Analysis of Electron Spin Evolution Using Monte Carlo Methods

Neema Rafizadeh*
Department of Physics and Astronomy,
University of Kansas, Lawrence, KS, 66045, USA

(Dated: May 8, 2023)

The main question being asked in this analysis is whether we can model computationally the magnetic parameters of a material, and then have those parameters match up with experimentally derived values. Magnetization, magnetic susceptibility, and specific heat are computed for a simulated spin lattice system and then compared to experimentally measured curves. The experimental curves were nearly identical to the computed ones, which both match the theoretically predicted curves from statistical mechanics. Thus, the answer to this question is a very strong 'maybe.' The magnetic parameters for a generic lattice can certainly be simulated, but trying to compute these curves for a specific atomic structure is quite involved and something that people probably devote years of study to.

I. INTRODUCTION

One of the emerging topics in the field of condensed matter physics today is known as spintronics, a topic which delves into the study of the spin of particles to process and store information. To give a brief introduction and analysis to this topic, we will investigate a foundation of statistical mechanical systems: magnetization.

Studying the magnetization state of a system can give us insight into how electron spins evolve in a crystal lattice and how that evolution affects the system's thermodynamic properties. This is a process that has been experimentally demonstrated many times in condensed matter physics and will be discussed in detail later in the Results and Analysis section.

II. HYPOTHESES OF MAGNETIZATION

Consider a system made of electrons that have the set spins $\{s_i\}$. The magnetization and susceptibility of that system are then,

$$M = \left\langle \sum_{i=1}^{N} s_i \right\rangle \quad \text{and} \quad \chi = \left.\frac{\partial M}{\partial H}\right|_T,$$

where N is the number of total spins in the system, M is the magnetization, χ is the magnetic susceptibility, and H is the external magnetic field that will be applied to the system.

For any material with collective spin, there are three possible magnetic properties it can have. The first is para-magnetism, which is the case when $\chi > 0$, and $M = 0$ when $H = 0$. In other words, it has no intrinsic magnetic field but one can be induced when an external field is applied to it. The next is ferro-magnetic, which is the case when $\chi > 0$, and $M \neq 0$ when $H = 0$. This means the material has an intrinsic magnetic field, whether there is an external field or not. The last is dia-magnetism, which is the case when $\chi < 0$, and $M = 0$ when $H = 0$. This is similar to the para-magnetic classification, but the induced field is anti-parallel to the external field instead of parallel, such as in the para-magnetic case. To summarize:

$$\begin{aligned} &\text{Paramagnetism :} && M = 0 \text{ when } H = 0 \text{ and } \chi > 0, \\ &\text{Ferromagnetism :} && M \neq 0 \text{ when } H = 0 \text{ and } \chi > 0, \\ &\text{Diamagnetism :} && M = 0 \text{ when } H = 0 \text{ and } \chi < 0. \end{aligned}$$

The dia-magnetic case is very rare and not widely studied, so we will stick to the para- and ferro-magnetic cases.

The magnetic state of a material depends largely on its temperature. Earth magnets, for example, are ferromagnetic at low temperatures but transition to a paramagnetic state at sufficiently high temperature. The temperature at which this transition occurs is known as the Curie temperature, or colloquially as the critical temperature, as seen in Fig. (1).

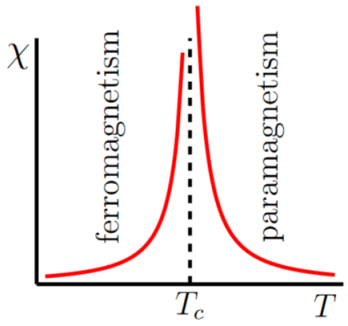

FIG. 1. A magnetic susceptibility (χ) v temperature graph that illustrates the Curie temperature, T_C, separating the ferromagnetic and paramagnetic states of a system.

As can be seen from the figure, the point at the Curie temperature itself is undefined and asymmetric. This is

* Email: rafizadehn@ku.edu

an order-disorder transition and thus discontinuous at the phase transition. The physics at this point is quite complex, but can be ignored for the purpose of this experiment and simulation.

Since the susceptibility is defined as the derivative of the magnetization with respect to the magnetic field, the magnetization is not discontinuous and, for a ferromagnet, has a curve of the form shown in Fig. (2).

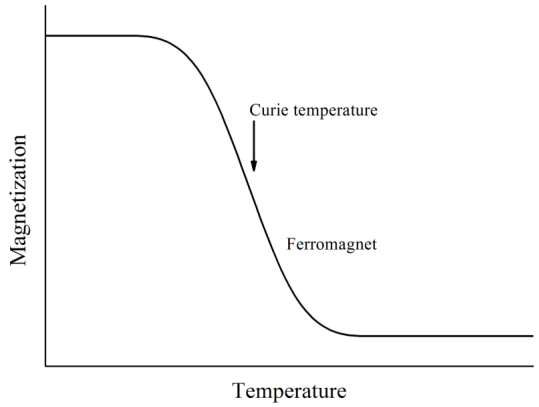

FIG. 2. A magnetization (M) v temperature graph that illustrates the change of the magnetization before and after the Curie temperature.

We are able to study these thermodynamic properties in materials by taking measurements and then fitting our measurements to specific models of lattice structures. For this experiment, we will consider each electron spin to interact with its nearest neighboring electron. This can be modeled with the Hamiltonian,

$$\mathcal{H} = -H\sum_{i=1}^{N} s_i - \sum_{i,j} J_{ij} s_i s_j,$$

where J_{ij} is the interaction term between electrons, known as spin-spin coupling. This value depends largely on the distance between the two spins in reality, but we will simply assign an arbitrary value to this for our own simulation.

III. EXPERIMENTAL DATA

A study performed by a group of condensed matter physicists at the State University of Maringa in Brazil investigates these properties for a crystal with a molecular formula of $(Zn_{1-x-y}Fe_xCo_y)O$. They studied the magnetic order of the material at various temperatures that allowed them to plot the magnetization and magnetic susceptibility of their material, which is what we are trying to simulate for this project.

As seen in Fig. (3), the magnetization curve is quite similar in structure to the idealized curve shown in Fig. (2). Since this effect is due to the interactions of the electrons in the lattice, simulating this should be relatively

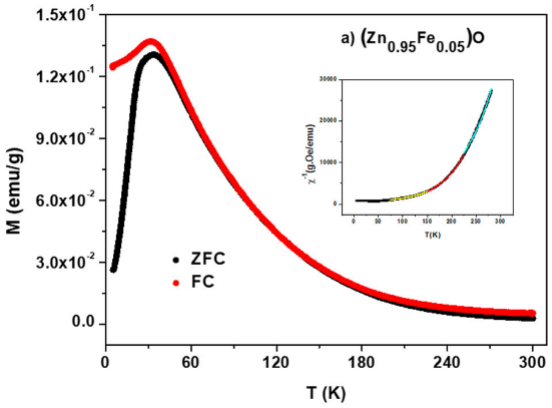

FIG. 3. Magnetization (M) v temperature curves for one particular $(Zn_{1-x-y}Fe_xCo_y)O$ sample. The red curve shows a Zero Field Cooling (ZFC) magnetization curve, while the black curve shows a Field Cooling (FC) magnetization curve.

straight forward depending on the interaction model we use.

The researchers also measure the magnetic susceptibility of this material and plotted the curve, shown in Fig. (4).

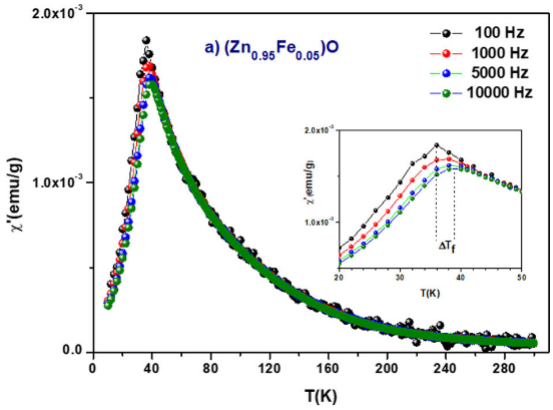

FIG. 4. Magnetic susceptibility (χ) v temperature curves for one particular $(Zn_{1-x-y}Fe_xCo_y)O$ sample taken at varying frequencies.

The experimentally determined curve in Fig. (4) differs slightly from the idealized curve in Fig. (1), simply due to the fact that the discontinuity is a product of our mathematical notation and derivations and can't exist fully in reality. Thus, what appears as the discontinuity in the idealized plot is simply a global maximum in the experimentally determined data. From this, we are able to find the Curie temperature for this material, which is our goal with our simulated data as well.

IV. CODE AND EXPERIMENTAL SIMULATION

The first part of the script uses a Monte Carlo method to calculate a 'configuration' of electron spin states that are used to then determine thermodynamic and magnetic properties from. This starts by first generating a uniform distribution, shown in Fig. (4), that is then used as input for the Monte Carlo simulation for calculating the electron spin configuration. The code for this section of the script is,

```
for i in range(N):
       for j in range(N):
              a = np.random.randint(0, N)
              b = np.random.randint(0, N)
              s =  config[a, b]
              nb = config[(a+1)%N,b] +
                   config[a,(b+1)%N] +
                   config[(a-1)%N,b] +
                   config[a,(b-1)%N]
              cost = 2*s*nb
              if cost < 0:
                  s *= -1
              elif rand() < np.exp(-cost*beta):
                  s *= -1
              config[a, b] = s
```

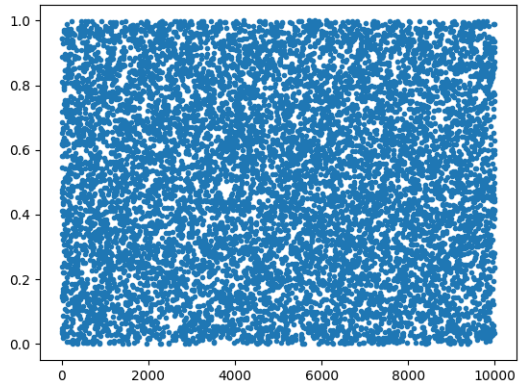

FIG. 5. Plot of values used in uniform distribution input.

Then, the energy and magnetization are calculated in similar fashion through use of the configuration output of the Monte Carlo simulation,

```
    for i in range(len(config)):
        for j in range(len(config)):
            S = config[i,j]
            nb = config[(i+1)%N, j] +
                 config[i,(j+1)%N] + config[(i-1)%N,
                 j] + config[i,(j-1)%N]
            energy += -nb*S

mag = np.sum(config)
```

It becomes easier to do this over multiple iterations if we define a function to do this as a loop (as this process has to enter a new configuration for each spin-state evolution). The function we define is written as,

```
class Ising():
    ## monte carlo moves
    def mcmove(self, config, N, beta):
        for i in range(N):
            for j in range(N):
                    a = np.random.randint(0, N)
                    b = np.random.randint(0, N)
                    s =  config[a, b]
                    nb = config[(a+1)%N,b] +
                         config[a,(b+1)%N] +
                         config[(a-1)%N,b] +
                         config[a,(b-1)%N]
                    cost = 2*s*nb
                    if cost < 0:
                        s *= -1
                    elif rand() < np.exp(-cost*beta):
                        s *= -1
                    config[a, b] = s
        return config

    def simulate(self):
        # initilizes the lattice simulation
        N, temp    = 64, .4
        config = 2*np.random.randint(2,
            size=(N,N))-1
        f = plt.figure(figsize=(15, 15), dpi=80);
        self.configPlot(f, config, 0, N, 1);

        # want to create values at different times
            to be able to
        # observe the evolution of the spins inside
            the lattice
        msrmnt = 1001
        for i in range(msrmnt):
            self.mcmove(config, N, 1.0/temp)
            if i == 1:      self.configPlot(f,
                config, i, N, 2);
            if i == 4:      self.configPlot(f,
                config, i, N, 3);
            if i == 32:     self.configPlot(f,
                config, i, N, 4);
            if i == 100:    self.configPlot(f,
                config, i, N, 5);
            if i == 1000:   self.configPlot(f,
                config, i, N, 6);

    def configPlot(self, f, config, i, N, n_):
        # actually creates the plots mentioned
            before
        X, Y = np.meshgrid(range(N), range(N))
        sp =  f.add_subplot(3, 3, n_ )
        plt.setp(sp.get_yticklabels(),
            visible=False)
        plt.setp(sp.get_xticklabels(),
            visible=False)
        plt.pcolormesh(X, Y, config,
            cmap=plt.cm.RdBu);
        plt.title('Time=%d'%i); plt.axis('tight')
```

```
plt.show()
```

This code was pieced together from a couple different sources, which are listed in the GitHub link in the Appendix. This section of the script will iterate the Monte carlo simulation over many different configuration of electron spin states and create a coherent evolution of electron spins that can be used to calculate magnetic properties. In other words, the uniform distribution that is entered as the input for the original iteration will evolve, and those evolutions are used for the next run through the loop. If we graph the evolution of the spin states throughout the loop iterations, we are able to plot them together in something like Fig. (6).

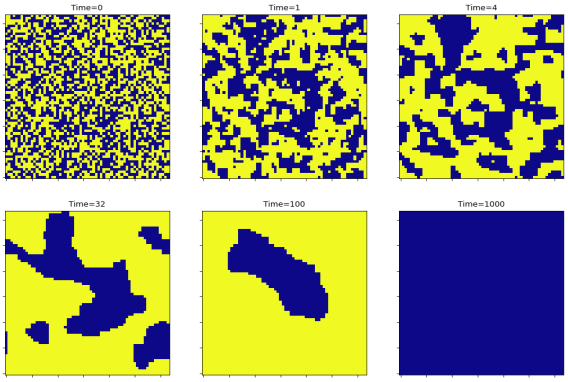

FIG. 6. Loop iteration evolution of electron spin states.

As can be seen in Fig. (6), the spins start out randomly distributed and tend toward lining up with one another in large time scales due to their interaction factors. This doesn't illustrate the calculation of the magnetic properties, but rather the evolution of spin that results in the magnetization itself.

V. RESULTS AND ANALYSIS

The first calculation we can check is the magnetization of the material as a function of the temperature the lattice is simulated, shown below in Fig. (7).

This plot is very similar to the idealized plot in Fig. (2) and the experimentally measured data shown in Fig. (3). As the simulation is being carried out through a Monte Carlo method, the error in each of the points is larger than with other methods. To reduce the error of the points and achieve a smoother curve, one would need to increase the accuracy and precision of the Monte Carlo simulation, which can be computationally taxing.

Next, we can compare the simulated specific heat capacity to see if it makes sense with the system we are studying. The plot for this is shown in Fig. (8).

Comparing Figs. (7) and (8), the transition between the two sections of the curves happen at the same arbitrary temperature value. This value is the Curie temperature of the system in study. This transition temperature

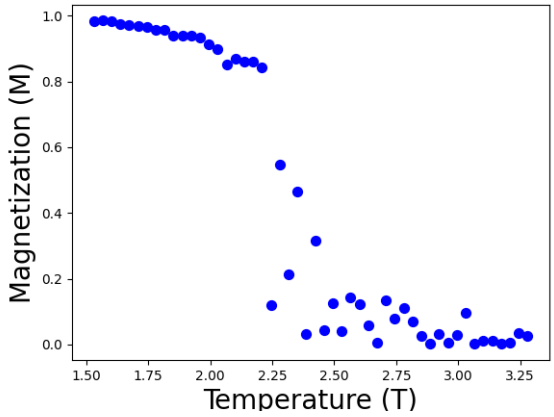

FIG. 7. Magnetization vs. temperature of simulated lattice. Temperature is in arbitrary units, not necessarily kelvin.

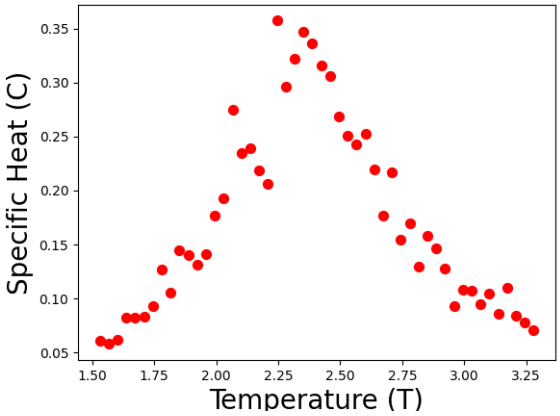

FIG. 8. Specific heat vs. temperature of simulated lattice. Temperature is in arbitrary units, not necessarily kelvin.

can also be seen clearly in the magnetic susceptibility plot below, in Fig. (9).

This plot does not match the idealized curve shown in Fig. (1) perfectly, as the simulation can not really account for the ferro-magnetic and para-magnetic cases at the same time. Another configuration of parameters would be able to recreate the former half of the plot shown in Fig. (1).

The plots shown here are mostly the same as the idealized plots and experimentally plotted data, which shows that the computational analysis for complex systems such as lattice spin evolution can be made quite easily with some particular approximations.

VI. CONCLUSION

The simulated properties of a lattice are nearly identical in shape to our theoretical calculations from statistical mechanics and the experimentally measured data

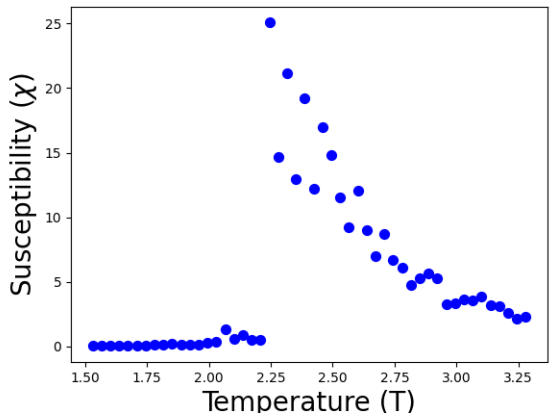

FIG. 9. Specific heat vs. temperature of simulated lattice. Temperature is in arbitrary units, not necessarily kelvin.

from a crystal lattice. The sources of uncertainly mainly come from uncertainty in knowing the temperature of the lattice itself.

To measure temperature, a nano-scale probe has to be connected to the lattice to monitor its heat. The problem with this is, the probe has a significant impact on the system on such a small scale. Because of this impact, there is non-negligible uncertainty in the temperature measure. This uncertainty most likely follows a Boltzmann distribution as the electrons function as a free gas in the lattice, rather than acting as a stochastic process.

The main question being asked in this analysis is whether we can model computationally the magnetic parameters of a material, and then have those parameters match up with experimentally derived values. The answer to this question is a very strong 'maybe.' The magnetic parameters for a generic lattice can certainly be simulated, but trying to compute these curves for a specific atomic structure is quite involved and something that people probably devote years of study to. I did this project in a week, so I unfortunately could not look into specifics regarding this. It is absolutely something I would be interested in doing in the future.

ACKNOWLEDGMENTS

I would like to thank Prof. Chris Rogan for his continued support in the course. I would also like to thank Daniel Murray who assisted me with his peer review.

Appendix A: Sources and Raw Data

All python scripts and data generated are contained in Neema Rafizadeh's GitHub repository for this project. Instructions for running the scripts and changing the inputs are listed on the README.md file listed in that repository.

The `MySort` and `Random` classes of functions can be located at Prof. Christopher Rogan's GitHub. More information regarding magnetization can be sourced from the Wikipedia article discussing the subject.