Project 4 FYS3150

ULRIK BERG ARUP SEIP

Abstract

In this project I will be implementing the metropolis algorithm for the Ising model, to study phase transitions in ferromagnets. The resulting data will be measured against analytical values for a simple system.

Introduction

The first part of the project explains the theory behind what we are trying to simulate numerically, starting with the basics thoughts behind the Ising model. By starting with an analytical approach to the problem we obtain baseline values to measure the results of our algorithm against. Having found out tha

is dedicated to simulating the Earth's orbit around Sol, in order to test our two algorithms. Once these algorithms were properly tested, the velocity-Verlet was deemed the most suitable of the two, and we dropped using the forward-Euler algorithm from then on. For the second part of the project, we expanded our algorithm such that it would take the gravitational fields of Sol, all eight planets, and Pluto into account for each body. Finally, we ran a series of tests to confirm the functionality of each component in our simulations.

Theory

The Ising model

Analysing and understanding magnetic fields analytically is complicated and requires too much knowledge of mathematical techniques for it to be a reasonable approach for the lay people. To get around this problem, we can instead use a numerical approach. In this project we will be using a 2-D model composed of a $L \times L$ lattice S of binary spins S_{ij}.

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,N} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N,1} & s_{N,2} & \cdots & s_{N,N} \end{bmatrix}$$

A spin in this lattice has an energy described by its own spin, and its surrounding spins. For example, the energy of $S_{2,2}$, marked in green, would depend on the spins marked in red:

$$\begin{matrix} \uparrow & \uparrow & \downarrow & \uparrow \\ \downarrow & \downarrow & \uparrow & \downarrow \\ \downarrow & \uparrow & \downarrow & \uparrow \\ \uparrow & \uparrow & \downarrow & \downarrow \end{matrix} \Leftrightarrow \begin{matrix} +1 & +1 & -1 & +1 \\ -1 & -1 & +1 & -1 \\ -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 \end{matrix}$$

This energy is then found by taking the sum over the nearest neighbours

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

Using the above example s_k becomes $s_{2,2}$, while s_l takes the values of $s_{1,2}, s_{2,3}, s_{2,4}$ and $s_{1,3}$. J is an unspecified constant representing interparticle interactions. This gives us $E_{2,2} = -2J$. The energy of the entire lattice then becomes the sum of all the particles connections' energies:

$$E_{tot} = -J \sum_{i=1}^{L} \sum_{j=1}^{L} \sum_{<kl>}^{L} s_k s_l \tag{2}$$

For the edge cases, where there are less than 4 bordering spins, we apply periodic boundary conditions, meaning $s_{-1,L+1} = s_{L,1}$ and $s_{L+1,-1} = s_{1,L}$. While calculating the total energy, we also make sure not to double count the connected spins, meaning if we find the energy of one spin, as a function of the surrounding spins, we need to skip the connection between said spin and its neighbours when calculating the spin of the neighbours.

Microstates and macrostates

Before we begin exploring an analytical solution to a simple system we need to introduce the concept of microstates and microstates. Consider the four following systems:

↓ ↑ ↑ ↓ ↑ ↑ ↑ ↑
↑ ↑' ↑ ↑' ↓ ↑' ↑ ↓

Using (2) they all have the same total energy, which makes them the same macrostate, even though they are all different, meaning they are different microstates. In these systems however:

↓ ↓ ↑ ↓ ↑ ↑ ↑ ↑
↑ ↑' ↑ ↑' ↓ ↓' ↑ ↓

1 and 3, and 2 and 4 are in the same macrostate, but they are all different microstates. In a system where you know the probability of every microstate these states can be put together in a partition function to determine several properties of the system.

An analytical approach

When performing an analytical analysis of any system, it is ideal to have some analytical values to measure against. This way we know that we are on the right track before we continue to more advanced systems that we simply cannot effectively model analytically. The four properties we will be measuring are:

Mean energy	$\langle E \rangle$
Mean absolute magnetisation	$\langle \lvert M \rvert \rangle$
Heat capacity	C_V
Susceptibility	χ

The partition function

When working with a thermodynamic system, one of the most important function for any further analysis is the partition function:

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

Here, Z is the partition function for a set of microstates N, $\beta = \frac{1}{k_B T}$ and E_i is the energy of the i-th microstate given by (2). The value of Z is simply a scaling factor for the probabilities of the microstates.

Mean Energy

Now that we have found the partition function we can use it to find our desired properties starting with mean energy. The mean energy can be found by dividing the sum of the product of the probability of each microstate and said microstates energy by the partition function:

$$\langle E \rangle = \frac{1}{Z} \sum_{i=1}^{N} E_i e^{-\beta E_i} \tag{4}$$

Mean absolute magnetisation

The mean magnetisation works the same way. The simple explanation is that M is the sum of all spins, and so to find the mean M we do the same as with $\langle E \rangle$ in (4):

$$\langle M \rangle = \frac{1}{Z} \sum_{i=1}^{N} M_i e^{-\beta E_i} \tag{5}$$

Heat capacity

Heat capacity is the correlation between heat radiation from an object, and the resulting temperature change, and is quantified (Hjort-Jensen, 2015):

$$C_V = \frac{1}{k_B T^2} \sigma_E^2 \tag{6}$$

Here k_B is the Boltzmann constant, T is temperature, and σ_E the standard deviation of the energy across all microstates.

Susceptibility

The magnetic susceptibility of a material is that materials' tendency to become magnetised when exposed to an external magnetic field. The susceptibility (χ) is quantified (Hjort-Jensen, 2015):

$$\chi = \frac{1}{k_B T} \sigma_M^2 \tag{7}$$

Where σ_M is the standard deviation of the energy across all microstates.

A simple 2×2 lattice

Now that we have established analytical expressions for our values we can start with 2×2 lattices to get the basics of our model right, before we move on to larger lattices. We use a lattice that is as small as possible to make the analytical work easier for us. The analytical solutions for a $L \times L$ lattice require us to consider 2^{L^2} different microstates, which quickly gets out of hand. Even for our 2×2 lattice we get 16 different possible microstates:

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

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

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

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

Applying (3) to the above lattices we can generate a new matrix consisting of the corresponding total energies:

$$\begin{matrix} -8J & 0 & 0 & 0 \\ 0 & 0 & 0 & 8J \\ 8J & 0 & 0 & 0 \\ 0 & 0 & 0 & -8J \end{matrix}$$

Once again, I must mention that these values are obtained using periodic boundary conditions. Essentially what this does is simulating a larger lattice, of which our 2×2 lattice is a small part.

Similarly, we can obtain a matrix of the magnetization ($|M|$):

$$\begin{matrix} 4 & 2 & 2 & 2 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 2 & 2 & 2 & 4 \end{matrix}$$

With these values we can find proper expressions for $\langle E \rangle, \langle |M| \rangle, C_V$ and χ for a given as a function of T, for our 2×2 lattice. Firstly (3) gives us a partition function:

$$Z = 4\left(\cosh\left(\frac{8J}{k_B T}\right) + 3\right) \tag{8}$$

With a partition function we can use (4) to find $\langle E \rangle$:

$$\langle E \rangle = \frac{32}{Z}\sinh\left(-\frac{8J}{k_B T}\right) \tag{9}$$

And (5) to find $\langle |M| \rangle$:

$$\langle |M| \rangle = \frac{8}{Z}\left(e^{\frac{8J}{k_B T}} + 2\right) \tag{10}$$

To progress further, and find C_V we need the variance of E:

$$\sigma_E^2 = \langle E^2 \rangle - \langle E \rangle^2 = \frac{1}{Z}\sum_{i=1}^{N} E_i^2 e^{-\beta E_i} - \left(\frac{1}{Z}\sum_{i=1}^{N} E_i^2 e^{-\beta E_i}\right)^2$$

Which we can then plug into (6):

$$C_V = \frac{256}{k_B T^2 Z}\left(\cosh\left(-\frac{8J}{k_B T}\right) - \frac{1}{Z}\left(\cosh\left(-\frac{16J}{k_B T}\right) - 2\right)\right) \tag{11}$$

Similarly, we use the variance of $|M|$ to find χ:

$$\sigma_M^2 = \langle M^2 \rangle - \langle M \rangle^2 = \frac{1}{Z}\sum_{i=1}^{N} |M_i|^2 e^{-\beta E_i} - \left(\frac{1}{Z}\sum_{i=1}^{N} |M_i|^2 e^{-\beta E_i}\right)^2$$

Plugged into (7) we get:

$$\chi = \frac{32}{k_B T Z}\left(e^{\frac{8J}{k_B T}} + 1 - \frac{2}{Z}\left(e^{\frac{16J}{k_B T}} + 2e^{\frac{8J}{k_B T}} + 4\right)\right) \tag{12}$$

Algorithm

Having derived the baseline analytical expressions, we can now move on to the main objective of our project: to study phase transitions in ferromagnets numerically, using the Ising model. To do so we need an appropriate algorithm, and the Metropolis algorithm (Hjort-Jensen, 2015) is just what we want.

Simplified, one step of the Metropolis algorithm goes like this:

1. We start with a binary lattice (± 1).
2. We generate a list (W) of possible values for $w = e^{-\beta \Delta E}$, where $\beta = \frac{1}{k_B T}$
3. Generate L^2 random index pairs in S
4. For each random index pair:
   4.1. Find ΔE
   4.2. If $\Delta E < 0$: flip the polarity of the spin at the given index
   4.3. If $\Delta E > 0$:
      4.3.1. Generate a random number $q \in [0, 1]$.
      4.3.2. Set $w = e^{-\beta \Delta E}$ from W
      4.3.3. If $r < w$, flip the spin at the given index

This is a slightly modified version of the Metropolis algorithm. In step 2 we generate the list W, so that we can extract the desired value in step 4.3.2, instead of calculating $e^{-\beta \Delta E}$ every single time. This cuts the number of flops, per spin, per cycle roughly in half (depending on the temperature).

Parallelisation and multithreading

For repeatable tasks that can happen simultaneously we can make the computer perform them simultaneously instead of one by one linearly. One "line" of executions is called a thread, and by dividing the tasks into several such threads we can perform the tasks as many times faster as there are available threads in the processor of the computer. For most standard desktops there are 1-4 threads per core, and usually 2-8 cores. Intel's mid-range processors (which is what I happen to have in both my desktop and my laptop) all have a total of 8 threads, giving us a potential eightfold increase in efficiency. In our case the flipping of the spins can happen in any order because the many different random lattices are independent of each other, and so that part of the algorithm can be assigned to different threads.

Results

Analytical

The benchmark analytical values given by (9), (10), (11) and (12), are:

$$\begin{aligned} \langle E \rangle &= -7.984 \\ \langle |M| \rangle &= 3.994 \\ C_V &= \text{-0.003} \\ \chi &= 0.027 \end{aligned}$$

Numerical

For an equivalent simulation of a 2×2 Lattice, the numerical results are:

$$\begin{aligned} \langle E \rangle &= -1.994 \\ \langle |M| \rangle &= \text{-1.799} \\ C_V &= \text{-0.045} \\ \chi &= 0.006 \end{aligned}$$

Plots

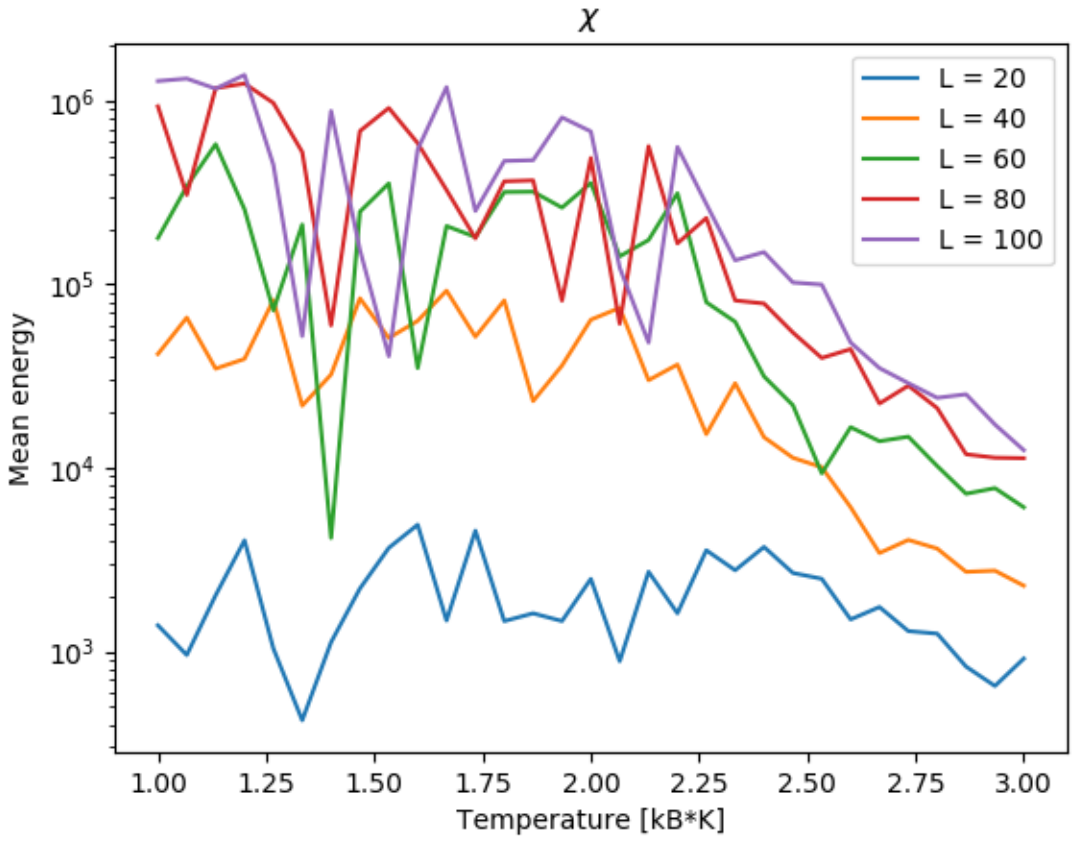

Figure 1 $\chi(t)$, with 5 monte carlo cycles

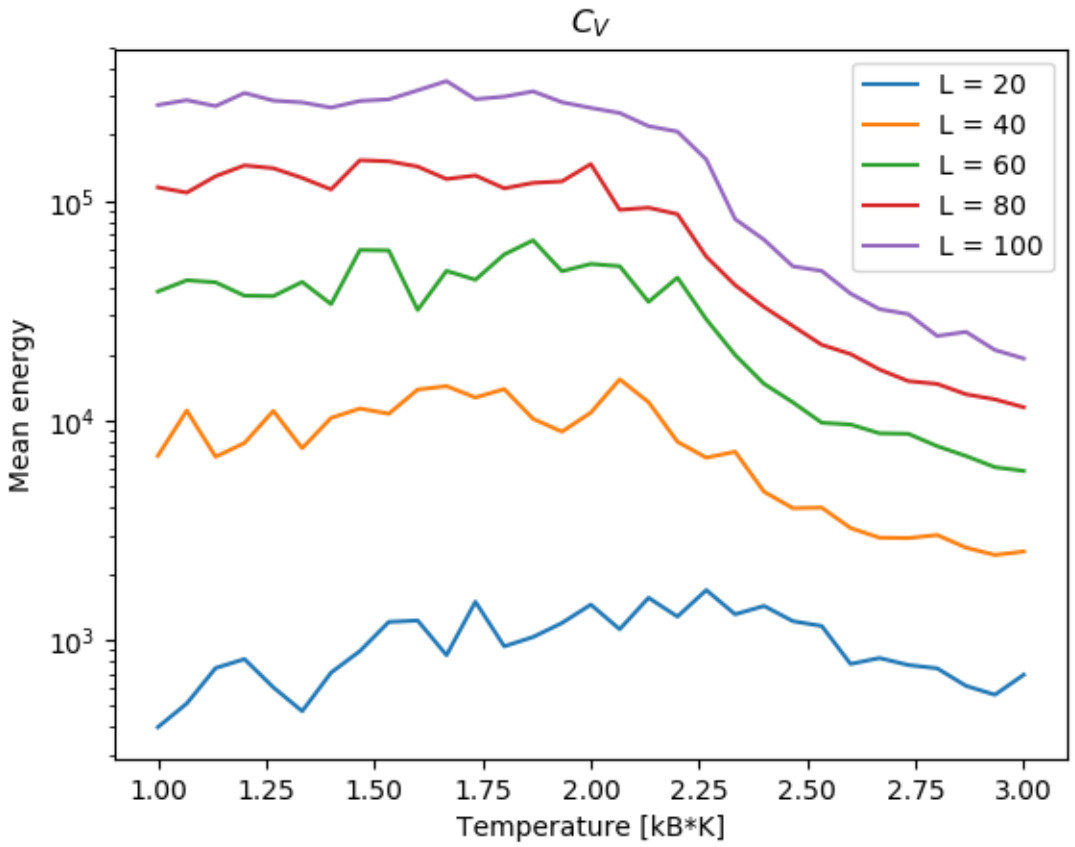
C_V
L = 20
L = 40
L = 60
L = 80
L = 100
10^5
10^4
10^3
Mean energy
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
Temperature [kB*K]

Figure 2

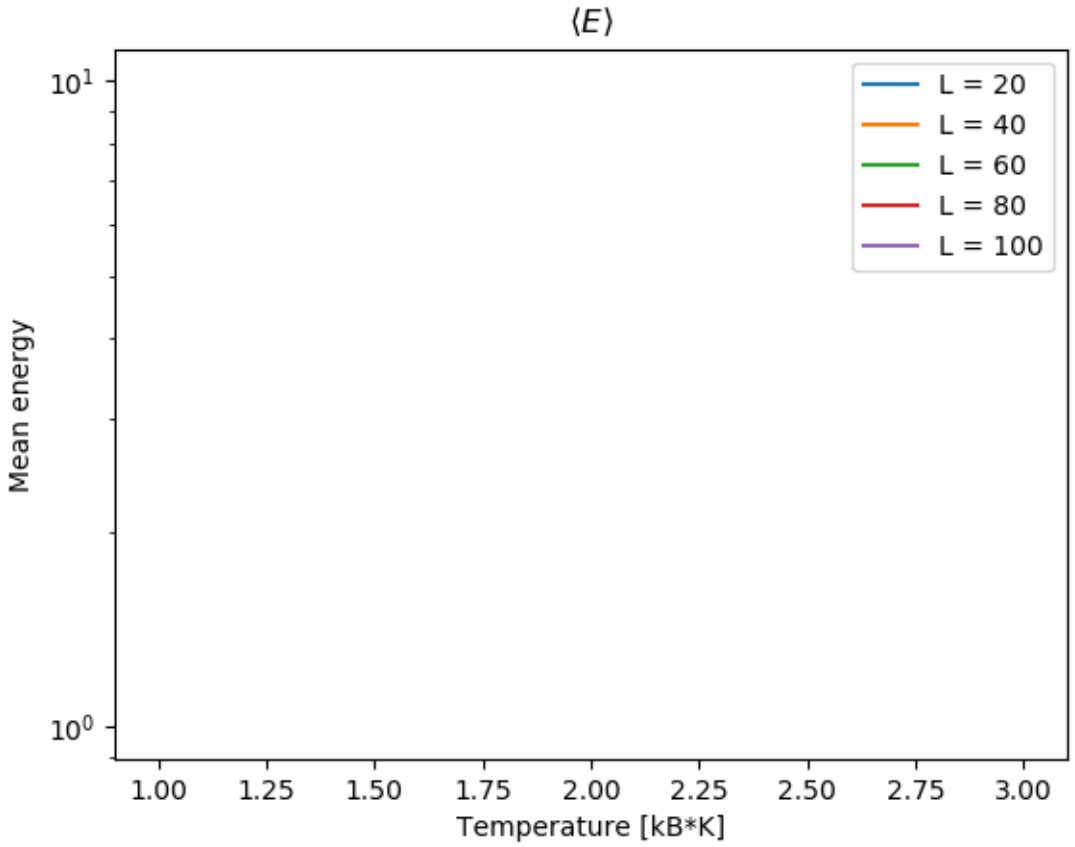

Figure 3 No time to rerun.

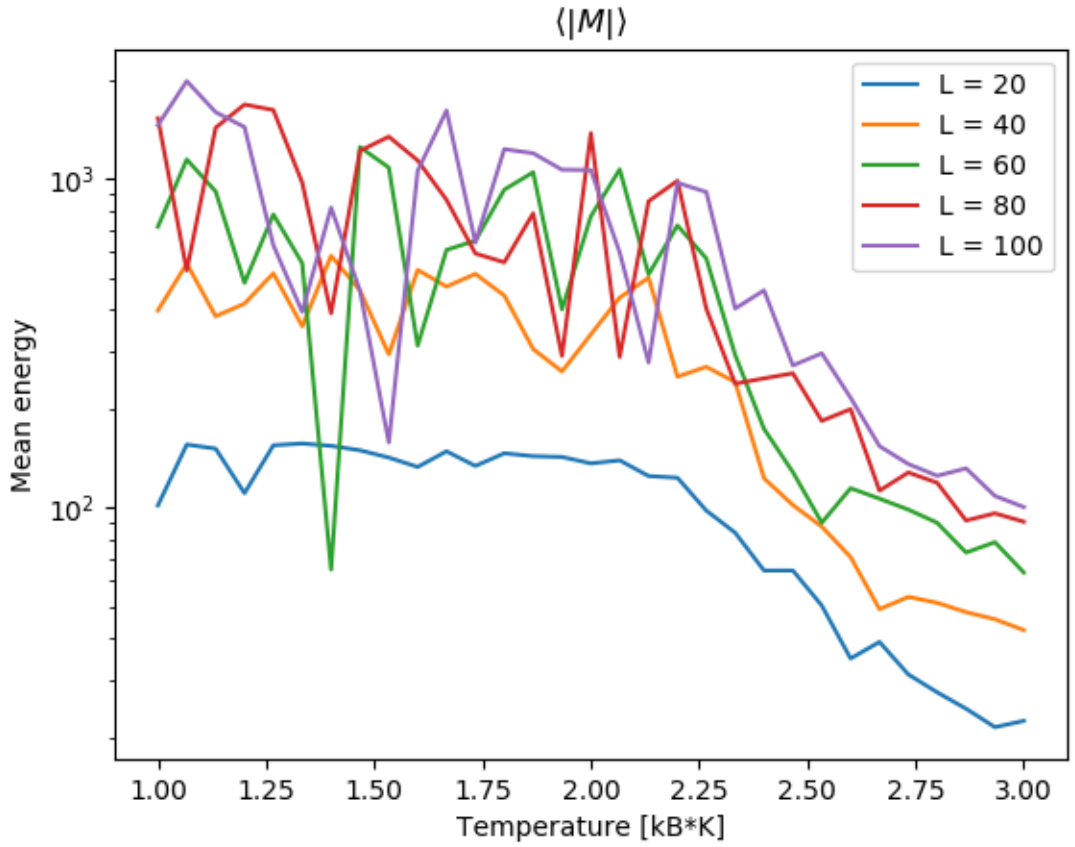
⟨|M|⟩
L = 20
L = 40
L = 60
L = 80
L = 100
10^3
10^2
Mean energy
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
Temperature [kB*K]

Figure 4

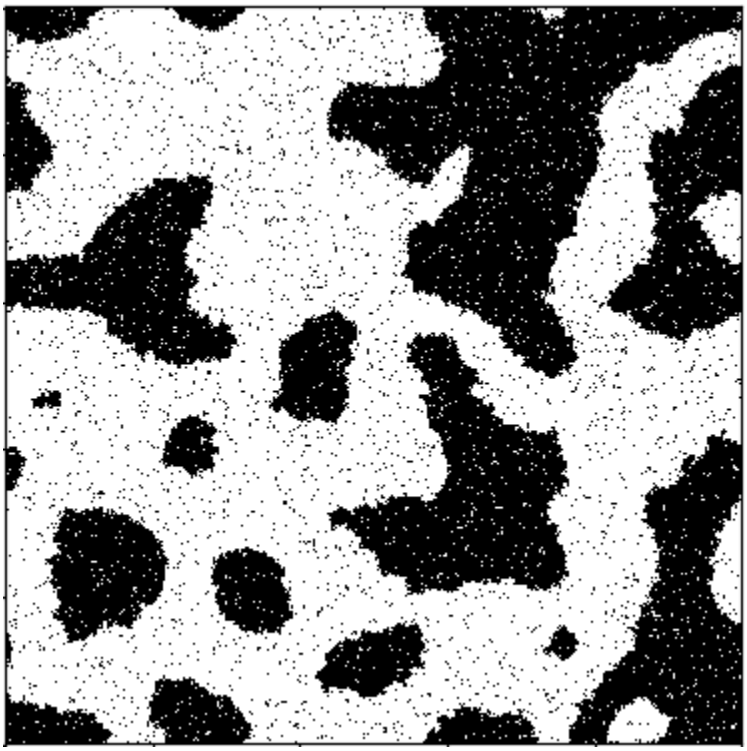

Figure 5 Resulting lattice after 1e4 monte carlo cycles. T = 1.5, L=1000

Conclusion

Comparison of analytical and numerical values

It is obvious that something went terribly wrong during the calculation of the values in the 2×2 lattice. However, as one can see from the simulation videos in the GitHub repo (Seip, 2018), and also the still image of a finished simulation in figure 5, the lattices do reach a steady point, which should give the values we get from the analytical model. In figure 2, we can see the magnetisation hovering at about $\frac{1}{4}$ of what it should for all values of L. I therefore believe that the error lies in calculation of the numeric values, and not in the algorithm itself. What we can take away from this is that although the different plots aren't necessarily correct, their shape should still be able to give us information about the critical temperature.

The most likely state

We can see from the videos (Seip, 2018), and also from figure 1-4 that the time it takes for the system to stabilise scales with both L and T. A higher L means that the magnetic fields that form in the lattice take more time to eat each other up, because they are bigger. A higher T creates more disturbance, which again affects the time used for these magnetic fields or flakes to consume each other. As far as I can tell, it takes around 500-3000 runs of the Monte Carlo cycle to completely stabilise the system.

Critical temperature

In figure 3 and 4 we see a sharp bend in the graphs around $T = 2.2$, which indicates a phase transition here, which is quite close to the actual analytical number of $T = 2.269$.

References

Hjort-Jensen, M. (2015). Computational physics. https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf.

Seip, U. (2018). Appended files. https://github.com/UlrikSeip/Projects/tree/master/prosjekt4.