

Carter Gustin  
PHY 490  
Professor Freeman  
Final Report

## 1. Overview of the Metropolis-Hastings Algorithm

### a. The Algorithm

- i. The Metropolis-Hastings Algorithm is a Markov Chain Monte Carlo random sampling method designed to find the ideal state of a system based on individual independent random moves. The general steps in the method are to define an initial state, choose a random adjustment to the initial state that will produce a new and different state, calculate an acceptance probability depending on the nature of the state, then either accept the state and allow this to be the new configuration or reject it and revert back to the state prior to the suggestion. A move can be accepted in two ways: first, it is accepted if it allows the system to move towards the ideal state that is desired, however, it can also be accepted even if it moves away from this ideal state. Its acceptance in this manner is determined on its acceptance probability which is calculated based on the probability for it to be in some state  $s$  ( $P(s)$ ), the current state. This probability will be a number between 0 and 1. When this probability is calculated, another random variable with uniform distribution between 0 and 1 is calculated. If  $P(s)$  is greater than this randomly distributed variable, then its state is accepted, even if it brings the system further from its ideal state. If  $P(s)$  is less than the random variable, then the state is rejected and the system is reverted to the state before the suggestion. There are two criteria that the Metropolis-Hastings Algorithm has to follow. First, it must be reversible such that the probability of obtaining state  $s'$  from  $s$  is equal to the probability of obtaining state  $s$  from  $s'$ . Second, it must be ergodic so all states will eventually be accessible.

### b. How the Algorithm is used in these Projects

- i. Both Projects 1 and 2 are governed by the laws of statistical mechanics. Because of this, Boltzmann statistics are used to calculate the probabilities of the states of the system. The system is trying to get in its lowest energy state so any suggestion that lowers the energy of the system will automatically be accepted and become the new state. If it raises the energy of the system, it will be accepted based on the Boltzmann probability

function. From statistical mechanics, the ratio of probabilities between two distinct states is given by the formula:

$$\frac{P(s')}{P(s)} = e^{-\frac{(E(s') - E(s))}{kT}}$$

Where  $E(s)$  is the energy of state  $s$ ,  $T$  is the temperature of the system which is the independent variable, and  $k$  is the Boltzmann constant,  $k = 1.381 * 10E-23 \text{ J/K}$ . If this quantity calculated is greater than the random variable chosen uniformly between 0 and 1, then the move is also accepted, even though it raises the energy of the system. Both projects 1 and 2 are temperature-dependent models. The acceptance probability that is calculated is proportional to  $\exp(-1/kT)$ . This means that at lower temperatures, the acceptance probability is low while at high temperatures, the acceptance probability is high.

## 2. How Statistics are Calculated

### a. The Statistics Code

- i. A separate .py file was created in order to calculate statistics on both models in projects 1 and 2. Each model has an interesting observable quantity to study and how it varies with the temperature. First, data of the property of interest for the model is sent to a .txt file at the specific temperature being studied. Once there is a significant number of .txt files, each with a different temperature. Each file is graphed to see at what point the model has thermalized. Each system starts in the same state regardless of the temperature. Because of this, it takes time for the system to get to the true tendency of the model at the particular temperature. Once this thermalization point is noted, the data after this point is placed into a list. The average values and standard deviation of these post-thermalization values are of interest but since each value is correlated with the values next and before it, these data points must be blocked. Blocking is a method that allows for a chunk of data points to be averaged and then all of these new values are then averaged.

## 3. Project 1: The Ising Model

### a. The Model

- i. The Ising Model of a Ferromagnet is a representation of atomic spins inside of a material and how their individual spins contribute to the total magnetization of the material. Although in reality, the atomic spins are generally not confined to one axis, for the sake of simplicity, the Ising Model only allows for each atom to be spin up or spin down. A spin up

atom is given  $\sigma = +1$  whereas a spin down atom has  $\sigma = -1$ . This distinction is important because the energy is dependent on pairs of neighboring spins. The energy of state  $s$  is given as:

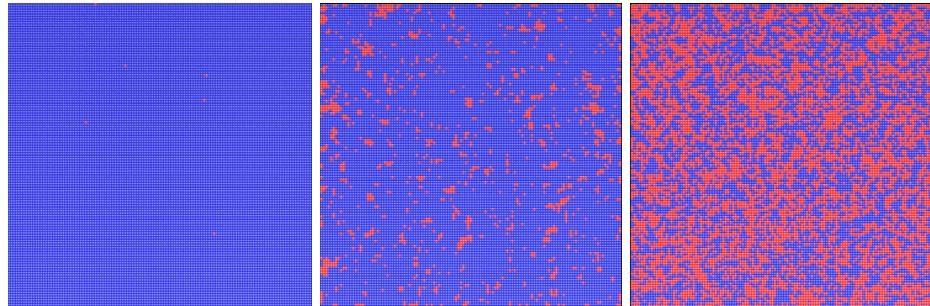
$$E(s) = - \sum_{i,j}^n \sigma_i \cdot \sigma_j$$

i.e., the energy contribution of one spin,  $\sigma_i$ , depends on the spins surrounding this atom. If  $\sigma_i$  has neighboring spins of the opposite spin, this will make  $\sigma_i \cdot \sigma_j < 1$  otherwise, if the spins are the same then  $\sigma_i \cdot \sigma_j > 1$ .

The total energy of state  $s$  is then the negative sum over all neighboring spins  $i,j$ . What is of interest in this model is the overall magnetization of the entire system which is dependent on each individual spins. The magnetization is calculated as the absolute value of the difference between the up spins and the down spins divided by the total number of spins. This gives the restriction:  $0 < M < 1$ . If all spins are spin up (or spin down) then the magnetization will be number of up spins divided by the total number of spins which are the same so  $M$  will be 1. The other extreme will be if there are an equal amount of spin up atoms and spin down atoms. In this case, spin up atoms minus spin down atoms will be zero so  $M$  will be 0.

### b. Coding the Model

- i. The initial system is a configuration of  $n \times n$  spin up atoms initialized. After the energy of the initial state is calculated, the algorithm begins. The metropolis algorithm for this model is to randomly choose one spin, flip it to the opposite spin, and then calculate how this change affects the overall energy of the system. If the energy decreases after the move, this becomes the new state of the system, otherwise, the acceptance probability is calculated to determine if a move that increases the energy will also be accepted. Because the acceptance probability depends on the temperature of the system, different temperatures will yield different total magnetization values of the model:



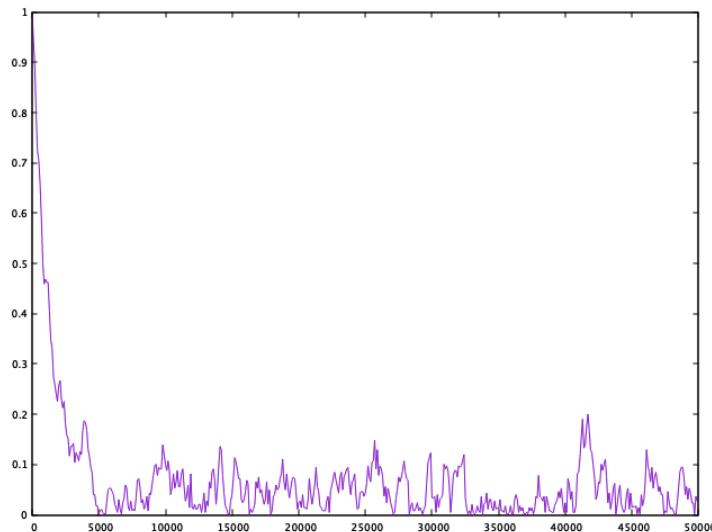
kT = 1.0

kT = 2.25

kT = 5.0

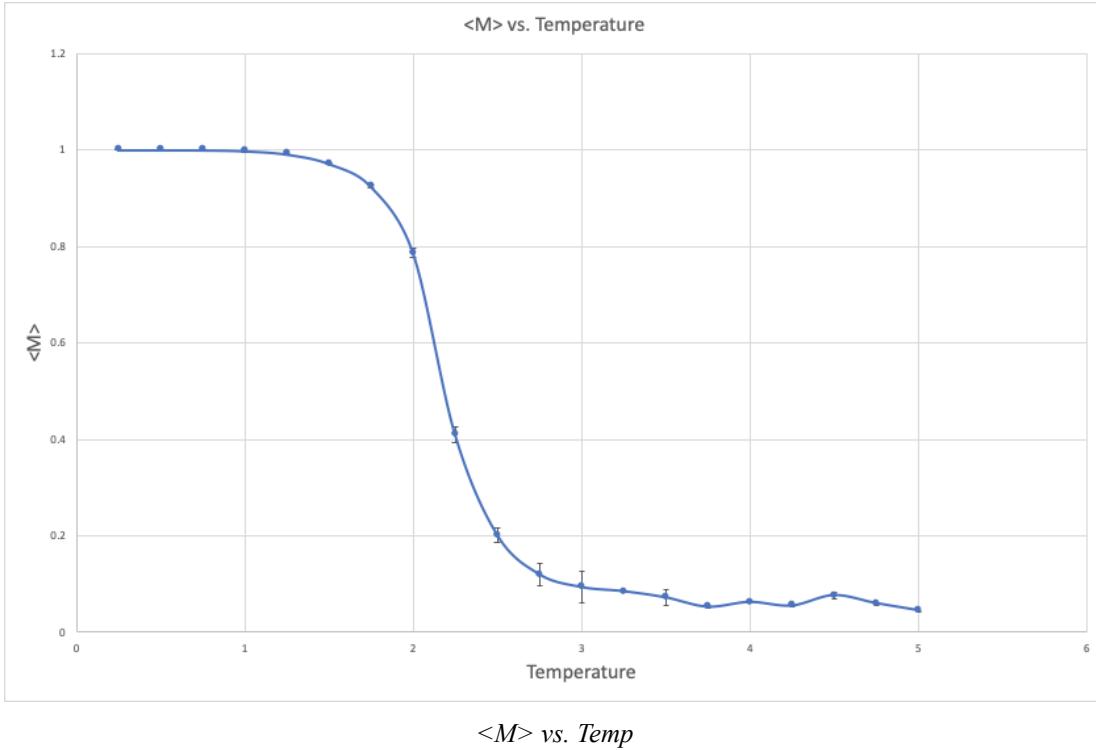
### c. Results

- i. The quantity of interest for this ferromagnet model was average magnetization,  $\langle M \rangle$ . Magnetization values for each temperature were placed into a .txt file and after the thermalization point was determined, and the new data was blocked, an average value for the magnetization value and a standard deviation for the particular temperature was acquired. To determine the point of thermalization, the magnetization data points must be analyzed. The following graph shows a standard magnetization versus Metropolis suggestions:



*Absolute Magnetization vs. Metropolis Suggestions ( $kT = 5.0$ )*

Thermalization occurs after the system has reached a point that the system behaves independent of its initial state and behaves solely on how the inputted temperature allows it to act. For this graph, it is clear that thermalization occurs after around 5,000 suggestions. Then, data post-thermalization was blocked and averaged. After this was completed for every temperature, a graph of  $\langle M \rangle$  vs.  $kT$  was graphed:



$\langle M \rangle$  vs. Temp

It is clear from this graph that there is a certain point where the phase change occurs and the magnetization goes from nearly 1.0 to almost zero. This phase change occurs around a value of  $kT = 2.25$ . From this, it can be concluded that at lower temperatures, the ferromagnet is highly magnetic but as the temperature increases, there is a point where the magnetization begins to rapidly fall to zero.

#### 4. Project 2: The Phase Change Model

##### a. The Model

- i. This model examines the interaction between neutral particles and how the system behaves at various temperatures. The Lennard Jones 12-6 potential gives the energy between two neutral particles. This potential is given by:

$$V(r) = k \left( \left(\frac{r_0}{r}\right)^{12} - \left(\frac{r_0}{r}\right)^6 \right)$$

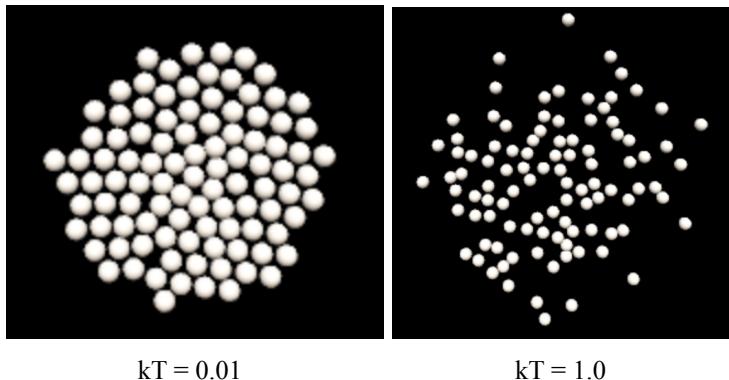
Where  $k$  is a proportionality constant,  $r_0$  is the minimum of the potential and  $r$  is the distance between the two particles. The total energy of the system at a particular state then becomes the sum of the Lennard Jones Potential over all particles (making sure not to overcount). The average energy of the system versus temperature is of interest in this model.

Depending on the temperature, the system will behave as one of the states

of matter and when a wide-enough range of temperatures is tested, an obvious phase change occurs.

b. Coding the Model

- i. The particles in the model are randomly initialized by randomly choosing an x and y coordinate for each individual particle between (-1, 1). After each particle has been placed in the xy plane, the initial energy of the system is calculated. The Metropolis Algorithm for this model is to choose one particle, displace it by a small distance (while making sure it stays in the “box” from (-1, 1)), re-calculate the energy of this new configuration and then accept or reject based on the previous criteria. The acceptance probability formula is the same as for the previous model. It is assumed that at low temperatures, the system will behave as a solid with particles clumping together into one region and as the temperature is increased, there will be a phase transition to a liquid state where the particles will be in clusters throughout the phase space. This is evident through the animation of the code:

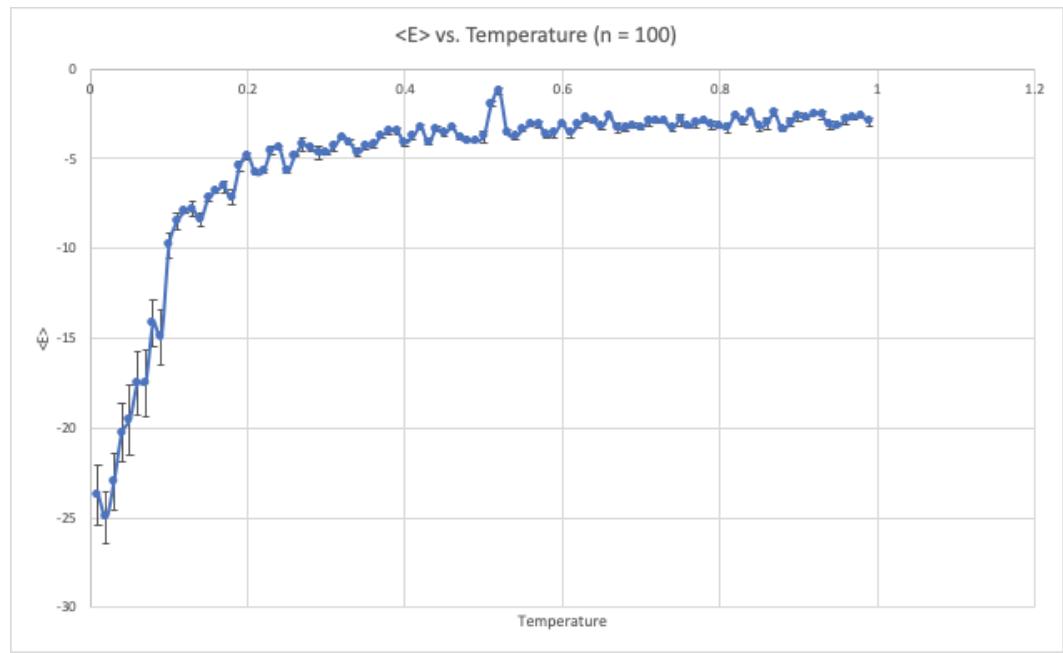


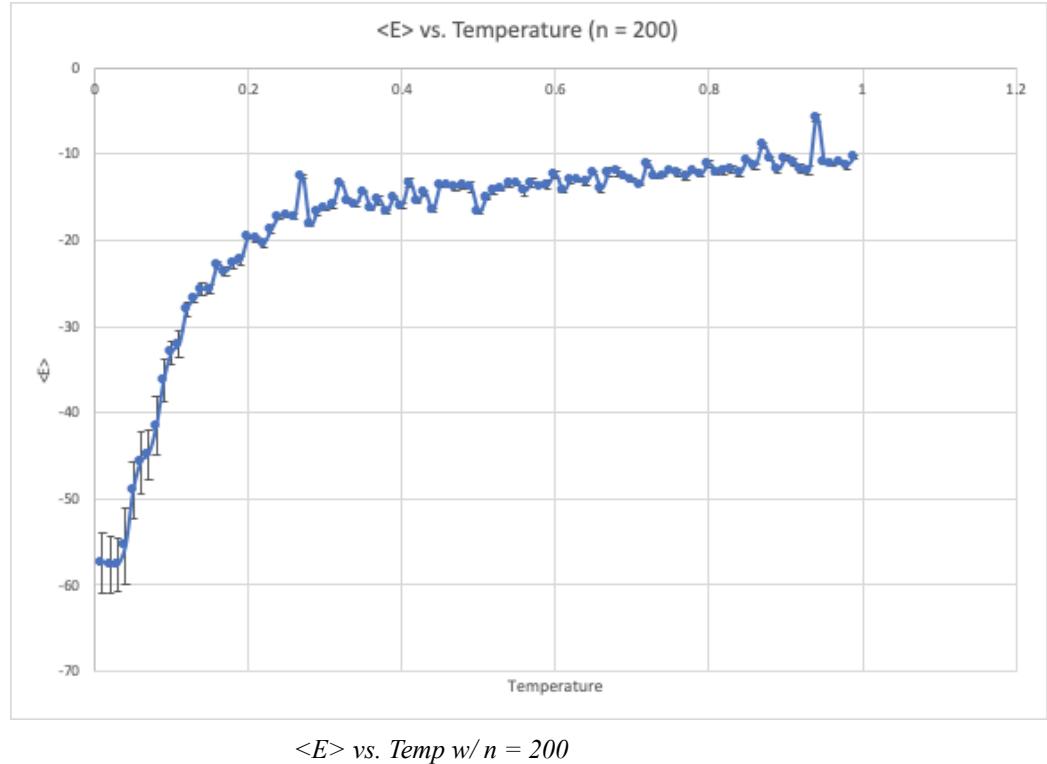
At  $kT = 0.01$ , all of the particles want to be close to each other so this state is a solid while at  $kT = 1.0$ , the particles are still close but they are more spread out throughout the phase space so this is classified as a liquid (It is not a gas due to tendencies for particles to still remain close to each other). Clearly, there is a phase transition between  $kT = 0.01$  and  $kT = 1.0$ . Graphing the average energy of the system as a function of the system’s temperature allows for the location of the phase change to be determined.

c. Results

- i. The method of calculating statistics for this model was the same as for the Ising model. Energy values are sent to a .txt file and they are opened in a separate python script with the thermalization point noted (The magnetization data points versus Metropolis suggestions graph looks very

similar to that of the Ising Model in project 1. Because of this, the method of choosing a thermalization point and blocking is the same). After the values are blocked, an average energy of the system at the particular temperature is taken down. This is repeated for each temperature (this process was automated through the os and sys python modules which allowed for temperatures from  $kT = 0.01$  to  $kT = 1.0$  to be tested automatically). The following graphs are the average energy versus temperature with  $n = 100$  atoms and  $n = 200$  atoms (Note: The phase change occurs at the same place regardless of the number of atoms. The only thing that changes is the average energy so the entire graph is shifted down for  $n = 200$ ):





The shape of the graph is as expected. There is a vertical line where the phase change (the melting point) occurs which is roughly around  $kT = 0.1$  and then a positively sloped line where the liquid state resides. The phase change is not perfectly vertical due to imperfections in the simulation.

These graphs corroborate the original assumption that at low temperatures, the system is a solid and there is a phase change where the system becomes a liquid state after the melting point of around  $kT = 0.1$  to  $0.2$ .

## 5. Final Thoughts

- The Metropolis-Hastings Algorithm is a very powerful tool in statistical mechanics simulations. The Boltzmann Statistics method deals with studying individual states of a larger system, for example one spin in a ferromagnet or one atom in the phase change model, and how changing this state affects the entire system. The Metropolis-Hastings Algorithm works in a very similar way. Because of this, the algorithm is very effective in solving problems in statistical mechanics. The algorithm provides results that are predicted by theory for both the ferromagnet and the phase change model.