. The spesific heat capacity for a constant volume is given as:

$$C_V(T) = \frac{1}{k_b T^2}(\langle E^2 \rangle - \langle E \rangle^2) = \frac{1}{k_b T^2}\sigma_E^2 \tag{17}$$

Using what we found in 13 and in 16, we get:

$$\begin{aligned} C_V(T) = \frac{1}{k_b T^2}\Bigg(&\frac{128J^2}{Z}\left(e^{8\beta J} + e^{-8\beta J}\right) \\ &- \left(-\frac{16J}{Z}\left(e^{8\beta J} - e^{-8\beta J}\right)\right)^2 \end{aligned} \tag{18}$$

Spins (s)	Magnetization ($\mathcal{M}_i$)	Counts
4 ↑	4	1
3 ↑ 1↓	2	4
2 ↑ 2 ↓	0	6
1 ↑ 3 ↓	-2	4
4 ↓	-4	1

Table III. All possible magnetization for different states.

$$= \frac{1}{k_bT^2}\left(\frac{128J^2}{Z}\left(e^{8\beta J}+e^{-8\beta J}\right) - \frac{256J^2}{Z^2}\left(e^{16\beta J}-2e^0+e^{-16\beta J}\right)\right. \tag{19}$$

$$= \frac{128J^2}{k_bT^2Z}\left(e^{8\beta J}+e^{-8\beta J}+4-\frac{2}{Z}\left(e^{16\beta J}+e^{-16\beta J}\right)\right) \tag{20}$$

From [1] we have an expression for the magnetization as the sum over all the spins in the lattice:

$$\mathcal{M}_i = \sum_{j=1}^{N} S_j \tag{21}$$

The possible magnetization for the 2 × 2 system is shown in Table III. The expectation value of the absolute magnetization will be the sum over the product of the magnetization and the PDF for each spin in all possible states:

$$\langle|\mathcal{M}|\rangle = \sum_i^M |\mathcal{M}_i| P_i(\beta) = \frac{1}{Z}\sum_i^M |\mathcal{M}_i| e^{-\beta E_i} \tag{22}$$

$$\begin{aligned} &= \frac{1}{Z}(|4|e^{8\beta J}+4\cdot|2|e^0+0+0 \\ &\quad +4\cdot|-2|e^0+|-4|e^{8\beta J}) \end{aligned} \tag{23}$$

$$= \frac{8e^{8\beta J}+16}{Z} \tag{24}$$

The susceptibility is defined as:

$$\chi(T) = \frac{1}{k_bT}(\langle\mathcal{M}^2\rangle - \langle|\mathcal{M}|\rangle^2) = \frac{1}{k_bT}\sigma_{\mathcal{M}}^2 \tag{25}$$

We have already found $\langle|\mathcal{M}|\rangle$ so we only need to find $\langle\mathcal{M}^2\rangle$ (the absolute sign is dropped because of obvious reasons). This is done by the same method as used previously:

$$\langle\mathcal{M}^2\rangle = \frac{1}{Z}\sum_i^M \mathcal{M}_i^2 e^{-\beta E_i} \tag{26}$$

$$\begin{aligned} &= \frac{1}{Z}(4^2e^{8\beta J}+4\cdot 2^2e^0+0+0 \\ &\quad +4\cdot(-2)^2e^0+(-4)^2e^{8\beta J}) \end{aligned} \tag{27}$$

$$= \frac{32e^{8\beta J}+32}{Z} \tag{28}$$

Thus with 24 and 28 giving us the susceptibility:

$$\chi(T) = \frac{1}{k_bT}\left(\frac{32e^{8\beta J}+32}{Z} - \left(\frac{8e^{8\beta J}+16}{Z}\right)^2\right) \tag{29}$$

$$= \frac{32}{k_bTZ}\left(e^{8\beta J}+1-\frac{2e^{16\beta J}+8e^{8\beta J}+8}{Z}\right) \tag{30}$$

So now we want to write a code for the Ising model to see if we can get the same results numerically. We will do so by implementing the Metropolis algorithm and using periodic boundary conditions for the lattice. So what it does is that we will first initialize a random configuration and then the algorithm decides if the configurations should be accepted or not. The configuration is changed based on the difference in energy of the system. The algorithm goes like so:

1. Generate a lattice with random spins and calculate the energy E_i of that state.

2. Choose a random particle from the lattice and flip that spin and calculate the new energy E_f. Since the spin is flipped, the change of energy of that particle is just two times it's current energy.

3. Find $\Delta E = E_f - E_i$. For the special case of the Ising model with just two dimension, ΔE is limited for only 5 different energies as shown in Table I.

4. If $\Delta E \leq 0$ we will accept the configuration since we want the energy to converge to a minimum. The next step is then 6. If $\Delta E > 0$ we will calculate $w = e^{-\beta\Delta E}$.

5. Again, a normal distributed random number $R \in [0, 1]$ is generated. If $R \leq w$, we accept the configuration.

6. If the configuration is accepted, the expectation values are updated.

7. Step 2-6 are repeated so that we hopefully get a desired result. Every iteration where we sum over the spins are called a cycle (measurement). We will at the end divide every expectation values with the number of cycles and if we want, we may also divide with the number of spins. The reason for dividing with the number of spins is so that we get some unit per spin.

The random number generator (RNG) used is the Mersenne Twister(MT19937). To make sure that we do not count any energy in the lattice twice, we will only summarize the neighbours that are under or to the right of the measured particle. That means that when we look at the particles at the bottom or at the right side of the lattice, they will need to follow the periodic boundary condition. This is taken care of in the program by the function called periodic$(x, y, -1)$. The algorithm will look something like this in C++:

```
// metropolis algorithm
for (int i=0; i < cycles; i++){
    for (int spin=0; spin < N; spin++){
        int ran_x = dis(gen);           // RNG in [-1,L]
        int ran_y = dis(gen);           // RNG in [-1,L]

        int del_energy = 2*spin_matrix *
            (spin_matrix(per(ran_x,L,-1),ran_y) +
            spin_matrix(per(ran_x,L,1),ran_y) +
            spin_matrix(ran_x,per(ran_y,L,-1)) +
            spin_matrix(per(ran_x,ran_y,L,1));

        double R = r_dis(gen);          // RNG in [0,1]
        double w = exp(-beta*del_energy);
        if (R <= w){
            spin_matrix(ran_x,ran_y) *= -1;
            energy += del_energy;
            magnetization = += 2*spin_matrix(ran_x,ran_y);
        }
    }
    // Updating the expectation values
    ave_energy += energy;
    ave_energy_sq += energy*energy;
    ave_mag += abs(Magnetization);
    ave_mag_squared += Magnetization*Magnetization;
}
  spec_heat_cap = (ave_energy_sq - ave_energy*ave_energy)/(k_b*T*T);
  susceptibility = (ave_mag_sq - ave_mag*ave_mag)/(k_b*T);
```

This code follow step 2-6. One of our unit test will be comparing the results gained from our code with the analytical solution. If they more or less agree, we may assume that our algorithm is valid for larger lattices.

B. Reaching equilibrium state

We will now increase the length to $L = 20$ and study more carefully how many Monte Carlo cycles are needed before we reach a steady state. We will show this by plotting $\langle E \rangle$ per particle and the accepted configuration with respect to the number of Monte Carlo cycles. We will do this for $T = 1$ and $T = 2.4$ in units of k_bT/J and then roughly estimate the steady state when the measurements begin converging towards some value. It would be interesting to plot $\langle |\mathcal{M}| \rangle$ as well, but it seems to be quite the same as the plot of $\langle E \rangle$ as they converge towards the same steady state.

C. Probability distribution

The system will fluctuate around the equilibrium as soon as it reaches the steady state. As the temperature of the system increases we can expect an increased number of probable energy states. In order observe this behaviour we will count all the unique energies in each simulation. Since all energies should be integers, this is easily done by storing a unique energy in a list and then count the number of energy occurrences. The probability of the system being in a certain energy state will be the number of times this unique energy is occurs and divided by the total number of energies.

$$P(E_i) = \frac{N_{E_i}}{N_{total}} \tag{31}$$

We know from Equation 17 that if we have the specific heat capacity and the temperature of the system, we may find the energy variance as $\sigma_E^2 = C_V k_b T^2$. We will compare the variance to the computed probability distribution. To do this we will see if $\approx 68\%$ of the total energy lies inside the standard deviation σ.

D. Phase transitions

Our next step is to study the Ising model close to the critical temperature. We will therefore run our algorithm for different values of T. In this case $T \in [2.0, 2.4]$. This is done for various sizes of the lattice $L = 40, 60, 80, 100$. We will then calculate the expectation values $\langle E \rangle$, $\langle |\mathcal{M}| \rangle$, C_V and χ for each of the lattice sizes and plot it with respect to T. Also the number of Monte Carlo cycles should be increased in order to get a good estimate of the critical temperature. To be sure that we get decent results, we will set the number of Monte Carlo cycles to 100 000. The number of iterations the algorithm will compute is $L^2 \cdot N_{cycles} \cdot T_{steps}$ where the last term is the number of temperatures between the max and minimum temperature. So for $L = 40$, $N_{cycles} = 10^5$ and $T_{steps} = 200$, the number of iterations is $32 \cdot 10^9$ iterations. This computation can take some time so we have parallelized the program with MPI to run on multiple processors. The parallization is done by running the ising model independently on all available processors for the entire range of the problem. The data on each processor is then reduced to one processor and averaged. We should do a performance test to to make sure that we get significant improvement from this parallization. Compiler flags are used for further optimization. Our code is tested for various compiler flag for optimal optimization. It is found from [2] that compiler flag -O3 is a good choice, so this will be one of the flag tested. Runs are done using one and four cores. The data for $L = 40, 60, 80, 100$ is then stored in a text file which will be used for determine the critical temperature. This is done by plotting the data using the MatPlotLib library in Python and identifying behaviour that indicates a phase transition. If we have enough data points, we should quite easily estimate where the critical point in the plot. As the lattice size increases we expect the critical temperature to move towards a known value $T_C(L = \infty)$. We will perform an regression analysis using the critical temperature scale [3]:

$$T_C(L) - T_C(L = \infty) = aL^{-1/\nu} \tag{32}$$

where T_C is the critical temperature at different values of L, $a \in [0, 1]$ [4] is a unknown constant and ν is a correlation constant which in this case will be $\nu = 1$. Since Equation 32 is a linear function, we should find $T_C(L = \infty)$ by regression analyzing using the estimated critical temperature. $T_C(L = \infty)$ is the value of the regression function where it crosses the y-axis.

III. RESULTS

A. Setting up a simple model

If we assume that the temperature is $T = 1$ [k_bT/J] and that $k_b = 1$, it follows that $\beta = 1$. Also we should assume for now that $J = 1$ as well. With these assumption we may now calculate the expected value of the energy $\langle E \rangle$, energy squared $\langle E^2 \rangle$, absolute value of magnetic moment $\langle |\mathcal{M}| \rangle$ and the magnetic moment squared $\langle \mathcal{M}^2 \rangle$ using Equation 13, 16, 24 and 28. These values can be found in Table IV. Also the analytic specific heat capacity and the susceptibility found in Equation 20 and 30 is included in the same table. TThis is what we should get with our algorithm and will be the results that we should compare with for the special case $L = 2$. The computed results that are later produced are also implemented in the table for comparing:

Parameter	Analytic	Computed
$\langle E \rangle$	-1.9960	-1.9970
$\langle E^2 \rangle$	15.9679	15.976
$\langle \lvert\mathcal{M}\rvert \rangle$	0.9987	0.9991
$\langle \mathcal{M}^2 \rangle$	3.9933	3.9952
C_V	0.0321	0.0959
χ	0.0040	0.0096

Table IV. Analytical solution of the different expected values per spin including both the specific heat capacity and the susceptibility. Values are obtained for the case $L = 2$. Note that the susceptibility is found using $|\mathcal{M}|$. The computed values are also implemented for comparing.

B. Reaching equilibrium state

Figure 1, 2, 3 and 4 show results for a lattice of size $L = 20$ with both ordered and random initial states. If we look at Figure 1 of the average energy for $T = 1$, we observe that the lattice with random orientation reaches equilibrium at approximately 3200 cycles. There are however still significant fluctuations away from the steady state even with a higher number of cycles. Looking at the ordered matrix makes us believe that it is already in a steady state. The energy barely changes in value after the first configuration. The average magnetization (not shown) exhibits an identical behaviour.

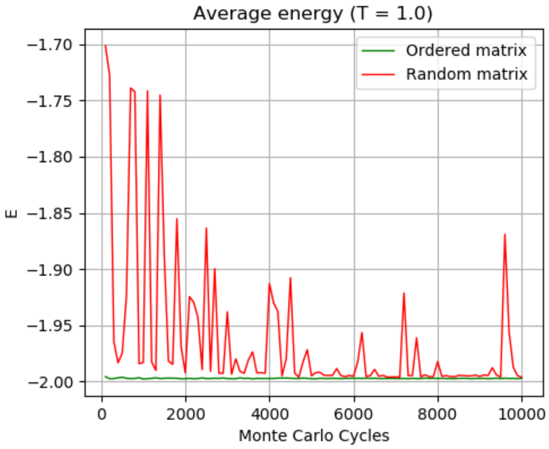

Figure 1. Shows how the average energy per particle develops as the number of Monte Carlo cycles increases for temperature $T = 1$. Each time the algorithm is run, we create a new randomly ordered matrix.

The number of accepted configurations follows Figure 1 closely. As the system moves towards an equilibrium state the algorithm will begin accepting fewer and fewer configurations. In Figure 2 we see that as the number of Monte Carlo cycles increas the random lattice has fewer accepted configurations, excluding some spikes.

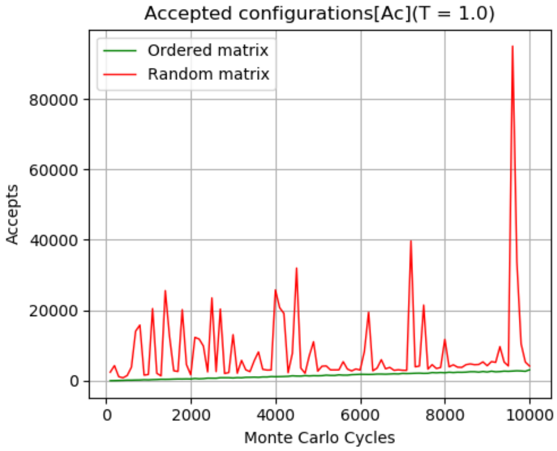

Figure 2. Shows how many configuration that was accepted using the Metropolis algorithm for a temperature $T = 1$ as the number of Monte Carlo cycles increases.

Figure 3 shows the average energy for the systems with a temperature of $T = 2.4$. We observe that both the random and ordered lattices quickly reach a situation where they fluctuate around an average energy per particle of $\langle E \rangle = -1.24$. Purely based on visual observation there is little difference between the random and ordered cases especially as compared with the case with $T = 1$. Both reach an equilibrium state at around 1000 cycles.

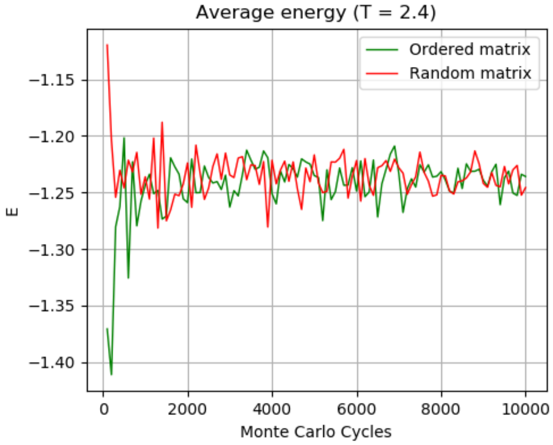

Figure 3. Shows how the average energy per particle develops as the number of Monte Carlo cycles increases for temperature $T = 2.4$. With a higher temperature there is a greater number of configurations available to achieve a steady state.

The number of accepted configuration for both ordered and random initial state at $T = 2.4$ are basically the same for increasing Monte Carlo cycles. As the number of iterations increases the number of accepted configurations increases linearly. As compared with Figure 2 there is a much larger amount of accepted configurations.

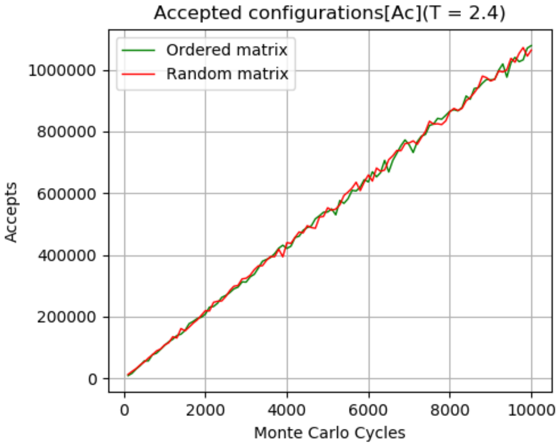

Figure 4. Shows how many configuration that was accepted using the Metropolis algorithm for a temperature $T = 2.4$ as the number of Monte Carlo cycles increases.

C. Probability distribution

The following plots show the probability of finding the system in a given energy state after equilibrium is reached. The probabilities are based on data from the Ising model using 10000 Monte Carlo cycles.

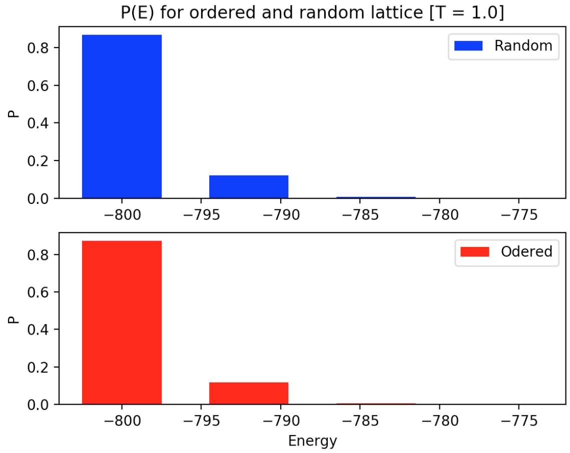

Figure 5. This figure shows the probability distribution of different total energies of a 20 × 20 lattice when the system is in equilibrium. The results are generated using 6000 measurements.

Figure 5 shows that after reaching equilibrium there are in total 6 unique energy states in the system. Though there is only a significant probability of finding the system with an energy of -800 (87%) and -792 (12%). The remaining four energy states account for only 1% of the measurements. The distribution is almost identical for a random and ordered initial lattice.

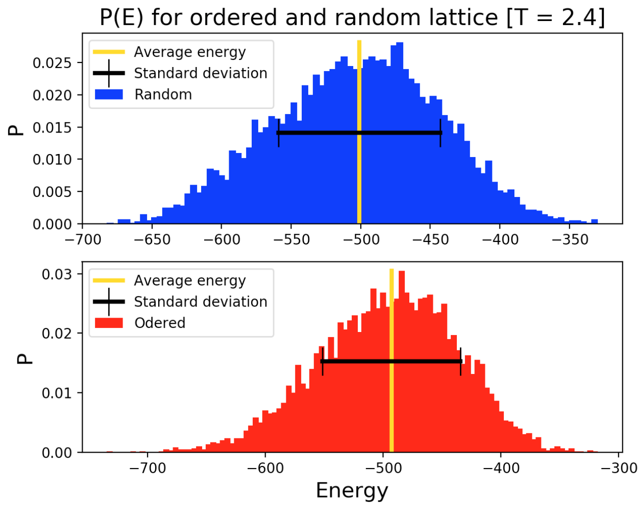

Figure 6. Probability distribution for a random(blue bars) and ordered(red bars) initial lattice with an initial temperature of T=2.4. The distribution is based on 6000 measurements after the system is in equilibrium.

The probability distribution for the same system with a temperature of T=2.4 is quite different (Figure 6). The probability distribution has the shape of a Gaussian curve. The Gaussian nature of this distribution is shown by the fact that for the random and ordered states respectively 65% and 67% of the counted energy states lie within one standard deviation of the mean value. This is close to the theoretical 68% that one would expect from such a curve.

D. Phase transitions

Running our code with different compiler flags at given lattice size and cycles, produces results shown in Table V. The number of cores used are also shown in this table:

Cores	Compiler flag	Time taken (mm:ss)
1	None	08:07
1	-O2	02:08
4	-O2	00:35
1	-O3	02:17
4	-O3	00:37

Table V. Time taken using different compiler flags and cores. Tests are run with lattice size $L = 30$ at 10 000 Monte Carlo cycles. Specs on computer used: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 4 cores.

Figure 7 shows the heat capacity and susceptibility as a function of temperature for different lattice sizes. There are clear tops on both graphs that are indicative of a phase transition. The point at which the graph shows a phase transition moves as L increases. It is found using Equation 32 and with regression analysis that the critical temperature $T_C(L = \infty) \approx 2.260$. It is also found that the constant $a \approx 0.5519$. We notice that a lies between zero and one which is satisfied by article [4].

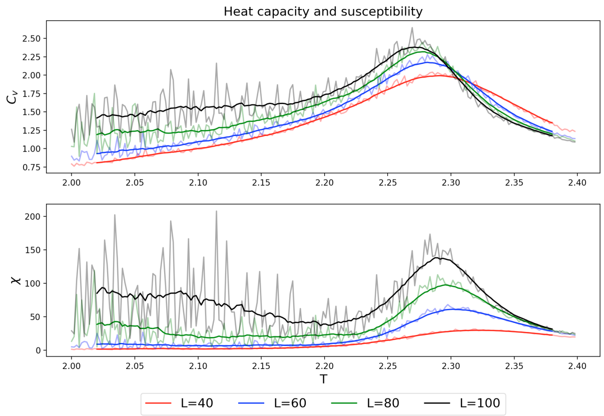

Figure 7. This figure shows the heat capacity on top and the susceptibility on the bottom for various lattice sizes L. The pale lines represent the raw data, the opaque lines represent a moving average of the raw data.

From Figure 8 we see that as T increases the average energy per particle also increases. The Average energy seems to be independent of the lattice size. As T increases the magnetization per particle drops. The slope of the decrease is increasingly steep for higher lattice sizes.

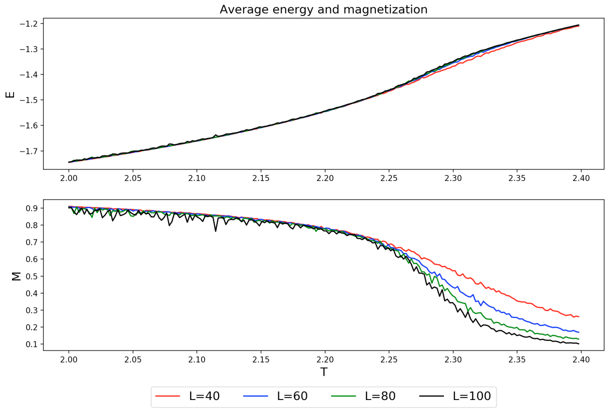

Figure 8. The top graph represents the average energy per particle in the system and the bottom graph, the average magnetization per particle. The different colored lines represent different lattice sizes.

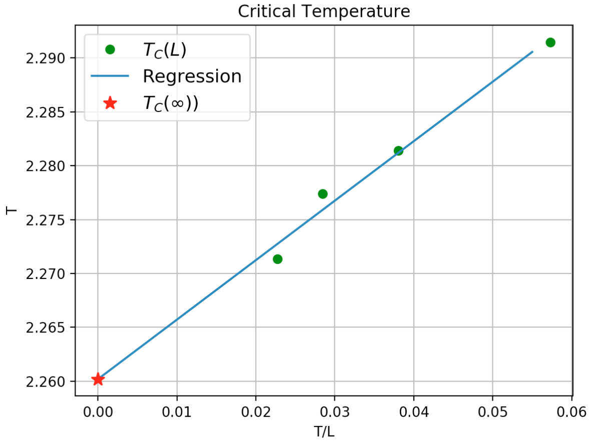

Figure 9. In this figure we see a linear regression based on the critical temperatures we got from the heat capacity in Figure 7. The green dots show the measured $T_C(L)$ and the red star shows what $T_C(\infty)$ should be based on this linear regression.

IV. DISCUSSION

A. Setting up a simple model

In order to test if our model gives sensible results we used a unit test to see if our model results match those of an analytical solution for L=2. We found that there was an error of under 10% between the analytical and numerical expectation values. Although the error is usually much smaller. The problem is that we are working with random numbers and we may in some cases get some large fluctuations. We believe this to be satisfactory results since for a large number of simulations the average error is below 1%. From Table IV we see that our computed results coincide with the analytical values. This assures that our code is working properly. Note that both C_V and χ from the table are relatively far off from the analytical values. Random numbers may in fact cause major errors when the lattice size are small.

B. Reaching equilibrium state

Figure 1 shows some steep oscillations in the average energy of a particle for few Monte Carlo cycles. We can see from Figure 5 that there are just a handful of unique energy states that are present in an equilibrium situation. Since the lattice can start in any possible spin configuration it is unlikely that it will start in a configuration which is close to equilibrium. Given enough cycles though this oscillating behaviour begins to even out. This is however not the case for the ordered lattice. The average energy per particle stays fairly constant at around $\langle -2.0 \rangle$ which indicates that the systems initial state is in fact close to the equilibrium.
With an increased temperature in Figure 3 we see an initial phase on both the random and ordered matrix where they need a few oscillations to stabilize around a equilibrium line. After they reach equilibrium they oscillate around this line in a range of energies. We see from Figure 6 that there is a much wider range of probable energy levels, meaning that you are likely to find the average energy of a particle somewhere within this.
It makes sense that we find a wider range of particle energies for an increased temperature because as you increase the temperature of the system, the entropy is increased and with it the amount of disorder.

C. Phase transitions

It is shown from Table V that -O2 is the compiler flag that runs the algorithm with greatest speed for this computational problem. And also, by parallelizing the code and using this flag decreases the time usage by a factor of 13. This is a huge speedup considering the number of iterations that are needed for analyzing the critical temperature. We have found that the critical temperature for this system goes to $T_C \rightarrow 2.260$ as $L \rightarrow \infty$. This corresponds to the paper [4] which states that $k_b T_C / J = 2/\ln\big(1+\sqrt{2}\big) =\approx 2.269$ as $L \rightarrow \infty$. The relative error is 0.4 % which gives us confidence in our results. The method with which we chose the critical temperature is however prone to error. We chose the critical temperature from the maximum value of the averaged heat capacity. This means that we are in danger of choosing the wrong temperature, because the raw data fluctuates a lot as seen in Figure 7.

If we think about why the average magnetization decreases with increased temperature we realise that magnetism arises from many dipoles pointing in the same direction. When the temperature of the system increases the amount of disorder also increases so the dipoles are not aligned and the net magnetic moment is decreased. The reason why the magnetization decreases faster for a larger lattice could have to do with that a larger system is more likely to have a configuration of spins that are not aligned than a smaller lattice.

V. CONCLUSION

Studying a phase transition close to the critical temperature has led to some interesting results. There is first of all little difference in whether the system starts at an ordered or random generated initial state as the system depends mostly on the temperature. Our results converge towards the exact value of the critical temperature as the lattice size increases. We may conclude that the methods used in this project leads to reasonably good results even with small error between the analytical and numerical solution of the various parameters. Optimization in terms of parallelization is a key point in this project as we are dealing with a great number of iterations.

It would have been interesting to delve a little deeper into what is physically going on during the phase transition, but we did not have time for this. We would also have been wise to pay more attention to the efficiency of our code. We did not do a thorough analysis of where we could have saved time in our code. For the large simulation this could have saved us hours of cpu time.

[1] Hjorth Jensen M. Computational Physics, Lecture Notes; 2015. Available from: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf.

[2] Cowie GS, Faber JT. Project 3 in Computational Physics, Numerical Integration; 2019. Available from: https://github.com/JonasTFab/Project_3.

[3] Hjorth Jensen M. Computational Physics Lectures: Statistical physics and the Ising Model; 2017. Available from: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/pub/statphys/pdf/statphys-print.pdf.

[4] Onsager L. Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition. Phys Rev. 1944 Feb;65:117–149. Available from: https://link.aps.org/doi/10.1103/PhysRev.65.117.