Phase Transitions and the Ising Model of Ferromagnetism

Nicolas Dronchi
Thanks given to my CMSE 202 group: Joseph Slivka, Casey Chartier, and Zhiyang Yu Michigan State University - Computational Physics - PHY480
Dronchin@msu.edu
Git Hub link: https://github.com/dronchin/Project4
(Dated: May 4, 2018)

The goal of this project is to study the Ising model. This is a model that has many far reaching applications but in this case the Ising model will be used to phase transitions as well as taking a peak into statistical mechanics. The analytical solutions to the $2x2$ case of the Ising model confirm that the algorithm that we use is correct. This allowed moving on to larger lattice sizes where the emerging properties of energy and net magnetism can be measured. Statistical mechanics helps with finding the derivative in terms of temperature to find the next two properties of heat capacity and susceptibility. With the largest lattice size used, $80x80$, you where able to estimate the critical temperature at which a phase transition will occur. This estimation can get very close to anaylitical critial temperature of $T_c = 2.269$. The distribution of energies was also looked at for different temperatures to look the relation between variance and the expectation value.

Usage: Ising model of ferromagnetism, phase changes, magnetic spin, critical temperature

I. INTRODUCTION

The Ising model is a widely used model in condensed matter and statistical physics. While it was initially developed to study ferromagnetism, it has been found to be useful in fields ranging from studying bird songs [1], to Econophysics, to sociology [2]. In condensed matter physics, the Ising model is useful for looking at phase transitions. In this case, the phase transition of spontaneous magnetization.

The Ising model was originally developed by Ernst Ising for the 1D case. In the 1D case, the solution lacks a phase transition. This lead to him to incorrectly concluding that the higher dimensions would also lack a phase transition behavior.

In 2D, the Ising model really starts to shine. The model is set up as a nxn lattice of spin states. At each lattice site, there is a spin value, σ, such that $\sigma \in \{1, -1\}$. Between any two spins at position (i, j) there is an interaction value of $J_{i,j}$. The energy of a configuration is given by the Hamiltonian [3].

$$H = -\sum_{\langle i,j \rangle} J_{i,j}\sigma_i\sigma_j + \mu\sum_j \sigma_j$$

$\langle i, j\rangle$ indicates you are looking at the nearest neighbors as you sum over spin pairs. The second sum is looking at the magnetic moment μ.

The probability for the lattice to be in a certain state is given by the boltzman distribution. With $\beta = k_bT$ and the partition function as Z, we get the following:

$$P = \frac{e^{-\beta H}}{Z}$$

When the system is at a low temperature, the probability of spins flipping to a lower energy is higher than the probability of the spin flipping to the higher energy state. This leads to a system where the spins align to minimize the energy and we get what we see in figure 1 a. When the system is at a high temperature, the probability of spins flipping approaches 1 and we get a system with random spins which is what we see in figure 1 b.

Currently, only the $2x2$ lattice has analytical solutions to the four quantities that are typically measured. Developed in 1944 by Onsager, analytical solutions for the expected value of energy, magnetization, heat capacity, and susceptibility can be found using statistical mechanics.

The critical temperature of the system is the temperature in which we get a phase transition, or in the case of the Ising model, where you get spontaneous magnetization. Many of the physical quantities like

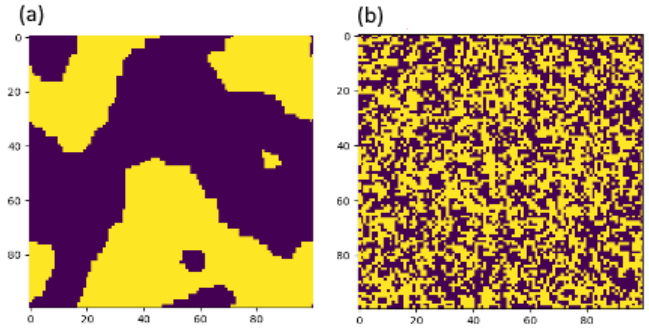

FIG. 1. (a) Lattice at low temperature (b) lattice at high temperature

energy, magnetization, heat capacity, and susceptibility can be described by power rules.

Because we work with finite sized lattices, size L, a finite sized scaling relation connects the size of the lattice to the results of the infinite lattice [4]. Here we have:

$$T_c(L) - T_c(\text{inf}) = aL^{-1/v}$$

Using this relation it is possible to find the analytical solution to the critical temperature of the system when $v = 1$ [5].

$$kT_c/J = \frac{2}{\ln 1 + \sqrt{2}} \approx 2.269$$

In this project, I go on to discus the methods used to calculate and run the Ising model. Then we take a quick technical look at the code and tests. Finally, I review the results and conclusions of the system in terms of the critical point and energy distributions.

II. METHODS

The first item to take care of is setting up the lattice of spin states. I chose to randomly select the starting spin state, either +1 or -1, for each spot in the lattice. Another strategy would have been to start out with a lattice full of one spin state. This requires more cycles near the critical point of our phase transition. Another way we set up the lattice is using periodic boundary conditions. We have to imagine the grid's left and right sides are connected as well as the top and bottom are connected in torus like shape. We take care of this using modulus math when referencing the nearest neighbors.

The first way to describe the algorithm is that it is a Markov chain Monte Carlo. When we set up our Ising model, we loop over changes in the lattice. The current state of our lattice depend on the state of our lattice before we make the change. If you pick a position to look at the spins, the current spins around determine whether there will be a change.

We also say that we are using a Monte Carlo Metropolis-Hastings algorithm. One way to do this is by running the system for a set number of steps doing the spin flip each time. At the end you would compare the final and initial energy. This uses much more computational time though because finding the energy is computationally expensive. The way we chose to do a spin flip is by looking at how the spin flip changes the energy. We look at the nearest neighbors to determine the change in energy. In this model, there are is only 5 possible values for dE: -8J,-4J, 0J, 4J, and 8J. These can be seen in figure 2.

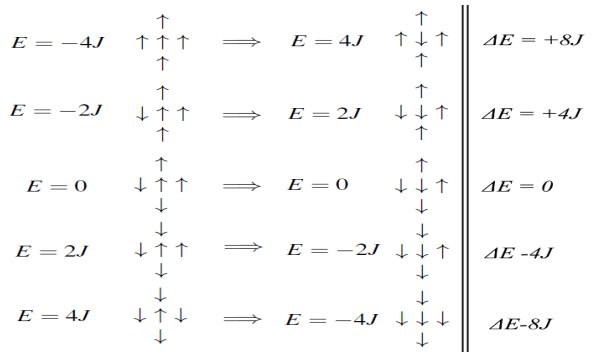

FIG. 2. Changes in energy with different neighboring spin conditions.

If the change in energy is negative, then the change is automatically accepted. If the change is positive in energy, we randomize a number on the range $[0, 1)$. This is compared to the boltzman distribution given by:

$$P = e^{-dE\beta}$$

This gives a chance to break out of local minimums in total energy in the search for the most probable state at the specific temperature given as an input.

After a spin flip is accepted, the energy and magnetization of the system changes. We can also change the instantaneous energy of the system by simply adding on the dE calculated before the spin flip. The change in the instantaneous net magnetization is 2x the current value of the spin in the position you are looking at. Looking at the change in these values saves a lot of computational power. Calculating the total energy and magnetization of the system is an $O(n^2)$ operation for each.

After running through a set number of random spin flips, we start to average the energy and magnetization. The expectation value of the energy squared and magnetization squared is also calculated at this point as they are needed for calculating the heat capacity and susceptibility. After running through a large amount of these Monte Carlo cycles, the values start converging. It is here that we normalize the values by how many loops we did while also calculating the heat capcity and susceptibility using the following [3]:

$$\langle C_v \rangle = \frac{1}{k_b T^2} \left[\langle E^2 \rangle - \langle E \rangle^2 \right]$$

$$\langle \chi \rangle = \frac{1}{k_b T} \left[\langle M^2 \rangle - \langle |M| \rangle^2 \right]$$

III. CODE AND TESTS

The code for the Ising model is relatively compact. This doesn't mean that it isn't powerful nor does it mean that is quick to run. A psudo-python code can be seen in listing 1 bellow.

Listing 1. Ising General Algorithm

```
for mcstep in range(mcc):
    for step in range(n**2):
        i,j = random_point()
        dE = energydifference(i,j)
        if rand() < exp(-dE/T):
            lattice[i,j] *= -1
            energy += dE
            mag += 2*lattice[i,j]
    E,E2,M,M2 = average()
Cv = (E2 - E**2)/T**2
Sus = (M2 - M**2)/T
```

In the program, I use numpy's pseudorandom number generator. This is the if statement in the code where I use rand(). Numpy uses the Mersenne Twister algorithm which has a period of $2^{19937}1$. This is much larger than the number of times we are calling it which is only on the order of 10^10.

The run time for this algorithm is dominated by the double nested loop as well as the calculation of 3 random numbers within each iteration. In the general algorithm, an exponential is called. This can be avoided as there are only 5 possible values for dE: -8J,-4J, 0J, 4J, and 8J. This means that you can pre-calculate exp(-dE/T) for each an only look up the value when needed.

Thermalization time is the largest factor in how long it takes to complete the program. The more cycles that you run the model for, the better averages you get but the more time it takes. For small grid sizes, around $10x10$ you can run for 10^3 - 10^4 cycles and get accurate results. For larger grid sizes like $100x100$, it takes upwards of 10^6 - 10^7 cycles before the numbers become useful. Figure 3 shows the relationship between the size of the matrix and how many cycles you need to run through to get the correct end values.

There are are a few ways that try to reduce the amount of cycles need for thermalization time. The first of which is hot starting the lattice. If you are scanning over temperatures that are very similar, then using the end product of the temperature before it as a starting point can reduce a lot of the time getting to the equilibrium. The issue with this method is that it requires bumping up the number of cycles when you are close to the critical point. Another method used for reducing the number of cycles is to only run the program until you are below a certain error in the calculated values. This ensures that you aren't running unnecessary cycles that don't contribute to the final answer.

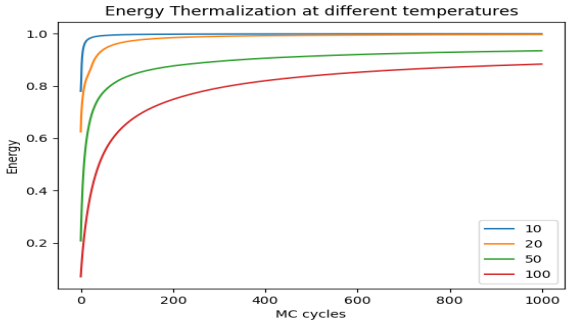

FIG. 3. Thermalization time with varying grid sizes. All gird sizes should approach an end value of 1

For tests on the Ising model, the program was run as a $2x2$ lattice. The end values that the program calculates should match with the analytical solutions described below [3]. Because the $2x2$ case matches the analytical solutions, it is assumed that the algorithm works for larger size matrices.

$$\langle E\rangle = \frac{-8J\sinh 8\beta J}{\cosh 8\beta J + 3}$$

$$\langle |M|\rangle = \frac{2e^{8\beta J}+4}{\cosh 8\beta J + 3}$$

$$\langle C_v\rangle = \frac{1}{k_b T^2}\Big[\frac{256J\cosh 8\beta J}{\cosh 8\beta J + 3} - \langle E\rangle^2\Big]$$

$$\langle \chi\rangle = \frac{1}{k_b T}\Big[\frac{8e^{8\beta J}+8}{\cosh 8\beta J + 3} - \langle |M|\rangle^2\Big]$$

IV. RESULTS

The phase change of the system should happen around the critical temperature of 2.269 according to the analytical solutions determined by Onsager. Because of this, we look for changes in the the four main quantities energy, magnetization, heat capacity, and susceptibility.

The first quantity measured was the energy of the system. It makes intuitive sense that at lower temperatures, the energy of the system is low, and at higher temperatures, the energy of the system is higher. In figure 4, the energy of the system is plotted on the

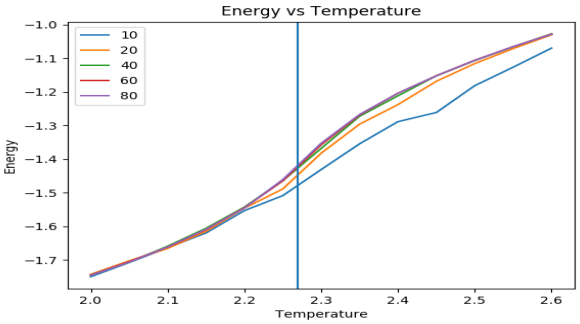

FIG. 4. Energy of the system between $T \in [2, 2.6]$. T_c is indicated by the blue vertical line

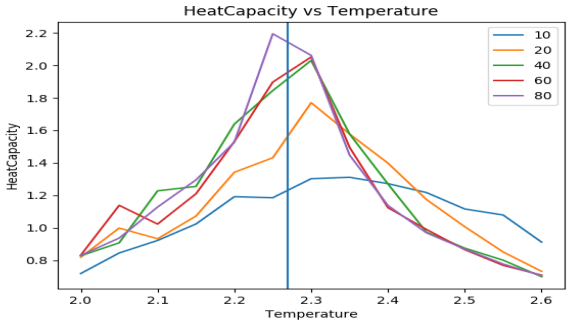

FIG. 6. Heat Capacity of the system spikes at the critical temperature.

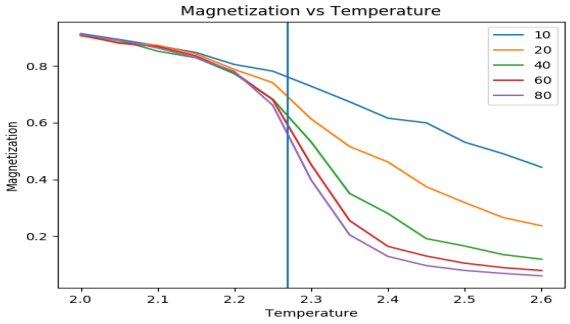

FIG. 5. Net magnetization of the system drops at the critical temperature.

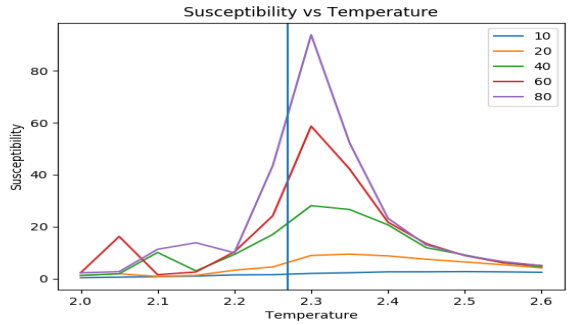

FIG. 7. Magnetic Susceptibility spikes at the critical temperature.

temperature interval $T \in [2, 2.6]$. The blue vertical line indicates the theoretical critical temperature of $T_c = 2.269$. With only the graph of the energy it would be hard to determine the critical temperature because the energy has a much more gradual change.

The second quantity was the net magnetization of the system which can be seen in figure 5. At low temperatures, the system spontaneously magnetizes making the net magnetization closer to 1. This happens when all the spins align in the same direction. At high temperatures, the system is past the critical temperature and it loses it's magnetization. This means the magnetization is closer to 0 because all of the spins cancel each other out.

The magnetization is the first look at increased accuracy as a result of increasing the lattice size. In the $80x80$ case, you can see the distinct drop around the critical temperature. Without knowing the analytical critical temperature, the guess of where the critical temperature would be about 2.3.

The last two physical quantities are heat capacity and magnetic susceptibility. The heat capacity is the first derivative of the energy in terms of temperature while the susceptibility is the first derivative of the magnetization in terms of temperature. It is in these two quantities that the critical temperature becomes apparent. They both hover closer to zero when not around the critical temperature but spike when right over. These quantities also greatly improve when the lattice size is increased. When the lattice size is increased, the peaks slowly shift to the left towards the critical temperature which matches the description given in the intro.

Finally, a look at the energy distributions at different temperatures can be found in figure 8. When the temperature is low, the average energy drops to around $-800J$. When the temperature is high, the average energy goes up to around 0. But it is the average for a reason. The instantaneous energy of the system will fluctuate around where the expected value is.

This all relates back to the Boltzman distribution. As the limit of T approaches 0, the expectation value for the energy becomes -800 in this case as, that is the lowest possible energy. As the limit of T approaches inf, the expectation value for the energy approaches 0.

Finally, it's also important that you can see the variance in the distributions. Ideally, the variance should follow distribution similar to the heat capacity and susceptibility graphs. The high and low temperatures have the lowest variance while the temperature near the

critical point in green has the greatest variance in the the energy distribution.

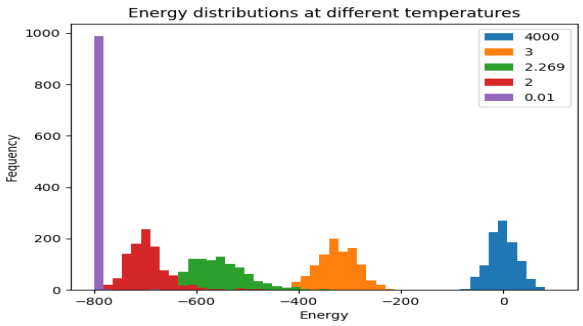

FIG. 8. Energy distribution at different temperatures of the system.

V. CONCLUSION

The first thing to look at is the accuracy of the critical temperature from the graphs of the four measurable quantities. With the largest matrix size of $80x80$, I got the closest to the theoretical value of the critical temperature. This is obvious in the Heat capacity where because of the step in temperatures, the heat capacity peaks in front of the critical temperature. The susceptibility also makes it clear that the larger matrix size gives the closest answer to the theoretical critical temperature of 2.269.

The increasing accuracy as the lattice goes up also comes with the downfall of requiring a lot more computational time. Due to my lack of optimizations and using python, the $80x80$ took upwards of 24 hours to complete with the Monte Carlo cycles set to 500,000 for each experiment. Each experiment loops over 80^2 chances at a spin flip. For smaller matricies, this can be turned down to only 1000 Monte Carlo cycles. Because of this need for larger lattice sizes, there is a need for developing the code in a different language like c++, where substantial speed gains can be made compared to python. Features like parallel computing also makes this a perfect job for submitting jobs to computing centers or super computers. Finally there is room to optimize the code to cut down on the ammount of time that the system thermalizes. As mentioned earlier, you can hot start the system from the previous temperature and less cycles are needed.

The energy distribution is also something with a lot of ability to improve with more optimization or computing power. With more computations of the energy distribution, you could measure the variance or standard deviation at each temperature. Increasing the lattice size would also have an effect on the dominance of the $-800J$ energy level. There could be more of a finer structure in that region than what is shown in figure 8. As well as affecting the low temperature regions, there could be some light shed on the distribution's proximity to a net energy of zero.

One of the areas that you can continue the work on the Ising model is studying it in 3D. Again this would require some of the optimization's mentioned earlier but looking at a system of spin flips. Also there was the realization of phase changes as you moved from the 1D to 2D Ising model so there could be higher order phenomena that only appear in the 3D system.

Also mentioned in the intro, the work being done in other fields could expand. The Ising model could apply to many other fields and only requires one connection for there to be another use.

Finally, I believe there is an interesting connection between the Ising model and machine learning. In a system with a varying temperature there can be boundaries of where the phase change is occurring. With machine learning, these boundaries could be identified and the drawn. This could be a bridge that helps identify phase changes in other systems. This could also be useful for examining how large of a region that a phase transition is happening in.

VI. CITATIONS

[1] Burridge, James Kenney, Steven. (2016). Birdsong dialect patterns explained using magnetic domains. Physical Review E. 93. 10.1103/PhysRevE.93.062402.
[2] Stauffer, D., (2008) Social applications of two-dimensional Ising model. American Journal of Physics, Volume 76, Issue 4, pp. 470-473
[3] Hjorth, Morten (2015) Computational Physics Lecture Notes Fall 2015 can be found on github at https://github.com/CompPhysics/
[4] Hjorth, Morten (2018) Project4 description can be found on github at https://github.com/CompPhysics/ComputationalPhysicsMSU/tree/master/doc/Projects/2018/Project4/IsingModel/pdf
[5] Onsager, Lars (1944) Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition Phys. Rev. 10.1103/PhysRev.65.117