Computationally Solving the $2D$ Ising Model with the Metropolis Algorithm

Tahoe Schrader
PHYS566

Abstract

In this paper I depict the results of computationally exploring the $2D$ Ising Model for various lattices with an interaction strength of $J = 1.5$ and the Boltzmann constant set to one, solely using Python. Each lattice was cooled from $T = 5 \to 0$. For $n = 50$, the magnetization as a function of temperature was plotted to determine a critical temperature of $T_c = 2.65$. For $n = [5, 10, 20, 30, 40, 50, 75, 100, 200, 500]$, the maximum specific heat per lattice size as a function of temperature was plotted to determine the finite-size scaling relation, $C_{\max}/N \approx \log(n)$. The data matches the theory for magnetization measurements and specific heat. Nonetheless, the finite-size scaling relation, $C_{\max}/N \approx \log(n)$, does not fit perfectly for either large or small n.

1 Theory

The Ising model is one of the few models of a system with a phase transition that can be treated analytically. In 1925, it was introduced to describe the phase transitions of ferromagnets at the curie temperature. Nowadays, it is used to describe much physical phenomena, such as order-disorder transitions in binary alloys, spin glass, and certain neural network classes.

Lattices that obey the Ising model have periodic boundary conditions. In $1D$ and $2D$, the lattice resembles a circle or toroid, respectively. The Hamiltonian takes the form:

$$H_N(\sigma_1, \ldots, \sigma_N) = -J \sum_{\text{n.n}} \sigma_i \sigma_k - \mu B \sum_{i=1}^{N} \sigma_i, \tag{1}$$

where μ is the magnetic moment, σ is spin up or down, $N = n^2$ is the dimension of the lattice, J is the interaction strength, and n.n refers to a sum across nearest neighbors. I have chosen to study the case with no external magnetic field, which causes the second term in Equation 1 to fall out. From now on, I will refer to the Hamiltonian as the energy of the system.

1.1 Specific Heat and Magnetization

The energy of the system is important because a large number of parameters can be solved for once the energy is obtained. An in-depth derivation of these parameters can be found in any statistical mechanics textbook. The important thing to note here is that by obtaining the partition function from the energy, the free energy can be derived. The free energy can then be manipulated to derive any thermodynamic property.

The thermodynamic property studied in this paper is specific heat. Specific heat is given by the equation:

$$C = \frac{(\Delta E)^2}{k_B T^2}, \tag{2}$$

where $(\Delta E)^2$ is determined computationally during a sweep through temperatures. The $(\Delta E)^2$ parameter will be discussed more thoroughly in Section 1.2.

The second major property studied in this paper is the magnetization of the system. Magnetization depends on the over-arching spin alignment of the lattice. Individual spins in the lattice can have a value of $+1$ (spin up) or -1 (spin down). Magnetization per spin is given by,

$$M_{\text{per spin}} = \langle \sigma \rangle = \frac{1}{N} \sum_i \sigma_i, \tag{3}$$

and the total magnetization is given by,

$$M_{\text{total}} = N \langle \sigma \rangle = \sum_i \sigma_i. \tag{4}$$

The reason why I am focusing on specific heat and magnetization is that by juxtaposing the two against one another, a phase transition can be proven to occur. This phase transition occurs at the critical temperature, T_c, exhibited by a step function in magnetization and a peak in specific heat.

Lastly, the Ising model can be shown to obey a finite-size scaling relation, given by

$$\frac{C_{\text{max}}}{N} \approx \log(n), \tag{5}$$

where n is simply the length of the lattice. This implies that although an infinite lattice will have an infinite specific heat, all finite lattices have a realistically measurable specific heat that scales by $N * \log(n)$. This relation is tested in Section 3.

1.2 Determining $(\Delta E)^2$

The starting point of investigating the $2D$ Ising model is a lattice of spins that can only assume two directions with respect to the $\hat{z}$-axis. These two values were described above to be $\sigma_i = \pm 1$. Energy is determined by Equation 1 with no external field to be:

$$E = -J \sum_{\text{n.n}} \sigma_i \sigma_k. \tag{6}$$

This sum is over all nearest neighbor pairs. Note that when computing this type of sum for a $2D$ lattice, each pair is double counted. Therefore, the nearest neighbor sum is divided by two.

Given N_m total microstates, the energy expectation value is

$$\langle E \rangle = \frac{1}{N_m} \sum_\alpha E_\alpha. \tag{7}$$

The energy squared expectation value is

$$\langle E^2 \rangle = \frac{1}{N_m} \sum_\alpha E_\alpha^2. \tag{8}$$

Using Equation 2 with

$$(\Delta E)^2 = \langle E^2 \rangle - \langle E \rangle^2 \tag{9}$$

will then give the specific heat of the system at a given temperature.

2 Code

Actually obtaining the parameters above in Python is quite difficult due to the level of randomness a heat bath adds to the system. Each individual spin in the lattice has an interaction strength of $J = 1.5$. Because $J > 0$, the system favors a parallel alignment of spin ($\sigma = +1$). In the absence of a heat bath, the system evolves into a fully aligned state, either spin up or down. With a heat bath, the alignment is governed by the probability

$$P_\alpha \approx e^{-E_\alpha/k_B T}, \tag{10}$$

where α is an individual microstate.

Flipping spins according to Equation 10 is a process known as the Metropolis Algorithm or Markov Chain analysis. The logic needed to run the Metropolis Algorithm is enumerated below:

1. Initialize the system by assigning random spin orientations to each lattice point and compute the initial energy using Equation 6.
2. Relax system by:
   (a) Instigate a hypothetical spin flip at a random location, and calculate the hypothetical change in energy this flip would cause.
   (b) Use the Metropolis algorithm to decide if the spin flip is kept.
   (c) If kept, update the lattice and energy by adding the hypothetical change.
   (d) If not kept, do not update the lattice or the energy.
   (e) Repeat 2a-2d $10,000$ times.
3. Measure and store E and M. This is the first microstate.
4. Measure and store another N_m microstates of energy and magnetization by:
   (a) Repeat 2a-2d 100 times. This helps give a statistically significant measurement since nearby measurements are heavily correlated.
   (b) Store energy and magnetization.
   (c) Repeat 4a-4b N_m times. I chose $N_m = 1,000$
5. Find $\langle E\rangle$, $\langle E^2\rangle, (\Delta E)^2$.
6. Average M
7. Repeat steps 2-6 for each temperature.
8. Repeat 1-7 for each lattice size.

This process was done for a number of different temperatures from $T = 0 \to 5$ for each lattice with dimensions $n = [5, 10, 20, 30, 40, 50, 75, 100, 200, 500]$. The Metropolis algorithm itself says:

1. If $\Delta E \leq 0$, keep the spin.
2. If $\Delta E > 0$:
   (a) Pick a random number, r, in the range $[0, 1]$.
   (b) If $r \leq e^{-\Delta E/k_B T}$, keep the spin.
   (c) Otherwise, do not keep the spin.

3 Results

It is expected that at a high initial temperature, the magnetization of a system is at around zero. After cooling the lattice down, the magnetization should fully align to $M = n^2$. Note that plotting $\frac{|M|}{N}$, instead, will show the spins to fully align at +1 and remove some noise. Noise is removed because there is a degeneracy in the magnetization measurements near T_c. This degeneracy says that there is an equal probability for M measurements to be negative. So, plotting the absolute value of magnetization clears up this degeneracy. My data for $n = 50$ is given in Figure 1.

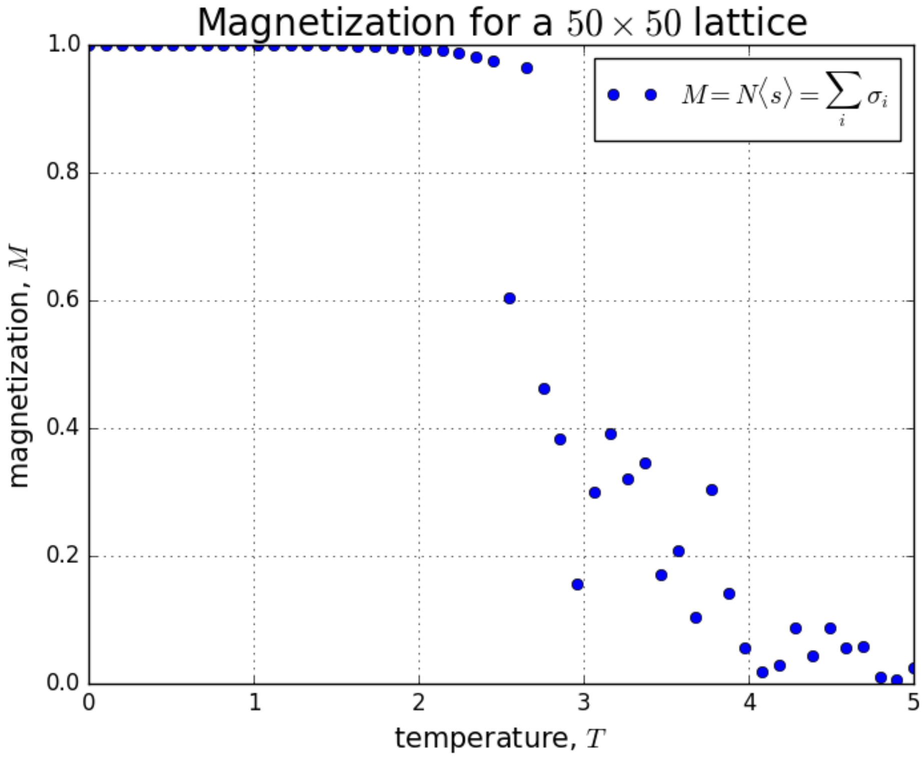

Figure 1: The magnetization of a $2D$ system, divided by N, as a function of temperature. At $T = 5$, the magnetization is zero. At $T = 0$, it is near 1. The critical temperature, $T_c = 2.65$, is found by noting the location of the step function at $M = 0.5$, by eye.

When a lower temperature resolution is used, the data becomes much worse. This can be seen in Figure 2. I used this temperature resolution for the rest of my calculations, which may explain some slight deviations from what is expected from the theory.

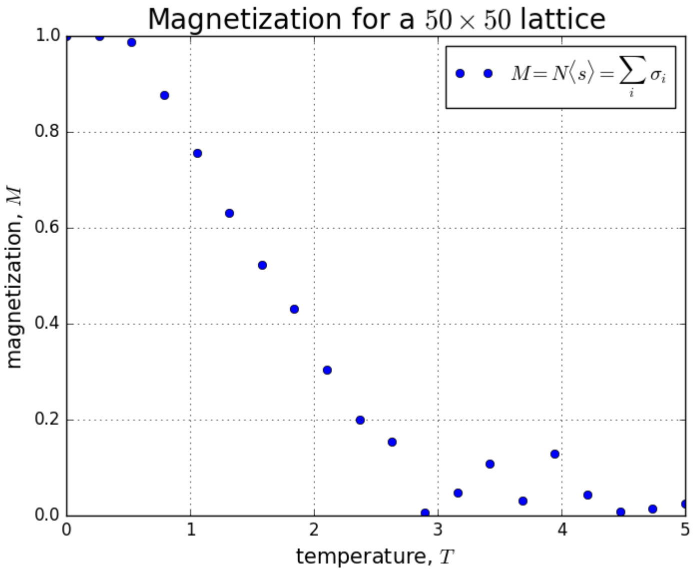

Figure 2: The magnetization of a $2D$ system, divided by N, as a function of temperature. At $T = 5$, the magnetization is zero. At $T = 0$, it is near 1. Due to the low temperature resolution, the results are much worse than those displayed in Figure 1

The specific heat for each lattice is expected to resemble a curve with a peak near its critical temperature. This peak location is called C_{max}. A couple example plots of specific heat are given in Figures 3-5.

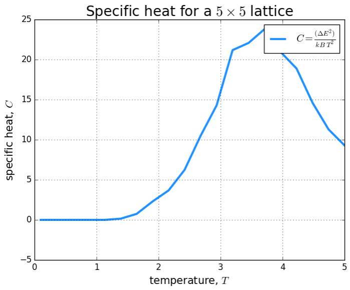

Figure 3: C as a function of T for $n = 5$.

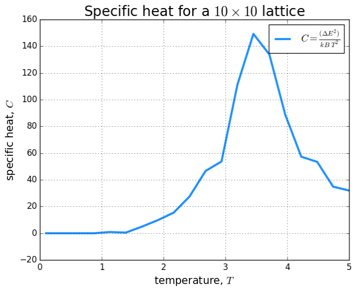

Figure 4: C as a function of T for $n = 10$.

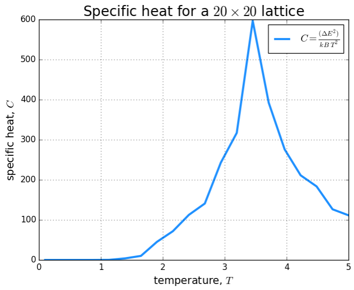

Figure 5: C as a function of T for $n = 20$.

Lastly, the maximum of each specific heat function was obtained and divided by N in order to obtain the finite-size scaling relation. A plot of this relation is given in Figure 6. The fit does not work well for large values of n. One possible reason for this is that larger values of n need a longer relaxation time. This may be because the code sees the lattice as "infinite" with the given relaxation time of $10,000$ flips. Due to time constraints, I was not able to run the code for a longer relaxation time in order to test this. The data could have also been improved by using a finer temperature resolution. Again, due to time constraints, I was only able to solve for specific heat across 20 temperatures from $T = 5 \rightarrow 0$.

Zooming in to Figure 6, shows what the data looks like for small n. These results are shown in Figures 7 and 8.

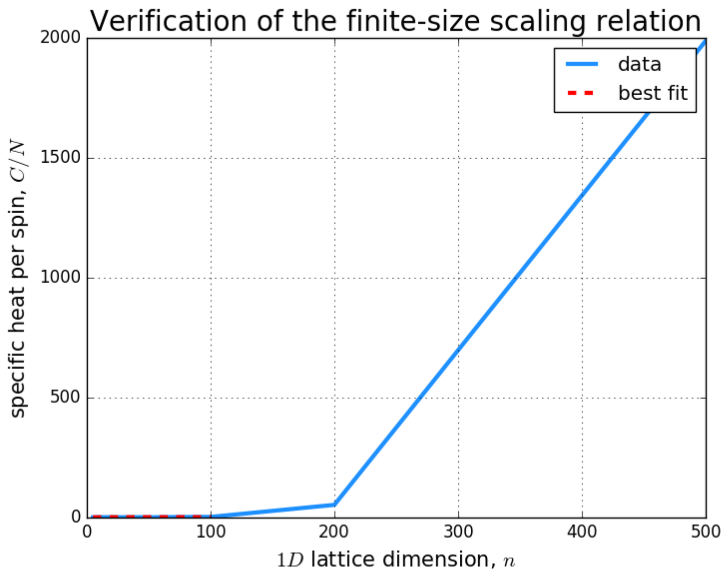

Figure 6: The scaling relation for $C_{\max}/N$ is depicted for $n = [5, 10, 20, 30, 40, 50, 75, 100, 200, 500]$. The data should resemble $\log(n)$, as determined by a fit, but it clearly does not for large values of n.

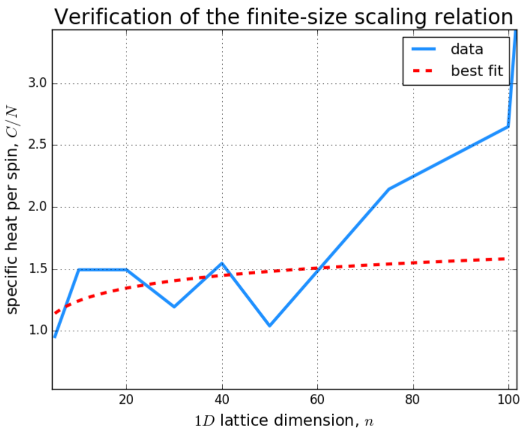

Figure 7: The scaling relation for $C_{\max}/N$ is depicted for $n = [5, 10, 20, 30, 40, 50, 75, 100]$. The line of best-fit does not match the data very well. A number of different factors could have caused this, such as a low relaxation time and low temperature resolution.

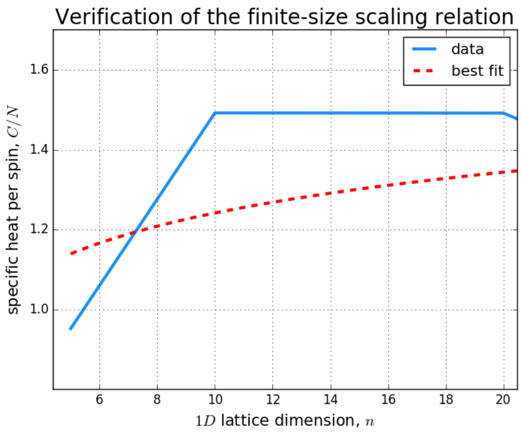

Figure 8: The scaling relation for $C_{\max}/N$ is depicted for $n = [5, 10, 20]$. The line of best-fit does not match the data very well, because it includes $n = [30, 40, 50, 75, 100]$, which skewed the results. Due to time constraints, I was not able to refit the results for fewer data points.