,

FYS4150
Project 4 - deadline November 15

Sander W. Losnedahl

University of Oslo, Autumn 2017

Introduction

Method

To be able to start programming, one first needs to find the expressions for the partition function Z with its corresponding energy values E, the mean magnetic moment M, the specific heat C_V and the susceptibility X as functions of the temperature T. All of this using the periodic boundary condition. The partition function Z is giving by:

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

where β is the inverse temperature given by $\beta = \frac{1}{kT}$ where k is the Boltzmann constant and T is the temperature, so every expression containing β is dependent on the temperature T. E_i is the energy for different spin settings given by:

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

where J is the coupling constant and N is the total number of spins. s_k and s_l are the spins of two neighbouring objects in a lattice. Since we are working 2x2 lattice, the total number of combinations are given by $2^4 = 16$, considering we are working with the Ising model where the spins can only be -1 or 1.

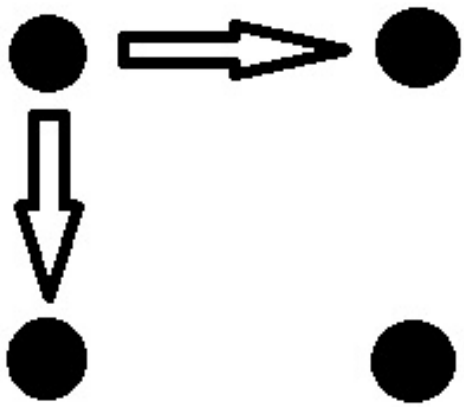


Figure 1: A two by two lattice and how they interact using the Ising model

From figure 1 we see how the Ising model works on a 2x2 lattice and from this the energy is calculated. One can observe that energy is non-zero on lattices where all the objects have the same spin $(-8J)$ and where the two diagonals have opposite spin from each other $(8J)$. All other settings have zero energy. Knowing the energy values we can calculate the mean energy $< E >$ with our specific partition function:

$$< E >= \frac{1}{Z}\sum_{i=1}^{M} E_i e^{-\beta E_i}$$

$$= \frac{1}{2e^{-8} + 2e^{8} + 12}\sum_{1}^{16} E_i e^{-\beta E_i}$$

$$= \frac{16e^{-8} - 16e^{8}}{2e^{-8} + 2e^{8} + 12} = -7.983928$$

Now we can simply put the same energy values into our partition function:

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

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

The magnetization is given by:

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

Unlike in the energy case, the magnetization does not depend the lattice having the same spin or that the diagonals have opposite spin. The only case where the magnetization is zero in a 2x2 lattice is when half of the objects in the lattice opposite spins. Therefore we get the magnetization values:

$$M_i = [4 + 2 + 2 + 2 + 2 + 0 + 0 + 0 + 0 + 0 + 0 + -2 + -2 + -2 + -2 + -4]$$

To calculate the mean magnetic moment or mean magnetization we use the equation below with our calculated magnetic moment and energy values:

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

$$= \frac{1}{2e^{-8} + 2e^{8} + 12}\sum_{1}^{16} M_i e^{-\beta E_i}$$

$$= \frac{8e^{8} + 16}{2e^{-8} + 2e^{8} + 12} = 3.994643$$

To calculate the the specific heat, one only needs to know the total number of spins of the lattice. In the 2x2 lattice case the total number of spins is 4, and remember also that $\beta J = 1$ and $k = 1$. The specific heat is then:

$$C_V = \frac{1}{kT^2}(< E^2 > - < E >^2)$$

$$= \frac{1}{kT^2}(\frac{128e^{-8} - 128e^{8}}{2e^{-8} + 2e^{8} + 12} - (\frac{16e^{-8} - 16e^{8}}{2e^{-8} + 2e^{8} + 12})^2)$$

$$= 0.128329$$

The term $< E^2 > - < E >^2$ is also called the variance of the energy and is precisely calculated just as shown above. The same variance calculation can be applied to calculating the susceptibility, but the energy terms have to be substituted for mean magnetization. Let's calculate the variance separate from the desired equation this time:

$$\sigma_M^2 =< M^2 > - < M >^2 = \frac{1}{Z}\sum_{i=1}^{M} M_i^2 e^{-\beta E_i} - (\frac{1}{Z}\sum_{i=1}^{M} M_i e^{-\beta E_i})^2$$

$$= \frac{32}{2e^{-8} + 2e^{8} + 12}(e^8 + 1) - (\frac{8e^8 + 16}{2e^{-8} + 2e^{8} + 12})^2 = 0.016004$$

Remember that the $\beta = 1$ in the 2x2 lattice case, so the susceptibility would be the same as the variance in that case. We would usually have to calculate the susceptibility with the following equation:

$$X = \frac{1}{k_b T}(< M^2 > - < M >^2) = \frac{1}{k_b T}\sigma_M^2 = 0.010853$$

To compute the above values, a standard Metropolis algorithm were implemented (kilde). This algorithm takes in a matrix, only ones or negative ones in this case, and then decides to flip or not flip the current value at a random location in the matrix (lattice). The probability of when to flip or not is in this case given by the Boltzmann distribution $e^{-E_i/T}$. If the algorithm decides to flip, the computed energy at that position in the lattice is added to the total energy of the system. The same happens for the magnetization. Initial values of the energy and magnetization is calculated beforehand. What initial matrix to use, that is a randomly generated matrix or a matrix with only positive spin, is determined by the user. After the Metropolis algorithm is finished, the cumulative energy, squared cumulative energy, cumulative magnetization and cumulative magnetization squared is calculated which is needed to later calculate the mean energy, mean magnetization, specific heat and susceptibility. This algorithm is based on random events and so one needs to perform the algorithm many times before the results are stable. Therefore the whole Metropolis algorithm is looped inside what is called a Monte Carlo loop. When the number of cycles are increased, the more stable the results will be. However, many cycles require a lot of processing power when the number of cycles get to around 10^6.

To be able to run the program in a time efficient manner, one needs to parallellize the program, and this was done by using MPI with 8 processors. Using MPI made the processing time much smaller, but it still took a long time for large lattice sizes.

Raw data were produced using c++ while plotting and further calculations were done using MatLab.

Further calculations include finding the probability distribution and the thermodynamic limit.

Results

The specific analytical value found in the method is compared to the numerical value calculated in c++ in the table below:

Table 1: Analytical and numerical solutions

Type	Analytical	Numerical	MC cycles needed
$< E >$	-7.983928	-7.98437	10 000
$< M >$	3.994643	3.99483	10 000
C_V	0.128329	0.124812	100 000
$< X >$	0.016004	0.0153613	100 000

To get good results for mean energy and mean magnetization took very few Monte Carlo cycles, only about 10^4. Then the values would be off only by a factor of about 10^{-3}. The susceptibility and the specific heat took more Monte Carlo cycles to get a precise and stable result. Correct results started showing around 10^5. For 10^4, both the specific heat and susceptibility gave unstable results.

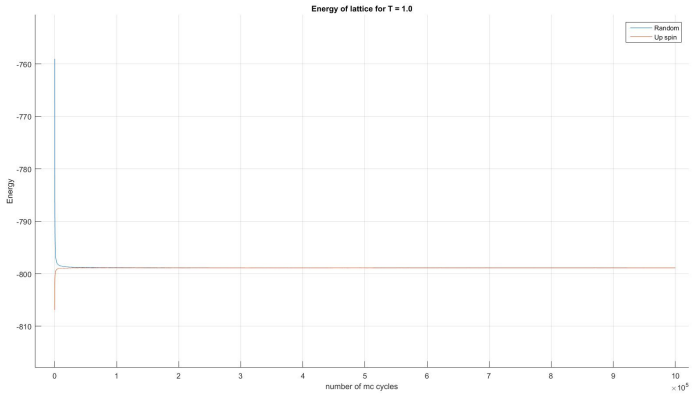


Figure 2: Energy versus Monte Carlo cycles for T = 1.0

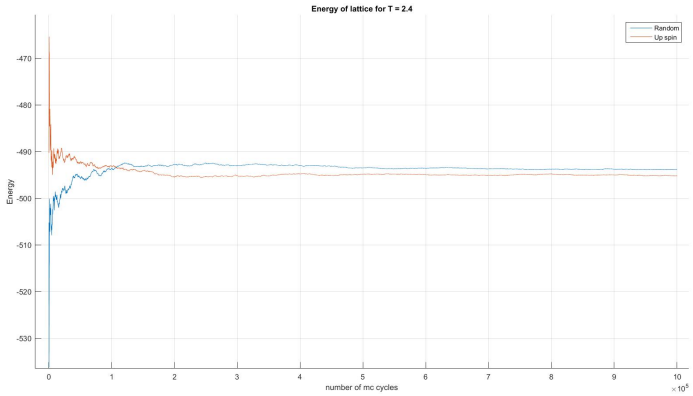


Figure 3: Energy versus Monte Carlo cycles for T = 2.4

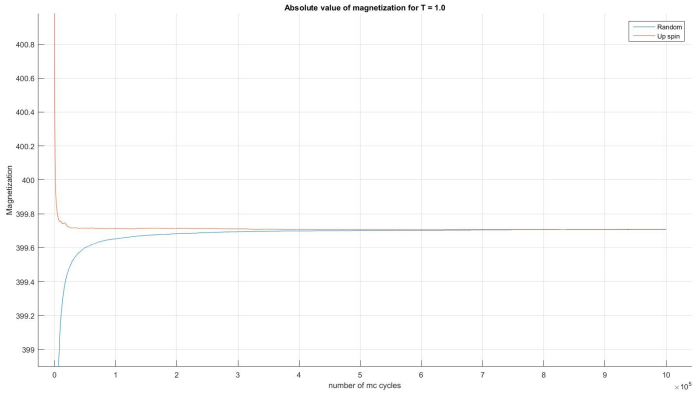


Figure 4: Absolute value of magnetization versus Monte Carlo cycles for T = 1.0

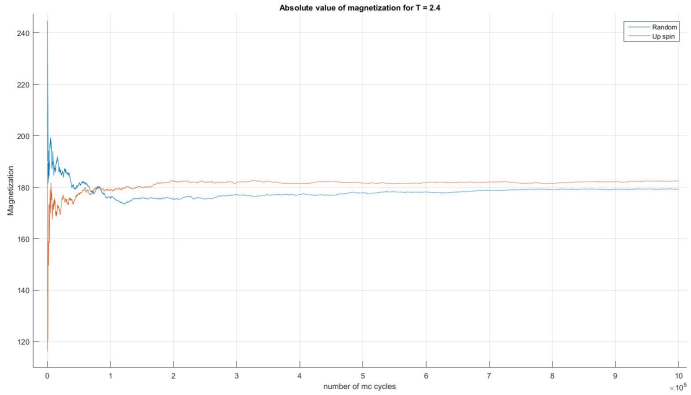


Figure 5: Absolute value of magnetization versus Monte Carlo cycles for T = 2.4

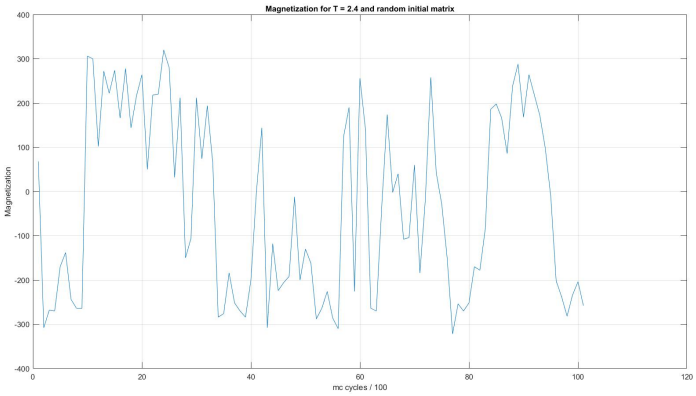


Figure 6: Mean magnetization versus Monte Carlo cycles for T = 2.4

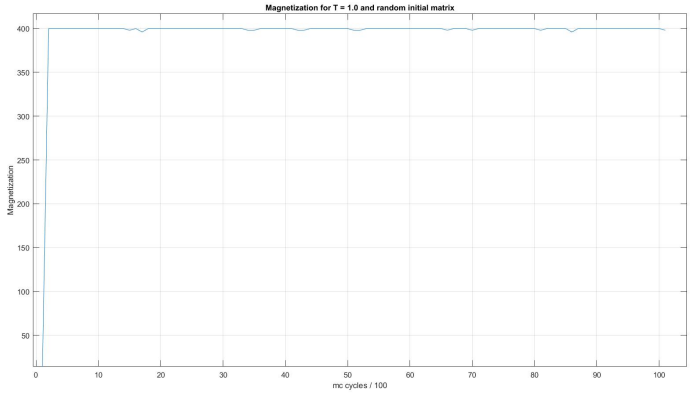


Figure 7: Mean magnetization versus Monte Carlo cycles for T = 1.0

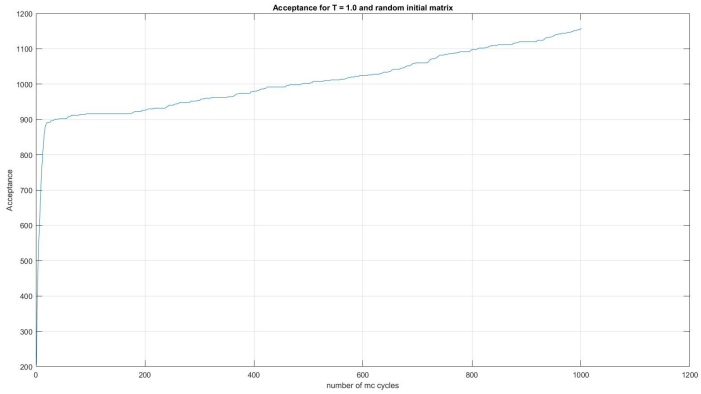
One can see from figure 2, figure 3, figure 4 and figure 5 that the mean energy and magnetization approaches it's equilibrium state. There are fluctuations after the mean energy and magnetization has reached it's equilibrium state, but the value seems to jump back to to the equilibrium state quickly. The values become more unstable when $T = 2.4$, and also the value of the equilibrium state is changed. The number of Monte Carlo cycles needed to hit equilibrium is different for different temperatures, but the initial matrix that is used do not seem to affect the time it takes to reach equilibrium. The table below gives an approximation of how many Monte Carlo cycles are needed:

Table 2: EQ values for energy

Orientation	Temperature	EQ state	max MC cycles needed
Random	1.0	-790.881	$2 * 10^4$
Random	2.4	-496.494	$6 * 10^5$
Up spin	1.0	-798.96	$2 * 10^4$
Up spin	2.4	-493.996	$6 * 10^5$

Table 3: EQ values for magnetization

Orientation	Temperature	EQ state	max MC cycles needed
Random	1.0	398.762	$6 * 10^5$
Random	2.4	165.209	$2 * 10^5$
Up spin	1.0	399.732	10^5
Up spin	2.4	176.087	$2 * 10^5$



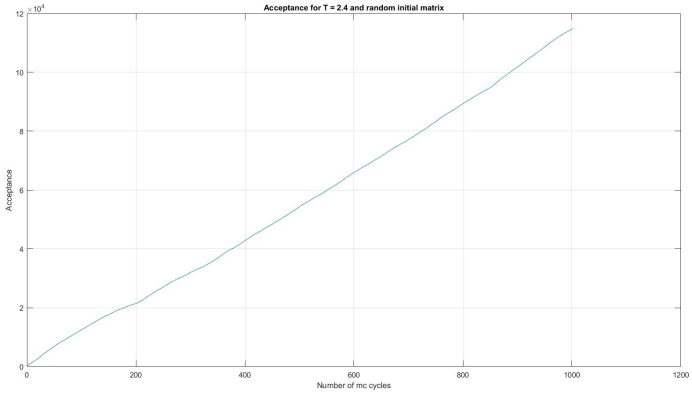
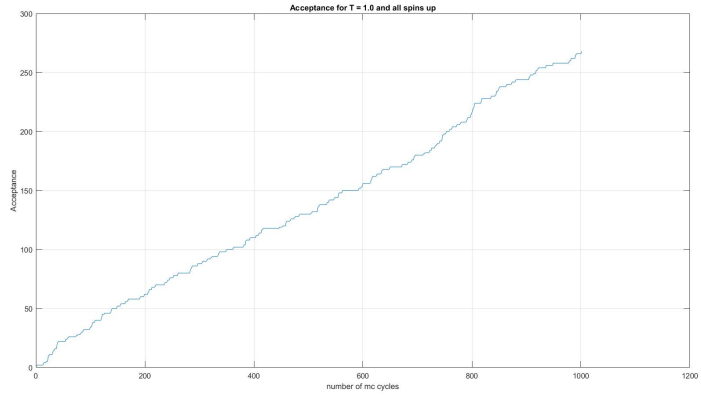


Figure 8: Acceptance versus Monte Carlo cycles for T = 1.0 and T = 2.4 with a random initial matrix



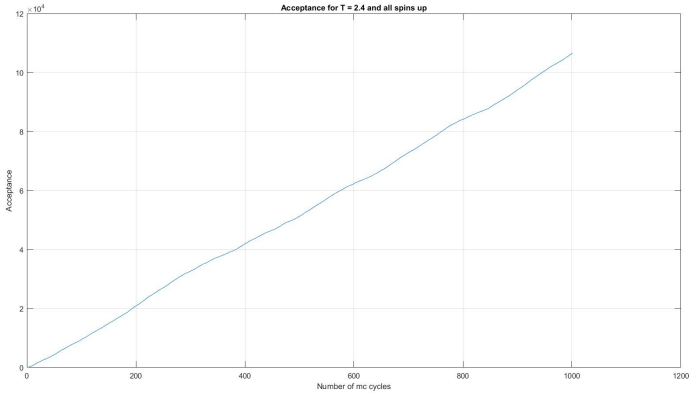
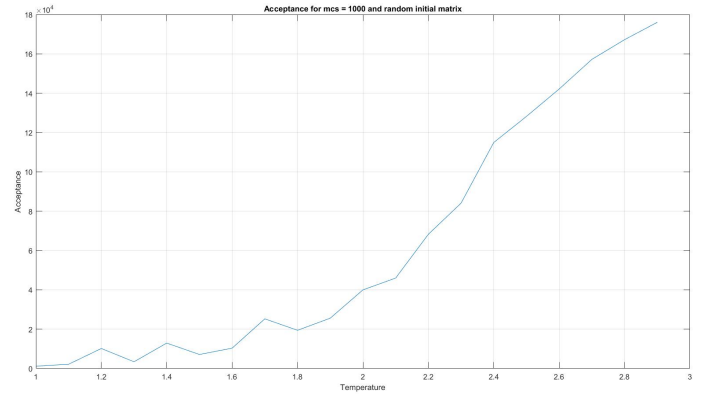


Figure 9: Acceptance versus Monte Carlo cycles for T = 1.0 and T = 2.4 with initial spin upwards



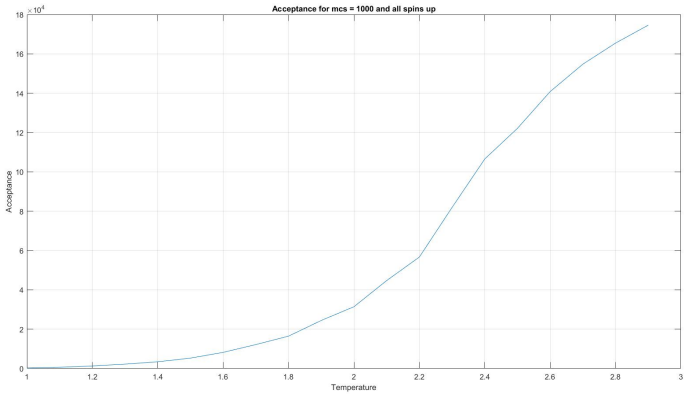
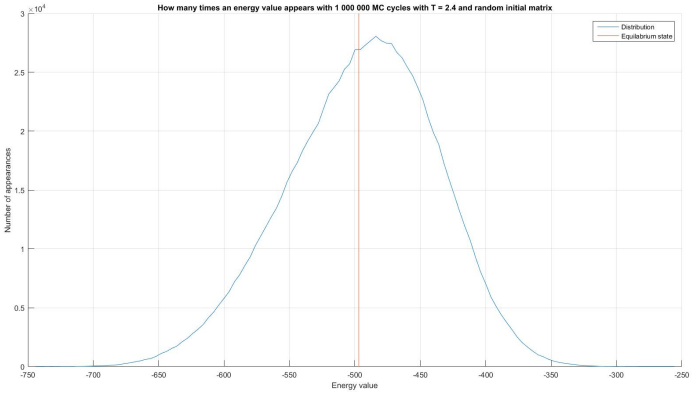


Figure 10: Acceptance versus temperature for a random initial matrix and with initial spin upwards

Acceptance is apparently dependent on both temperature and the initial state of the spins. From figure figure 8 one can observe that with higher temperature the number of accepted values increase by a factor of 10. For $T = 1.0$, one can also observe that the acceptance spikes for few Monte Carlo cycles.

If the initial spins are all positive like in figure 9, the acceptance becomes more linear. The line for



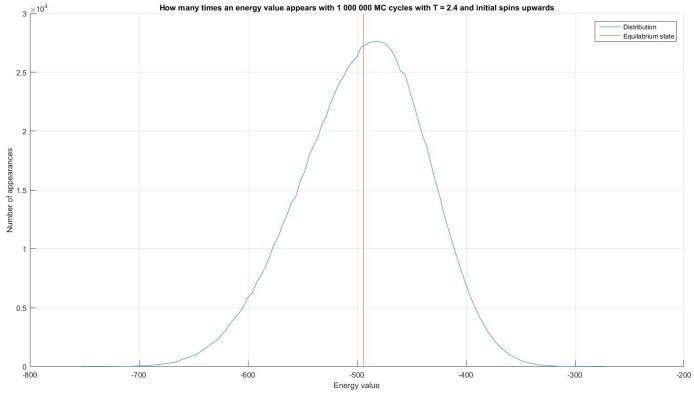


Figure 11: Number of appearances per energy value for a random initial matrix and with initial spin upwards when T = 2.4

$T = 1.0$ is more squiggly than the straighter line for $T = 2.4$. The ladder does not spike for few Monte Carlo cycles in this case.

From figure 10, one can observe that the curves are not linear. The curve based on a randomly generated initial matrix is uneven at low temperatures, but smooths out when T increases. When the initial matrix only consist positive spins, the curve is smooth throughout. These two plots are plotted to a maximum temperature of 3.0, so one would have an idea of how the acceptance would evolve after $T = 2.4$.

With a randomly generated initial matrix, the variance computed in c++ becomes 3226.62. The calculated variance from the energy becomes 3226.9 for 10^6 MC cycles (calculated in MatLab script "howmanytimes.m"). This gives great confidence in the variance. For an initial matrix where all the spins point upwards, the calculated variance in c++ becomes 3236.05. The variance calculated from energy values in MatLab becomes 3236.3, again showing the stability of the variance for 10^6 MC cycles. With fewer MC cycles, the two variances calculated does not correlate. Variances for other temperatures are shown in the table below:

Table 4: Variance for different temperatures

Temperature	Variance
T=1.0	10.1022
T=1.25	52.89957
T=1.50	181.264
T=1.75	478.217
T=2.0	1157.11
T=2.25	3145.31
T= 2.50	2479.27
T=2.75	1690.68
T=3.0	1445.93

One can observe from figure 11 that the curves are slightly skewed towards the higher energy values. The curve is bell shaped and somewhat centred around the equilibrium state for the lattice at temperature $T = 2.4$ which is -497 and -493.

The run time for higher order lattices increased as the lattice size increased. The run times are shown in the table below:

Table 5: Run time for different lattice sizes with 10^6

Lattice size	Run time in seconds
20x20	???
40x40	???
60x60	3996.57
80x80	7081.15
100x100	11176.2
120x120	???

Discussion

Comparing Table 2 and to figure 2 and figure 3 one can observe that the graph approaches the equilibrium value stated in the table. This is very clear for $T = 1.0$, but when $T = 2.4$, the energy seems to be oscillate which is natural due to the increased energy in the system (kilde). Even though the value slightly oscillates after reaching equilibrium, the energy is still said to have reached equilibrium as stated in Table 2 after a certain number of Monte Carlo cycles.

The mean magnetization plotted on figure 6 show the magnetization without the absolute value and tells us that the magnetization actually jumps from positive to negative magnetization, and vice versa, rather quickly and frequent after reaching equilibrium. The reason for this is because the energy values are equal for symmetric magnetization values around the equilibrium state (kilde). This makes it easy for the magnetization to make these kind of leaps. Looking at figure 4 and figure 5, the absolute value of the magnetization, we lose the above understanding, but we get the correct values, as calculated in c++, for the equilibrium state.

This statement is only true when $T = 2.4$, because one can observe on figure 7 that when $T = 1.0$ the magnetization does not fluctuate for positive and negative values. It seems as the magnetization reaches it equilibrium state and then shortly jump to some negative value, then right back to the equilibrium value. The reason for this is because the energy in the system is too low for the magnetization to make any great leaps (kilde). From the Boltzmann distribution we can see this mathematically:

$$w_i = e^{-E_i/T}$$

where w_i is the probability, E_i is the energy for and T is the temperature. One can immediately see that when the temperature increase, the probability decreases for low energy values.

This means that the likelihood of flipping the orientation of the objects in the lattice should be higher for $T = 2.4$ when the energy starts accumulating to higher values, since the probability of flipping should be higher than a random number. The magnetization would then fluctuate more when $T = 2.4$ and this is observed on figure 5 and figure 6. The same argument applies for the energy, when the temperature is high, the chance of flipping is higher for $T = 2.4$ compared $T = 1.0$ and one can see this on figure figure 2 and figure 3.

Following this train of thought, one would expect the rate of acceptance to be greater for higher temperature values and this can be seen by comparing the y-axis on figure 8 and figure 9. This is confirmed on figure 10 where the acceptance increases when the temperature increases. When the initial matrix is randomized, the results for low temperatures vary, due to the initial starting condition. Therefore, this does not happen when the initial matrix only has upwards spin since the acceptance does not have to stabilize.

Looking at figure 11, one clearly sees that both graphs are skewed towards higher energy values, when one may have expected the curve to center around it's equilibrium state.