Project 4 - Ising Model

Sam Edwards,[1] Eric Aboud,[1] and Parker Brue[1]

[1] *Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48823*

We present an algorithm that allows us to solve phase transitions within the two dimensional Ising model. Using Monte Carlo methods, specifically the Metropolis algorithm, we are able to compare our results to the standard analytical model and compute probability distributions and explicitly plot out the number of cycles needed for a system to reach equilibrium as well as study phase transitions close to a system's critical temperature. We found agreement with the standard analytical model, and found the Critical Temperature to be 2.281, which is in close agreement with the accepted values of previous experiments. Ultimately, the ground covered within this report gives us the skills to extend Monte Carlo methodology to the greater breadth of statistical systems.

INTRODUCTION

The Ising model, named after Ernst Ising, serves as a historically significant mathematical model of ferromagnetism, and the two dimensional model is one of simplest statistical models showing phase transitions. It is also an excellent showcase for Monte Carlo methods, specifically the elegant Metropolis algorithm. This project serves as an exploration into the versatility of the model and Monte Carlo methodology. We present analytical solutions to the Ising Model, and compare them to Monte Carlo methods, first studying the basic 2x2 model, and then delving into a finer analysis of the Monte Carlo method through cycle efficiency in reaching equilibrium and analysing the probability distribution of a steady state. Moving on to phase transitions, we numerically analyze the behavior of the Ising model close to the critical temperature, also extracting the critical temperature with various lattice sizes, approaching the thermodynamic limit of L = ∞. The integration of Monte Carlo methodology in our project not only allows for a wide analysis of the Ising model, but allows a glimpse into the power of solving deterministic systems through repeated random sampling.

Chronologically, our report first discusses the theory, algorithms and methods behind our project, then transitions into a structured analysis and discussion of the results to the posed questions. Finally, we summarize, and consider the consequences of our model and how we can improve our methods.

THEORY, ALGORITHMS AND METHODS

2-D Ising Model

Starting from the the standard given energy of the Ising model without an externally applied magnetic field:

$$E = -J \sum_{<kl>}^{N} s_k s_l \tag{1}$$

Where, $s_k = \pm 1$, N represents the total number of spins and J is a coupling constant expressing the strength of the interaction between neighboring spins. $< kl >$ indicates that we sum over nearest neighbors only, and we assume a ferromagnetic ordering, namely $J > 0$.

From this representation of the energy, we can build the partition function from the generic representation:

$$Z = \sum_{i=1}^{M} e^{-\beta E_i} \tag{2}$$

Where $\beta = \frac{1}{K_B T}$. Once the partition function has been found, it's simple to find $\langle E \rangle$, the expectation value of the system energy:

$$\langle E \rangle = \frac{1}{Z} \sum_{i=1}^{M} E_i e^{-\beta E_i} \tag{3}$$

From this, $\langle E^2 \rangle$:

$$\langle E^2 \rangle = \frac{1}{Z} \sum_{i=1}^{M} E_i^2 e^{-\beta E_i} \tag{4}$$

Similarly to find $\langle |M| \rangle$, the expectation of the absolute value of the system magnetic moment:

$$\langle |M| \rangle = \frac{1}{Z} \sum_{i=1}^{M} M_i e^{-\beta E_i} \tag{5}$$

and $\langle M^2 \rangle$:

$$\langle M^2 \rangle = \frac{1}{Z} \sum_{i=1}^{M} M_i^2 e^{-\beta E_i} \tag{6}$$

An important value in the system, the specific heat, or the heat required to raise the temperature of a unit mass of the system by one unit temperature is found by:

$$C_v = \frac{\beta}{T}(\langle E^2 \rangle - \langle E \rangle^2) \tag{7}$$

The magnetic susceptibility of the system χ, or the ease in which the system can be magnetized in relation to an applied magnetic field is found as:

$$\chi = \beta(\langle M^2 \rangle - \langle |M| \rangle^2) \tag{8}$$

Phase Transition

When looking at an Ising modeled system near critical temperature, T_c, the behavior morphs into that resembling power laws. Mean magnetization $\langle M(T) \rangle$ can be approximated as: [1]

$$\langle M(T) \rangle \sim (T - T_c)^{\beta}, (\beta = \frac{1}{8}) \tag{9}$$

similarly, heat capacity C_v:

$$C_v(T) \sim |T_c - T|^{\alpha}, (\alpha = 0) \tag{10}$$

and susceptibility $\chi(T)$:

$$\chi(T) \sim |T_c - T|^{\gamma}, (\gamma = \frac{7}{4}) \tag{11}$$

We can also look at the correlation length, which should on the order of lattice spacing for T >> T_c. Just as spins should become increasingly correlated at T approaches T_c, the correlation length also increases. We can map the divergent behavior of ξ:

$$\xi(T) \sim |T_c - T|^{-\nu} \tag{12}$$

In a second-order phase transition, in which the correlation length spans the entire system, ξ will be proportional to the lattice size, L, a finite value. This means, relating finite lattice behavior with an infinitely large lattice, we find the critical temperature scales as:

$$T_c(L) - T_c(L = \infty) = aL^{\frac{-1}{\nu}} \tag{13}$$

With a as a newly introduced constant. Setting T=T_c Mean magnetization $\langle M(T) \rangle$ now is valued:

$$\langle M(T) \rangle \sim (T - T_c)^{\beta} \rightarrow L^{\frac{-\beta}{\nu}} \tag{14}$$

similarly, heat capacity C_v:

$$C_v(T) \sim |T_c - T|^{-\alpha}, \rightarrow L^{\frac{\alpha}{\nu}} \tag{15}$$

and susceptibility $\chi(T)$:

$$\chi(T) \sim |T_c - T|^{-\gamma}, \rightarrow L^{\frac{\gamma}{\nu}} \tag{16}$$

Monte Carlo Methods

In contrast to conventional numerical discretization methodology, the statistical simulation methodology of Monte Carlo Methods tackle physical problems through random sampling of physical system's probability distribution functions. This means that it functions on the basis of running several random samplings and averaging over the number of total observations. Practically, one can extract the variance in the average result, and this can be used to estimate the total number of necessary Monte Carlo simulations to obtain a more accurate result.

The Metropolis algorithm in particular randomly selects from our matrix a spin as a candidate to flip. The energy at this state increases when the candidate spin is next to a spin pointing in the opposite direction (represented by positive or negative one), and decreases when it is next to a spin pointing in the same direction. In this way a variable "deltaE" is built up, with an inherent coupling constant "J" in its units describing the arbitrary interaction strength between two neighbors. For spins on the edge of the lattice, periodic boundary conditions are applied via the function seen in Figure 1.

```
// inline function for periodic boundary conditions
inline int periodic(int i, int limit, int add) {
    return (i+limit+add) % (limit);
}
```

FIG. 1: One of the more important assumptions to be made in the Ising model is periodic boundary conditions for the lattice with a size of our chosing. Because the energy of a state depends on neighboring spins, this function allows spins on the edges of the lattice to interact with a full set of neighbors through periodicity.

This "deltaE" value is an integer that can be used to find a corresponding energy in a predefined array of possible energy changes the system can make (this setup array can be seen near the top of Figure 3).

The randomness of the Monte Carlo approach hinges on this value, as it is compared to a random number in an inequality. Should the random number be less than the energy, the spin at the current index flips and the overall magnetization and energy are either lessened or increased, depending on the random outcome of the inequality. One application of the Metropolis function does this $nspins^2$ times. Pseudo-code can be seen in Figure 2.

```
void Metropolis(int n_spins, long& idum, int
    **spin_matrix, double& E, double&M, double *w,
    int&matrixcounter)
{
int babymatrixcounter = 0;
    // loop over all spins
        for(int y =0; y < n_spins; y++) {
            for (int x= 0; x < n_spins; x++){
            // Find random position
            int ix = (int)
                (ran1(&idum)*(double)n_spins);
            int iy = (int)
                (ran1(&idum)*(double)n_spins);
            int deltaE = 2*spin_matrix[iy][ix]*
            (spin_matrix[iy][periodic(ix,n_spins,-1)]+
            spin_matrix[periodic(iy,n_spins,-1)][ix]
```

```
               +
           spin_matrix[iy][periodic(ix,n_spins,1)] +
           spin_matrix[periodic(iy,n_spins,1)][ix]);
           // Here we perform the Metropolis test
               if ( ran1(&idum) <= w[deltaE+8] ) {
     babymatrixcounter += 1;
               spin_matrix[iy][ix] *= -1;
               // flip one spin and accept new spin
                   config
               // update energy and magnetization
               M += (double) 2*spin_matrix[iy][ix];
               E += (double) deltaE;
               }
           }
       }
   if(babymatrixcounter>0){
   //no spin flips does not update the total matrix
       counter
   matrixcounter += 1;
   }
}
```

FIG. 2: Metropolis algorithm. The algorithm flips the spin of one state at random to search for a lower energy state. We also apply a matrix counter to count the number of different matrices built by the metropolis algorithm.

```
for (double temp = initial_temp; temp <=
    final_temp; temp+=temp_step){
    // initialise energy and magnetization
    matrixcounter = 0;
    E = M = 0.;
    // setup array for possible energy changes
    for( int de =-8; de <= 8; de++) w[de+8] = 0;
    for( int de =-8; de <= 8; de+=4) w[de+8] =
        exp(-de/temp);
    // initialise array for expectation values
    for( int i = 0; i < 5; i++) average[i] = 0.;
    initialize(n_spins, temp, spin_matrix, E, M);
double normal = mcs;
double spins = n_spins*n_spins;
    // start Monte Carlo computation
    for (int cycles = 1; cycles <= mcs; cycles++){
   if(cycles == 1) int matrixcounter = 0;
       Metropolis(n_spins, idum, spin_matrix, E,
           M, w, matrixcounter);
       // update expectation values
       average[0] += E/normal; average[1] +=
           E*E/normal;
       average[2] += M/normal; average[3] +=
           M*M/normal; average[4] +=
           fabs(M/normal);
    }
```

FIG. 3: Segment of the main function, which displays the inclusion of the metropolis algorithm. This segment runs over a specified range of temperatures and calculated the mean energy and the mean magnetization.

ANALYSIS

2 x 2 case

Expanding upon theory, we look into the simple case of the 2x2 lattice, with only two spins in each dimension, so L=2.

$$\begin{bmatrix} \uparrow\downarrow & \uparrow\downarrow \\ \uparrow\downarrow & \uparrow\downarrow \end{bmatrix}$$

In a simple 2x2 matrix, a superposition all of the possible states is shown, in practice, there is going to be either an ↑ or a ↓ in each position of the matrix.

Total Spins Up	Degeneracy	Energy	Magnetization
4	1	-8J	4
3	4	0	2
2	4	0	0
2	2	8J	0
1	4	0	-2
0	1	-8J	-4

TABLE I: From the standard energy, evaluated in a 2x2 case. This demonstrates that most of the system configurations are constrained into degenerate cases of energies being (-8J,0,8J)

From these energy states, and enforcing the periodic boundary conditions, the partition function comes to:

$$Z = 12e^{\beta(0)} + 2e^{\beta(-8J)} + 2e^{-\beta(8J)} \tag{17}$$

which simplifies to:

$$12 + 4cosh(8\beta J) \tag{18}$$

We find $\langle E \rangle$:

$$\langle E \rangle = \frac{-32Jsinh(8\beta J)}{12 + 4cosh(8\beta J)} \tag{19}$$

and $\langle E^2 \rangle$:

$$\langle E^2 \rangle = \frac{1024J^2 sinh^2(8\beta J)}{(12 + 4cosh(8\beta J))^2} \tag{20}$$

We find $\langle |M| \rangle$:

$$\langle |M| \rangle = \frac{8e^{\beta J} + 16}{12 + 4cosh(8\beta J)} \tag{21}$$

and $\langle M^2 \rangle$:

$$\langle M^2 \rangle = \frac{32(1 + e^{8\beta J})}{12 + 4coshh(8\beta J)} \tag{22}$$

Knowing this, calculating out C_v:

$$C_v = \frac{\beta[64J^2(3cosh(8\beta J) + 1)]}{T(cosh(8\beta J) + 3)^2} \tag{23}$$

and χ:

$$\chi = \frac{\beta 4(e^{-8\beta J} + 3e^{8\beta J} + 3)}{(cosh(8\beta J) + 3)^2} \tag{24}$$

Finding the Most Likely State

Moving on to a two dimensional square lattice L=20, we are looking into how many Monte Carlo cycles it takes for the system to reach the most likely state. In order to do this, we study the number of Monte Carlo sweeps of the lattice it takes to reach equilibrium and compute various expectation values for plotting. We do with with both ordered and random spin matrices, measuring both T = 1.0 and T = 2.4.

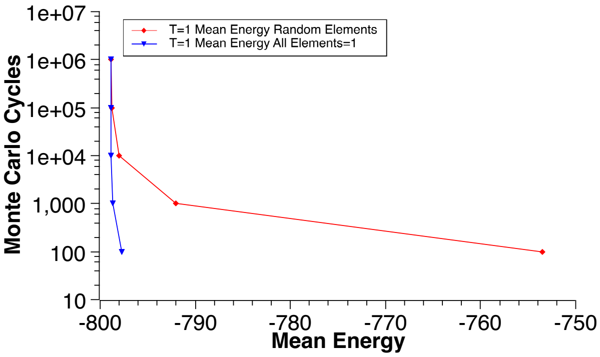

FIG. 4: Monte Carlo cycles vs. the mean energy. The blue is the mean energy for an ordered initial matrix - all matrix elements set to 1 - for the temperature T=1. The red is the mean energy for a random initial matrix for the temperature T=1.

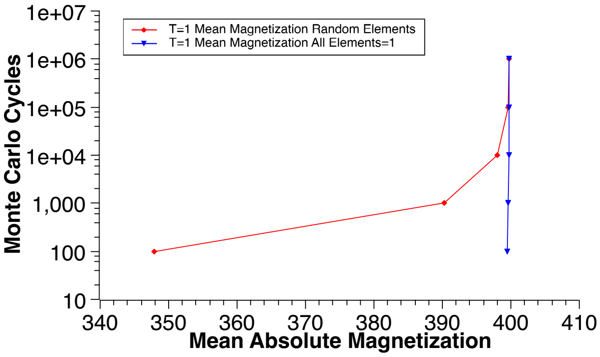

FIG. 5: Monte Carlo cycles vs. the mean absolute magnetization. The blue is the mean absolute magnetization for an ordered initial matrix - all matrix elements set to 1 - for the temperature T=1. The red is the mean absolute magnetization for a random initial matrix for the temperature T=1.

Starting with T=1, we computed both the expectation value of the energy (figure 4) and the absolute magnetization (figure 5) with respect to the number of Monte Carlo cycles. We computed these with both an ordered and random initial matrix, where an ordered matrix has all elements of one, or spin up. Both of these figures show that with an ordered initial matrix the mean energy and absolute magnetization was continually consistent with the expected value, -800 for energy and 400 for magnetization. The random initial matrix provided a poor approximation with a low number of Monte Carlo cycles, but the approximation improved as the Monte Carlo cycles increased towards one million cycles. At the point near one million cycles the random inital matrix produced an identical result as the ordered initial matrix.

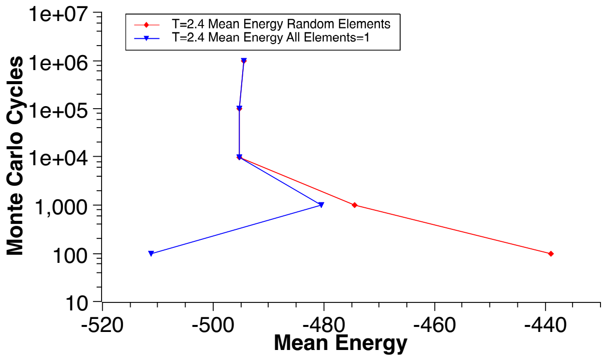

FIG. 6: Monte Carlo cycles vs. the mean energy. The blue is the mean energy for an ordered initial matrix - all matrix elements set to 1 - for the temperature T=2.4. The red is the mean energy for a random initial matrix for the temperature T=2.4.

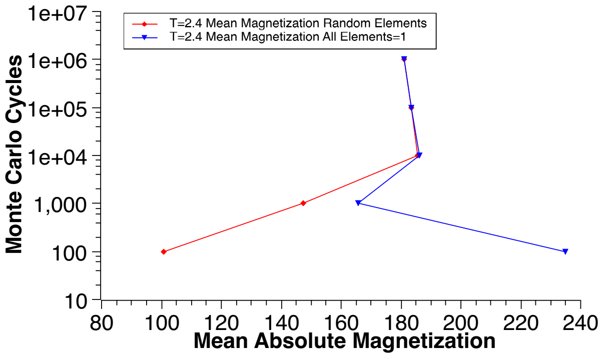

FIG. 7: Monte Carlo cycles vs. the mean absolute magnetization. The blue is the mean absolute magnetization for an ordered initial matrix - all matrix elements set to 1 - for the temperature T=2.4. The red is the mean absolute magnetization for a random initial matrix for the temperature T=2.4.

With T=2.4, we observed an jump in the ordered initial matrix for both the mean energy (figure 6) and the mean absolute magnetization (figure 5). The random initial matrix trends smoothly and obtains similar results as the ordered initial matrix at ten thousand Monte Carlo cycles. At that point, both means and both initial matrices behave in almost identical (but opposite for the means) manners. For the mean energy, the two matrices converge to an energy of approximately -495. For the mean magnetization, the two matrices converge to a value of approximately 180.

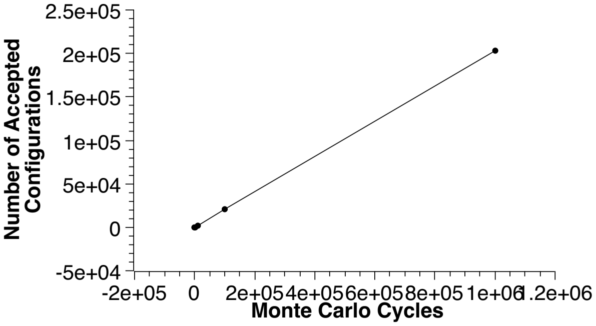

FIG. 8: Monte Carlo cycles vs. the number of accepted configurations for temperature equal to 1 and for a random initial matrix. The behavior of the number of number of accepted configurations with respect to the number of monte carlo cycles is exactly linear. The plot for an ordered initial matrix is the same and would overlap is plotted together.

Another interesting test that we were able to perform is to calculate the number of different matrices produced by the metropolis algorithm to converge to the values that we obtained. Figure 8 displays how the number of accepted configurations has a linear dependence on the number of Monte Carlo cycles. This graph was produced for the condition of the temperature equal to one and for both an ordered and random initial matrix, with both initial matrices producing the same curve.

Probability Distribution

We computed the probabilities for the 20x20 system to take certain values for its total energy. We did this at two temperatures: T=1 and T=2.4. The measurement of the counts for each energy was taken after the system was at a steady state; so we set the grid to all the same value and ran 10^5 Monte Carlo cycles from there.

Two plots are included in figure 9 and figure 10 showing the distribution of the energies, and the most likely state. Table 3 has also been included, showing the probabilities for each energy value.

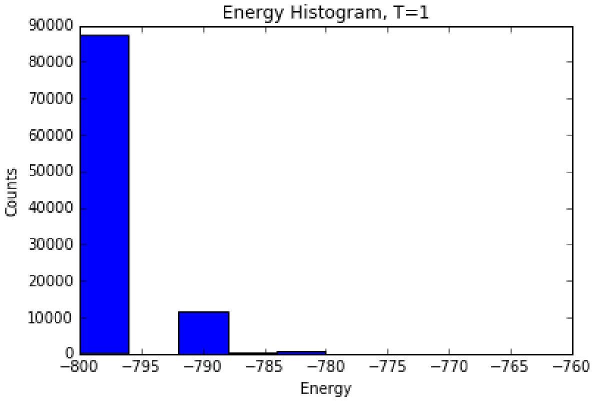

FIG. 9: After 10^5 mc cycles for a 20x20 lattice at T=1, the most likely total energy for the system is easily -800.

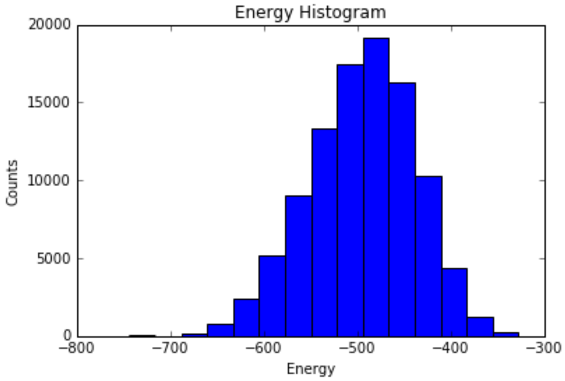

FIG. 10: After 10^5 mc cycles for a 20x20 lattice at T=2.4, the most likely energy state is near -522, falling off on either side. Thermal fluctuations cause the lattice of spins to take on a more disordered state.

Numerical Look at Phase Transitions

An area of interest for the Ising model in two dimensions is close to the critical temperature. We are looking at it as a function of lattice size, with temperature steps $\delta T = .02$.

Using a range of temperatures of T=2.1 − 2.4 and a range of matrix sizes L=20, 40, 60, and 80, the expectation value of the energies (figure 11), the expectation value of the absolute magnetizations (figure 12), the specific heats (figure 13), and the susceptibilities (figure 14) were found.

The latter three figures adequately display how the functions are dependent on the critical temperature. The plot of the expectation value of the absolute magnetization tends to drop towards an asymptote as the matrix

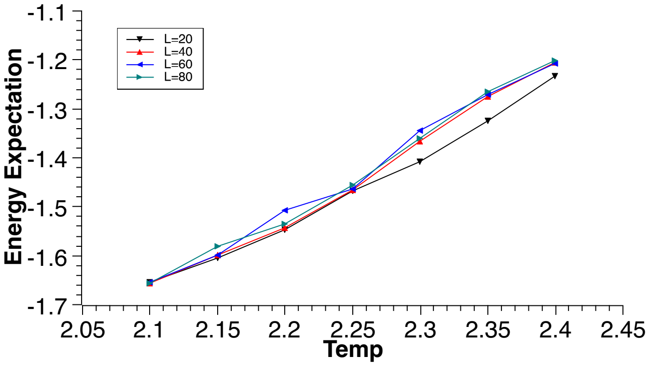

FIG. 11: Expectation value of energy vs. temperature. The black is for a matrix of size 20, the red is for a matrix of size 40, the blue is for a matrix of size 60, and the dark cyan is for a matrix of size 80. All for matrix sizes tend to behave linearly in an identical fashion.

size gets larger. Similarly, the plots of the specific heat and susceptibility show peaks, or spikes, that move towards a critical temperature as the matrix size increases.

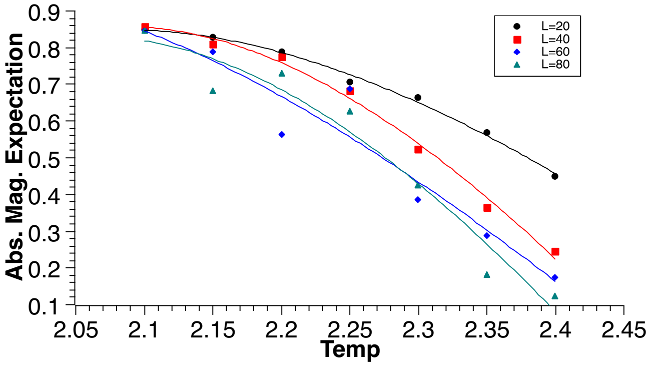

FIG. 12: Expectation value of the average magnetization vs. temperature. The black is for a matrix of size 20, the red is for a matrix of size 40, the blue is for a matrix of size 60, and the dark cyan is for a matrix of size 80. The data points were fit with functions of the form x^{-2} to display their behavior. As the matrix size increases, the behavior of the data points tends to trend towards a point, the critical temperature.

Extracting the Critical Temperature

Using equation 13, with the exact result that $\nu = 1$, we are estimating T_c in the thermodynamic limit of $L \to \infty$. For reference, the exact result for critical temperature with our specified result is $\frac{kT_C}{J} = \frac{2}{ln(1+\sqrt{2})} \approx 2.269$ [2]

By running our program for the matrix sizes of L=40, 80, 100, and 140, we can approximate how the algorithm will behave in the limit of $L \to \infty$. As shown in figure 15, the expectation value of the absolute magnetization was found with respect to temperature for these matrix sizes. The plot for these values for each of the matrix sizes reveals that there is a temperature, the critical temperature, in which the functions drip dramatically. We found that this temperature was at 2.281, which is very similar to the value of 2.269 [2].

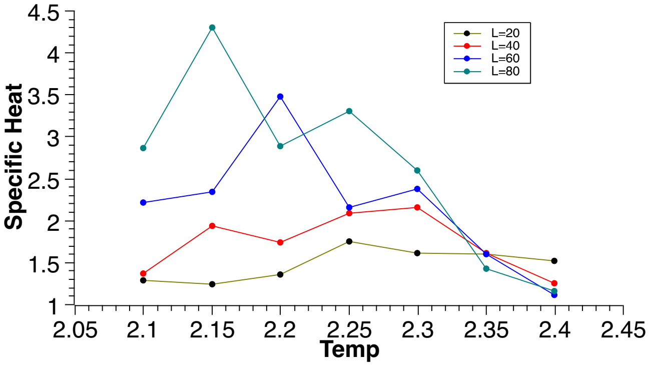

FIG. 13: Specific heat vs. temperature. The black is for a matrix of size 20, the red is for a matrix of size 40, the blue is for a matrix of size 60, and the dark cyan is for a matrix of size 80. As the size of the matrix increases, the spike in the specific heat tends to get larger, thinner, and move towards the top left of the graph. These spikes tend to be reaching a critical point, similar to figure 12.

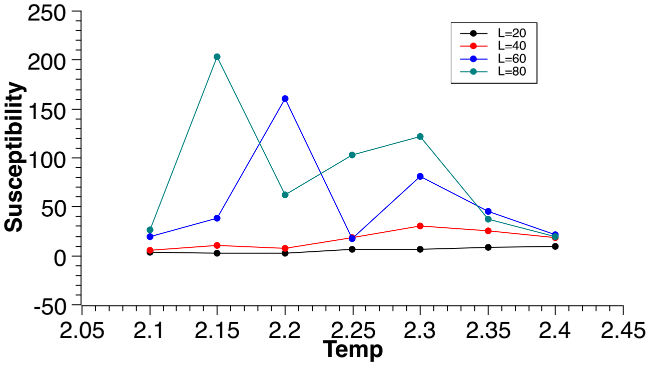

FIG. 14: Susceptibility vs. temperature. The black is for a matrix of size 20, the red is for a matrix of size 40, the blue is for a matrix of size 60, and the dark cyan is for a matrix of size 80. This graph appears to have similar properties to those of figure 13. The spikes in the susceptibility tend to be reaching a critical point.

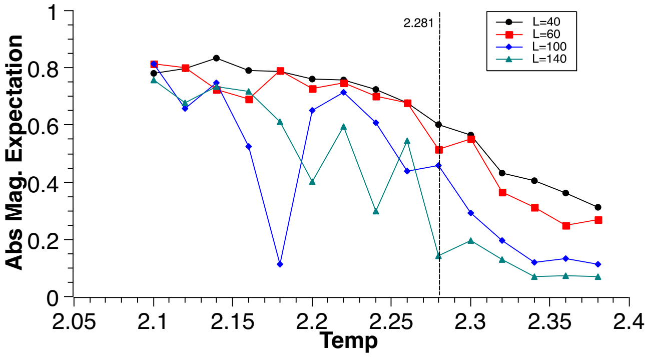

FIG. 15: Expectation value of the absolute magnetization vs. temperature. The black is for a matrix of size 40, the red is for a matrix of size 60, the blue is for a matrix of size 100, and the dark cyan is for a matrix of size 140. The data tends to drop, as expected, at the point of about 2.281, marked with a dashed line in the figure.

RESULTS AND DISCUSSION

Through implementation of our code for the Ising model, we test our algorithm on a 2x2 lattice with respect to the analytical solutions. Assuming that T= 1.0, and operating in temperature units of $(\frac{kT}{J})$ so our values come out to those displayed in table 3.

Values	Analytical	10 M.C.	10^3 M.C.	10^6 M.C.
$\langle E\rangle$	-7.9839	-8	-7.976	-7.9836
$\langle E^2\rangle$	63.7431	64	63.808	63.8691
$\langle \lvert M\rvert\rangle$	3.9946	4	3.992	3.9945
$\langle M^2\rangle$	15.9572	16	15.96	15.9726
C_v	0.1283	0	0.1914	0.1306
χ	0.01604	0	0.0239	0.01673

TABLE II: As demonstrated here, when 10^6 Monte Carlo cycles run we find significant agreement with the analytical results.

Our results agreed with our intuition; A higher number of Monte Carlo cycles resulted in closer approximations of our calculated quantities to the analytical expressions shown in the Theory section. Ten Monte Carlo cycles shows a very crude picture of our 2x2 system, results which can be shown to be true via drawings on paper for a frozen system.

More cycles give a better approximation of these quantities' averages, which deviate from the values seen after ten cycles due to more chances for thermal fluctuations to produces states with different energies and magnetizations. Once we saw the agreement of our calculations with our analysis, we felt comfortable using our code on larger lattices.

Using a matrix size of L=20 we found that the mean energy converges to -800 for both random and ordered initial matrices for a temperature of one. We also found that the mean average magnetization for both initial matrices converges to 400. Agreement between the two initial matrices provides verification of the mean energy and mean absolute magnetization for this temperature along with further verification of the values in table 3.

We also found that the same initial matrices, but for a temperature of 2.4, converges to an energy of approximately -495 and a magnetization of approximately 180. While the ordered matrix appears to have a spike in the data for both cases, both initial matrices appear to converge at a higher number of Monte Carlo cycles.

For the temperature of 1, the mean energy and mean absolute magnetization converges at one hundred thousand Monte Carlo cycles. For the temperature of 2.4, the mean energy and mean absolute magnetization converges at ten thousand Monte Carlo cycles. Lower temperatures cause the system to be less likely to flip a spin; the random value between 0 and 1 is compared to a number that decreases with temperature. Therefore, it is in agreement with theory that it would take longer for a system to reach a low energy state at T=1 than it would for T=2.4.

The number of accepted matrix configurations was also determined to be linearly dependent on the number of Monte Carlo cycles. This holds true for both ordered and random initial matrices. The number of accepted configurations was tested for both initial matrices by running the same parameters to test for agreement in the numbers. For the ordered matrix we received the same number of accepted configurations each time we ran it and for the random matrix we received numbers of accepted configurations that agreed within a certain error dependent on the number of Monte Carlo cycles, which was what we would expect. These tests provide verification of the linear dependence of figure 8.

The probability distributions of energies for T=1 and T=2.4 in initially ordered matrices met our expectations as well. In Figure 10, the majority of the energies remain at -800, the lowest acceptable energy for a 20x20 lattice. At such a low temperature, it is unlikely that any spins would flip to create a higher energy for the system.

In Figure 11, we attribute the wider distribution of energy values to the higher temperature. It is harder for the system to fall into its lowest energy state if the chance for spins to change configurations is more likely. We also examined the variance in the energy, given by:

$$\sigma^2 = (\langle E^2\rangle - \langle E\rangle^2) \tag{25}$$

In our C++ code and using numpy's method "var", we got perfect agreement. T=1 gave us a value of 9.20236 (most probable value -800) and T=2.4 gave us 3303.24 (most probable value -566).

Temp	Energies	Probabilities
1	-800	87.16%
1	-792	11.55%
1	-788	0.041%
2.4	-566	18.99%
2.4	-522	28.39%
2.4	-477	26.43%

TABLE III: The most three probable total energy values for the two systems and their probabilities are shown. It should be noted that for a 20x20 lattice, the smallest value the total energy can take is -800, when all spins point in the same direction.

Using large sized matrices we were able to find how different expectation values depend on temperature. In specific, we looked at the extreme of $L \rightarrow \infty$ to determine the critical temperature. Computing for a matrix of L=140, we were able to determine that the critical temperature was 2.281, which is in close agreement to the accepted value of 2.269 [2].

CONCLUSIONS AND OUTLOOK

Agreements in the analytic and simulation data - displayed in table 3 - provides verification of the presented algorithms and the results presented in this work. The presented algorithms have allowed for the testing of the number of Monte Carlo cycles needed to converge on various expectation values, how the number of Monte Carlo cycles affects the number of different matrices produced during the metropolis algorithm, the probability distribution of the of the energy, and the identification of the critical temperature.

Along with agreements between the analytic and simulation data, we were also able to determine the critical temperature of 2.281, which is in close agreement to the value of 2.269 present by [2]. Computation using larger matrices would likely produce a more accurate critical temperature and provide verification of the presented and accepted critical temperatures.

For future considerations, to improve the accuracy, we may increase the number of nearest neighbors included in the models and calculations.To expand upon what was covered here, we can explore the possibility of updating large clusters of spins near the critical point. Inspiration from this comes from a 1989 paper by Ulli Wolff exploring a novel Monte Carlo algorithm involving percolation theory to collectively update spin systems [3].

ACKNOWLEDGEMENTS

The authors would like to thank Morten for teaching us how to do C++ stuff. I would also like to thank my mom.

[1] "Project4," https://compphysics.github.io/ComputationalPhysicsMSU/doc/Projects/2017/Project4/IsingModel/pdf/IsingModel.pdf, accessed: 2017-04-29.

[2] L. Onsager, Phys. Rev. **65**, 117 (1944).

[3] U. Wolff, Phys. Rev. lett. **62**, 361 (1989).