Monte Carlo Simulations of the Phase Transition in Magnets

Aijia Yang

Dr. Yuxuan Wang

Department of Physics

University of Florida

Abstract

The purpose of this research project is to simulate the phase transition in magnets with the Metropolis Algorithm in Monte Carlo from the one-dimensional perspective. Phase transition may occur when the temperature of the magnet changes. To understand the implementation of the Metropolis Algorithm, a side project "computation of π" is done prior to the magnet simulation. With MATLAB, the code for computation is developed. We used the equation of the circle to calculate the possibility of a random dot landing under the curve of the function of a circle and used Metropolis Algorithm to update the location of each dot. The summation of dots is the area under the curve, which is the integral of the function. Having the numerical value of the area allows us to calculate backward the numerical value of π. A similar code is implemented in the n-site Ising Model. Bar charts are generated to test the reliability of the code. Magnetization determines the ability of a non-magnetic object to retain its magnetism after being magnetized by a magnet. With the code, we can generate the magnetization of each configuration and get the average value for magnetization. In the one-dimensional Ising Model, objects are hard to become magnets. Magnetic susceptibility is a property that demonstrates how big the influence of a magnet on a non-magnetic object is. By graphing the magnetization versus temperature and magnetic susceptibility versus temperature, we can see that in the one-dimensional Ising Model, the magnetic susceptibility decreases as the temperature increases.

Monte Carlo Simulations of the Phase Transition in Magnets

1. Introduction

In this research paper, section 1 is the general background information and introduction to the project. After that, section 2 introduces the methodology, followed by the results in section 3. In section 4, we will discuss the meaning of the results and the data analysis. Then, section 5 is the conclusion of the research and future work.

1.1 Background

Magnets are objects with north and south poles on their ends. All atoms within the magnet function as small magnets, pointing in the same general direction as the magnet. All magnets have magnetism: when you put a compass near the magnet, you can see the direction of the magnetic field and determine if the object is magnetic. However, magnets can lose their magnetism during phase transitions. Phase transition is a term used to describe the change of form of an object when its physical property changes. For instance, water changes from liquid to solid at a temperature of 0 ℃. Likewise, magnets lose magnetism at a specific high temperature.

In the scientific world, researchers have not yet done lots of analysis on the phase transition in magnets by simulating their transitional patterns on computers. This project aims to use computational methods to generate the magnetic information of each moment in the magnet and determine how well the data follows the Boltzmann distribution function. By having a deeper understanding of the phase transition in magnets, we can know the use of each magnet and thus have a deeper understanding of the physical properties of magnets, contributing knowledge to the broader Physics society.

1.2 Ising Model

To model magnetism in a 1-d lattice structure, the Ising Model is typically the most representative. Ising Model is a lattice structure that represents the atoms within a magnet. In a lattice of d dimensions, each magnetic moment has a spin, also considered the direction in which pointing up is +1 and pointing down is -1 (Tong, 2017). When a magnet has magnetism, all magnetic moments prefer to point in the same direction, corresponding to the magnet's direction, thus forming a specific pattern. Otherwise, different magnetic directions can show a non-magnetic feature of the object. Among the magnetic moments, each moment interacts with its nearest neighbors. For 1-d lattice structures, the number of a magnetic moment's nearest neighbors is 2. The number of neighbors increases as the dimension increases (Kochma, Paszkiewicz, Wolski, 2013).

1.3 Mean Field Theory

If the magnets do not have magnetism, there are many possible patterns. The probability of the spin of each site is represented by the Probability Distribution Function $p[s_i] = \frac{e^{-\beta E[s_i]}}{Z}$, which follows the Boltzmann distribution (Tong, 2017). β is the reciprocal of temperature T and Z is the partition function proportional to the number of dimensions of the lattice structure (Tong, 2011). To test the probability of the spin change as the temperature changes, we can use the Metropolis algorithm in Monte Carlo Method to simulate a sequence of numbers. Mean Field Theory states that we can take the mean value of the spin of each configuration in the lattice structure and assign it to the variable m, the magnetization (Nguyen, Berg, 2012). By using the mean value, the calculation for the phase transition will be simpler.

1.4 Monte Carlo and Metropolis Algorithm

The Metropolis Algorithm is the best algorithm in Monte Carlo since it has the highest accuracy compared to all other algorithms. The algorithm is coded in MATLAB. Firstly, we generate a random number i in the range of (-1, 1). Then, we propose a change, another random number j, and compare the probability $p(i)$ and $p(j)$. The probability is calculated by plugging the number into the specific Probability Distribution Function. If $p(i) < p(j)$, we accept the change, meaning the current i is the value of j. Else if $p(i) > p(j)$, the probability of accepting the change is $\frac{p(j)}{p(i)}$. For example, if $p(i) = 0.4$ and $p(j) = 0.3$, the probability of changing i to j is 75%. Notably, $\frac{p(i \to j)}{p(j \to i)} = \frac{p(j)}{p(i)}$ if both i and j are changing at the same rate. The purpose of generating and accepting values is to get a sequence of numbers that follows the predicted distribution function without bias. By repeating the previous step, we can get a sequence of i, and the sequence will satisfy $p(x)$, the Boltzmann Distribution (Shankar, 2018).

After getting the sequence of i, we can put them back into the original lattice structure. According to the equation $\beta(B + Jqm) = \frac{1}{2} log(\frac{1+m}{1-m})$ when functions on both sides are equal to each other, the graph will demonstrate an interception, which is the time when the phase transition occurs (Tong, 2017).

Therefore, with the help of graphs and numerical simulations, we can observe the impact of temperatures on magnets and record the time when they lose their magnetism.

2. Methodology

2.1 What is Monte Carlo

When the lattice structure of the magnet has a dimension $d \geq 1$, it is impossible to calculate its thermal averages by hand. Thus, we use the help of the computer to do the calculation, and the method is called Monte Carlo. Monte Carlo is a numerical method that generates possible results of an unpredictable situation and is technically a multidimensional integral (Shankar, 2018). In this case, when a magnet loses its magnetism, the configurations of all magnet moments can have different values, forming various patterns as a whole.

2.2 Why do we use Metropolis Algorithm in Monte Carlo

Metropolis Algorithm is the most accurate method in Monte Carlo due to its massive random data collection and loops. For magnets, each configuration of the magnetic spin is proportional to the Boltzmann distribution function. As we generate spins for each moment and collect them in a matrix, we can get the arithmetic average of the weighted configurations and then predict the pattern of the magnet, which allows us to determine the temperature when phase transition occurs.

2.3 Metropolis Algorithm Computing π

To become familiar with the Metropolis technique before beginning the magnet simulation, we follow a similar procedure to utilize the technique to obtain the numerical value of π. In the world of science and math, π always relates to circles. We calculate the area of a circle with the formula $A = \pi r^2$ and the circumference $C = 2\pi r$. In this side project, we focus on using the area of a quarter circle to determine π. If you draw a curve on a coordinate system in the first quadrant with points (0,1) and (1,0) that all points on the curve are one unit distance away from the origin, the area under the curve surrounded by the x and y axis is a quarter circle. The equation of the quarter circle is represented by $f(x) = \sqrt{1 - x^2}$.

Therefore, the area is $\int_0^1 af(x)\,dx$. Imagine cutting the quarter circle vertically with a tiny width, and if we put random dots on the graph, the probability of the dot landing on each specific rectangle is determined by $f(x)$.

To achieve this goal, we use the Metropolis Algorithm. Metropolis Algorithm uses random numbers following the probability distribution to simulate a sequence of i. In MATLAB, we first create an empty matrix to store the sequence. Then we generate a random number i as the starting value of the sequence. By plugging in the value of i into function $f(x)$, we can get the probability value $p(i)$. According to the algorithm, we then generate another random number j, and obtain $p(j)$ in the same way. If the value of $p(i)$ is less than $p(j)$, we change the current i to the value of j and record the updated value of i in the matrix as a second entry. If $p(i)$ is greater than or equal to $p(j)$, the probability to change i is given by $\frac{p(j)}{p(i)}$. As we repeat the previous steps, we will get a sequence of numbers at the end of the loop.

As soon as we have a sequence of numbers, we can plot them into a histogram to display the distribution of each number. If we make Δx small enough, the graph will be very close to a quarter circle (see Figure 1).

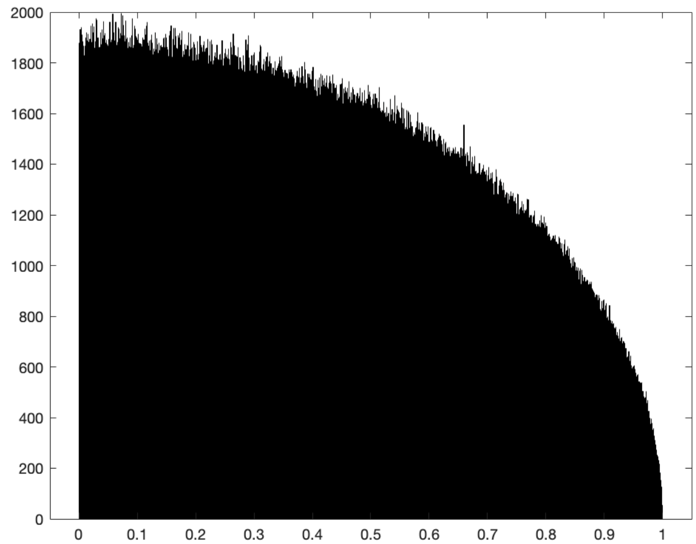

Figure 1. Histogram of the generated sequence

Since $p(x)$ is proportional to $f(x)$, which means $p(x) = a\sqrt{1 - x^2}$, we have to get the value of a so that we can get the approximation of π. Thus, by dividing $p(x)$ by $\sqrt{1 - x^2}$, we can get the value of a as a function of x (see Figure 2).

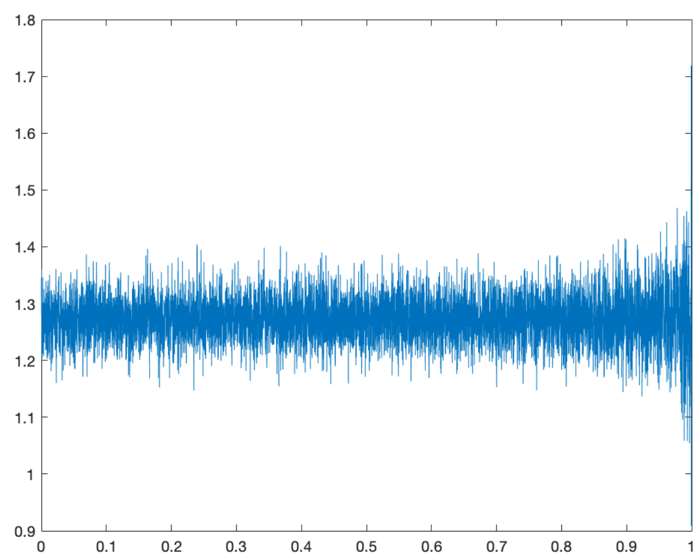

Figure 2. Value of a as a function of x

As mentioned earlier, if the circle has a radius of 1, the area is π. $1 = \int_0^1 a\sqrt{1 - x^2}dx$. And $\frac{1}{a}$ represents the area of the quarter circle. If we multiply the area of the quarter circle by 4, we get the area of the entire circle in which the radius is 1. That being said, the final answer is the approximation of π.

Eventually, the result of the approximation is very accurate (see Figure 3), and we have a deep understanding of how to apply the Metropolis Algorithm to the magnet simulation.

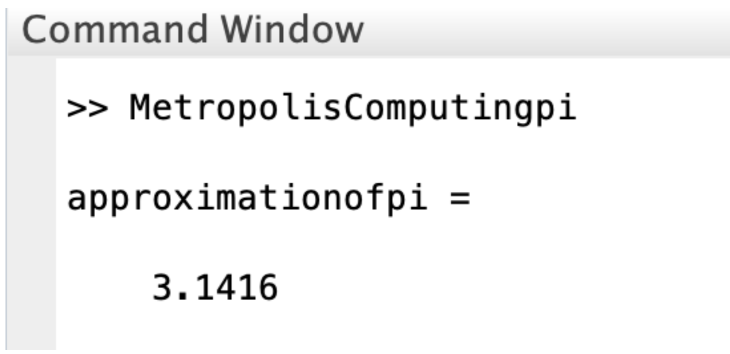

Figure 3. Final result of the approximation

3. Results

In an Ising Model, the function of the probability of occurrence of each pattern before normalization is represented by $p(i) = e^{(-\beta)*(-J)*\sum_{j} ij}$. In this case, if the Ising model has three sites, $\sum_{i} ij = s_1 * s2 + s_2 * s_3 + s_1 * s_3$. When $\beta = 1$ and $J = 1$, the probability of occurrence of the configurations [-1 -1 -1] and [1 1 1] is e^{3}, while all other cases– [-1 -1 1], [-1 1 -1], [-1 1 1], [1 -1 -1], [1 -1 1], and [1 1 -1]– has a probability of e^{-1}. The partition function Z is the sum of all possible probabilities of occurrence. Thus, $Z = 2 * e^{3} + 6 * e^{-1}$. The normalized probability of occurrence is $P(i) = \frac{p(i)}{Z}$, which is the theoretical value shown in the graph below (Figure 4). The experimental values are collected with the application of the Metropolis Algorithm.

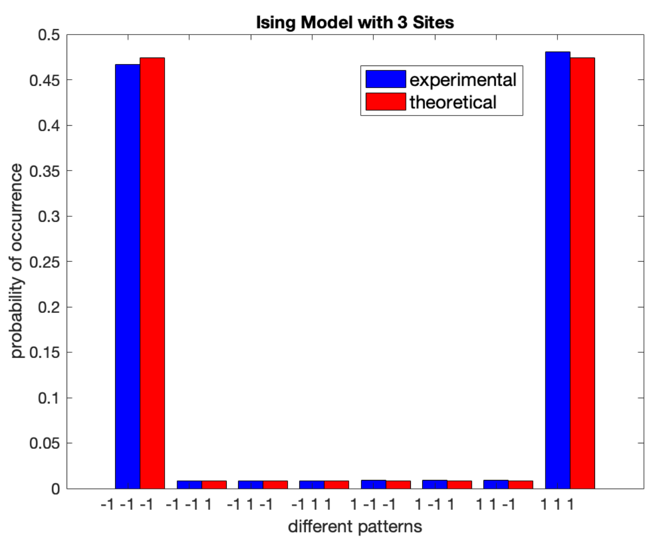

Figure 4. 3-Site Ising Model Probability of Occurrence

To extend the 3-site Ising Model to a broader range, a new code is created that implements the same algorithm and logistics to calculate the probability distribution of the patterns in the n-site Ising Model. As shown in Figure 5, a graph of the 4-site Ising Model with experimental and theoretical data is presented.

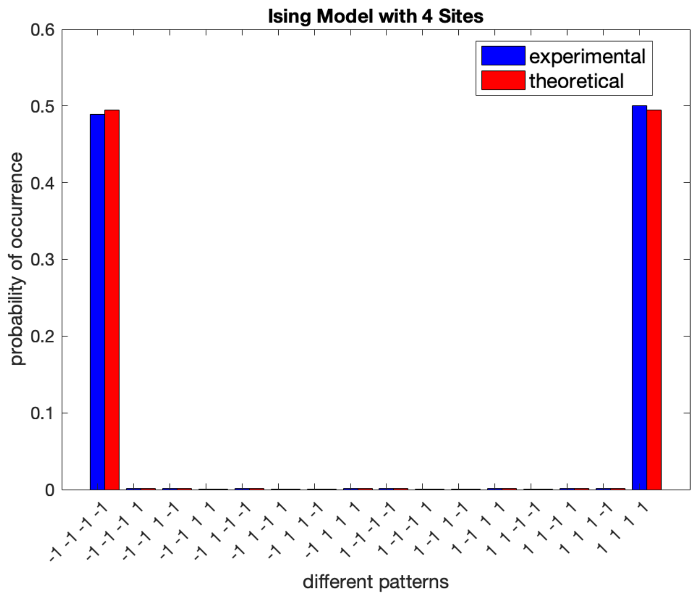

Figure 5. Ising Model with 4 Sites

With the code, the average magnetization (m) and magnetic susceptibility (χ) are calculated. In Figure 6, the relationship between magnetization and temperature is obtained.

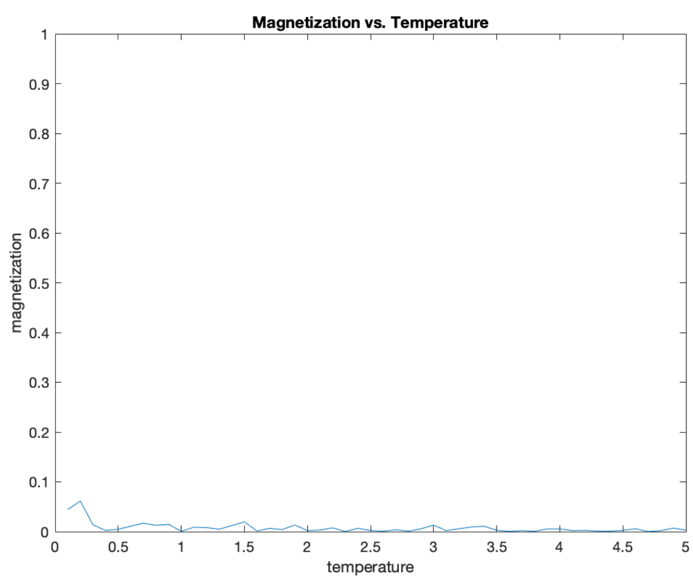

Figure 6. Magnetization vs. Temperature

It is also important to analyze the relationship between magnetic susceptibility and temperature. Magnetic susceptibility χ is calculated by $\chi = \ < M^2 > - \ (< M >)^2$, in which M is the magnetization. Magnetic susceptibility indicates the chance of a non-magnetic object being magnetized by a magnetic object. The higher the magnetic susceptibility is, the higher chance the non-magnetic object becomes and remains magnetized after the magnet leaves (Shankar, 2018).

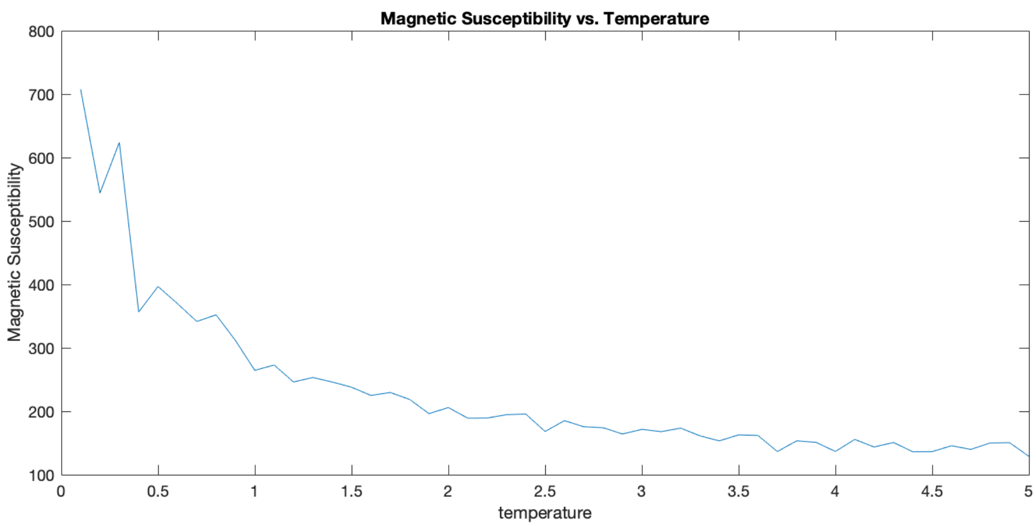

Figure 7. Magnetic Susceptibility vs. Temperature

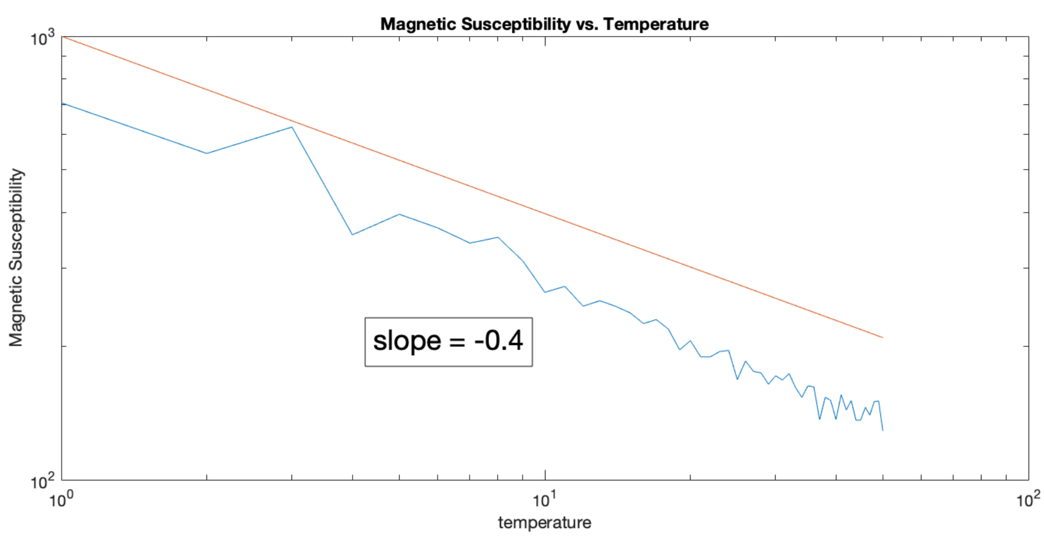

Figure 8. Magnetic Susceptibility vs. Temperature in the log scale

4. Discussions

In the previous section, two bar charts of the Ising Model are presented. From Figure 4, we can compare the experimental data to the theoretical data in a 3-site Ising Model. The experimental data is calculated by $\frac{number\ of\ occurrence}{total\ Monte\ Carlo\ steps}$. The sum of possibilities of each configuration is equal to 1 because the total possibility is 100%. The bar chart shows that objects tend to generate [-1 -1 -1] and [1 1 1] configurations both experimentally and theoretically. When each site is pointing in the same direction, the object is most likely a magnet. Similarly, in Figure 5 (Ising Model with 5 Sites), the object that has 5 sites is also very likely a magnet since [-1 -1 -1 -1] and [1 1 1 1] configurations have the highest probability of occurrence. However, although in these two cases the object is very likely to be a magnet, it only has three or four magnetic moments, which means from a microscopic perspective, the object is very tiny and not comparable to a real magnetic object. But since the experimental data and the theoretical data are very similar to each other and also very comparable, the code of the Metropolis Algorithm is implemented correctly.

As shown in Figure 6, the values of magnetization of an object from 1 to 5 degrees are very small and are almost zero. If an object is a magnet, the magnetization should be close to 1, which is the maximum magnetization in this case. Thus, the graph indicates that the object is hard to become a magnet in the 1-dimensional situation after being magnetized by induction. According to the Mean Field Theory, before a specific critical temperature, the magnetization should be relatively high and there should be a huge drop at the critical temperature (Zhang, 1992). From the graph, we can see that there is no drops or huge magnetization change. Therefore, the Mean Field Theory does not apply to the 1-dimensional Ising Model.

From the graph of magnetic susceptibility vs. temperature (see Figure 7), as the temperature increases, the magnetic susceptibility decreases, meaning that an object is getting harder to be influenced by an external magnetic field as the temperature increases. Thus, the object in 1-dimension is always paramagnetic (Pramanik, Banerjee, 2009). Figure 7 and Figure 8 are created from the same set of data, but Figure 8 is plotted on a log scale. With a log scale, we can calculate the slope of the curve, which is -0.4. In this case, the magnetization $m = \frac{1}{T^{0.4}}$. Compared to the Mean Field Theory, when the energy $J = 1$, $m = \frac{1}{T+2}$. The two magnetization functions are not the same, which means for 1-dimensional Ising Model, the Mean Field Theory fails.

All the simulations are based on the 1-dimensional Ising Model, so therefore, the results cannot represent all Ising Models with different dimensions, and future work is needed.

5. Conclusions and Future Work

Although the Mean Field Theory states that when the temperature is very low, an object can become a magnet, the theory does not apply to the 1-dimensional Ising Model. In the 1-dimensional Ising Model, objects are not very likely to become magnetized under all temperatures and are hard to retain their magnetism after being magnetized by induction. Since this research only focused on 1-dimension, future work will be conducted on 2 or more dimensional Ising Models.

References

Kochma ski, Paszkiewicz, T., & Wolski, S. (2013). Curie-Weiss magnet-a simple model of phase transition. European Journal of Physics, 34(6), 1555–1573. https://doi.org/10.1088/0143-0807/34/6/1555

Nguyen, H. C., & Berg, J. (2012). Mean-field theory for the inverse Ising problem at low temperatures. *Physical review letters*, *109*(5), 050602.

Pramanik, A. K., & Banerjee, A. (2009). Critical behavior at paramagnetic to ferromagnetic phase transition in Pr 0.5 Sr 0.5 MnO 3: A bulk magnetization study. *Physical Review B*, *79*(21), 214426.

Shankar, R. (2018). *Quantum Field Theory and Condensed Matter: An Introduction*. Cambridge University Press.

Tong, D. (Presenter). (2011). *Statistical Physics*. Lecture presented at University of Cambridge, Cambridge, United Kingdom.

Tong, D. (Presenter). (2017). *Statistical Field Theory*. Lecture presented at University of Cambridge, Cambridge, United Kingdom.

Zhang, J. (1992). The mean field theory in EM procedures for Markov random fields. *IEEE Transactions on signal processing*, *40*(10), 2570-2583.