PHYS 5640 - Project 1

Steven Stetzler

March 14, 2018

Metropolis and Wolff Algorithm for the Square Lattice Ising Model

We explore the Ising Model on a square lattice with periodic boundary conditions using the Metropolis algorithm and the Wollf clustering algorithm. These two algorithms provide a way to simulate and predict the energy E and heat capacitance of this system C as a function of the temperature T of the system using Monte Carlo simulations. We also explore how the efficiency of these two algorithms behaves as a function of system temperature T.

The Ising Model on a square lattice consists of a set of Ising spins ($\uparrow$, $\downarrow$) on a 2D (XY) lattice. This lattice has periodic boundary conditions which impose that $x + L_x = x$ and $y + L_y = y$, where L_x is the size of the system in the x dimension and L_y is the size of the system in the y dimension. In total, there exist $N = L_x \times L_y$ spins in this system. Each spin interacts with its nearest neighbors with the strength of this interaction parameterized by the spin exchange interaction parameter J. For a ferromagnetic interaction, $J > 0$. These conditions give rise to the Hamiltonian of the system which is

$$\mathcal{H} = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j \tag{1}$$

where σ_i and σ_j represent the value of the spin ($\uparrow = +1$ and $\downarrow = -1$) at the location i and j. $\langle ij \rangle$ signifies a nearest neighbors pair with the sum running over all sets of nearest neighbors in the system. The average energy of the system is given by the average value of the Hamiltonian

$$E = \langle \mathcal{H} \rangle, \tag{2}$$

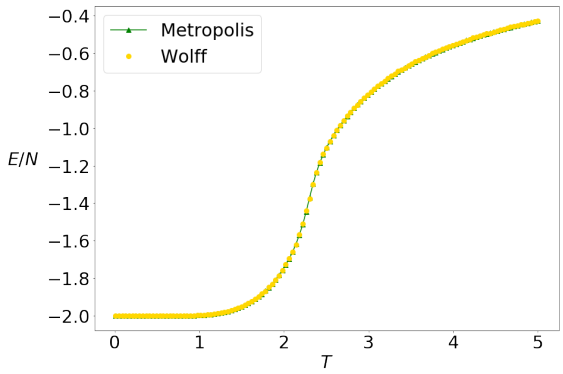
and the heat capacity is given by the variance of the Hamiltonian per spin:

$$C/N = \beta^2 \left(\langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2 \right). \tag{3}$$

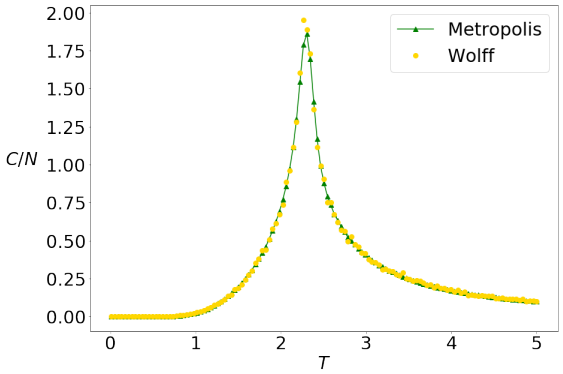
where $\beta = 1/T$. In both of these definitions $\langle ... \rangle$ signifies the sample average of the quantity of interest in the Monte Carlo simulation.

In both algorithms, in order to compute the relevant quantities E and C at a temperature T, we simply construct lattices that are consistent with being drawn from the probability distribution $exp\left(-\beta E\right)$, which is the Boltzmann distribution. The way that the algorithms differ is the way in which they construct these consistent lattices.

In the metropolis algorithm, a lattice is constructed by considering a series of spin flips at local sites on the lattice. First, one chooses a single site i on the lattice at random. Then, one computes the energy of the lattice using Eq. 1, giving an energy E_i. A spin flip is then considered at site i; either $\uparrow$ will convert to $\downarrow$ or vice versa. The energy of the lattice is computed again using Eq. 1, giving an energy E_f. The difference in energy $\Delta E = E_f - E_i$ is computed, and the spin flip is accepted as a permanent change with probability $\exp\left(-\beta \Delta E\right)$ and reversed with probability $1 - \exp\left(-\beta \Delta E\right)$. In following this procedure, we are able to construct lattices who's spins are arranged according to the Boltzmann distribution. In practice, and in our coded implementation, we first construct a completely randomized lattice and then, at a given temperature T, we "thermalize" the system by sweeping through the system and proposing a spin flip at every location in the lattice 10,000 times. Once this is done, we are guaranteed that new proposed spin flips will be producing spin configurations consistent with the Boltzmann distribution at the temperature we are interested in. After thermalizing the system, we then compute the energy of 100,000 lattices drawn



(a) The average energy per spin for the two simulation types.



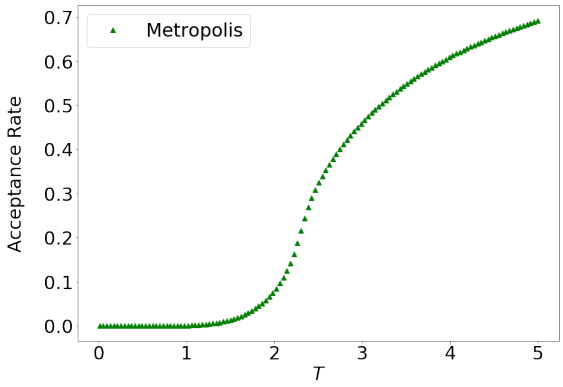
(b) The average heat capacitance per spin for the two simulation types.

Figure 1: Results of the per-spin energy (left) and the heat capacitance (right) from the Metropolis and Wolff simulations. Results from both are plotted overlayed on one another.

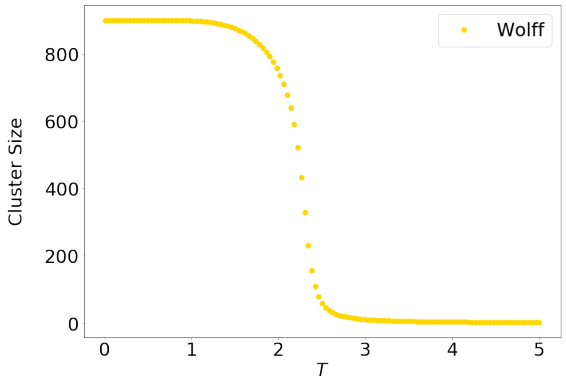
from the Boltzmann distribution, where each lattice is constructed independently. To construct independent lattices, we simply sweep through the system and consider 150 spin flips at every location in the lattice to separate and make independent any two lattices for which we would compute an energy E. These 150 sweep separated lattices should be independent draws from the Boltzmann distribution. Finally, using the 100,000 lattice draws, we compute E/N and C/N using Eq. 2 and Eq. 3. For $N = 30 \times 30$ and $J = 1$ at various values of $T \in (0, 5]$, we follow the above procedure to construct the plots of E/N and C/N shown in Fig. 1. The acceptance rate, simply the number of accepted spin flips divided by the total number of proposed spin flips, is plotted as a function of T in Fig. 2a.

The Wolff algorithm works differently than the Metropolis algorithm, as in a single iteration, the Wolff algorithm attempts to flip entire clusters of similarly pointed spins within the lattice. The procedure for the Wolff algorithm follows. First, pick a random location in the lattice to seed the cluster. Next, consider the neighbors of each point in the cluster. If the neighbor spin points in the same direction as the current spin under consideration, and the neighbor spin is not already in the cluster then include that lattice point in the cluster with probability $1 - \exp(-2\beta J)$ and exclude it with probability $\exp(-2\beta J)$. Once the cluster stops growing, all of the spins within the cluster are flipped at once. In doing this, the Wolff algorithm is able to make larger, non-local, changes to the lattice, allowing it to hopefully converge to a lattice consistent with the Boltzmann distribution faster than the metropolis algorithm can. To compute E/N and C/N at a given temperature T, we follow a similar procedure as described in the metropolis algorithm of thermalizing the system and then computing the energy of independent lattices. The only difference is that instead of considering a spin flip at a point in the lattice, we are starting the seeding process for a cluster at that point. In our implementation, for $N = 30 \times 30$ and $J = 1$ at several temperatures $T \in (0, 5]$ we sweep through the system 50 times to thermalize it from an initially randomized lattice configuration, then produce 2,000 independent lattices, each separated by 20 sweeps through the system. Using the values of the energies of the 2,000 lattices, we compute E/N using Eq. 2 and C/N using Eq. 3 at a temperature T. The values of E/N and C/N are plotted as a function of temperature in Fig. 1. The average cluster size produced at each of the temperatures considered is shown in Fig. 2b as well.

In observing Fig. 1, we can see that the results between the Metropolis algorithm and the Wolff algorithm are completely consistent. We can also see clearly the phase transition present in this system as it approaches smaller T. At small T, the system gains order, with all of the spins pointing in the same direction, achieving an energy per spin of $E/N = -2$. However, at high T, the order is lost and the energy escapes the ground state as more and more spins tend to be anti-aligned. We note that this phase transition is present as well in the plots of the acceptance rate for the Metropolis algorithm (Fig. 2a) and the cluster sizes produced by the Wolff algorithm (Fig. 2b). At low T, the Metropolis algorithm almost never succeeds in flipping a spin, indicative of the fact that spins will want to be aligned at low T and it is extremely unfavorable to anti-align them. However, as T increases, the probability of having a spin flip increases dramatically, indicative of the



(a) The average acceptance rate of the Metropolis algorithm.



(b) The average cluster size produced during each iteration of the Wolff algorithm.

Figure 2: The acceptance rate (left) of the Metropolis algorithm and the average cluster size (right) produced by the Wolff algorithm, both as a function of the temperature of the system.

spread of disorder in the system. This is also reflected in the cluster size produced by the Wolff algorithm. At low T, all of the spins point in the same direction, and they are likely to cluster together, producing an average cluster size of ~ 900, which is the number of spins in the system. As T increases, it become much less likely that spins will want to align, and thus unlikely that a cluster will grow beyond a single spin, shown by the decrease in cluster size at high T. These results produced by analyzing the phase transition are consistent with the behavior of a ferromagnetic material.