Ising Model Project

Wilson Stewart

Dec 22, 2016

1 Introduction

In statistical mechanics, most of the known problems cannot be explicitly solved. Statistical mechanics provides a guideline for these problems, but there is no generic formula that can describe all of the different statistical systems.

Since it is not possible to solve these problems directly we can resort to simulation methods. In this project we used the Monte Carlo method, which generates physical states of a given system with some probability distribution determined by a random number generator. The Monte Carlo method is one of the most popular simulation models in all of physics (and elsewhere) because it can be applied to many different scenarios: discrete and continuous, system either in equilibrium or not in equilibrium, QFT, etc. Here, we use the Monte Carlo method to look at the 2-dimensional Ising model. We do this by using the Metropolis algorithm.

1.1 Ising model

The Ising model, named after Ernst Ising, and it is most generally a mathematical model that has a discrete number of variables that represent magnetic dipole moments. These spins are represented to be in either the +1 state or the -1 state. Then these spins are set up on a lattice where each spin can interact with its neighboring spin. The importance behind this simple model is that it illuminates phase transitions, as well as many other interesting phenomena. In this particular project we are insterested the Ising model on a 2-dimensional square lattice. The corresponding Hamiltonian for the Ising model is:

$$\mathcal{H} = -\sum_{\langle ii \rangle} J s_i s_j - h \sum_i s_i \tag{1}$$

In the Hamiltonian the first sum over i and j is over the pairs of nearest neighbors spins, and J is a determination of how much this sum is weighted in the Hamiltonian (also known as the coupling between neighbors where J is also known as the interaction constant). If I have two neighboring spins then I have $-J$ if the spins are parallel $(\pm 1)^2$ and $+J$ if the spins are antiparallel -1×1. Therefor, if $J > 0$ then the model will favor parallel spins (lower E) and if $J < 0$ then the model will favor antiparallel spins.

1.2 Metropolis algorithm

The Metropolis algorithim, named after Nicholas Metropolis, was devoloped originally for the case of symmetrical proposal distributions, and it has since been extended to a much broader range of topics. Most generally, the Metropolis algorithim is a Markov chain Monte Carlo method that examines a probability distribution and from this it obtains a sequence of random samples. This method is implemented primarily when direct measurements are difficult.

A difficult problem commonly encountered in the Monte Carlo method is generating a set of states that correspond to the Boltzmann distribution. In order to get a set of states that follow the Boltzmann distribution the Metropolis algorithim takes a probability distribution $P(x)$ where you can determine a function $f(x)$ that is proportional to the density of the probability distribution $P(x)$. This method first generates

a sequence of sample values, and the way it does this is so that as more sample values are produced the artificial distribution created by the sequence gets closer to the desired distribution of $P(x)$. One nice factor about this process is that it only depends on the current sample value (i.e. it does not have any memory of the earlier states in its process). Eventually the process will produce a set of states that will appear generally consistent with probabilities given by the Boltzmann distribution.

The Metropolis algorithm is a useful method for achieving equilibrium because first off we know that the Markov chain satisfies detailed balance,

$$\omega_{ij}\delta p_i^* = \omega_{ij}\delta p_j^* \tag{2}$$

(here ω_{ij} is the transition rate from state i to j) which states that the probability flux from one state to another balances the probability flux back. Also, Markov chains are ergodic, where I can get from one state to every other state in a finite number of steps. Now let's say I have a transiition matrix T. Because I know the Markov chain satisfies the detailed balance then T must have a complete set of eigenvectors $\vec{a}$ with eigenvalue 1. I also have a stationary distribution $\vec{a}*$, and all the other eigenvalues are less than 1 in magnitude ($|\lambda|< 1$). Now I can decompose the initial condition $\vec{a}(0) = b_1\vec{a}^* + \sum_{|\lambda|<1} b_\lambda \vec{a}^\lambda$, this leads to:

$$\vec{a}(n) = P \cdot \vec{a}(n-1) = P^n \cdot \vec{a}(0) = b_1\vec{a}^* + \sum_{|\lambda|<1} b_\lambda \lambda^n \vec{a}^\lambda \tag{3}$$

This finite sum decays to zero which means the density must converge to $b_1\vec{a}^*$. This means that the system converges to equilibrium as our $n \to \infty$. If I have all of these conditions then equilibrium will indeed be met. [1]

2 Methods

2.1 Measurement

The first step to perform the measurements required in this project make sure that you create a random list for your lattice (for mine I chose $\pm$). Then you want to perform the Monte Carlo method to allow the configuration to reach an equilibrium. After the state has reached equilibrium (in my case this was around 500 iterations which I will discuss in more detail later) begin to calculate the energy of the configuration by the Hamiltonian seen:

$$\mathcal{H} = -\sum_{\langle ii \rangle} J s_i s_j - h \sum_i s_i \tag{4}$$

After each calculation of the energy at each iteration step I added the "new" energy to the previously calculated energy and then I divided this value by the number of lattice points (N^2). Similarly, at each iteration step I calculated the magnetization:

$$M = \sum_i \sigma_i \tag{5}$$

This equation simply says that I added all of the spins up in the configuration at each step. Again, after each calculation of magnetization I added the "new" magnetization to the previously calculated magnetization and then I divided this value by the number of lattice points (N^2). For both the energy and magnetization I calculated them for a range of temperatures from $T = 0.1$ to $T = 5$.

Also, I was tasked with calculating the specific heat and the magnetic susceptibility of the configuration. Since these are calculated from the energy and magnetization respectively, I needed to calculate both of

these after I had determined the lists for energy and magnetization over the specified range of temperatures. I know generically the specific heat C_v is found via:

$$C_v = \frac{\partial E}{\partial T} \tag{6}$$

and magnetic susceptibility $\mathcal{X}$ is found via:

$$\mathcal{X} = \frac{\partial m}{\partial h} \tag{7}$$

Since I don't have functional forms of energy or magnetization I cannot use the above equations. However, I can perform numerical differentiation, where I chose the central difference method:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} \tag{8}$$

This method allowed me to easily find both C_v and $\mathcal{X}$ from the energy and magnetization data I already calculated. Note that C_v and $\mathcal{X}$ both run over the same range of temperatures as energy and magnetization.

2.2 Determining Equilibrium

Since I start off in a random spin state I need to perform the Monte Carlo method in order to achieve equilibrium. In order to determine how many iterations I needed to perform in order to get to equilibrium I examined the energy as a function of time (iterations)with the energy set to $T = 2$.

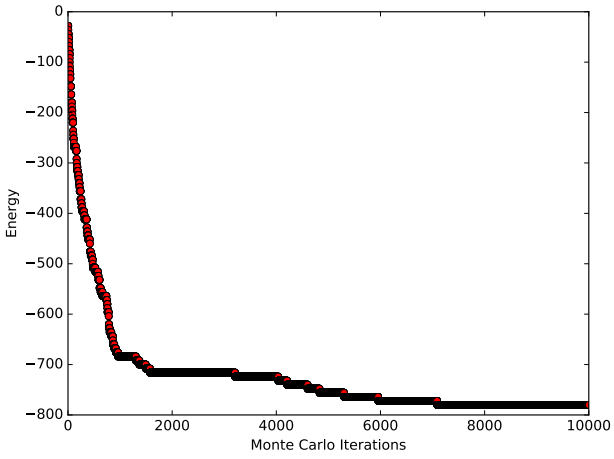

Figure 1: Energy versus Monte Carlo Steps for 15×15 lattice

As you can see in the plot, after about 5000 iterations the energy approximately stabilizes to a minimized energy state (equilibrium).

3 Algorithm

The main steps we need to take in order to solve for the properties of the Ising model are: pick a random spin (chose between -1 or $+1$), calculate the energy it costs to flip the spin ΔE, if $\Delta E < 0$ then flip this spin, if $\Delta E > 0$ flip the spin according to the probability $e^{-\beta\Delta}$. Here is a basic algorithm for the code:

Algorithm 1 main

```
define initial variables such as size of lattice and number of Monte Carlo Steps
creat a lattice of a random distribution of −1 and +1
for number of steps do
  examine a random spot on the lattice; config[a, b]
  try to change the configuration of the randomly picked state
  compute the cost of energy change ΔE
  if ΔE ≤ 0 then
    replace the state config[a, b] by -config[a, b]
  end if
  if ΔE > 0 and a random number less than one < e^{−ΔEβ} then
    replace the state config[a, b] by -config[a, b]
  end if
end for
```

4 Results

As seen in section 2.2 on determining the equilibrium where the equilibrium was achieved after 5000 Monte Carlo iterations. These plots seen below were run for 5000 Monte Carlo iterations. Also, note that if you set the dimension of your lattice above 30 (i.e. $N > 30$) then the code will run for an incredibly long time, therefore it is suggested to run this code for $N \leq 30$. Below you can see the magnetization as a function of temperature in the absence of an external field for temperatures ranging from $T = 0.1$ to $T = 5$:

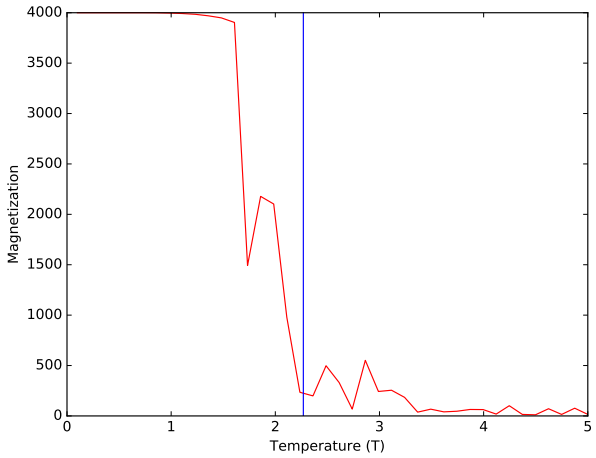

Figure 2: Magnetization versus Temperature for 5×5 lattice

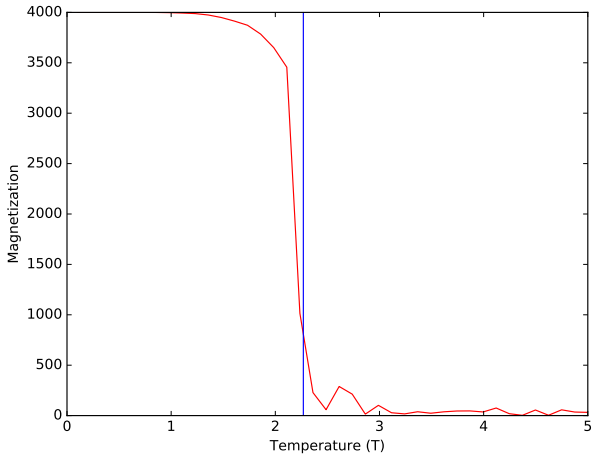

Figure 3: Magnetization versus Temperature for 10×10 lattice

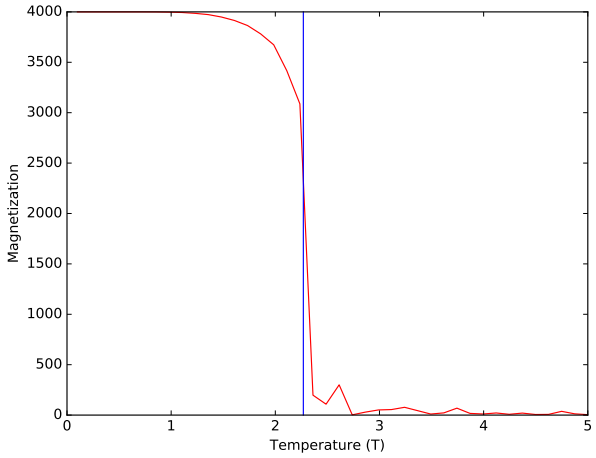

Figure 4: Magnetization versus Temperature for 15×15 lattice

In these plots you see a steep drop-off at around $T \approx 2.3$ which makes sense because the exact value of T_c (the critical temperature was calculated via:

$$\frac{k_B T_c}{J} = \frac{2}{ln(1+\sqrt{2}} \approx 2.269 \tag{9}$$

This $T_c = 2.269$ is denoted on the graphs as the vertical line.

Specific heat as a function of temperature, in the absence of an external field:

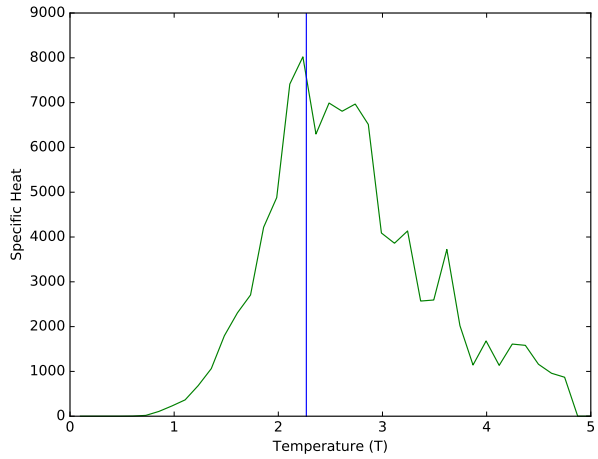

Figure 5: Specific Heat versus Temperature for a 5 lattice

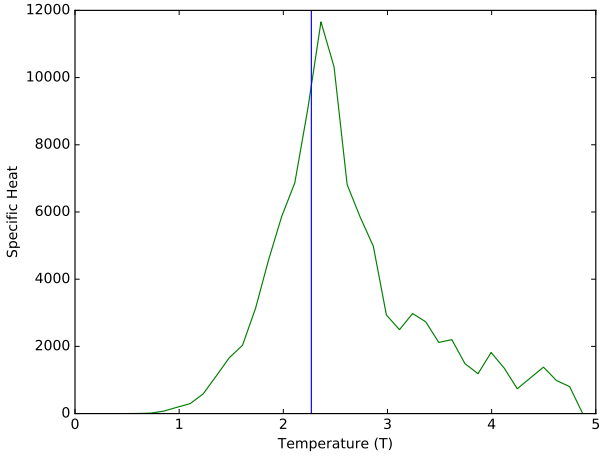

Figure 6: Specific Heat versus Temperature for a 10 lattice

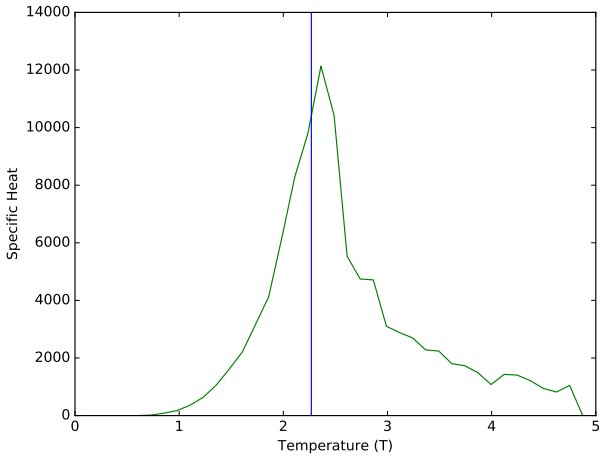

Figure 7: Specific Heat versus Temperature for a 15 lattice

As you can see in the above specific heat plots, as I increase the lattice size the peak at the critical temperature gets much steeper. This is concurrent with what we expected, and it appears that again, the critical temperature reaches a value of around $T_c = 2.3$, which again matches really well with the known value $T_c = 2.269$. This value is denoted on the graph again as a straight line.

Also, here are the plots of the magnetic susceptibility as a function of temperature, in the absence of an external field:

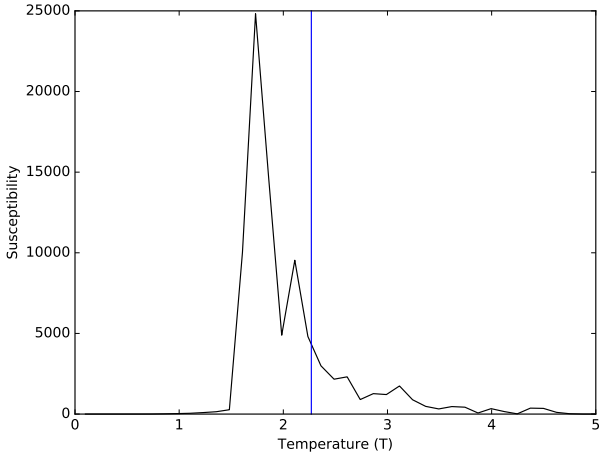

Figure 8: Magnetic Susceptibility versus Temperature for a 5×5 lattice

As can be seen in all of these plots, the trend becomes much more pronounced if the lattice size is increased (i.e. you have more data points). If you continue to increase the lattice size your trends will become even more pronounce, but this behavior would not be easily noticeable because the plots already shown the dramatic peaks predicted by the model. Therefore, a 15×15 lattice is sufficient.

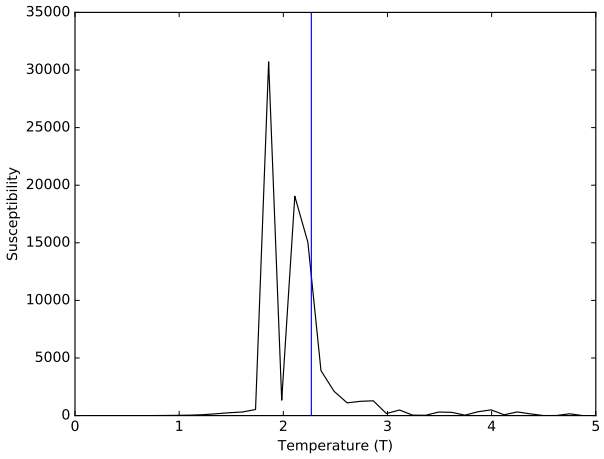

Figure 9: Magnetic Susceptibility versus Temperature for a 10×10 lattice

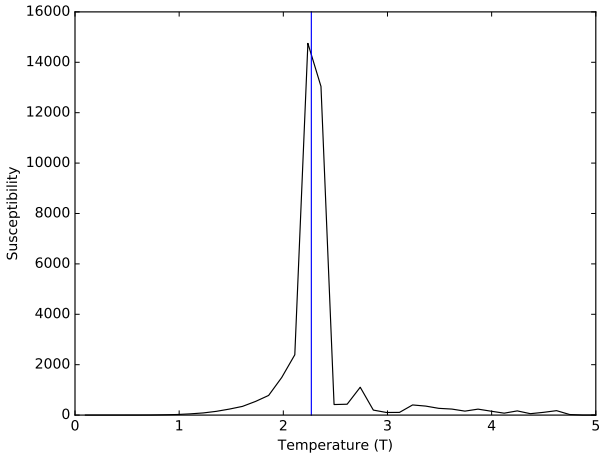

Figure 10: Magnetic Susceptibility versus Temperature for a 15×15 lattice

Finally, below is a phase diagram plot of the magnetization on the H, T plane for H ranging between -0.5 and $+0.5$, and T ranging between 0.1 and 5 (again 5000 Monte Carlo iterations):

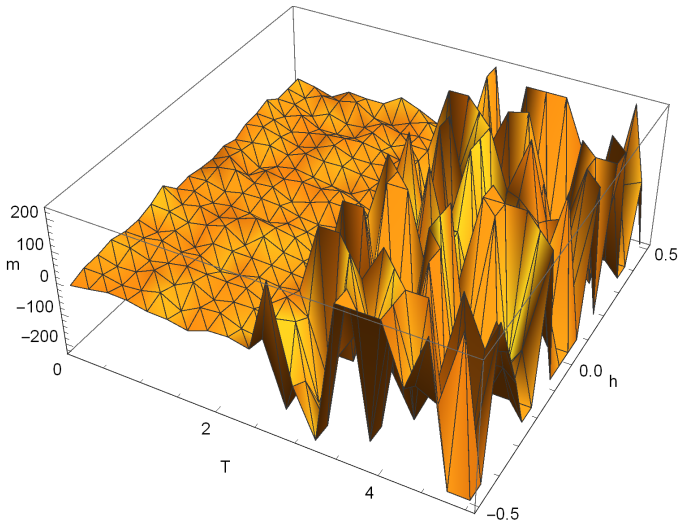

Figure 11: Phase Diagram in the H, T plane on a 15×15 lattice

Unfortunately, my phase diagram did not come out as I thought it would. I think there should appear smooth transitions between each region as opposed to my random-jagged-nature.