Monte Carlo Markov Chains

Implementations of MCMC algorithms
on the Ising Model

Benjamin Nguyen

February 7, 2019

1 What is the Ising Model?

A motivating question posed is what happens to a magnet as temperature changes? If a magnet were to be deposited into a furnace at high temperature and retrieved after some time, or conversely, if a magnet were deposited into a freezer at low temperature, what would happen to its magnetic properties?

The Ising Model is a model that tries to capture the properties of interacting particle systems, namely in ferromagnetic materials. The model allows us to study topics such as energy, magnetization, and phase transitions.

We would like to describe the Ising Model within a mathematical context so that we may more closely examine its features. However, depending on our approach, the difficulty of the procedure may vary. In the next few sections, the exact solution for the Ising Model will be sketched, followed by an easier approach using Monte Carlo methods.

1.1 Defining the Ising Model

The Ising Model is defined by the finite graph $G = (V, E)$. Let σ_i denote the spin value at vertex i on the lattice such that $\sigma_i \in \{-1, 1\}$. Let $(\sigma_1, \sigma_2, ..., \sigma_{|V|}) = \Lambda_i$ denote a configuration of the lattice in the set of all possible configurations of the graph $\Lambda = \{-1, 1\}^{|V|}$.

Let $\{i, j\} \in E$ denote the set of all neighboring vertices j that are connected to vertex i. The definition of a neighboring vertex is determined by the parameterization of the boundary conditions of the graph; the use of periodic boundary conditions will be used in this case, which wraps the graph around a torus. Occasionally, the neighboring relation will be described by the notation $i \sim j$.

The energy function is defined by the Hamiltonian function

$$H(\sigma) = -J \sum_{\{i,j\} \in E} \sigma_i \sigma_j - h \sum_{v \in V} \sigma_j \tag{1}$$

We may take a simplified version of the Hamiltonian by setting the coupling parameter J = 1 and setting the external field parameter $h = 0$. These parameters may be initialized to such values since they are only constant shifts in the simplified Hamiltonian, which becomes

$$H(\sigma) = -\sum_{\{i,j\}\in E} \sigma_i \sigma_j \tag{2}$$

At finite inverse temperature parameter $\beta = \frac{1}{k_b T}$ with Boltzmann constant k_b, we can define the probability measure

$$P_\beta(\sigma) = \frac{exp(\beta H(\sigma))}{\sum_\Lambda exp(\beta)} \tag{3}$$

The normalizing constant is often denoted as $Z(\beta)$ and called the partition function for the Ising model. It is summed over all possible $\{-1, 1\}^{|V|} = 2^{|V|}$ configurations. In practice, we consider finite graphs, but the calculation of the partition function can become computationally intractable even for small number of vertices. For example, a 5x5 configuration with 25 vertices would generate $2^{25} = 33,554,432$ possible different configurations. In the programming language R, the allocation of this many configurations into memory tally up to a total of 6.3 gigabytes, which is considerably large for a fairly uninteresting sized lattice[1]. Even if we could allocate enough memory so that we may take independently and identically distributed samples from the population of all configurations, we would run into the issue of drawing uncorrelated samples, which is of no interest on averaged quantities in thermodynamic equilibrium. We need to approach the problem of sampling from the Ising Model in a way that removes the necessity to compute the partition function and produce correlated samples that will allow us to make measurements of interest.

2 Monte Carlo Methods

We want to construct a method that will sample from the stationary distribution, which is a long run stable thermodynamic equilibrium for the system that allows us to characterize the proportion of time a system will spend in any given configuration. The framework of Markov chains will allow us to analyze the Ising Model in a well-defined probabilistic scheme and sample from the stationary distribution without the complexity of evaluating the normalizing constant in equation (3).

3 Ergodicity and Detailed Balance Conditions

To implement MCMC methods, we need to verify at least two conditions that guarantee convergence to equilibrium.

[1] For a nicely sized lattice, say 100 by 100, R would attempt to allocate approximately 1,070,000,000,000,000,000,000 TB in memory to store all the possible configurations.

Ergodicity is a criteria that is met when any configuration along the sample space S can be accessed from any other configuration through a sequence of flips on the lattice. The Markov chain for the Ising Model will be ergodic so long as we permit that all vertices on the lattice may be selected as a candidate to be flipped.

A stationary distribution π is said to satisfy detailed balance conditions if $\pi(x)P(x,y) = \pi(y)P(y,x)$ for $x, y \in S$. To frame the detailed balance conditions in the context of the Ising Model, let the configurations in consideration be denoted as σ and σ^v, where the latter configuration differs from σ by one flip on the lattice for simplicity.

Then the detailed balance conditions may be expressed as

$$\pi(\sigma)P(\sigma,\sigma^v) = \pi(\sigma^v)P(\sigma^v,\sigma) \tag{4}$$

Finally, we check that the detailed balance conditions hold.

$$\begin{aligned} \pi(\sigma)P(\sigma,\sigma^v) &= \pi(\sigma^v)P(\sigma^v,\sigma) \\ \pi(\sigma)Q(\sigma,\sigma^v)r(\sigma,\sigma^v) &= \pi(\sigma^v)Q(\sigma^v,\sigma)r(\sigma^v,\sigma) \\ \pi(\sigma)r(\sigma,\sigma^v) &= \pi(\sigma^v)r(\sigma^v,\sigma) \end{aligned} \tag{5}$$

The first line denotes the usual detailed balance conditions. The second line occurs by decomposing the transition probability $P(\sigma,\sigma^v)$ into a selection probability $Q(\sigma,\sigma^v)$ and an acceptance probability $r(\sigma,\sigma^v)$. The third line arises by noting that the selection probability is uniform, so that they cancel each other out.

$$r(\sigma,\sigma^v) = \frac{\pi(\sigma^v)r(\sigma^v,\sigma)}{\pi(\sigma)} \tag{6}$$

By solving for an expression for the transition probability $r(\sigma,\sigma^v)$ in terms of the right hand side of the detailed balance conditions, we can substitute the expression into the left hand side to confirm that equality holds.

$$\begin{aligned} \pi(\sigma)r(\sigma,\sigma^v) &= \pi(\sigma^v)r(\sigma^v,\sigma) \\ \pi(\sigma)\frac{\pi(\sigma^v)r(\sigma^v,\sigma)}{\pi(\sigma)} &= \pi(\sigma^v)r(\sigma^v,\sigma) \\ \pi(\sigma^v)r(\sigma^v,\sigma) &= \pi(\sigma^v)r(\sigma^v,\sigma) \end{aligned} \tag{7}$$

The importance in the detailed balance conditions is that it permits the system to reach an equilibrium state. It essentially states that if we were to observe a system over time, it will reach equilibrium and stay in steady state with no significant change so long as all the parameters of the system are held fixed. In other words, if a system begins at stationary distribution $\pi(\sigma)$ and attempts to become a configuration σ^v, it should be the same as if the system were to begin at stationary distribution $\pi(\sigma^v)$ and attempt to become a configuration σ.

3.1 Describing Detailed Balance for Ising Model

It has been shown that the detailed balance conditions hold. However, the expressions for the stationary distribution, selection probability, and acceptance probability were generalized. In this section, I will more explicitly define these probability measures so that the model may be practically implemented.

In statistical mechanics, the stationary distribution follows the Boltzmann distribution. For a configuration σ, the Boltzmann distribution is $e^{-\beta H(\sigma)}$ so we have that $\pi(\sigma) \sim \frac{1}{Z}e^{-\beta H(\sigma)}$ for arbitrary configuration σ and normalizing constant $Z = \sum\limits_{\sigma \in S} e^{-\beta H(\sigma)}$ on the state space S.

The transition probability $P(\sigma, \sigma^v)$ may be decomposed into a selection probability Q and an acceptance probability r.

$$P(\sigma, \sigma^v) = Q(\sigma, \sigma^v) r(\sigma, \sigma^v) \tag{8}$$

Similar to a hill-climbing algorithm in optimization, we would like to avoid completely deterministic procedures in which we only accept better outcomes as to avoid becoming stuck in non-optimal regions. This motivates our probabilistic expression for the acceptance probability which strictly accepts better outcomes and occasionally accepts worse outcomes:

$$r(\sigma, \sigma^v) = \min\left\{1, \frac{Q(\sigma^v, \sigma)\pi(\sigma^v)}{Q(\sigma, \sigma^v)\pi(\sigma)}\right\} \tag{9}$$

Since the candidate site flip is chosen uniformly across the lattice, the selection probability Q is the same for all vertices on the lattice and cancels out in calculation. The acceptance probability simplifies to the symmetric Metropolis algorithm

$$r(\sigma, \sigma^v) = \min\left\{1, \frac{\pi(\sigma^v)}{\pi(\sigma)}\right\} \tag{10}$$

We have seen before that the stationary distribution $\pi(\sigma)$ follows the Boltzmann distribution. The ratio of the stationary distributions at configuration σ^v and σ may be expressed as

$$\frac{\pi(\sigma^v)}{\pi(\sigma)} = \frac{\frac{1}{Z}e^{-\beta H(\sigma^v)}}{\frac{1}{Z}e^{-\beta H(\sigma)}} \tag{11}$$

As it turns out, we do not have to trouble ourselves with the computation of the normalizing constant Z, as they cancel each other out in computation. Furthermore, the single-flip algorithm will affect the total energy of the lattice only through the neighbors of the flipped vertex, as seen through the Hamiltonian function $H(\sigma)$. The Hamiltonian with the flipped spin $H(\sigma^v)$ will introduce a negation of the spin in the original Hamiltonian $H(\sigma)$, so that the ratio can be expressed as

$$\frac{\pi(\sigma^v)}{\pi(\sigma)} = \frac{e^{-\beta \sum\limits_{i \sim j} -\sigma_i \sigma_j}}{e^{-\beta \sum\limits_{i \sim j} \sigma_i \sigma_j}} = e^{-2\beta \sum\limits_{i \sim j} -\sigma_i \sigma_j} \tag{12}$$

In the case of neighboring sites on a two-dimensional lattice, there are four candidate neighbors for a vertex provided that we have periodic boundary conditions. Among the neighbors, we are interested in the set of neighbors that have spins aligned with the flipped spin, since this corresponds to a reduction in energy in the system. The set that describes the number of neighbors that may agree with the flipped vertex is $\{0, 1, 2, 3, 4\}$.

Defining a function $K(\sigma, v)$, which counts the number of neighbors in agreement with the flipped vertex v on the configuration σ, we can simplify the ratio into the following reduced form :

$$\begin{aligned} \frac{\pi(\sigma^v)}{\pi(\sigma)} &= e^{-2\beta(K(\sigma,v)-(4-K(\sigma,v))} \\ &= e^{-2\beta(2K(\sigma,v)-4)} \\ &= e^{-4\beta(K(\sigma,v)-2)} \end{aligned} \tag{13}$$

The first line is the Boltzmann distribution with a new expression for the Hamiltonian function, which is counting the number neighbors which agree with the flipped site value and subtracting away the number of neighbors which disagree with it. The second and third line follow from algebraic simplifications. Concisely, the acceptance probability may be expressed as $\min\{1, e^{-4\beta(K(\sigma,v)-2)}\}$. [5]

Alternatively, we may re-express the Hamiltonian in a way that uses an indicator function that checks whether or not neighboring spins have the same alignment as the flipped spin.

$$I(\sigma_i = \sigma_j) = \left\{ \begin{array}{ll} 1, & \text{if } \sigma_i = \sigma_j \\ 0, & \text{if } \sigma_i \neq \sigma_j \end{array} \right\} \tag{14}$$

We want to produce an energy function that assigns decrease in energy to more fully aligned configurations between a selected site and its neighbors; conversely it should assign an increase in energy to less aligned configurations. One possible re-expression is

$$H(\sigma) = 2 - 2 \sum_{i \sim j} I(\sigma_i = \sigma_j) \tag{15}$$

In general, as long as we assign lower energy to aligned configurations and higher energy to unaligned configurations, the expressions for the Hamiltonian function may be numerous. Its choice of presentation may be chosen to robustly define the Ising Model or may be chosen to improve computational efficiency in simulation.

4 Temperature Parameterization

Since the Ising Model is characterized by the temperature parameter, it is worth discussing the probabilistic behavior at the lowest and highest limits of temperature. As seen before, the probability measure of the Ising Model may be expressed as

$$P_{\beta}(\sigma) = \frac{exp(\beta H(\sigma))}{\sum_{\Lambda} exp(\beta)}$$

In the infinitely high temperature regime ($\beta = 0$), the stationary distribution behaves as a uniform measure on $\{-1, 1\}^{|V|}$ for finitely many vertices, i.e.

$$P_{\beta}(\sigma) = \frac{1}{\{-1, 1\}^{|V|}} \tag{16}$$

In the infinitely low temperature regime ($\beta = \infty$), the stationary distribution is partitioned into

$$P_{\beta}(\sigma) = \frac{1}{2}\sigma_{(+)} + \frac{1}{2}\sigma_{(-)} \tag{17}$$

where $\sigma_{(\pm)}$ denotes a configuration that is totally aligned in a spin up or spin down configuration. Equation (17) captures the behavior of the Ising Model such that the configuration will be totally aligned at the limit of lowest temperatures, so that it does not consider intermediate configurations.

5 Glauber Dynamics and Metropolis Algorithm

Now that the detailed balance conditions have been verified to hold, the algorithm may be stated with the assurance that the end result will converge to equilibrium by sampling from the stationary distribution. The following MCMC algorithm is a general algorithm that may be applied on many applications that may be modeled as a Markov chain. Here is its implementation in the case of the Ising Model:

Algorithm 1 Glauber Dynamics

- 1 Initialize a lattice of size n x m with spins $\sigma_i \in \{\pm 1\}$.
- 2 Uniformly sample a random vertex from the lattice σ_{smp}.
- 3 Find the neighboring spins of the sampled spin. Compute the Hamiltonian function using the site values of the sampled spin and its four neighbors. Flip the sampled spin value and then re-compute the Hamiltonian function.
- 4 Compute the Boltzmann distribution using the difference in energy ΔE. If $\Delta E < 0$, then update the lattice by keeping the spin flipped; otherwise check if $r = unif(0, 1) \leq e^{\beta \Delta E}$. If $r \leq e^{\beta \Delta E}$, then update the lattice by keeping the spin flipped, otherwise un-flip the spin and retain the original configuration.
- 5 Repeat steps 2-4 for some desired amount of simulations.

5.1 Glauber Dynamics Lattice Simulation

The graphical output of the lattice under simulation using Glauber Dynamics will be shown for the low, critical, and higher temperature regimes, along with discussion about features of the plots.

5.2 Low Temperature

In the low temperature regime, the number of MC steps required to show interesting developments on the lattice is considerably large relative to higher temperature regimes. As time progresses forward, the lattice has a tendency to form clusters in the same alignment. Conceivably, the lattice would become more magnetized in a single direction given sufficiently more time to develop.

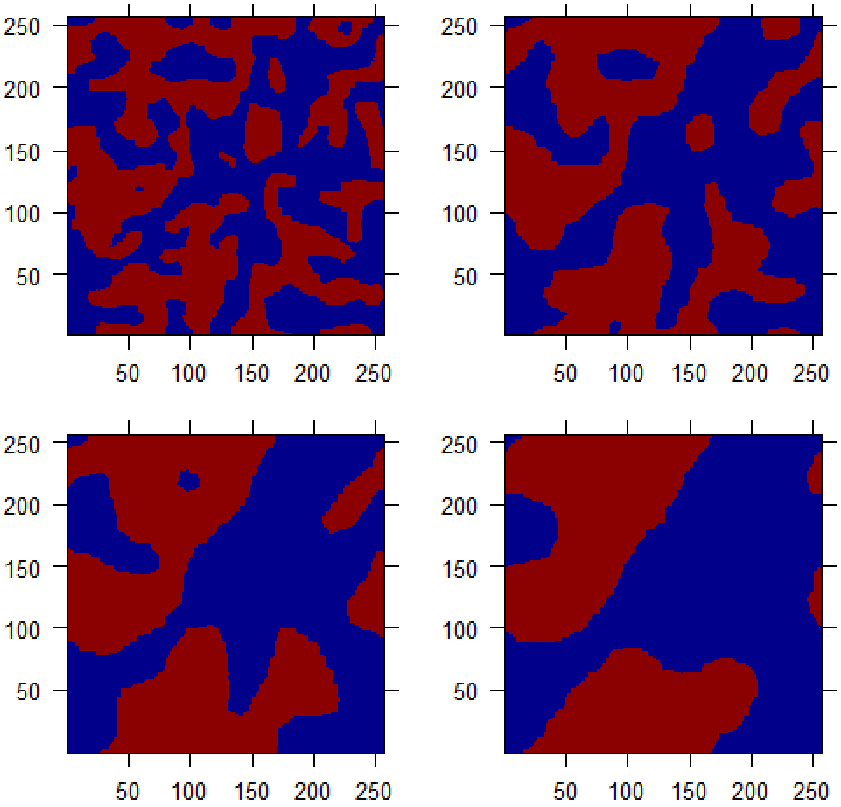


Figure 1. The four plots were generated on a 250 by 250 lattice at a temperature of 0.5. The number of iterations required to generate each plot were 0.5x10^6, 1.5x10^7, 3.0x10^7, and 1.0x10^8 respectively for the plots seen from left to right and top to bottom.

5.3 Critical Temperature

The behavior of the lattice near the critical temperature characterizes the process of critical slowing down. The lattice is at a junction in time where the lattice could potentially choose one alignment over the other provided that the temperature is incrementally decreased below T_c, e.g. temperature approaches zero. If the temperature is kept nearby the critical temperature, its behavior can be seen in figure 2, where we can see that the alignment of the lattice is approximately split in terms of total magnetization.

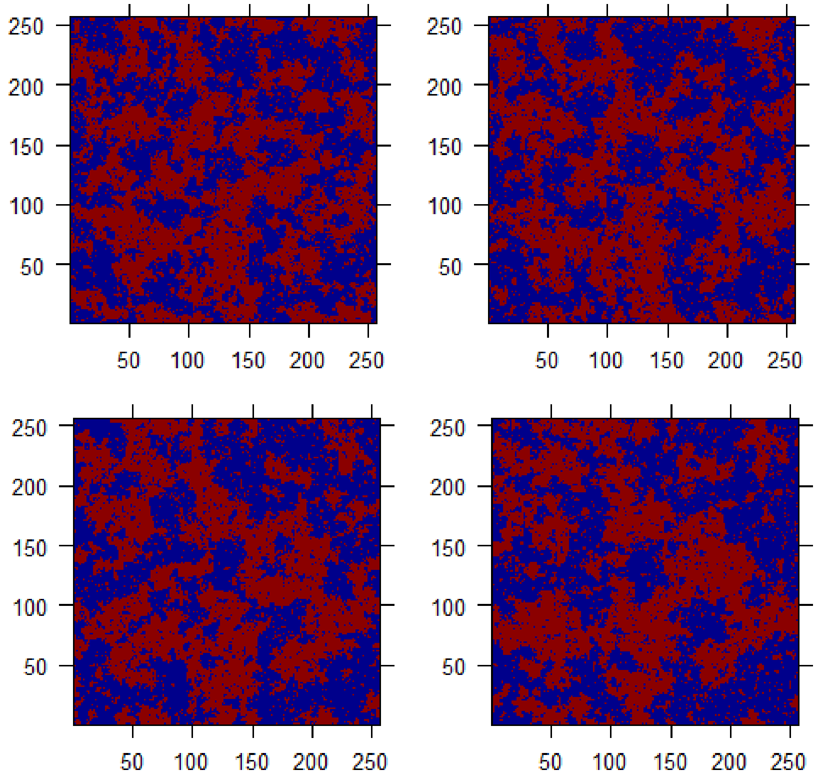


Figure 2. The four plots were generated on a 250 by 250 lattice near the critical temperature $T_c \simeq 2.26$. The number of iterations to produce the plots were 0.5x10^6, 1.0x10^6, 1.5x10^6, and 2.0x10^6 respectively.

5.4 High Temperature Regime

In the high temperature regime, the probability measure was observed to be a uniform measure on $\{-1, 1\}^{|V|}$. The implication is that the vertices of the lattice may be thought to be independent of each other; in other words, the site value σ_v can be thought of as having independent value from a neighboring site ν with site value σ_ν. There is no preference by Glauber Dynamics to align the lattice in an up or down spin configuration, so the lattice remains completely disordered, which may be observed in figure 3.

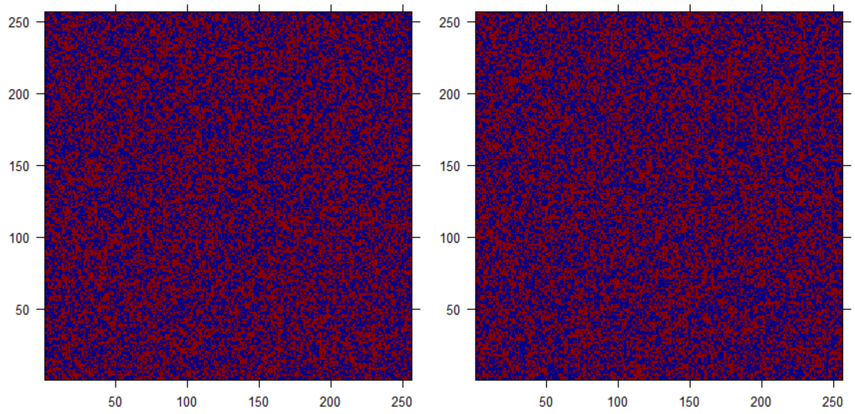


Figure 3. The two plots were generated at a temperature of 10 on a 250 by 250 lattice. The number of iterations required to produce the plots were 5.0x10^6 and 3.0x10^6 respectively. Even after many iterations, there is little distinction between the two plots.

A concise way to describe these last two plots is by noting that the behavior of ferromagnetic materials under high temperature has the tendency to lose their magnetic potency. Measuring the total magnetization of these plots would result in recorded values $\simeq 0$.

5.5 Glauber Dynamics Sampling Simulation

To verify that the algorithm is continuing to sample from the stationary distribution, we can produce time series data from the data generating process of the tail end of the simulation. By inspecting this data, we would hope to see that the algorithm is still making acceptances and rejections of flipping single sites.

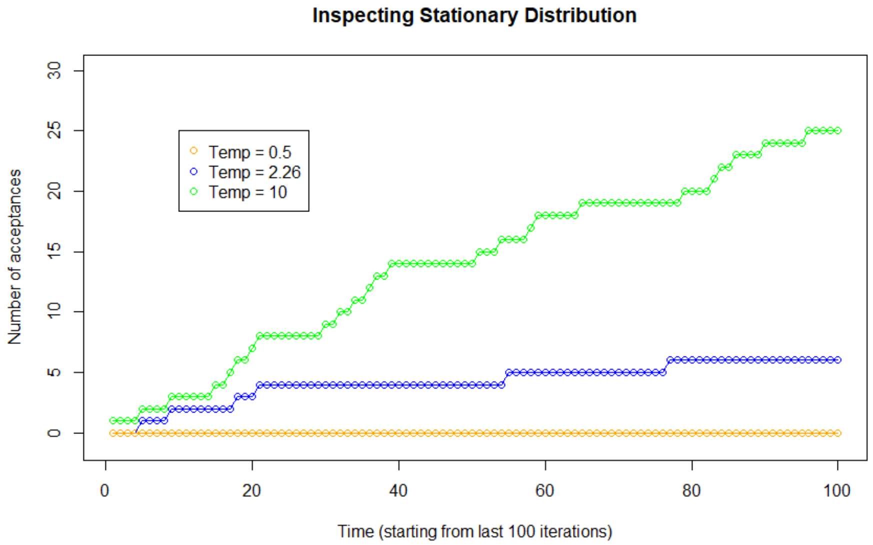


Figure 4. Plot of total number of acceptances over the course of the last hundred iterations on a 50 by 50 lattice using Glauber Dynamics [2]

When the temperature is low, the simulation will produce larger clusters of spins in the same alignment. Once it has reached total alignment of the lattice, it will have low susceptibility to undergoing spin flips. When the temperature is at critical temperature, the mechanism occasionally permits acceptances and rejections of site flips. When the temperature is high, the probability of flipping the site is high; there are many acceptances of site flips even at the tail end of the simulation. These traits may be observed in figure 4. The choice of using a 50 by 50 lattice is to facilitate making observations on systems that have nearly converged; it takes a considerably long time for low temperature regime systems to converge, especially on larger lattice structures.

6 Non-Local Cluster Algorithm [3]

The Wolff Algorithm was introduced by Uli Wolff in his 1989 paper titled "Collective Monte Carlo Updating for Spin Systems", which built upon the Swedsens-Wang non-local clustering algorithm. Before implementing the algorithm, we need to verify that it also guarantees convergence to equilibrium.

[2] The choice of using a 50 by 50 lattice is to facilitate making observations on systems that have nearly converged; it takes a considerably long time for low temperature regime systems to converge, especially on larger lattice structures, so the choice of a smaller lattice structure is taken to be as remedial action to spend less time waiting on making observations on the stationary distribution. The measurements may be taken on larger lattice structures, but it is uncertain about when these systems will converge.

7 Detailed Balance Conditions for Wolff Algorithm

To assure ourselves that the Wolff Algorithm will sample from the stationary distribution, we have to check for detailed balance conditions.

To begin, suppose we have configurations σ and σ^v which differ by a cluster flip. Then the selection probability $Q(\sigma, \sigma^v)$ is proportional to the the probability of selecting the same cluster and omitting spins outside of the cluster with the same alignment. Letting m and n denote the number of bonds that did not become incorporated into the cluster, we see that

$$\begin{aligned} Q(\sigma, \sigma^v) &= (1 - P_{add})^m \\ Q(\sigma^v, \sigma) &= (1 - P_{add})^n \end{aligned} \tag{18}$$

Following the usual detailed balance conditions,

$$\begin{aligned} \pi(\sigma)P(\sigma, \sigma^v &= \pi(\sigma^v)P(\sigma^v, \sigma) \\ \frac{\pi(\sigma)}{\pi(\sigma^v)} &= \frac{P(\sigma^v, \sigma)}{P(\sigma, \sigma^v} \\ &= \frac{Q(\sigma^v, \sigma)r(\sigma^v, \sigma)}{Q(\sigma, \sigma^v)r(\sigma, \sigma^v)} \\ &= (1 - P_{add})^{n-m}\frac{r(\sigma^v, \sigma)}{r(\sigma, \sigma^v)} \\ &= e^{-\beta(E_{\sigma^v} - E_\sigma)} \end{aligned} \tag{19}$$

By permitting the energy difference $(E_{\sigma^v} - E_\sigma) = 2J(n-m)$, we can simplify the detailed balance conditions through a sequence of rearrangements

$$\begin{aligned} (1 - P_{add})^{n-m}\frac{r(\sigma^v, \sigma)}{r(\sigma, \sigma^v)} &= e^{-\beta(E_{\sigma^v} - E_\sigma)} \\ \frac{r(\sigma^v, \sigma)}{r(\sigma, \sigma^v)} &= (1 - P_{add})^{m-n}e^{-\beta(E_{\sigma^v} - E_\sigma)} \\ &= (1 - P_{add})^{m-n}e^{-2\beta J(n-m)} \\ &= (1 - P_{add})^{m-n}e^{2\beta J(m-n)} \\ &= [(1 - P_{add})e^{2\beta J}]^{m-n} \end{aligned} \tag{20}$$

The detailed balance conditions will be satisfied by setting $P_{add} = 1 - e^{-2\beta J}$, which simplifies the acceptance ratio to be equal to one. That means to have detailed balance conditions satisfied for the Wolff Algorithm, the cluster flips may be done with complete certainty, i.e. with probability one.

7.1 Wolff Algorithm

Now that we have verified that the Wolff Algorithm satisfies the detailed balance conditions, we may outline the structure of the Wolff Algorithm:

Algorithm 2 Wolff Algorithm

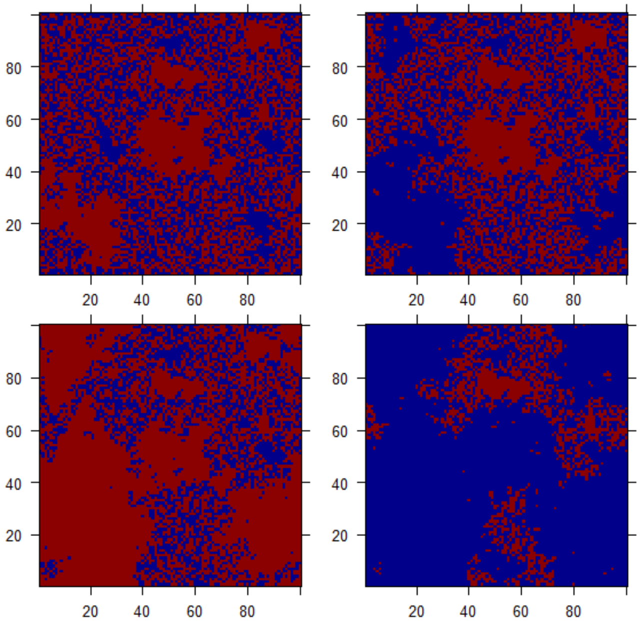
```
• 1 Initialize a lattice of size $n$ x $m$ with spins $\sigma_i \in \{\pm 1\}$.
• 2 Uniformly sample a random spin from the lattice $\sigma_{smp}$ to be the first element of a cluster that will be built.
• 3 Inspect the neighboring spins of $\sigma_{smp}$ and determine which neighbors have the same spin. Among these neighbors, add it to the cluster with probability $P_{add} = 1 - e^{2\beta J}$.
• 4 For each neighboring site adjoined to the cluster, we repeat the previous step until no more sites are added to the cluster.
• 5 Flip the spin of the cluster.
• 6 Repeat the steps 2-5 for some desired amount of simulations.
```

8 Wolff Simulation

The graphical output of the lattice under simulation using the Wolff Algorithm will be shown for the low, critical, and higher temperature regimes, along with discussion about features of the plots.

8.1 Low Temperature

Figure 5. The four plots below were generated by the Wolff Algorithm on a lattice size 100 by 100 at temperature 0.5. The plots were generated from iterations 16, 17, 21, and 22 respectively.



The tendency for large clusters to form on the lattice becomes greater as the temperature parameter approaches zero. The likelihood of the cluster being re-selected by the algorithm increases proportionally with the cluster size, which characterizes the behavior of oscillating alignments of spins of formed clusters. In figure 5, the Wolff Algorithm can be observed selecting a large cluster and flipping the spin value of the entire cluster in one step; which occurs in iterations 16-17 and 21-22 respectively.

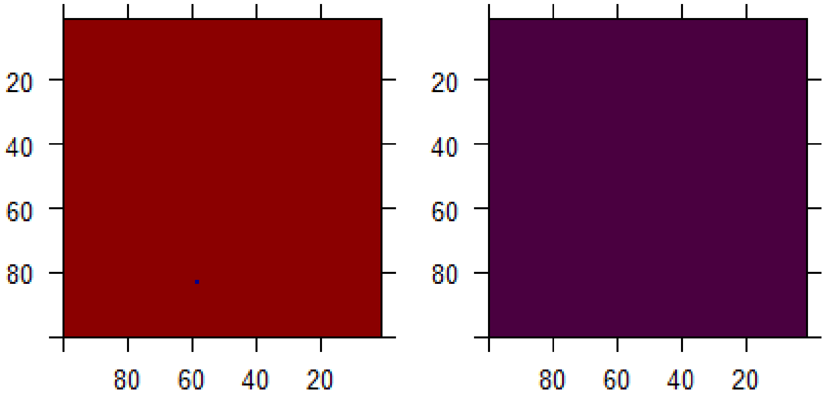


Figure 6. Two consecutive Wolff Steps; continuation from the chains generated by the Algorithm in Figure 5. The cluster size is (nearly) the whole lattice size in one iteration; the cluster is selected and flipped, causing the whole lattice to be aligned.

Given that the algorithm runs for a longer duration at lower temperatures, an individual cluster may take on the size of the entire lattice and additional steps in the algorithm will lead to total oscillation in spins of the lattice in each of the following Wolff steps. This phenomena can be seen in figure 6. It can also be seen in figure 11 (presented in later sections), by the rapid oscillating magnetization between ±1, which is characterized by the Wolff Algorithm collecting the whole lattice as the cluster and flipping the cluster's spin value.

8.2 Critical Temperature

Using the critical temperature parameter, the size of the clusters are not as extreme as the low temperature variety since the granularity of the lattice is far from uniform, causing the candidate sites to be added into a cluster to be less pronounced.

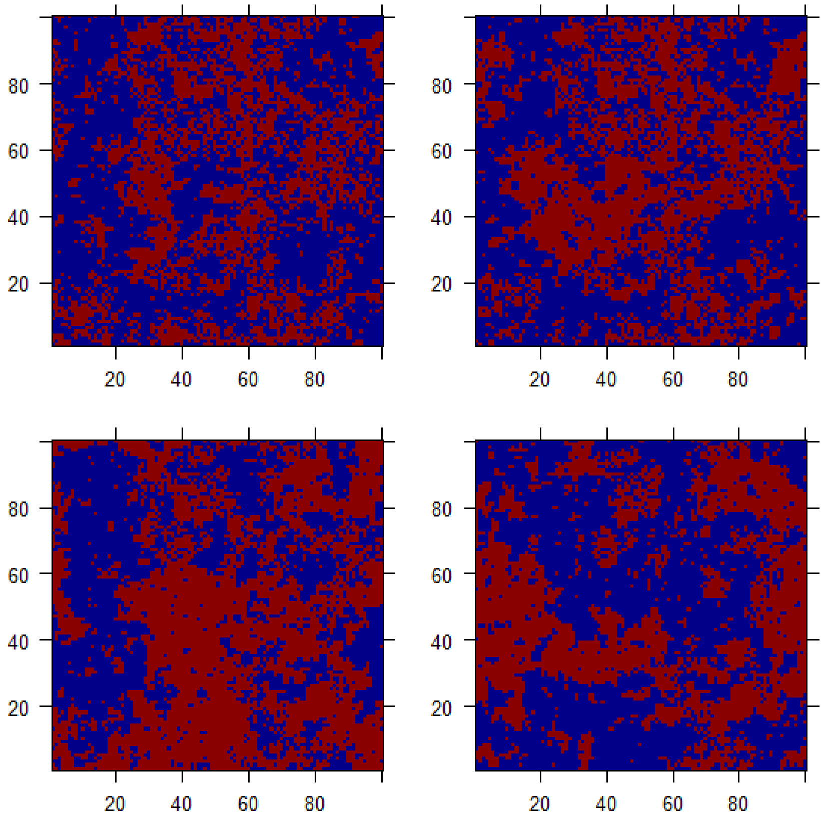


Figure 7. The four plots were generated at a temperature near the critical temperature T_c at Wolff steps 500, 550, 600, and 650 respectively on a 100 by 100 lattice.

On the other hand, the clusters are large enough such that if we were to decrease the temperature parameter below the critical temperature, the Wolff Algorithm would quickly be able to bring the lattice into to total alignment.

8.3 Cold Start [3]

To see the probabilistic mechanism of forming clusters in action, it might be an interesting procedure to initialize the starting lattice to be fully magnetically aligned in one spin value. If we allow for this initial configuration, the algorithm will choose a site whose neighbors all have the same spin value and we will be

[3] The term cold start comes from the concept that ferromagnetic materials tend to have all their particles fully aligned in the same magnetic direction; this section has the initial lattice structure as an all 'up' or all 'down' configuration.

able to see the extent of the formation of a cluster solely through the temperature parameter in $P_{add} = 1 - exp(2\beta J)$.

We can observe that as temperature increases, the algorithm prefers to form progressively smaller clusters. It is interesting to note that under the critical temperature regime, the cluster nearly becomes the whole lattice after one iteration. This is the result of each neighboring site having multiple chances to be appended to the cluster despite facing multiple rejections.

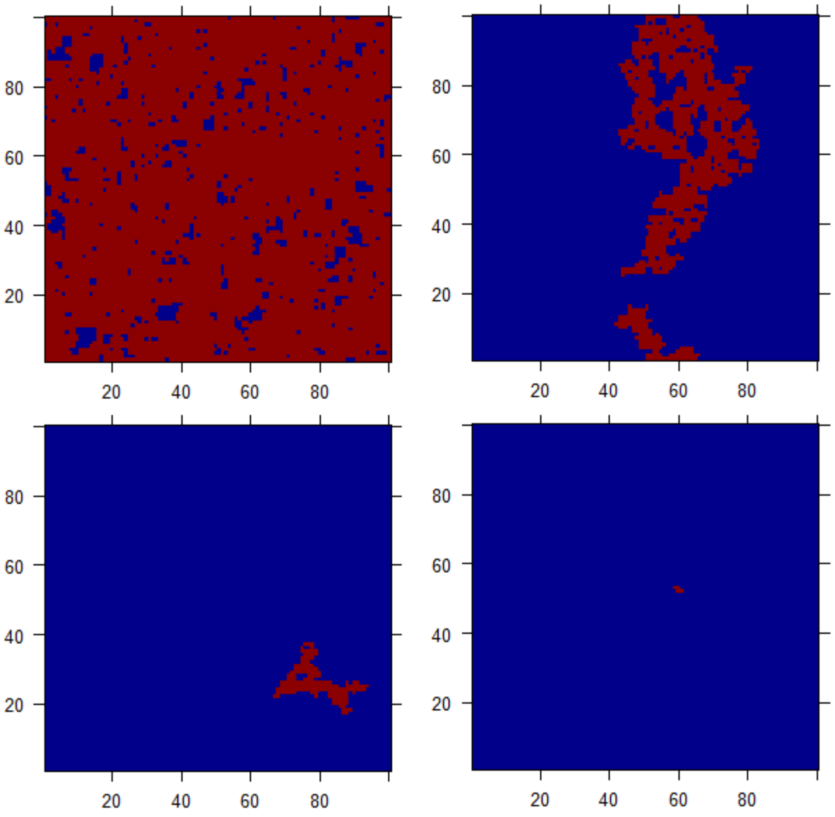


Figure 8. The four plots were generated at temperatures 2.26, 3.00, 4.00, and 10.0 respectively on a 100 by 100 lattice. The initial configuration was selected to be a lattice of all negative ”blue” spin values and the formed cluster after one Wolff step is the positive ”red” spin values.

Running the algorithm under a high temperature regime would produce plots similar to the high temperature regime of the single-site flip algorithm. Its performance may also be observed in the last plot above under a cold start on the lattice, which shows a final cluster formation that is approximately of size four.

9 Discussion

An issue with single-site flip algorithms arises when we allow temperatures to vary during the simulation, which is often the case in simulated annealing schedules. At temperatures nearby the critical temperature, the lattice arrangement is split between positive and negative spins, as seen in figure 2. As temperature is changed to $T < T_c$ and $T \to 0$, the lattice will tend to converge toward a single alignment; however the time it takes for a certain spin configuration preference to gain traction can be quite long. This is due to an increased difficulty for randomly selected sites to become flipped since its neighbors are typically aligned in the same direction. Since there are numerous clusters of same spin values, often times the selected site will refuse to flip.

This described pheneomenon can be concisely described by correlation length ξ, which measures the rate at which correlation between configurations vanish. More specifically, we could look at correlations between spins across distance $||i - j||$. Under the single flip spin algorithm, the asymptotic behavior of ξ is characterized by the cases of zero temperature, critical temperature, and infinite temperature. When temperature is at infinite temperature, the spins may be thought of as independent from each other so that their correlation is identically equal to zero across any distance between sites. At the critical temperature, the correlation length is infinite, so that there is correlation between all sites on the lattice. At low temperatures, the correlation length returns to zero, as the influence of one spin on another is determined locally through neighboring spin values so that correlations between sites across large distances may be thought of as zero.

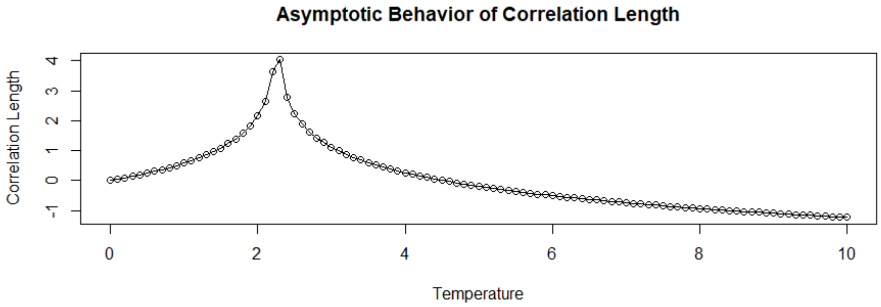


Figure 9. Asymptotic behavior of correlation length computed using theoretically determined dynamical exponents for the 2D Ising Model.

From a configuration stand-point, if we started at high temperature T, and let $T \downarrow T_c$, we would see the correlation between spins increase and fluctuations in spin values will increase to make the configuration more aligned. If we started from a low temperature T and let $T \uparrow T_c$, we would see clusters of spins begin to break up into smaller clusters as the correlation of each spin increases with other spins across the entire lattice.

While my simulations did not vary temperature over MC steps, we may still observe the phenomenon of critical slowing down from a magnetization versus time graph for different temperature regimes. A low temperature will lead to quicker magnetization as the lattice prefers to become aligned over time. A high temperature will result in a lattice that usually will not converge to ±1 magnetization.

A critical temperature will lead to a total magnetization whose central tendency is split between positive and negative lattice alignments. Given that the proportion of up spins and down spins may be near equal, if we allowed temperature to decrease below the critical temperature, it would take a considerable length of time before the total magnetization approached a ±1 value; for considerably high temperatures, the total magnetization tends to stay around zero.

With this information, we can observe the amount of time it takes for the lattice to converge to a single spin alignment as a surrogate for observing the process of critical slowing down under Glauber Dynamics.

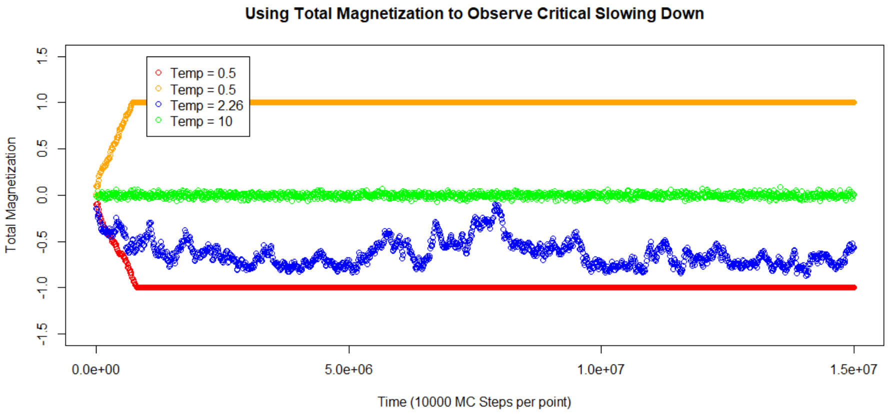


Figure 10. Glauber Dynamics: The main feature of interest occurs when temperature is near the critical temperature; it does not converge and has a central tendency to be split between positive and negative spins for many number of iterations which is proportional to the auto-correlation length.

A non-local clustering algorithm solves the issue of critical slowing down at critical temperatures by updating potentially large clusters of spins at a time. Instead of selecting single-sites that are hesitant to flip, the clustering algorithm selects a cluster of sites and flips its spin value with probability one.

The Wolff Algorithm works well for temperatures around the critical temperature; however the single-site flip algorithm will often perform better at lower and higher temperatures. When low temperatures are used in the Wolff Algorithm, the tendency is for the whole lattice to become a cluster, causing oscillating total magnetization values; this is a non-sense result in reality as freezing a magnet will not cause it to spontaneously change total alignment between ±1 in rapid succession. When high temperatures are used in the Wolff Algorithm, the tendency is that many spins are rejected from joining a cluster, causing the algorithm to behave like a sub-optimal single-flip algorithm.

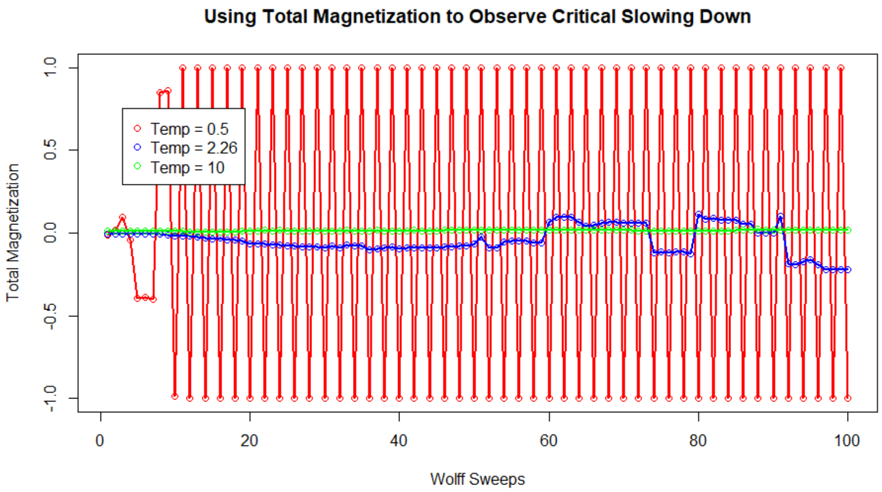


Figure 11. Using the Wolff Algorithm on a 50 by 50 lattice, we can see that temperatures near the critical temperature have the same tendency to split between positive and negative spin alignments. However, if we were to decrease the temperature below critical temperature, the lattice would quickly undergo a phase transition, which can be observed by the line following the low temperature regime which abruptly becomes totally magnetized in one direction.

Figure 11 demonstrates the magnetic behavior of the lattice at a low, critical, and high temperature regime. Furthermore, it can be used to extrapolate the non-issue that the Wolff Algorithm has with regards to the process of critical slowing down; if temperature were initialized at $T := T_c$ and allowed to approach the low temperature regime, it would spontaneously magnetize.

10 Time Comparison

To make the comparison between the single site flip algorithm and the Wolff Algorithm, we could try to make the units of time comparable in some way. The adjusted unit of time needs to account for the differences in behavior of the Wolff Algorithm at various temperature regimes.

Set the unit of time $\tau_W = \tau_{flip}\frac{<n>}{L^d}$, where τ_{flip} is the length of time it takes for the Wolff Algorithm to flip one cluster, $< n >$ is the temperature dependent average cluster size, and L^d is the lattice size in $d = 2$ dimensions. [3].

The average cluster size $< n >$ may be thought as a computational cost for the program to run, and L^d may be thought as the size of the system to be updated so that the ratio $\frac{<n>}{L^d}$ may be thought of as the proportion of the lattice under consideration on a given iteration of the Wolff Algorithm.

By computing τ_W, we may compare the unit of time produced by Glauber Dynamics denoted τ_G, which may be thought of as the run-time for one iteration in the single-flip dynamics.

More generally, it is difficult to make comparable time steps since the unit of time for the Wolff Algorithm is dynamically changing with respect to average cluster sizes $< n >$ and the execution time for one step in the Wolff Algorithm τ_{flip}.

10.1 Bounds on unit of time for the Wolff Algorithm

Generally speaking, τ_{flip} is the amount of time for one iteration of the Wolff Algorithm to be executed τ_{flip} which is dependent on the implementation of the algorithm itself and may be highly variable. On the other hand, the autocorrelation time also depends on some attributes which we can determine to hold approximately true. For simplicity, hold τ_{flip} at some fixed value determined by the run-time of one iteration which depends on temperature, lattice size, and implementation of the algorithm.

In the case of $T := \infty$, the average cluster size $< n > \approx 1$; it is not computationally costly to form a cluster of size 1 on a lattice of size L^2, so that the time unit $\tau_W \approx \tau_{flip}\frac{1}{L^2}$.

On the other hand, if $T := 0$, the average cluster size $< n > \approx L^2$; the computational cost of forming the cluster is proportional to the lattice size; the algorithm would typically have to do a full sweep of the lattice before making an update, $\tau_w \approx \tau_{flip}$, i.e. the time unit under the Wolff Algorithm will be approximated by the run-time of one iteration of the algorithm.

11 References

[1] “Exact Solutions of the Ising Model in 1 and 2 Dimensions.” NYU. ↪

[2] ”Markov Chain Monte Carlo: Metropolis and Glauber Chains.” Markov Chains and Mixing Times, by David Asher Levin et al., American Mathematical Society, 2017, pp. 37-45.

[3] Carlon, Enrico. “Computational Physics: Advanced Monte Carlo Methods.” Institute for Theoretical Physics. ↪

[4] Durrett, Richard. ”The Metropolis-Hastings Algorithm.” Essentials of Stochastic Processes. pp. 35-38.

[5] Based on lecture notes from MATH 5652 taught by Professor Arnab Sen in Fall 2017 semester. ↪

12 Attempted Changes to the Draft

- **1** Figures were captioned and numbered. Lattice size and number of iterations were deposited into the captions to decrease clutter in main body of the text.

- **2** Sections (supplmentary material, ising significance, appendix references) were omitted.

- **3** The topic of critical slowing down was re-written to hopefully be more explanatory in the section titled "Discussion".

- **4** The section titled "Exact solutions" has been re-titled as "Defining the Ising Model" and the section was rewritten to be more concise (omit discussion of infinite ensemble).

- **5** Subsections were created for each algorithm w.r.t. different temperature parameters to try to keep these features self-contained.

- **6** Algorithms were put into a 'box' to be more visible.

- **7** Sections were re-arranged in attempt to be more cohesive.

- **8** (Not shown); Wolff Algorithm re-run at temperature 0.8 to see if the configuration could look more similarly to low temperature Ising. The result looked much like the temp 0.5 Wolff configuration and the whole lattice became aligned.

- 9 Time comparison section re-written, although a preliminary search has indicated to me that there might be more robust ways to compare the two algorithms.