FYS3150 - Project 4

Daniel Heinesen

November 15, 2016

Abstract

In this article we are going to investigate the Ising model, and its application to phase transitions. We are going to simulate grids of spin up to 140 for temperatures around $3K$ and use the results to find the critical temperature for phase transition.

Contents

1 Introduction

The Ising model is a simple but very popular model for modelling phase transitions (see e.g. **HERE**). The model consists of a lattice of atomic spins which can have one of two possible values ("up" or "down"). The energy of each spin in their configuration is determined solely by their nearest neighbors. This model is widely used in statistical physics, in particular to study ferromagnetic phenomena.

We will investigate how multiple interesting thermodynamic quantities, including heat capacity, magnetic susceptibility, mean magnetization and mean energy, behave over time in the Ising model. We will investigate both ordered and disordered initial states of the lattice, checking the equilibration time in each case. Finally, we will also investigate phase transitions in the Ising model. We implement periodic boundary conditions and simulate temporal progression in the

lattice by means of the Metropolis algorithm.

We begin with a discussion of the thermodynamic quantities that we will study, and how they are manifested in the Ising model. We then briefly discuss our chosen boundary conditions. Subsequently, we present the theory of phase transitions and the Metropolis algorithm, and discuss how we could identify the equilibrium of our system. We then discuss some technicalities relating to our implementation of the physical model, before presenting and discussing our results..

2 Theoretical model

2.1 Thermodynamic quantities and the Ising model

We study a square lattice consisting of $N \times N$ magnetic dipole moments (atomic spins) that can be in one of two states (+1 or -1). We wish to study how energy, magnetization, heat capacity and magnetic susceptibility develop in this system. This system, the Ising model, is a microcanonical system, i.e. we keep the temperature fixed. Then it is known from basic thermodynamics that this system can be described by Boltzmann statistics, i.e. by a distribution function given by:

$$P(E) = \frac{1}{Z} e^{-\frac{E}{k_B T}} \tag{1}$$

Where E is the energy of a microstate, k_B is Boltzmann's constant and T is the temperature of the system. We define $\beta = 1/(k_B T)$. Then Z, the partition function, is a normalization constant defined by:

$$Z = \sum_i e^{-\beta E_i} \tag{2}$$

Where the sum is over all microstates, i. As $P(E)$ is a probability function, it may be used to define moments of a quantity X according to the general scheme:

$$\langle X^n \rangle = \frac{1}{Z} \sum_i X_i^n e^{-\beta E_i} \tag{3}$$

Where X_i is a quantity associated with the microstate i. This will be useful later, when computing the heat capacity and the magnetic susceptibility.

From this probability distribution, and the partition function, we can compute all thermodynamic quantities of interest.
In the simplest model of the Ising system, the energy is given by:

$$E = -J \sum_{\langle kl \rangle} s_k s_l \tag{4}$$

Where the sum is over all the nearest neighbors in the lattice. J is a constant which is determined by the quantum mechanical details of the system. We will simply set it to 1. The magnetization, on the other hand, is given as:

$$\mathcal{M} = \sum_i s_i \tag{5}$$

Where the sum goes over all spins, i.

It is shown **here** that this gives the following relation for thermodynamical quantities:
Heat capacity, C_V

$$C_V = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2} \tag{6}$$

Magnetic susceptibility, χ:

$$\chi = \frac{\langle \mathcal{M}^2 \rangle - \langle \mathcal{M} \rangle^2}{k_B T} \tag{7}$$

The first moment of the energy, $\langle E \rangle$, can also be computed in another way, which will be useful later, namely:

$$\langle E \rangle = -\frac{\partial \ln Z}{\partial \beta} \tag{8}$$

This is proved **here**.

2.2 Periodic boundary condition

One problem of this model is how to treat the boundaries. The easiest way is to say that the spins on the boundaries have no neighbors. Real materials have close to infinite atom spins, so the boundaries are negligible. But we can't simulate infinite lattices, so we have to find another. With periodic boundary conditions, the spins at the boundary treats the spins at the opposite boundary as its neighbor. This will make a good approximation to an infinite lattice, since every spin has a neighbor. Because of this does the boundary condition corresponds better with reality.

2.3 Phase transitions in the Ising model

It is well known from literature **REFERENCE** that there exists a critical temperature, T_C, at which phase transitions can be observed in the Ising model. At this temperature, one can observe spikes in the thermodynamic quantities. Near T_C,our thermodynamic quantities can be modelled as simple power laws. As shown **HERE**, these laws take the following forms:

$$\begin{aligned} \langle \mathcal{M}(T) \rangle &\sim (T - T_C)^{\beta} \\ C_V(T) &\sim |T_c - T|^{-\gamma} \\ \mathcal{X}(T) &\sim |T_C - T|^{-\alpha} \end{aligned} \tag{9}$$

Where α, β and γ are critical exponents, given by: $\alpha = 0,\ \beta = 1/8,\ \gamma = 7/4$

The temperature T_C will depend upon the number of spins in our lattice, N. Ideally, the number of spins should be close to infinite. However, our computational capacities limit us to systems with a maximum size of about $N = 140$. Luckily, however, we can estimate the critical temperature for $N = \infty$, $T_C(N = \infty)$ from the critical temperature at a finite N, $T_C(N)$ by the equation shown **HERE**, which we reproduce:

$$T_C(N) - T_C(N = \infty) = aN^{-1/\nu} \tag{10}$$

Where a is a constant, which can be determined from:

$$a = \frac{T_C(N_1) - T_C(N_2)}{N_1^{-1/\nu} - N_2^{-1/\nu}} \tag{11}$$

ν is another critical temperature, which has the exact result $\nu = 1$. Thus we can estimate the critical temperature by computing a from equation 11 for two different lattice sizes, N_1 and N_2, and then solving equation 10 for $T_C(N = \infty)$. Note that we can also use the equations in 9 near the critical temperature as analytic solutions to which we can compare our numerical results.

2.4 The Metropolis Algorithm

We will use Monte-Carlo simulations to model the time development of our spin system. Assume that we have an initial random configuration of spins, with energy E_b (which we have calculated from equation 4). We employ the famous Metropolis algorithm to achieve this. The Metropolis algorithm is described in detail **HERE**, but it can be briefly summarized in the following steps:

- Randomly select N^2 spins from the $N \times N$ lattice.
- For each spin, calculate the change in energy, ΔE, that the system would experience if we were to flip it
- Draw a random number, ζ uniformly between 0 and 1, and compare it to $e^{-\beta \Delta E}$.
- If $\zeta \leq e^{-\beta \Delta E}$, flip the spin, else do not flip the spin (note that any spin flip with $\Delta E < 0$ will always be performed).
- Update the thermodynamic quantities.

It can be shown (**REFERENCE**) that this simple scheme will evolve the system towards its equilibrium state (dictated by the Helmoltz free energy). Thus, repeating the above steps many times, which corresponds to performing many Monte-Carlo cycles, will evolve our system towards equilibrium. We will therefore use the number of Monte-Carlo cycles synonymously with time in our simulation.

2.5 Equilibrium of the system

We choose multiple different starting configurations of our lattice, including a homogeneous configuration (all spins initially point the same way), and a random configuration (spins drawn from a uniform distribution). We expect that our system should approach a stationary state after some time, T. There are multiple indicators of this state. First of all, we expect the number of accepted spins to reach a low, constant value, i.e. the total number of accepted spins per time steadily decreasing curve, approaching an equilibrium value. Most spins will be in their stable configuration at equilibrium, and therefore few flips will be occur. We also expect all thermodynamical quantities to reach an equilibrium state, where the variations are small. Both of these will be good indicators for equilibrium.

2.6 Analytic solution for the 2x2 case with periodic boundary conditions

It is not too difficult to find an analytic solution for a small latice consisting only of 2×2 spins. This analytic solution provides an important consistency check for our algorithm. Assume therefore that we have a lattice consisting of 4 magnetic dipole moments, labelled as shown in figure 1:

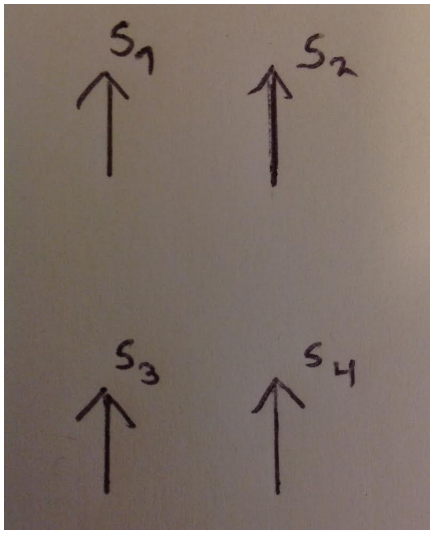


Figure 1: One microstate of our 2×2 lattice, illustrating how we label the spins.

We notice first there are only 5 distinct macrostates, namely 0 to 4 spins up. There should be a total of $2^4 = 16$ microstates. It is clear that the extremes (0 spins up and 4 spins up), have only one microstate associated with them. The states 1 spin up and 3 spins up each have 4 associated microstates (you can flip one of the four spins). This implies that there are a total of 6 ways to have the state 2 up and 2 down. This is summarized in the table below:

Spins up	Number of microstates
4	1
3	4
2	6
1	4
0	1

Now we must investigate the energy of each of these states. In general, the energy with periodic boundary condition for a 2×2 lattice can be written as:

$$E = -J\left[s_1 s_2 + s_2 s_1 + s_1 s_3 + s_3 s_1 + s_2 s_4 + s_4 s_2 + s_3 s_4 + s_4 s_3\right]$$

From this it follows immediately that the energy for the states for 4 spins up or 4 spins down is $-8J$. If we flip one spin, we will flip the sign of exactly half the terms (because each spin occurs in four of the eight pairs). Thus the energy will be zero. If we flip two spins, we can either flip spins that are nearest neighbors (such as s_1 and s_2, s_1 and s_3, s_2 and s_4 or s_3 and s_4), or we can flip spins that are not nearest neighbors (such as s_1 and s_4 or s_2 and s_3). In the first case, we flip the sign of four of the terms in the sum (namely all those that contain one of the two spins we flip). Thus the total energy will be zero. In the other case, we flip the sign of all terms. Thus the total energy will be 8J.

To find the magnetization, we must simply sum the total number of spin up and subtract the number of spin down. This is all summarized in the table below:

Spins up	Energy [J]	Magnetization[J/T]	Number of microstates
4	-8J	4	1
3	0	2	4
2	0	0	4
2	8J	0	2
1	0	-2	4
0	-8J	-4	1

This enables us to calculate the partition function as:

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

Where $\beta = 1/k_B T$. As most of the energies are zero, this is relatively straightforward and gives:

$$Z = 2e^{8\beta J} + 2e^{-8\beta J} + 12 = 4\cosh(8J\beta) + 12$$

We can now compute the expected value of the energy as:

$$\langle E\rangle = -\frac{\partial \ln Z}{\beta} = -\frac{\partial}{\partial\beta}\ln\left(2e^{8\beta J} + 2e^{-8\beta J} + 12\right) = \frac{-16Je^{8\beta J} + 16Je^{-8\beta J}}{2e^{8\beta J} + 2e^{-8\beta J} + 12} = -\frac{8J\sinh(8\beta J)}{\cosh(8\beta J) + 3} \tag{12}$$

Similarly, we can compute the heat capacity, C_V, as:

$$C_V = \frac{d\langle E\rangle}{dT} = \frac{d}{dT}\left(\frac{-16Je^{\frac{8J}{k_B T}} + 16Je^{-\frac{8J}{k_B T}}}{2e^{\frac{8J}{k_B T}} + 2e^{-\frac{8J}{k_B T}} + 12}\right)$$

However, this is an ugly differentiation. Therefore, we will instead use the relation:

$$C_V = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}$$

The square of the expected value of the energy is given by:

$$\langle E^2 \rangle = \frac{1}{Z} \sum_{i=1}^{16} E_i^2 e^{-\beta E_i}$$

The sum is easy to compute:

$$\sum_{i=1}^{16} E_i^2 e^{-\beta E_i} = 64J^2 e^{8J\beta} + 128J^2 e^{-8J\beta} + 64J^2 e^{8J\beta} = 128J^2 \left(e^{8J\beta} + e^{-8J\beta} \right)$$

Which gives:

$$\langle E^2 \rangle = \frac{128J^2 \left(e^{8J\beta} + e^{-8J\beta} \right)}{2e^{8J\beta} + 2e^{-8J\beta} + 12} = \frac{256J^2 \cosh(8J\beta)}{4\cosh(8J\beta) + 12}$$

From which it follows that:

$$C_V = \frac{1}{k_B T^2} \left(\frac{128J^2 \left(e^{8J\beta} + e^{-8J\beta} \right)}{2e^{8J\beta} + 2e^{-8J\beta} + 12} - \left(\frac{-16Je^{8\beta J} + 16Je^{-8\beta J}}{2e^{8\beta J} + 2e^{-8\beta J} + 12} \right)^2 \right)$$

$$C_V = \frac{1}{k_B T} \left(\frac{256J^2 \cosh(8J\beta)}{4\cosh(8J\beta) + 12} - \left(\frac{-8J \sinh(8J\beta)}{\cosh(8J\beta) + 3} \right)^2 \right) \tag{13}$$

Magnetization can be computed as:

$$\langle M \rangle = \frac{1}{Z} \sum_{i=1}^{16} M_i e^{-\beta E_i}$$

The sum can be computed as:

$$\sum_{i=1}^{16} M_i e^{-\beta E_i} = 4e^{8J\beta} + 8 - 8 - 4e^{8J\beta} = 0$$

Thus the expected value of the magnetization is zero. The expected value of the absolute value of the magnetization on the other hand is:

$$\langle |M| \rangle = \frac{1}{Z} \sum_{i=1}^{16} |M_i| e^{-\beta E_i} = \frac{1}{Z} \left(4e^{8J\beta} + 4 \cdot 2 + 4 \cdot 2 + 4e^{8J\beta} \right) = \frac{2e^{8J\beta} + 4}{\cosh(8J\beta) + 3} \tag{14}$$

The square of expected value of the magnetization can be computed as:

$$\langle M^2 \rangle = \frac{1}{Z} \sum_{i=1}^{16} M_i^2 e^{-\beta E_i} = \frac{1}{Z} \left(16e^{8\beta J} + 4 \cdot 4 + 4 \cdot 4 + 16e^{8\beta J} \right) = \frac{8e^{8\beta J} + 8}{\cosh(8J\beta) + 3}$$

Now it is easy to compute the susceptibility:

$$\chi = \frac{\langle M^2 \rangle - \langle M \rangle^2}{k_B T} = \frac{1}{k_B T} \left(\frac{8e^{8\beta J} + 8}{\cosh(8\beta J) + 3} \right) \tag{15}$$

We can now simulate a 2×2 lattice, and check if equations 12, 13, 14 and 15 hold true, which gives us a consistency check for our methods.

3 Methods

3.1 Energy in our system

The fastest and easiest way of implement the Ising method is to flip one spin at the time. This is because the energy difference ΔE in the system only will depend on the energy difference on a single spin. ΔE of a spin only depends on its closest neighbors, and can only have one of five values:

$$\Delta E = \{-8J, -4J, 0, 4J, 8J\}$$

This means that all the values for $e^{\beta \Delta E}$ can be precomputed. This saves us a lot of this in large simulations.

For each spin flip, we calculate the ΔE by checking the spin of the neighbor spins, and using 3.3.

Before starting the Metropolis algorithm the energy of the system has to be calculated this is done by looping though all the spins and calculating:

```
for i in range(Nspins):
    for j in range(Nspins):
        energy -= spinMatrix(i,j)*(spinMatrix(i-1,j)+ spinMatrix(i,j-1));
```

3.2 Implementing periodic boundary conditions

The easiest way of implementing the boundary conditions is with modular division. Every time we want to calculate ΔE, we have to use four nearest neighbors, but this neighbor may not exist. If the spin we are calculating ΔE is x_{N-1}, then its neighbor x_N does not exit. Instead we will check $periodic(N)$, where

$$periodic(i) = (i + N) \text{mod} N \tag{16}$$

We can check this we can test with $periodic(N)$ which should be 0

$$periodic(N) = (N + N) \text{mod} N = 0 \tag{17}$$

So as we can see this loops to the other side of the lattice.

3.3 Implementing the Metropolis algorithm

We implemented the Metropolis algorithm by looping over two nested for-loops. The outer for-loop loops over the number of Monte Carlo cycles, while the inner loops over the number if spin. Inside the loops a spin in the lattice is chosen to be flipped. The then energy difference is calculated from

$$\Delta E = 2J \sum_k s_i s_k$$

we use this to to find the precalculated Boltzmann factor, and compare this with a random variable $\zeta \in [0, 1]$, as described in 2.4, and check if this flip is acceptable.

Here is psudo-code describing the implementation of the Metropolis algorithm. Periodic boundary conditions are not implemented:

```
for i in range (MCSteps):
    for k in range(N*N):
        x=int(random.uniform(0,1)*N)
```

```
y=int(random.uniform(0,1)*N)
dEnergy=2*grid(x,y)*(grid(x+1,y)+grid(x-1,y)+grid(x,y+1)+grid(x,y-1))
boltzmann=boltzmann_factor(dEnergy) //Precomputed factors
if random.uniform(0,1)=<boltzmann: //Metropolis test
    lattice(x,y)*=-1
    energy +=dEnergy
    magnetic_moment=2*lattice(x,y)
```

3.4 Investigating the time to reach at the most likely state, and probability distribution

We are interested in seeing how long it takes the system to reach the most likely state, and if this depends on the temperature. We also wish to see if the start configuration change this time. We want to check when the mean energy and the mean absolute value of the magnetization smooth out and converges to a value. Since is difficult to do in a qualitative way, so we are just going to plot these values against the Monte Carlo cycles, and look for the cycle where this happens. We will also look at the number of accepted flips, since, as discussed in section 2.5, we also want the accepted flips to converge as we reach equilibrium.

We also want to look at the probability distribution of the equilibrium state. This is done by counting the number of times a certain E appears. We then have to use what we found above, and start when the equilibrium state is started.

3.5 Investigating phase transitions

We already have an analytical temperature for $T_C(N = \infty)$ from section 2.3, $\sim 2.269K$. We are there for only going to look at temperatures in an interval around this temperature while investigating the phase transitions. At the critical temperature the heat capacity and the magnetic susceptibility are going to diverge, but since we are using finite lattices, they aren't going to go to diverge, but they are going to show a value much higher then for the surrounding. This spike will give the critical temperature, which we can compare with the equations from section 2.3.

3.6 Parallelizing our code

Monte Carlo methods have an error(variance)

$$\sigma^2 \sim \frac{1}{\sqrt{N}}$$

This means that the higher the number of cycles we use, the more accurate the simulation will be. One way of getting more cycles without increasing the run time is with parallelization. MPI lets us use all the core of the CPU. The Monte Carlo calculation as sent to the cores, and when the they are done they(the slaves) will send to data to a master core/node. This node will find calculate the mean of all the data. We are mainly using 4 cores, so with MPI, we will get 4 times the Monte Carlo cycles, and more accurate data. [1]

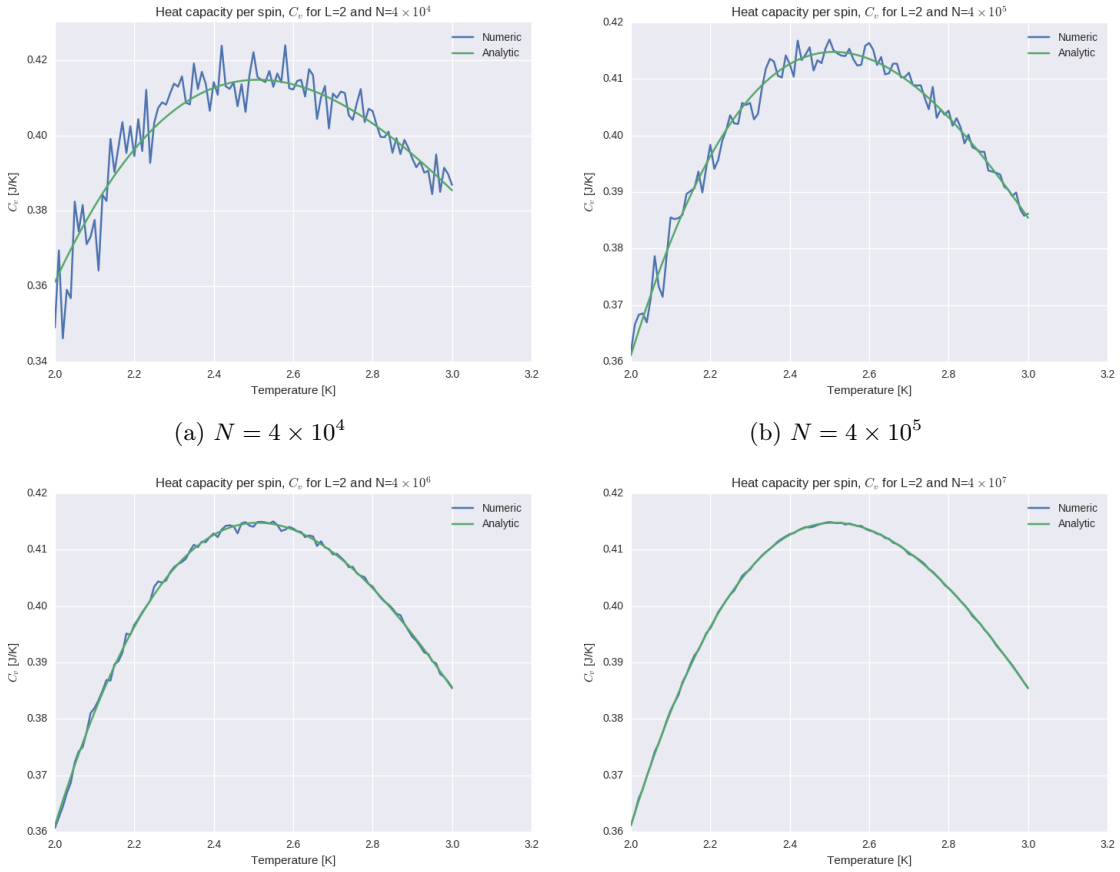
4 Results

In this section, we present the results from our investigation. We postpone an extended discussion of these results until the next section.

[1] Note that we initially intended to run on a larger cluster but, due to time constraints, this was not possible.

4.1 Results from the 2 × 2 lattice

We begin by investigating the number of steps required to get a good correspondence between the analytic expressions from section 2.6 and our simulations. We choose to compare the heat capacity of the system, as this has the most complex form. We let each core run an N that is divisible by 10, so that the total number of steps will be 4 times a number divisible by 10. Note that we plot all of these quantities *per spin* to ease readability.



(a) $N = 4 \times 10^4$

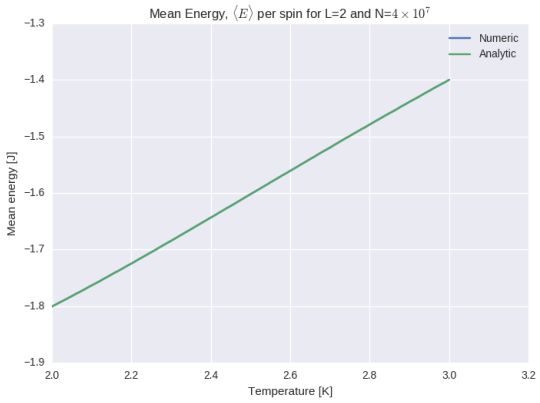
(b) $N = 4 \times 10^5$

(c) $N = 4 \times 10^6$

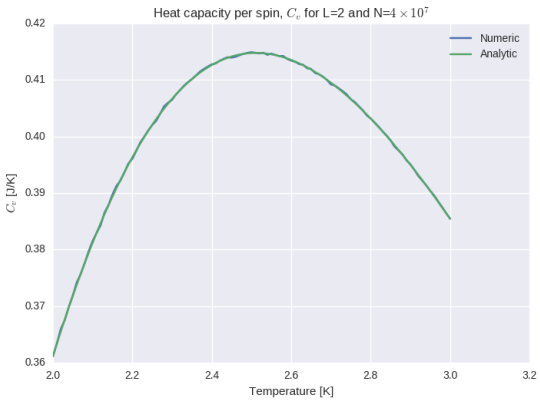
(d) $N = 4 \times 10^7$

Figure 2: Comparison of the numeric and analytic solution of heat capacity and mean energy per spin for the 2×2 lattice. We plot multiple Monte Carlo steps, N, to find where the correspondence becomes good.

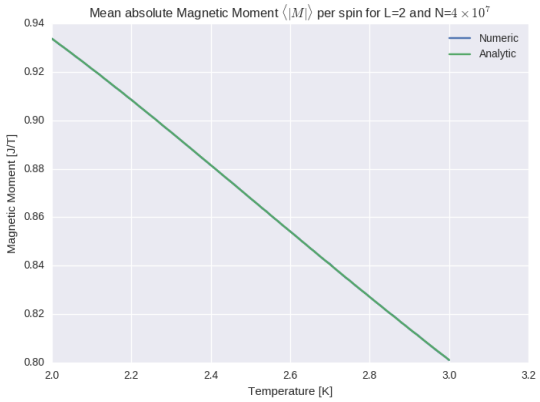
Looking at the plots, we choose $N = 4 \times 10^7$ as the required N, as this gives an excellent correspondence between the analytic and the numeric solution. The plots of our thermodynamic quantities in the 2×2 lattice for this value of N are shown below:



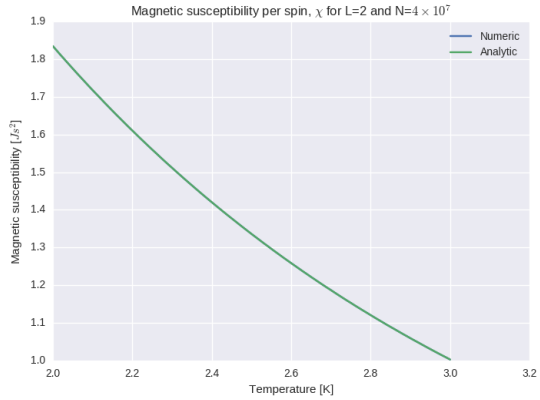
(a) Mean energy per spin



(b) Heat capacity per spin



(c) Mean absolute magnetic moment per spin



(d) Magnetic susceptibility per spin

Figure 3: Comparison of the numeric and analytic solution for the mean energy per spin, heat capacity per spin, mean absolute magnetic moment per spin and magnetic susceptibility of the 2×2 lattice.

4.2 Results from our investigation into the equilibration time

4.3 Results from our investigation into the probability distribution

4.4 Results from our investigation of phase transitions

5 Discussion

5.1 The 2 × 2 lattice

Figure 2 compares the analytic and numeric solution for a different number of Monte Carlo cycles. As expected, our numeric solution approaches the analytic solution with increasing N.

5.2 Results from paralellizing our code

6 Conclusion