

Project 3: Monte Carlo Simulations

CHEN 4880

Jack Vogelgesang
4-11-2022

Problem 1: Monte Carlo Integration:

A. Accuracy of Monte Carlo Integration

- a) In order to plot the mean unsigned error as a function of the number of samples N the code seen in Figure 1 of Appendix A was used. This code was ran with increasing sample sizes starting from 10 and increasing to 1,000,000 samples. Each sample size was repeated ten times so that good data with as little error as possible was obtained. In Table 1 the results of the integral for each sample size can be seen. Also, in Table 1 the mean unsigned error of each sample size is given which was calculated by summing all of the integral calculations then subtracting the analytical solution times ten and finally dividing the resulting number by ten to get the mean unsigned error of that sample size. The analytical solution was provided and was found to be 21.333. An example of the mean unsigned error calculation can be seen below.

Table 1: Monte Carlo Approximation of Integral and Mean Unsigned Error Data

	MC Samples, N					
	10	100	1000	10000	100000	1000000
Trial #						
1	12.8	24.32	21.248	21.0688	21.42976	21.352256
2	19.2	19.2	21.248	21.216	21.34208	21.304512
3	19.2	24.32	22.72	21.4208	21.3632	21.357568
4	25.6	23.68	22.08	21.3248	21.29728	21.40896
5	32	22.4	19.84	21.3056	21.24288	21.284352
6	19.2	16	20.608	21.5296	21.40352	21.333312
7	25.6	23.68	21.76	21.7664	21.39008	21.376384
8	6.4	24.32	20.032	21.4272	21.35168	21.300224
9	32	19.84	21.952	21.4976	21.40672	21.351424
10	38.4	19.2	22.656	21.2608	21.2768	21.3632
MUE	1.707	0.363	0.0814	0.04876	0.0174	0.0102192

Calculation of Mean Unsigned Error for sample size of 100:

$$\frac{(24.32 + 19.2 + 24.32 + 23.68 + 22.4 + 16 + 23.68 + 24.32 + 19.84 + 19.2) - (21.333 * 10)}{10} = 0.363$$

Using the data from Table 1, a graph of the mean unsigned error vs sample size N was obtained and can be seen in Figure 1 below.

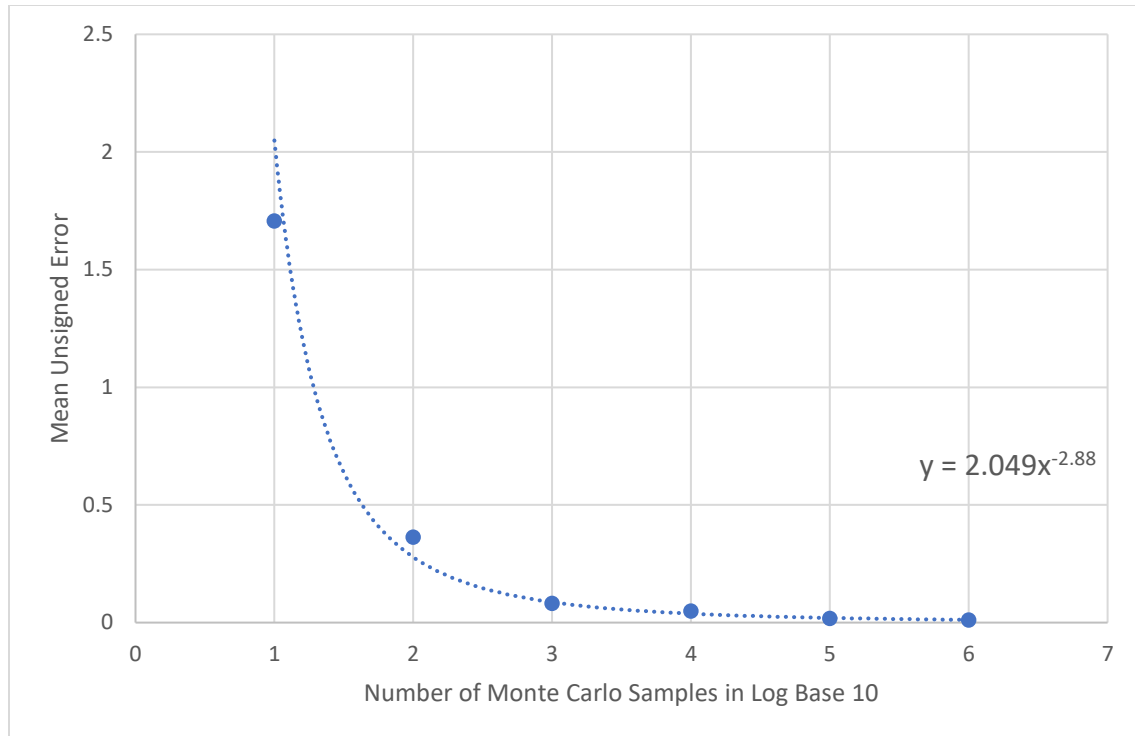


Figure 1: Mean Unsigned Error as a Function of Monte Carlo Steps, N (in Log Base 10)

- b) In order to see how the error improves as a function of increasing Monte Carlo steps we plotted the functions $1/N$, $1/N^2$, and $1/\sqrt{N}$ and obtained the following graph seen in Figure 2:

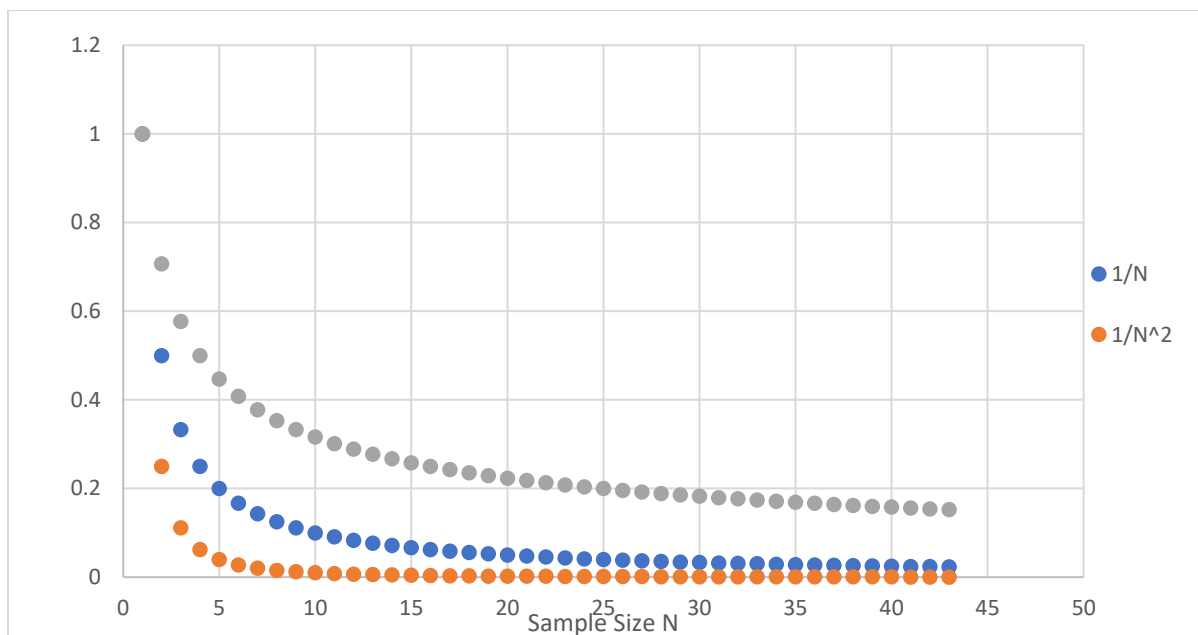


Figure 2: Graph of Functions for Sample Size N

Our graph obtained in part A and seen in Figure 1 seems to match closest with the $1/N^2$ function as Figure 1 approaches zero rapidly. $1/\sqrt{N}$ and $1/N$ lines do not converge fast enough. Also, a power trendline was plotted onto the graph in Figure 1 using Excel which gave the equation $y = 2.049x^{-2.88}$ which is closest to the $1/N^2$ line and confirms that the error improves as $1/N^2$.

B. Approximation of pi with Monte Carlo Integration

- a) In order to find a numerical approximation of pi the function $\sqrt{1-x^2}$ was integrated which can be seen in Figure 2 of Appendix A. This formula was used as it is a simple equation that gives us one quadrant of a circle which can then be multiplied by four to get our approximation of pi. Running the code with a sample size of 100,000 we obtained a value of 0.78344 which when multiplied by 4 gives us 3.13376.
- b) The bounding rectangle coordinates used were $[0, 1]$ in the x direction and $[0, 1]$ in the y direction which gives us an area of 1 unit².
- c) One hundred thousand MC samples would be required to calculate pi to the 5th decimal spot. Monte Carlo integration is an efficient method to approximate pi especially if we only need numerical certainty for the lower digit approximations. It is less efficient when we need a large number of digits of pi but can still be used.

Problem 2: Phase Transition of the 2D Ising Model

A. Ground State of the Ising Model

- a) The interaction strength term J is the interaction between two neighboring sites so if J is positive then a negative sign will be in front of the equation which causes a preference for both spins to point in the same direction and thus will eventually cause all spins to point one direction and thus cause ferromagnetism. If J is less than zero then the negative sign in front will vanish and in order to lower the energy it would be more beneficial for the system for σ_i and σ_j to have different signs and thus creates an anti-ferromagnetic state.
- b) The magnetic field parameter, h , allows us to control/favor either spin up or spin down as it directly affects σ_i by increasing or decreasing the magnetic field.
- c) In order to confirm our analysis, the code in the Problem 2 Jupyter notebook which can be seen in Figure 3 of appendix A was used. The number of sites per direction was set to 10, an initial temperature of 10 and final temperature 0.1 with 100 samples and 500 Monte Carlo steps were used for each simulation. Our first simulation was when $J = -1$ and $h = 0$ which gave us the results seen in Figure 3 below.

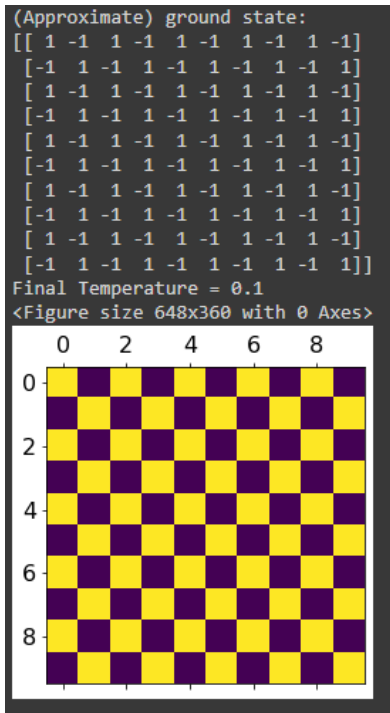


Figure 3: Ground State of Ising Model When $J = -1$ and $h=0$

This produced a clear anti-ferromagnetic state as spins are alternating between 1 and -1. Next, we simulated with parameters of $J = 1$ and $h=0$ which gave us the results seen in Figure 4.

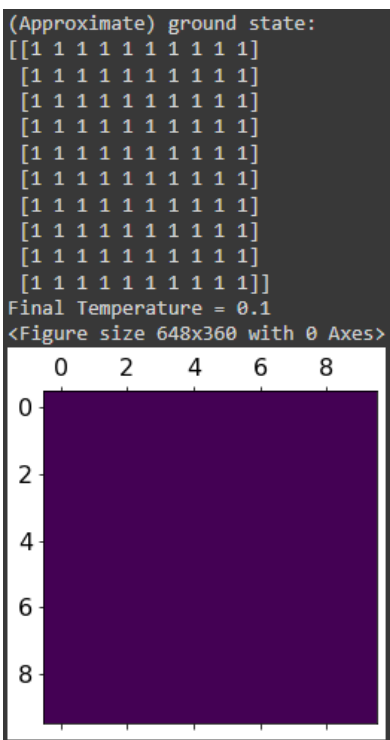


Figure 4: Ground State of Ising Model When $J = 1$ and $h=0$

This gave us a ferromagnetic state as all spins are aligned with a value of 1. Finally, we simulated $J=1$ and $J=-1$ when $h=20$ and when $h=-20$ and all simulations gave the same result which can be seen in Figure 5.

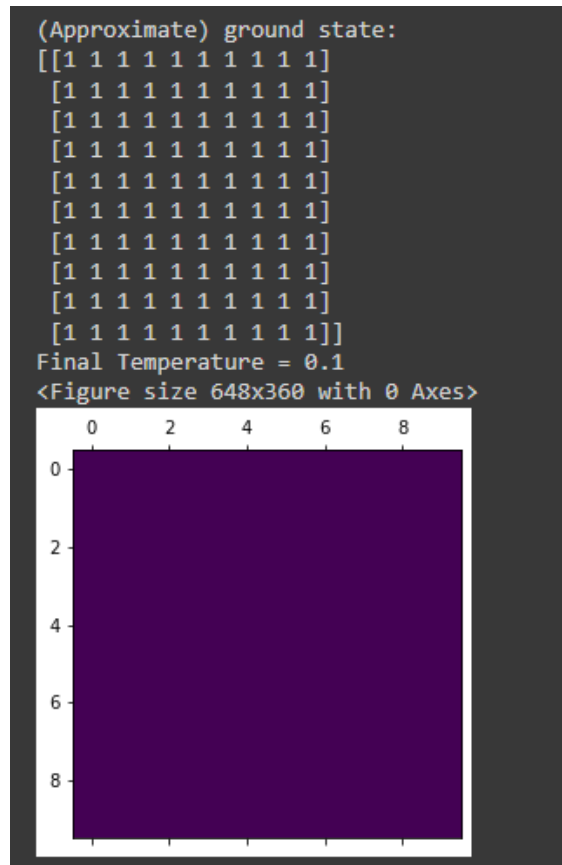


Figure 5: Ground State of Ising Model When $J= -1$ or 1 and $h= 20$ or -20

All simulations agree with our analysis in parts A and B as when $J = -1$ and $h = 0$ an antiferromagnetic state is calculated and when $J= 1$ and $h=0$ a ferromagnetic state is calculated. Then when we introduced a magnetic field by increasing h to 20 and leaving $J = -1$ a ferromagnetic state is calculated showing that h can cause favorability to spin up or spin down states if it is large enough.

B. Phase Transition

- a) In order to calculate the energy and specific heat of a second order phase transition we used code given to us in the Problem 2 Jupyter notebook seen in Figure 4 of Appendix A. We set $J= 1$ and $h = 0$ for all simulations. The first simulation ran was with an initial temperature of 5 and a final temperature of 0.1 with 10 steps, a size of 10 , and the

number of production steps was set to 100. This resulted in the graphs seen in Figure 6 below.

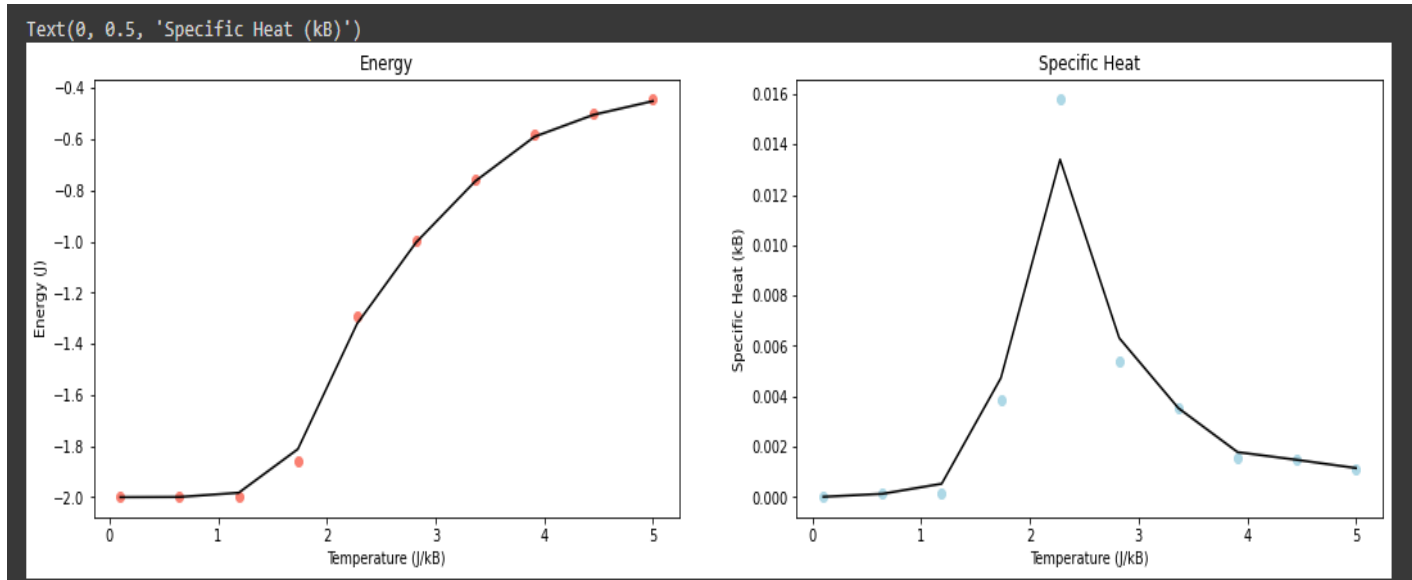


Figure 6: Graphs of Energy (J) and Specific Heat (kB) of 2D Ising Model vs Temperature (J/kB)

It is clear that the transition temperature occurs around 2.5 J/kB but we need to refine our settings for more accurate results. The next simulation was ran with an initial temperature of 3.5 and a final temperature of 1.5 with 100 steps, a size of 20, and the number of production steps was set to 1000 which gave us the graphs seen in Figure 7.

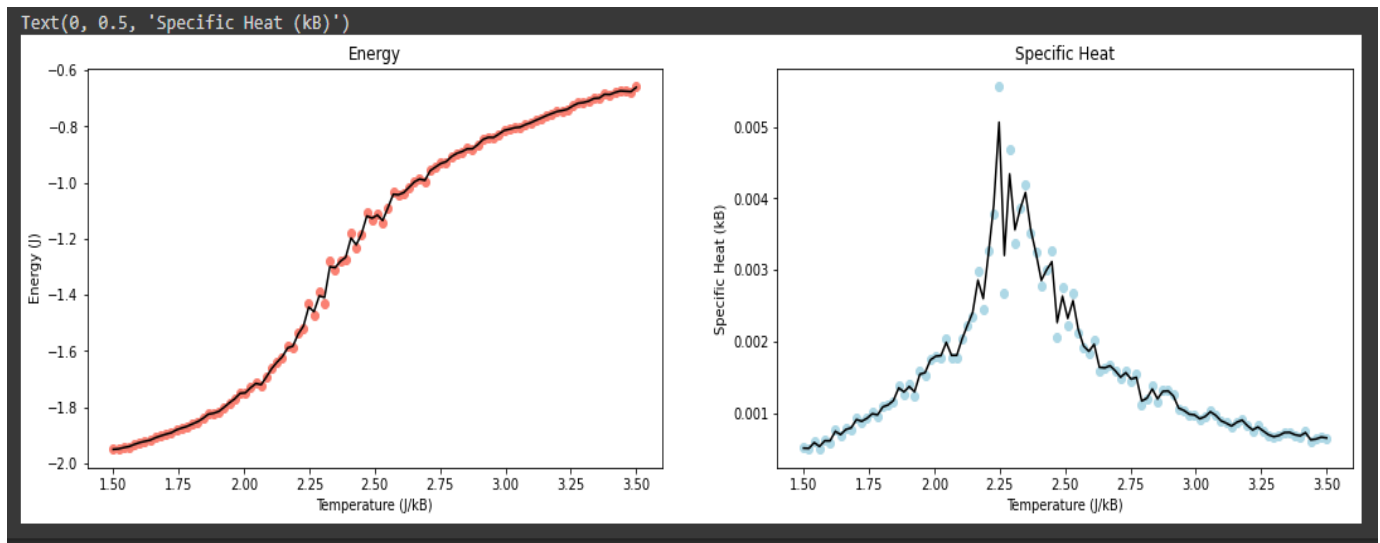


Figure 7: Refined Graphs of Energy (J) and Specific Heat (kB) of 2D Ising Model vs Temperature (J/kB)

With our refined settings our simulation calculated that the transition temperature is approximately 2.25 J/kB as that is where our specific heat graph peaks.

- b) We repeated the refined settings simulation of part A but we started with an initial temperature of 1.5 and had a final temperature of 3.5 instead which produced the graph seen in Figure 8.

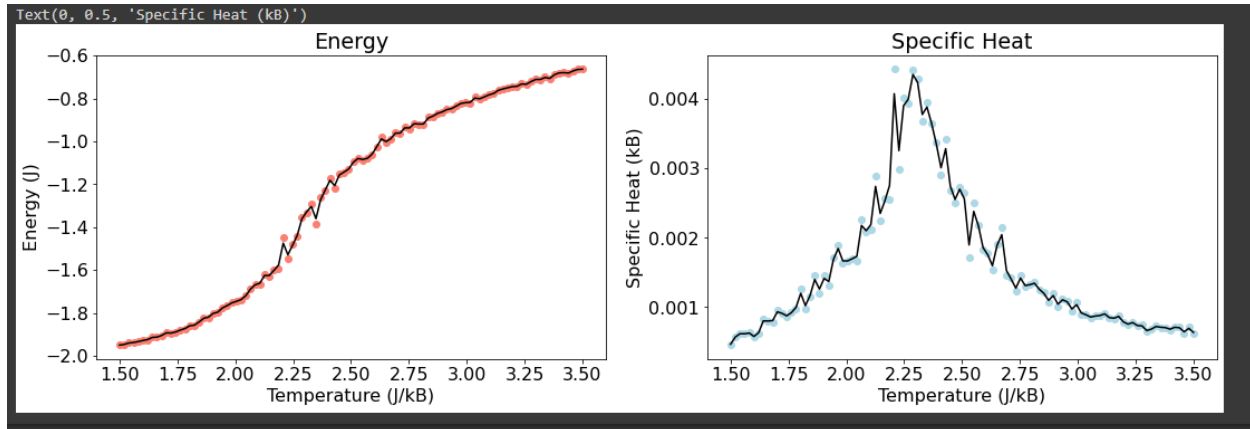


Figure 8: Graphs of Energy (J) and Specific Heat (kB) vs Temperature (J/kB) Calculated from Low to High Temperature

This shows that the phase transition temperature should be around 2.4 J/kB. The results are different compared to part A's graph as we started and ended in different states. In part A we started from a higher temperature so it was more disordered and cooled to a more ordered state. Whereas in part B we started from a colder and more ordered state and heated the system to a hotter and more disorderly state.

C. Order Parameter

- a) Magnetization is defined as the average spin of the system and is given by the equation $M = \frac{1}{N} \sum_i \sigma_i$ where N is the number of sites and σ_i is the spin at site i.

Magnetic susceptibility is defined in the Ising model as the variance of the magnetization which gives us the fluctuations of said magnetization. It is calculated by the equation

$\chi = \frac{1}{k_B T} (\langle M^2 \rangle - \langle M \rangle^2)$. Where M is the magnetization of the system, k_B is the Boltzmann constant, and T is the temperature.

Magnetization is a suitable order parameter because it changes from a positive number to zero when the phase transition occurs which makes calculations easier and cleaner to

execute. It is also much clearer to see the phase transition in the magnetization and magnetic susceptibility graphs as the transition is much sharper.

- b) To calculate the magnetization and susceptibility the code seen in Figure 4 of Appendix A was used. We started with the parameters of size equal to 10, a temperature range from 0.1 to 5 J/kB in 100 steps, and 100 production Monte Carlo Steps. These parameters and code calculated the graph seen in Figure 9 below.

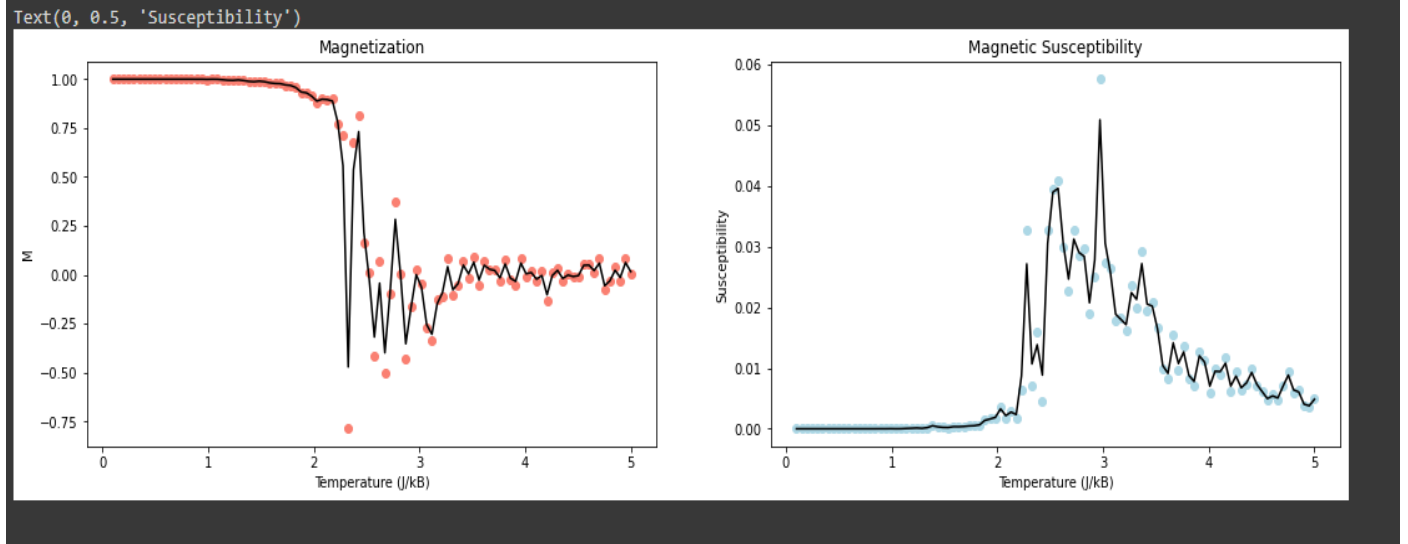


Figure 9: Graph of Magnetization and Susceptibility vs Temperature with Rough Parameters

These graphs do not agree nor give us a clear picture at what temperature the phase transition occurs at so we will need to refine our parameters further. Our next simulation was ran with parameters of size equal to 20, a temperature range from 1 to 4 J/kB in 100 steps, and 1000 production Monte Carlo Steps which resulted in the graph seen in Figure 10

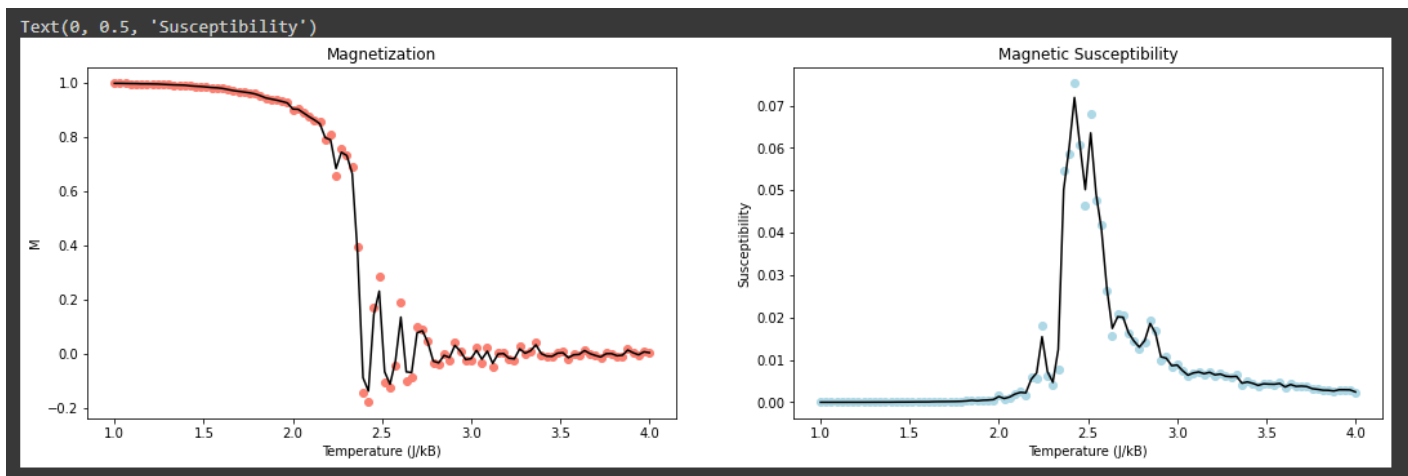


Figure 10: Graph of Magnetization and Susceptibility vs Temperature with Refined Parameters

c) According to our calculations in part B and the graph in Figure 10 the temperature of the phase transition is approximately 2.45 J/kB.

d) The real world phenomenon this simulation exhibits is called the Curie transition temperature of a ferromagnet. The name of the phase transition temperature is the ferromagnetic phase transition

Problem 3: Hydrogen Adsorption on the PT surface

A. H desorption with temperature

- a) I expect that the $T = 0\text{K}$ ground state H coverage to be FCC hollow.
- b) In order to complete the code in the Problem 3 Jupyter notebook the coverage, coverage susceptibility, and the graphing code must be created as well adding in our starting chemical potential. The initial chemical potential can be solved using the $\mu = -E_{ads} - TS$ where E_{ads} is equal to -0.45 eV and since $T = 0\text{K}$ then we know the initial chemical potential is 0.45 eV. To calculate the coverage and coverage susceptibility a similar process as the magnetization and magnetization susceptibility was used. The coverage and susceptibility were first initialized as zero and then using a for loop the coverage variable was added upon with the number of occupied sites and divided by the total number of sites to get the coverage. Then the susceptibility was calculated by taking the variance of the coverage and dividing it by the Boltzmann constant times the temperature. All of this was implemented into the Problem 3 Jupyter notebook which can be seen in Figure 5 of Appendix A.
- c) Our initial calculation of the coverage used a temperature from 0.1 to 1000 for the temperature in 200 steps, a size of 6 and a production Monte Carlo steps of 200. This produced the graph seen in Figure 11 seen below.

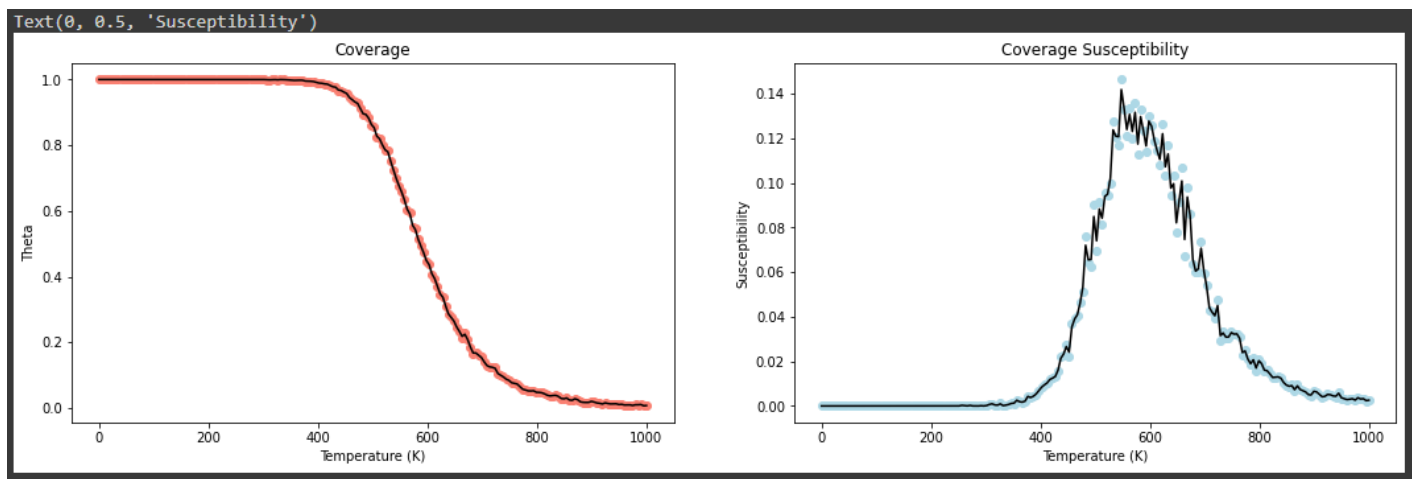


Figure 11: Coverage and Coverage Susceptibility vs Temperature (K) with Rough Parameters

The susceptibility spike isn't the clearest so we refined our parameters to get a clearer transition temperature. The next calculations were ran with 0.1 to 1500 for the temperature in 200 steps, a size of 10 and a production Monte Carlo steps of 1000. This gave us the graph seen in Figure 12 below.

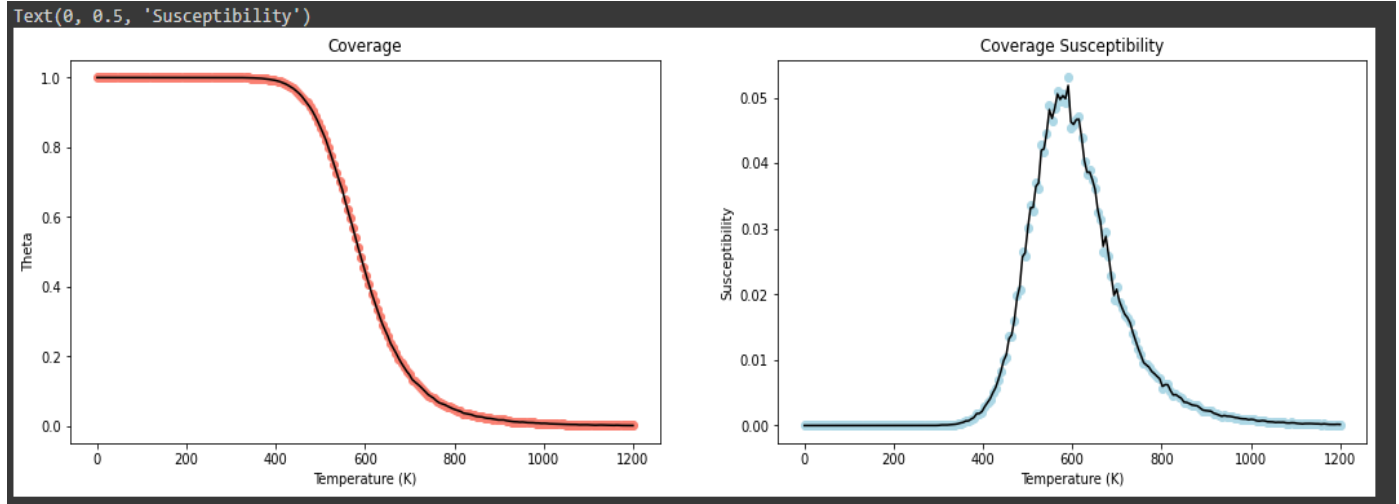


Figure 12: Coverage and Coverage Susceptibility vs Temperature (K) with Refined Parameters

There is a better coverage susceptibility spike and greater temperature range was used to show for certain that we had reached zero percent coverage. The coverage at room temperature is approximately a theta value of one which would mean approximately one hundred percent coverage. Desorption sets on around 400 degrees Kelvin.

B. Repulsion Between Adsorbed H Molecules

- a) We are given that for a single hydrogen atom the adsorption energy required to bond to the Pt(1 1 1) surface is -0.45 eV. We also know that when every adsorption site is occupied the adsorption energy of an H atom is -0.29 eV. In order to explain this difference in adsorption energy, we must consider the repulsive interaction that each neighboring hydrogen atom exerts on the hydrogen atom that we are observing. Thus, we can determine the repulsive interaction with the following equation:

$$\text{repulsive interaction} = \frac{(E_{\text{ads complete monolayer}} - E_{\text{ads of single atom}})}{\text{Number of neighboring atoms}}$$

For this problem we are told to use a square lattice so our Hydrogen atom has four neighbors. This gives us the following equation:

$$\text{repulsive interaction} = \frac{-0.29 \text{ eV} - 0.45 \text{ eV}}{4} = +0.04 \text{ eV}$$

Each neighboring atom emits a repulsive force of .04 and since there are four atoms that is a total repulsive force of .16 eV which is the difference observed between the adsorption energy of H atoms on the Pt(111) surface and the adsorption energy of an H atom within a complete monolayer on the Pt(111) surface.

- b) Now that we know $V = .04 \text{ eV}$ this was added to our code which can be seen in Figure 6 of Appendix A. When we ran our code again the graph seen in Figure 13 was produced.

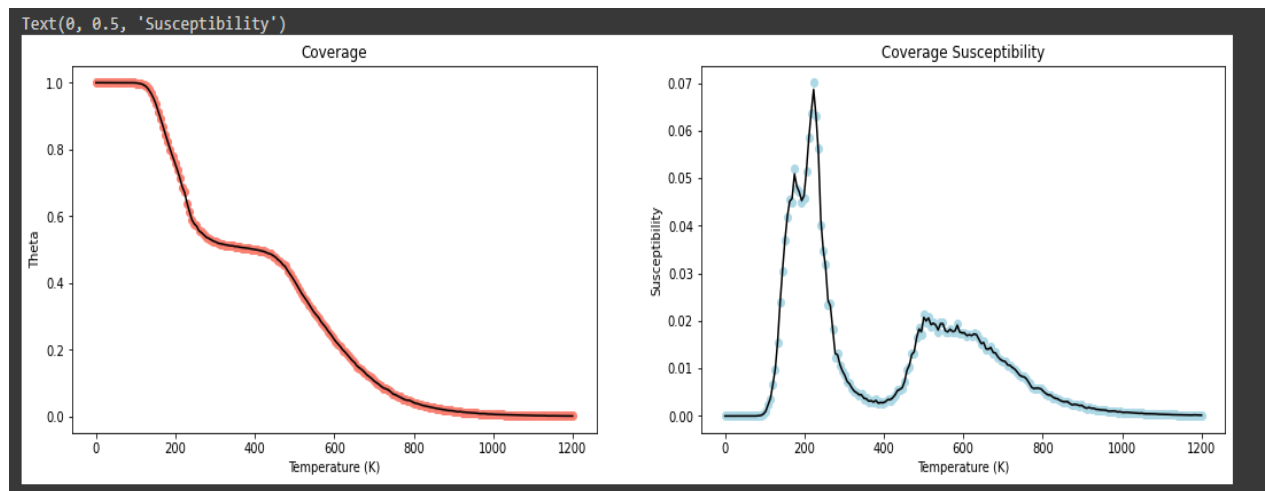


Figure 13: Coverage and Coverage Susceptibility vs Temperature (K) with Repulsive Force Added

The preferred state at room temperature has changed compared to part A. Before the preferred state was at 100% coverage at room temperature but after adding the repulsive force it is approximately 50% coverage. The coverage begins at 100% but then swiftly drops to about 50% at 200 degrees kelvin and then slowly drops to no coverage which begins around 450 degrees Kelvin. Qualitatively the coverage is inversely proportional to temperature. This is due to the extra thermal energy that a rising temperature provides to the system which makes bonding less likely as atoms are moving around more. Also, atomic bonds are more likely to break since atoms will be more excited from the increase in temperature.

Problem 4:

- A Markov chain method is a stochastic process that describes a sequence of events in which the probability of the $n+1$ event only depends on the current, n , state.
- Kinetic Monte Carlo simulations are used to simulate a process that evolves with time through the use of transition rates. It differs from Metropolis Monte Carlo as Metropolis MC is for equilibrium properties only and doesn't provide any time information whereas kinetic MC is for non-equilibrium processes such as nucleation.

- c) We used the microcanonical statistical ensemble in problem 3 because we had a changing chemical potential that was dependent on temperature. This was the correct choice because we were looking at H atom coverage as a function of temperature and canonical and grand canonical have temperature as a constant which would not make sense for problem 3. We also had a changing chemical potential which eliminates the grand canonical ensemble.

Problem 5:

Plugging in $p_i = \frac{\sigma_i+1}{2}$ and $p_j = \frac{\sigma_j+1}{2}$ into the Hamiltonian of the Lattice Gas Model we get:

$$\begin{aligned}
 H &= V \sum_i^{sites} \sum_j^{neighbors} \left(\frac{\sigma_i+1}{2} \right) \left(\frac{\sigma_j+1}{2} \right) - \mu \sum_i^{sites} \frac{\sigma_i+1}{2} \\
 &= V \sum_i^{sites} \sum_j^{neighbors} \left(\frac{\sigma_i \sigma_j}{4} \right) + \left(\frac{\sigma_i}{4} \right) \left(\frac{\sigma_j}{4} \right) + \left(\frac{1}{4} \right) - \mu \sum_i^{sites} \frac{\sigma_i+1}{2} \\
 &= \frac{V}{4} \sum_i^{sites} \sum_j^{neighbors} (\sigma_i \sigma_j) - \left(\frac{Vz}{2} + \frac{\mu}{2} \right) \sum_i^{sites} \sigma_i - \left(\frac{Vz}{8} + \frac{\mu}{2} \right) N
 \end{aligned}$$

Where z is the number of nearest neighbors.

- a) Plugging in (0, 1) site occupancies to $p_i = \frac{\sigma_i+1}{2}$ and $p_j = \frac{\sigma_j+1}{2}$ give us $0 = \frac{\sigma_i+1}{2}$ and $1 = \frac{\sigma_j+1}{2}$ which when solved for the spin states gives us $\sigma_i = -1$ and $\sigma_j = 1$
- b) the interaction parameters J and V are related by the equation $V/4 = J$ or $V = 4*J$ which means they are inversely proportional to one another
- c) The magnetic field h is equal to $\left(\frac{Vz}{2} + \frac{\mu}{2} \right)$ which means they are directly proportional to one another

Collaborations: I did not collaborate with anyone on this project

Appendix A:

```
[ ] def f(x):  
    return x*x  
  
    # area of rectangle  
    x_max = 4.0  
    y_max = 16.0  
    A = x_max*y_max  
  
    N = 10 # increase this number to improve accuracy  
    N_within = 0  
    for i in range(N):  
        # sample random point within rectangle  
        r_x = x_max*np.random.random() # r_x in [0, 4]  
        r_y = y_max*np.random.random() # r_y in [0, 16]  
        # count points within area I  
        if r_y <= f(r_x):  
            N_within += 1  
  
    print("The integral is approximately I = {}".format(N_within/N*A))  
  
The integral is approximately I = 25.6
```

Figure 1: Problem 1 Jupyter Notebook Integral Approximation Code

```
from math import sqrt  
  
def f(x):  
    return sqrt(1 - x*x) # COMPLETE  
  
    # area of quarter of circle  
    x_max = 1.0 # COMPLETE  
    y_max = 1.0 # COMPLETE  
    A=x_max*y_max  
  
    N = 100000 # increase this number to improve accuracy  
    N_within = 0  
    for i in range(N):  
        # sample random point within quarter of circle  
        r_x = x_max*np.random.random() # r_x in [0, 1]  
        r_y = y_max*np.random.random() # r_y in [0, 1]  
        # count points within area I  
        if r_y <= f(r_x):  
            N_within += 1  
    print("Pi is approximately = {}".format(N_within/N*A))
```

Figure 2: Problem 1 Jupyter Notebook Pi Approximation Code

```
# sites per lattice direction
size = 10
# initial/final T and number of samples
T0, T1, nT = 10, 0.1, 100
# MC steps
N = 500

kB = 1.0 # Boltzmann's constant
J = -1.0
h = 20.0

ising = Ising(size, J, h)
ising.decorate(spin_up=0.5)

temperature = np.linspace(T0, T1, nT)
for i, T in enumerate(temperature):
    beta = 1.0/(kB*T)
    for j in range(N):
        ising.muVT_sweep(beta)
```

Figure 3: Problem 2 Jupyter Notebook Ground State Code

```

size = 20 # COMPLETE
T0, T1, nT = 1.5, 3.5, 200 # COMPLETE
# equilibration and production MC steps
N_equil, N_produ = 100, 1500

kB = 1.0
J = 1.0
h = 0.0

ising = Ising(size, J, h)
ising.decorate(spin_up=1.0)

temperature = np.linspace(T0, T1, nT)
energy = np.zeros(nT)
specific_heat = np.zeros(nT)
magnetization = np.zeros(nT)
susceptibility = np.zeros(nT)

for i, T in enumerate(temperature):
    beta = 1.0/(kB*T)
    for j in range(N_equil):
        ising.muVT_sweep(beta)
    E = np.zeros(N_produ)
    M = np.zeros(N_produ)
    for j in range(N_produ):
        ising.muVT_sweep(beta)
        E[j] = ising.energy/ising.num_sites
        M[j] = np.sum(ising.sites)/ising.num_sites
    energy[i] = np.mean(E)
    specific_heat[i] = np.var(E)/(kB*T**2)
    magnetization[i] = np.mean(M) # COMPLETE
    susceptibility[i] = np.var(M)/(kB*T) # COMPLETE

```

Figure 4: Problem 2 Jupyter Notebook Integral Phase Transition Code


```

size = 10
T0, T1, nT = 0.1, 1200, 200
N_equil, N_produ = 100, 1000

kB = 8.6173303e-5 # eV/K
V = 0.00
mu0 = 0.45 # COMPLETE

lgm = LGM(size, V, mu0)
lgm.decorate(coverage=1.0)

temperature = np.linspace(T0, T1, nT)
coverage = np.zeros(nT)
susceptibility = np.zeros(nT) # COMPLETE

for i, T in enumerate(temperature):
    # entropy term for H2 gas and update of mu
    TS = 1.23697548e-03*T + 5.04703379e-07*T**2
    lgm.update_mu(mu0 - 0.5*TS)
    beta = 1.0/(kB*T)
    theta = np.zeros(N_produ)

    for j in range(N_produ): # COMPLETE
        lgm.muVT_sweep(beta)
        theta[j] = np.sum(lgm.sites)/lgm.num_sites

    coverage[i] = np.mean(theta)
    susceptibility[i] = np.var(theta)/(kB*T)

fig = plt.figure(figsize=(18, 5))

sp = fig.add_subplot(1, 2, 1)
smoothed = gaussian_filter1d(coverage, sigma=0.5)
plt.scatter(temperature, coverage, color="salmon")
plt.plot(temperature, smoothed, color="black")
plt.title("Coverage"); plt.xlabel("Temperature (K)"); plt.ylabel("Theta")

sp = fig.add_subplot(1, 2, 2)
smoothed = gaussian_filter1d(susceptibility, sigma=0.5)
plt.scatter(temperature, susceptibility, color="lightblue")
plt.plot(temperature, smoothed, color="black")
plt.title("Coverage Susceptibility"); plt.xlabel("Temperature (K)"); plt.ylabel("Susceptibility")

```

Figure 5: Problem 3 Jupyter Notebook H Desorption No Repulsive Force Code

```

size = 10
T0, T1, nT = 0.1, 1200, 200
N_equil, N_produ = 100, 1000

kB = 8.6173303e-5 # eV/K
V = 0.04
mu0 = 0.45 # COMPLETE

lgm = LGM(size, V, mu0)
lgm.decorate(coverage=1.0)

temperature = np.linspace(T0, T1, nT)
coverage = np.zeros(nT)
susceptibility = np.zeros(nT) # COMPLETE

for i, T in enumerate(temperature):
    # entropy term for H2 gas and update of mu
    TS = 1.23697548e-03*T + 5.04703379e-07*T**2
    lgm.update_mu(mu0 - 0.5*TS)
    beta = 1.0/(kB*T)
    theta = np.zeros(N_produ)

    for j in range(N_produ): # COMPLETE
        lgm.muVT_sweep(beta)
        theta[j] = np.sum(lgm.sites)/lgm.num_sites

    coverage[i] = np.mean(theta)
    susceptibility[i] = np.var(theta)/(kB*T)

```

Figure 6: Problem 3 Jupyter Notebook H Desorption With Repulsive Force Code