Raewyn Duvall
Jared Moskowitz
Anna Phillips

Ising Model Analysis

View this document online for better picture quality:
https://docs.google.com/document/d/1N1WFzZiibJoD8fILT2kGS9F5F5g64UhfOQo6EdiOuqQ/edit?usp=sharing

Introduction

We examined the Ising Model for various geometries for ferromagnetic and antiferromagnetic systems. The Ising Model of magnets assume that the energy of a magnet can be approximated by the following Hamiltonian:

$$H = J \sum_{\text{neighbors}} \sigma_i \sigma_j$$

where J=-1 for the ferromagnetic case and and +1 for the antiferromagnetic case, and σ_i, σ_j represent the spins of each state, either +1 or -1. The sum is executed over neighboring spins.

The majority of our analysis was done using the Metropolis Algorithm, which is described below. We have three sets of code that implement this algorithm: lattice.py, which runs on a “square” grid of arbitrary dimensions; regular2D.py, which does a square grid of 1 or 2 dimensions; and HexagonalLattice.py, which runs on a hexagonal grid of 2 dimensions.

We also implemented the Wolff algorithm. The code for this is in wolffModule.py. It draws on lattice.py to generate the grid and therefore can be run in an arbitrary number of dimensions. We chose to implement the Wolff algorithm as it is fundamentally different from the Metropolis algorithm and we could find clear explanations online. (We used this source).

Metropolis Algorithm Analysis

The Metropolis Algorithm gives us the ability to examine the behavior of the the magnet over time. For each time step, we move through our grid, checking the change in energy at each point caused by flipping that cell. If the change in energy is negative, we accept the flip. If it is positive, we accept the flip with probability $e^{-\Delta E/kT}$. Note that for $\Delta E < 0$, this quantity is always greater than 1. Therefore, we can perform one check: if our random number between 0 and 1 is less than $e^{-\Delta E/kT}$, we accept the flip.

To compute the change in energy, we did not use the full Hamiltonian. Instead, we looked only at the terms in the Hamiltonian that involve the cell of interest. This gives us

$$J\left(\sum_{\text{all neighbors}} (\text{test value})(\text{neighbor value})\right) - J\left(\sum_{\text{all neighbors}} (\text{current value})(\text{neighbor value})\right)$$

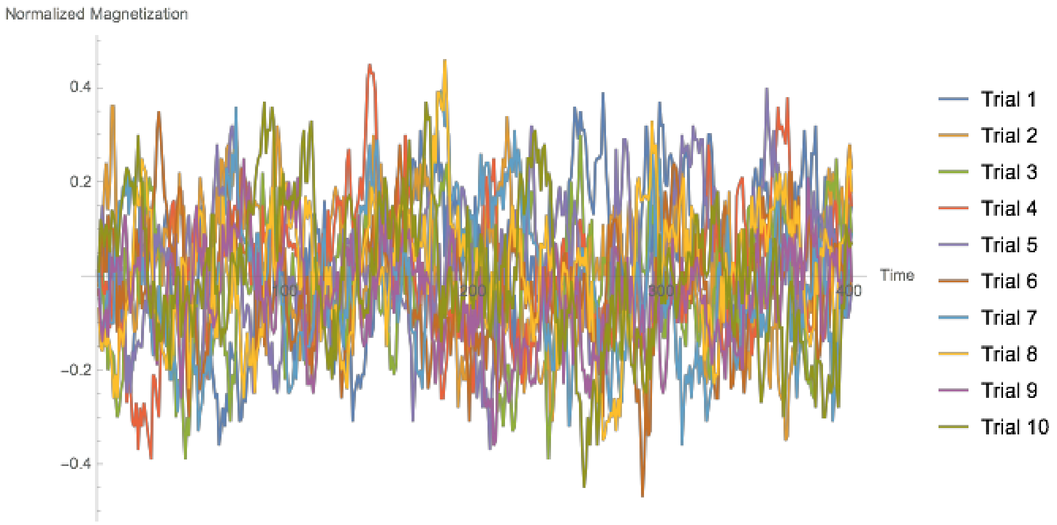
We can simplify this by noting that the test value is always -1 times our current value. Then this reduces to

$$-2J(\text{current value})\left(\sum_{\text{all neighbors}}(\text{neighbor value})\right)$$

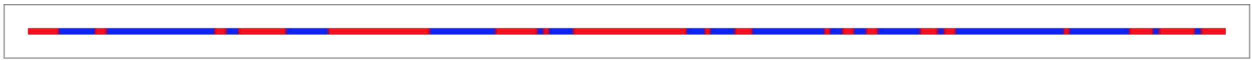
1D

Our primary trials on the 1D model were run with on a 200 cell long system for 400 time steps at temperatures between 0.05 and 3 in units where k=1. The Python code (regular2D.py) was run 10 times for each choice of parameters, and the data was analyzed in Mathematica. See the Mathematica notebook *DataAnalysis.nb* for more detail on each calculation. For all of our data, we examine the normalized magnetization, which is the sum of all cell values divided by the number of cells. This lets us compare across systems easily.

In one dimension, the Ising Model should not exhibit behavior consistent with a phase transition. (See Schroder, *Thermal Physics*, section 8.2 in the first edition). Indeed, this seems to be the case for our data. At high temperatures, we find that the magnetization fluctuates quickly as a function time. At the end of a trial, we also find small domains.

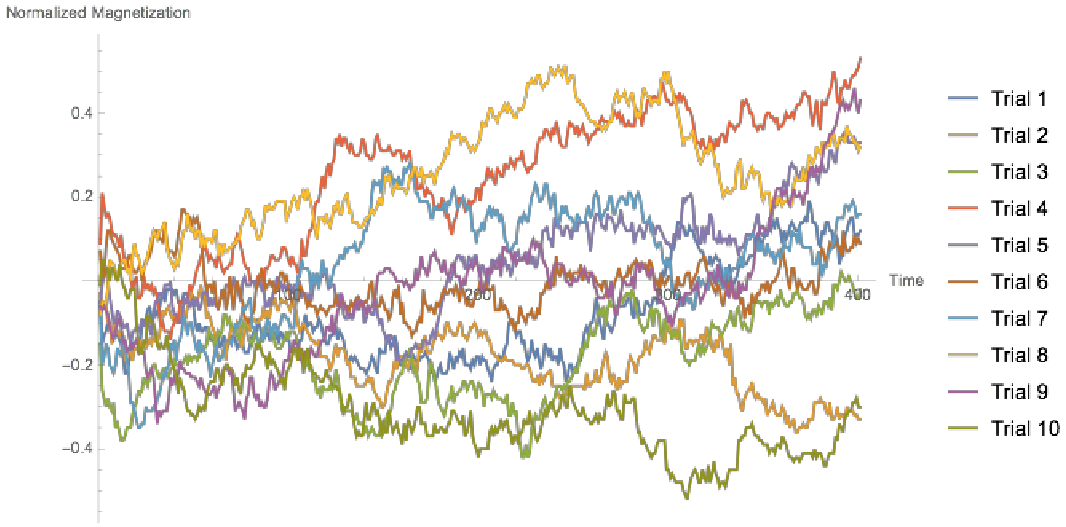


Magnetization vs time for ten trials for a 200 cell system with kT=3.

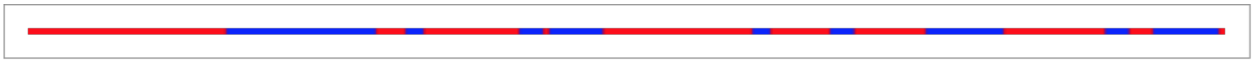


The last array in a one trial with kT=3.

At lower temperatures, the magnetization changes much more slowly. We also see larger domains forming.



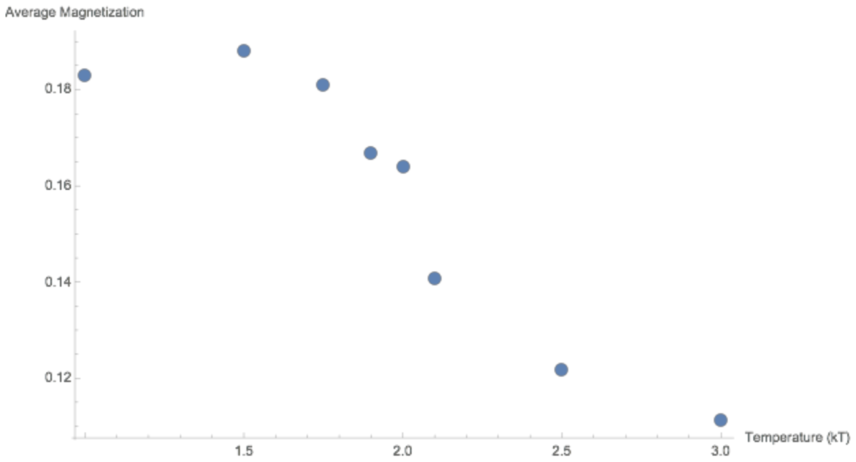
Ten trials on the same system at kT=1.5.
For each trial, the magnetization changes slowly over time.



The result of one trial at kT=1.5. The magnet is broken into relatively stable, slightly larger domains.

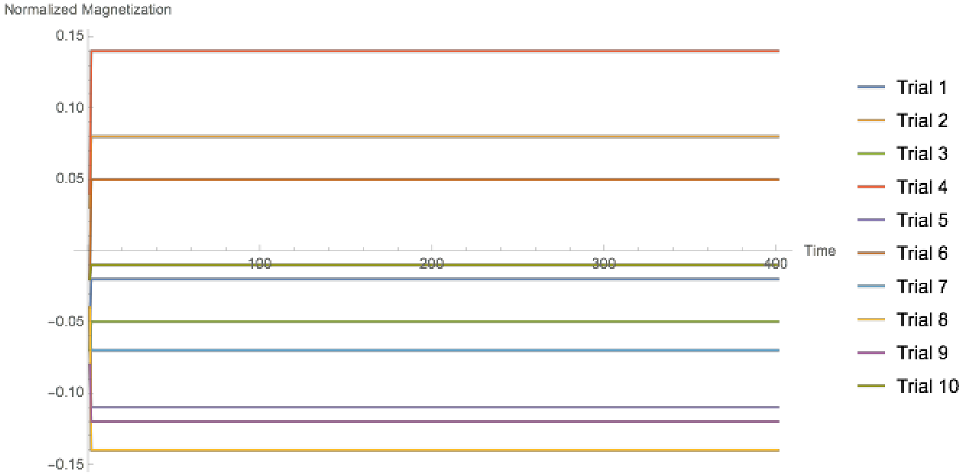
To look across trials, we found an average magnetization. Within each trial, we dropped the first 100 time steps to only look at the longer term behavior, taking the mean of the absolute values of the remaining magnetization values. The purpose of taking the absolute value was to take into account the symmetry of the system. We then averaged all 10 trials together.

Plotting magnetization as a function of temperature shows that the magnetization does decrease as the temperature increases. We see the beginnings of a flattening tail, which makes sense since our scheme is guaranteed to never give us a value of 0 due to the absolute value in the calculation.



For each trial, the average of the absolute value of the magnetization was calculated. This was then averaged over the 10 trials.

At very low temperatures, we saw that specific configurations locked into place in just a few time steps. These trials were omitted from the above plot because we see a very strong anchoring effect: the magnetization stays relatively close to the starting value and then is extremely constant.



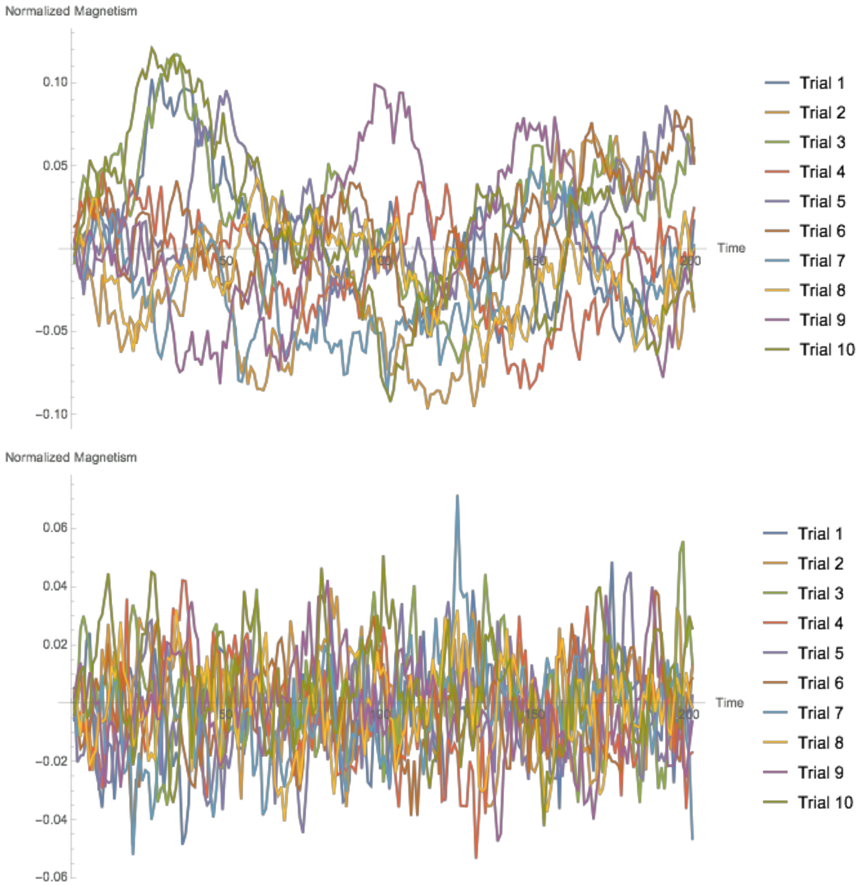
Trials at kT=0.05. We see certain configurations lock into place within 10 time steps.

At no point on the 200 cell 1D system did we see all of the spins aligned. The largest magnetization values seen were around 0.4, which corresponds to about 70% of spins pointing in one direction versus 30% in the other. We do see complete alignment (corresponding to a normalized magnetization of 1) in smaller systems of around 10 cells.

2D

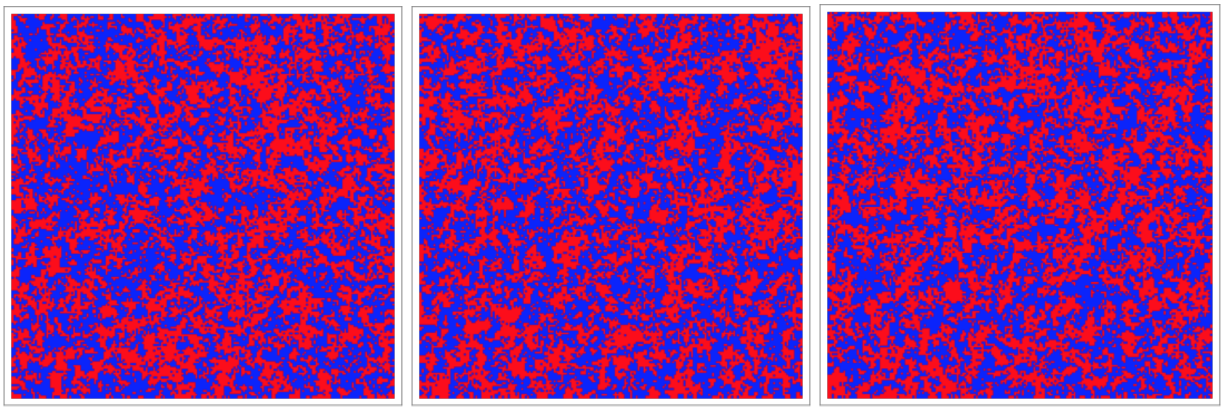
For the 2 D Ising Model, we do expect to see a phase transition in the data. The majority of these trials were run on a 200x200 grid for 200 time steps. Some trial and error revealed that 200 time steps was a sufficiently long duration for us to see the long term behavior.

As with the 1D model, at higher temperatures, we see that the magnetization is close to zero and changes rapidly.



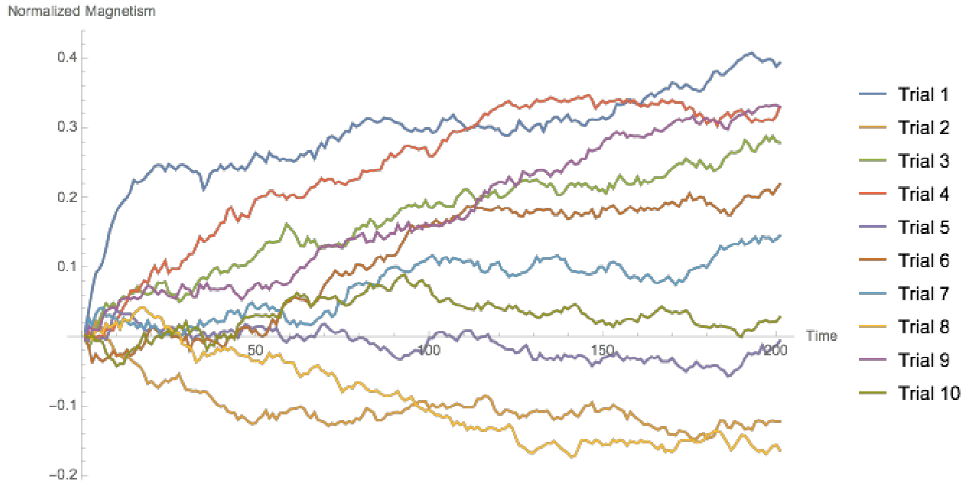
Magnetization vs time for 10 trials at 2.5kT (top) and 3kT (bottom).
The magnetizations fluctuate more rapidly with increasing temperature.

Visualizations reveal that the system does not appear to become significantly more ordered over time at high temperatures.

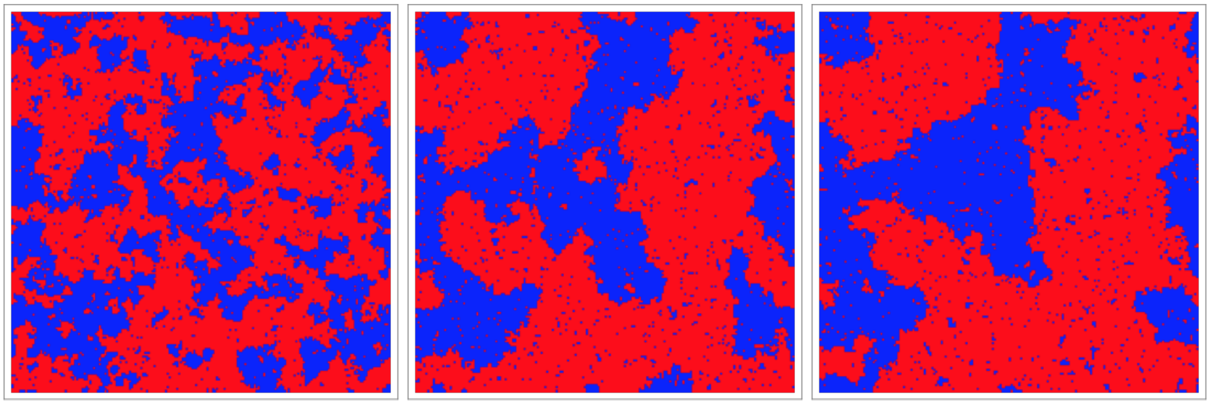


A trial at kT=3 at time steps 10 (left), 50 (center), and 100 (right)
No domains are visible.

At lower temperatures, we do see domains develop, though they are not uniformly smooth. The appearance of these domains suggests that the critical temperature is near kT=2. The magnetization does not change as quickly over time.

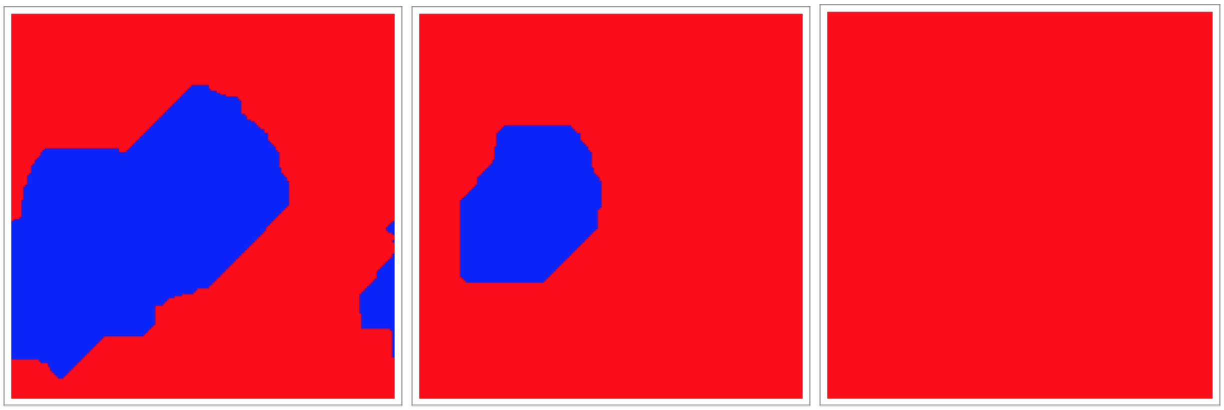


Magnetization versus time for 10 trials at kT=2.
Trial 1 is the same data as the snapshots below.

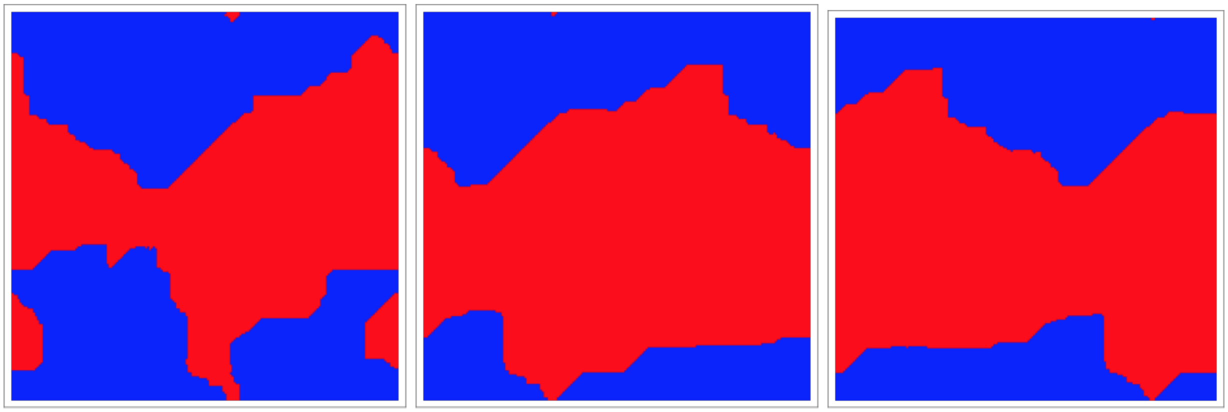


Trial 1 at kT=2 at time steps 10 (left), 50 (center),100 (right).
Two domains are visible after 100 time steps.
Those domains remained relatively stable for the next 100 steps.

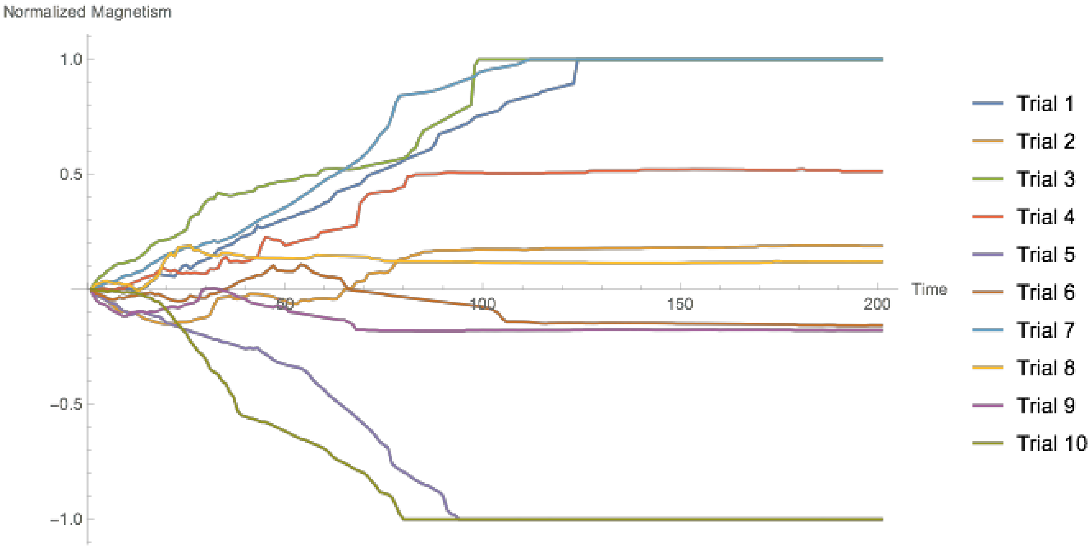
At even lower temperatures, we find smooth domains that stabilize quickly. Some stabilize into one domain and other stabilize into two domains.



Trial 1: A 200x200 grid at 0.5kT at time steps 50 (left) and 100 (middle) and 200 (left). This trial settled into one domain

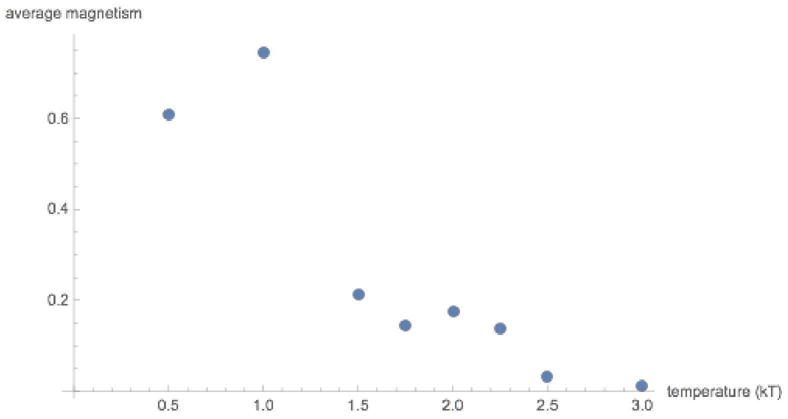


Trial 2: A 200x200 grid at 0.5kT at time steps 50 (left) and 100 (middle) and 200 (left). This trial settled into two domains



10 trials of 200x200 grid at 0.5kT. Trials 1 and 2 are the same data as the above snapshots.

The plot of magnetization versus temperature is more difficult to interpret. The average magnetizations for very low temperatures are high, well above the highest values for the 1D case. We do see a small bump around kT=2, which is around where we expect the phase transition based on the magnetization versus time graphs.

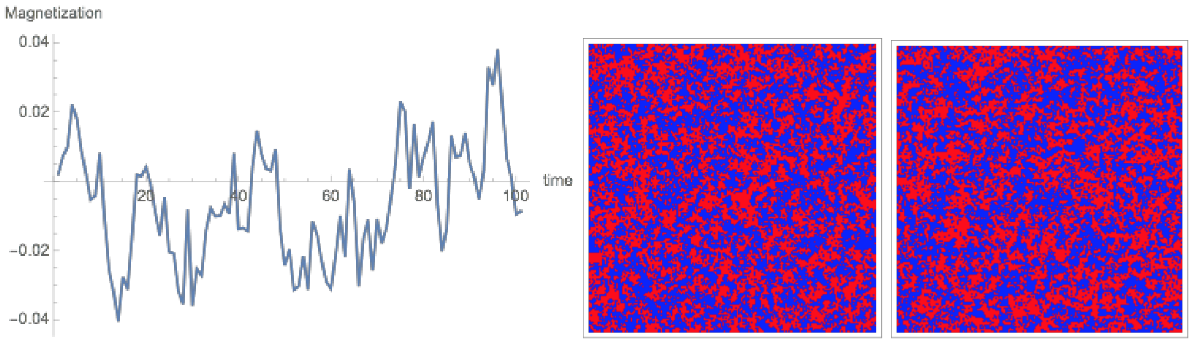


3D and beyond

As of 9pm on Sunday, our generalized code isn't working correctly, and so we can't complete this analysis.

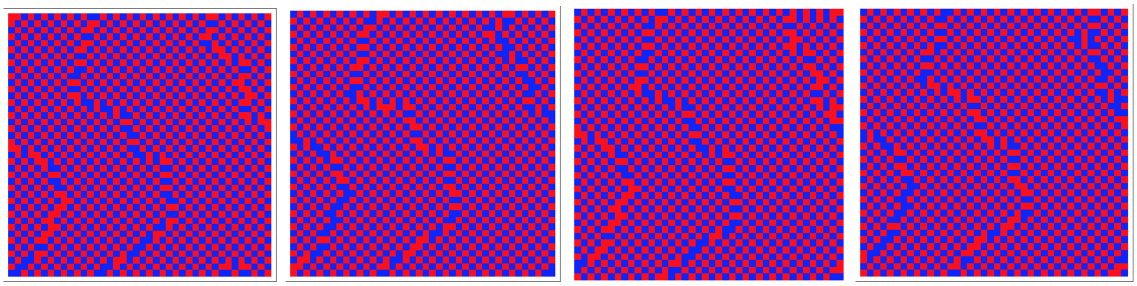
Antiferromagnetic cases in 1 and 2 D

At high temperatures, the 2D antiferromagnetic system behaves much like the ferromagnetic system, though with lower average magnetization.

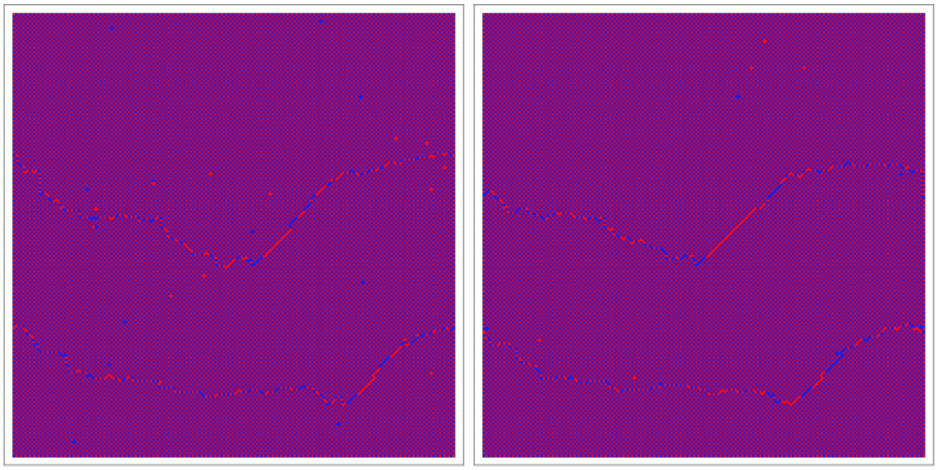


A 2D antiferromagnetic square grid at kT=3.
The magnetization versus time (left) and snapshots at timesteps 50 (center) and 100 (right).

At lower temperatures and on smaller grids, we see easily certain patterns lock into place as either one or two domain results. The one domain result seems more common, though the two domain solution is more fun to watch. The same thing happens on larger grids (with 2 domain solutions being more common), though it is harder to see.

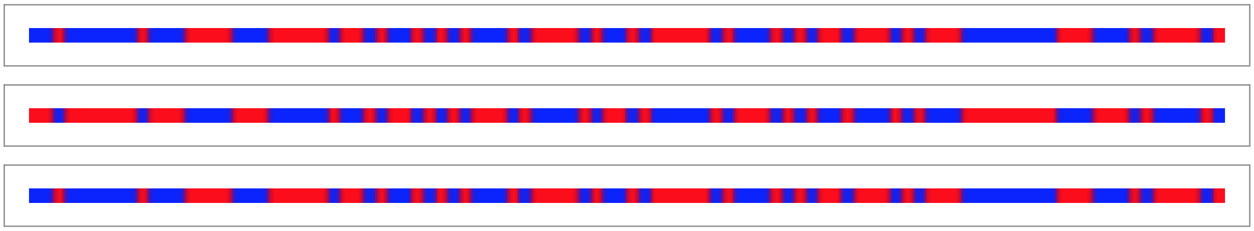


A 40x40 antiferromagnetic grid with kT=1 at time steps 15, 16, 17, and 18.
This is a two domain final state, and the boundaries between the domains seems to move as a wave.



A 200x200 antiferromagnetic grid at time steps 150 and 160 with two domains.

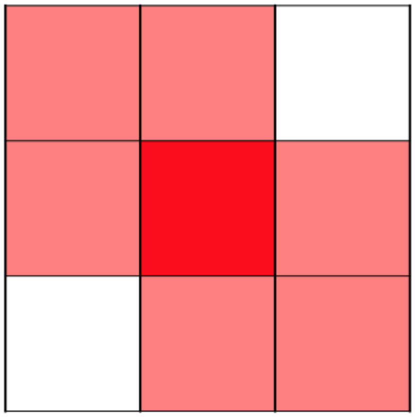
The 1D case seems to exhibit the same behavior regardless of temperature. It settles into a pattern where everything flips from one timestep to the next. This seems odd, and I don't have an explanation.



Time steps 20, 21, and 22 of a 1D antiferromagnetic grid at kT=3.
The system alternates between two states.

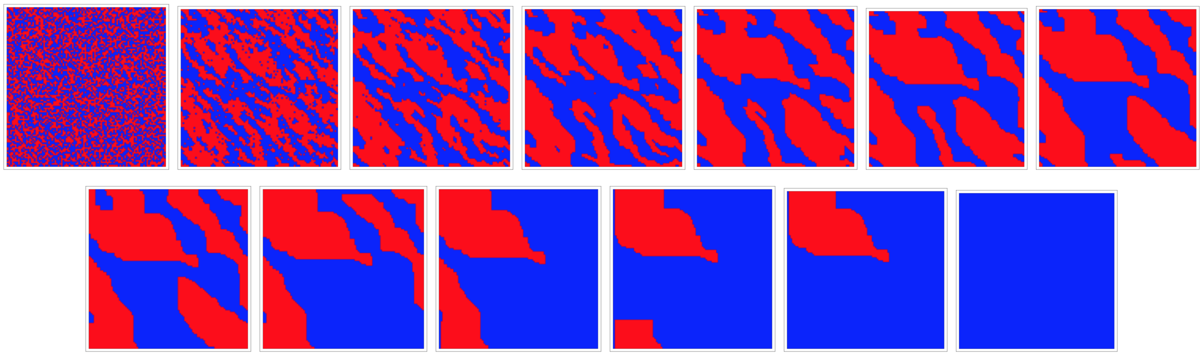
Hexagonal Lattice: Ferromagnetic and Antiferromagnetic cases

The hexagonal lattice pattern behaves much like the square lattice in the 2D case. We can run our code on a square grid with new adjacency rules:

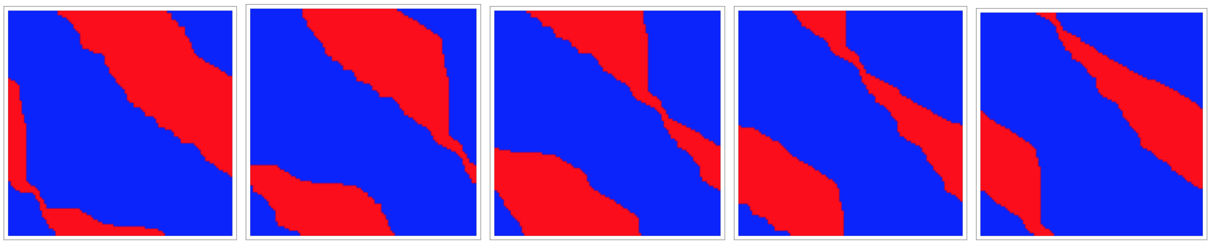


Each cell (red) has 6 neighbors (pink).

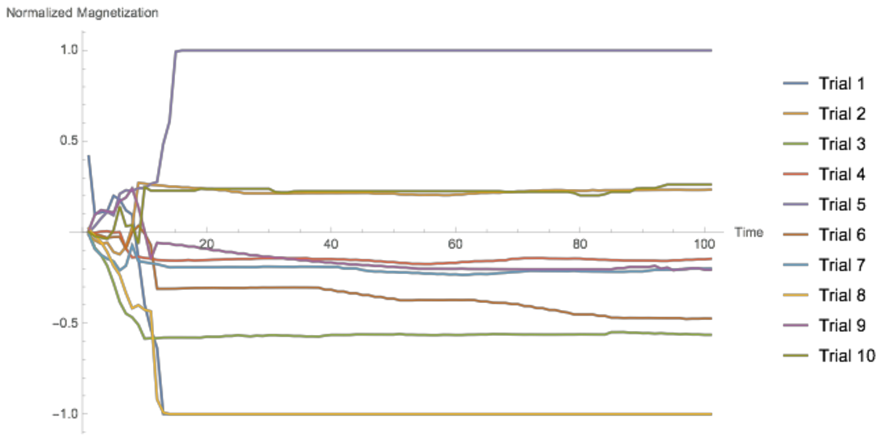
In this way, we only need to change our code slightly. At low temperatures, the system tends to align, just as in the square lattice case. At kT=0.5, 2 domain solutions seem common. When only 1 domain forms, it forms faster than the square case at the same temperature. In two domain systems, the borders between the two move in wave like patterns, which is interesting.



The first 13 time steps of a 100x100 hexagonal grid at kT=0.5.
The system settles in only 13 time steps.
Note that boundaries tend to form on diagonal lines as a result of the adjacency rules.

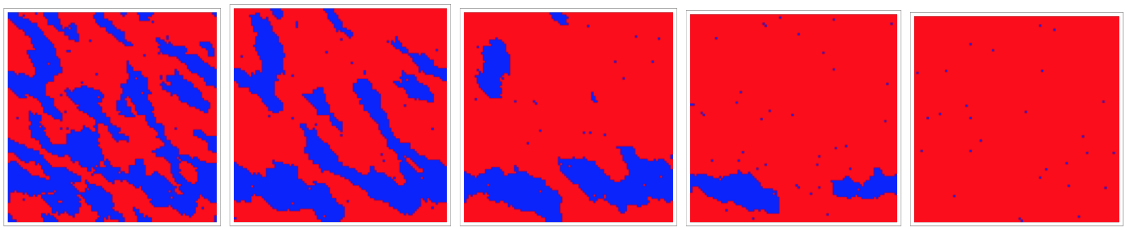


From left to right: time steps 20, 40, 60, 80 and 100 of a 100x100 hexagonal grid at kT=0.5.
The system settled into two dynamic domains.

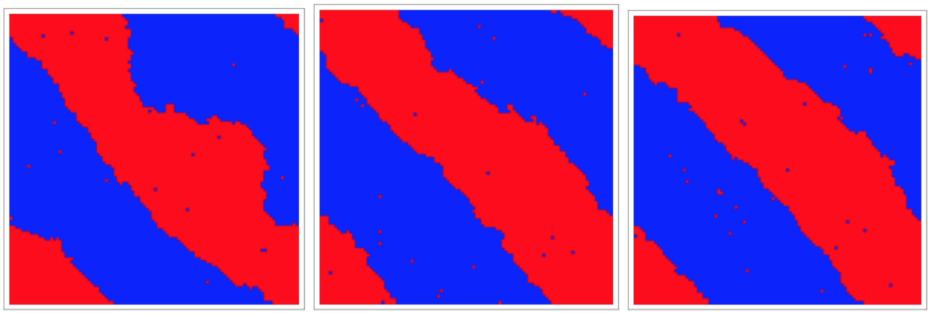


The magnetization versus time for 10 runs of a 100x100 hexagonal grid at kT=0.5. Three trials ended with one domain, the rest with two domains.

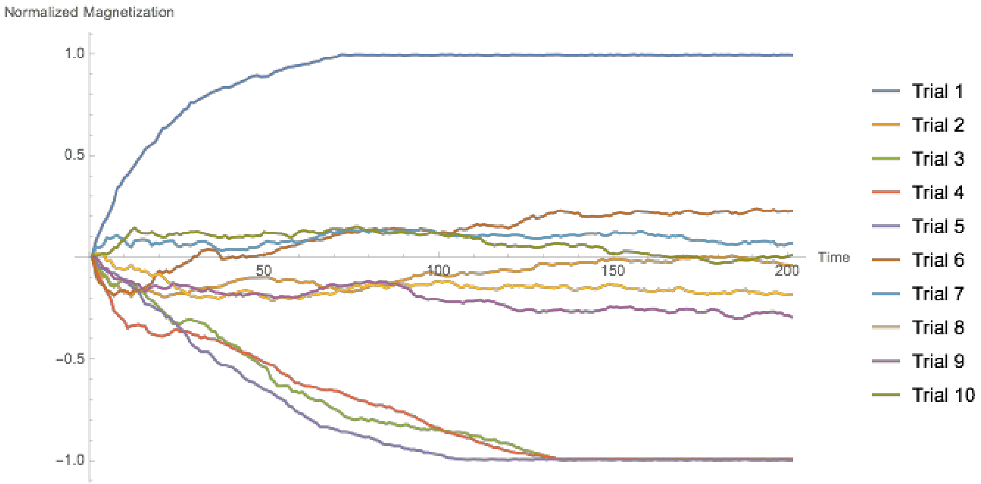
At kT=2, the system still settles into domains, though there are flecks of color remaining.



From left to right: time steps 5, 10, 20, 40 and 800 of a 100x100 hexagonal grid at kT=2. The system settled into one domain.



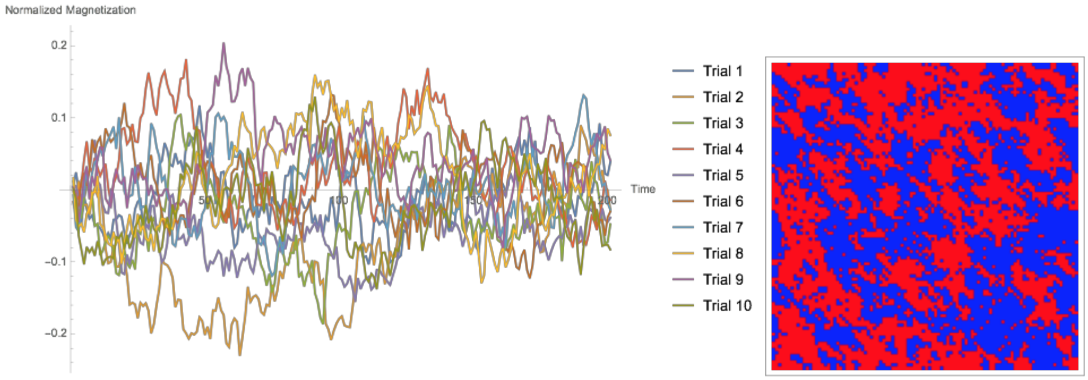
A 100x100 hexagonal grid at kT=2 at timesteps 50 (left), 100 (center) and 200 (right) The system settled into two domains.



The magnetization versus time for the 100x100 hexagonal grid at kT=2.

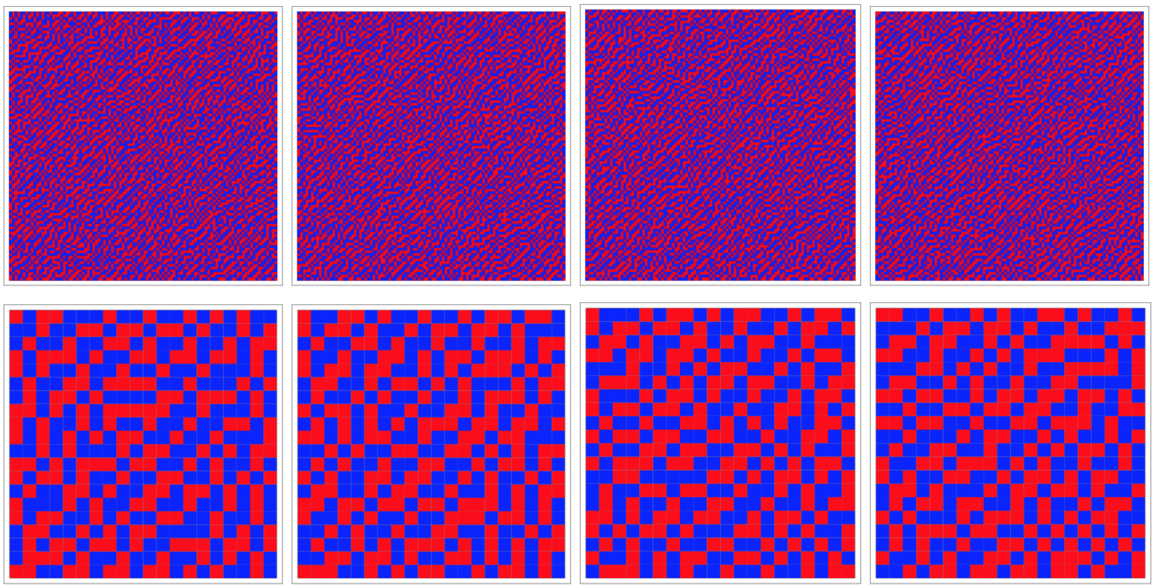
The hexagonal grid generally behaves as though it is at a lower temperature when compared to the square grid. This makes sense since each cell is attached to six others rather than four. This extra connectivity means that the grid is is more closely tied together.

By kT=4, we do see the type of behavior we expect above the phase transition. Thus, it seems likely that the critical temperatures of this model is somewhere in between kT=3 and kT=4.

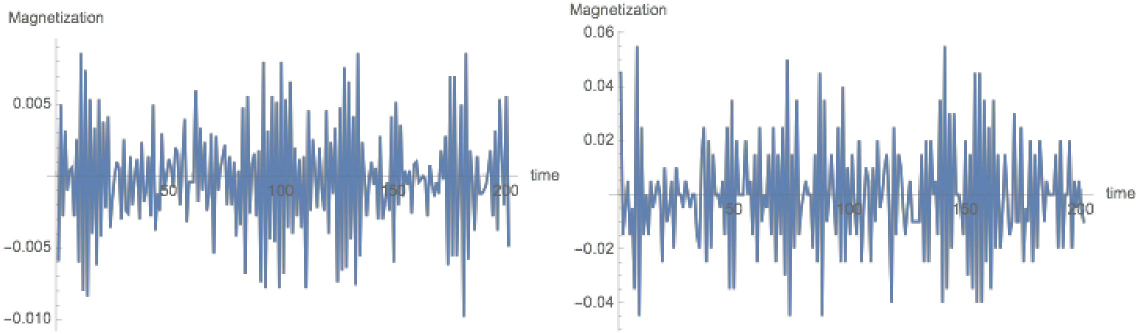


The magnetization over time (left) and a snapshot of a single trial at 200 time steps. The 100x100 grids at kT=4 do not settle into large domains, though certain areas are predominantly in one state.

The antiferromagnetic case for the hexagonal grid does not settle into a pattern like the square grid case. Instead, we see localized “W” shaped structures that fluctuate rapidly.



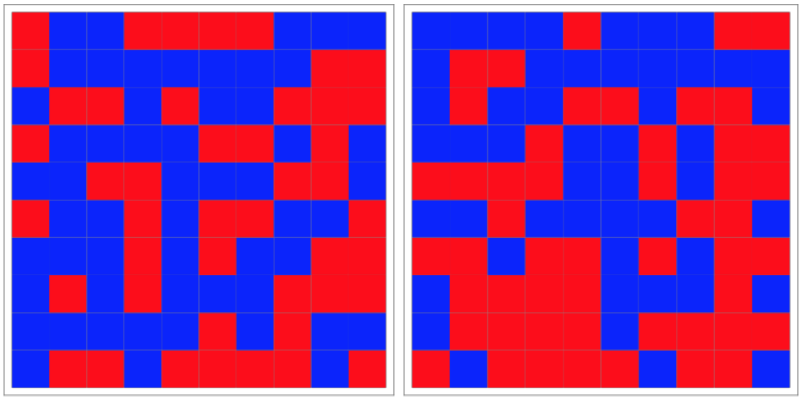
100x100 (top) and 20x20 anti ferromagnetic hexagonal grids at kT=1 at time steps 50, 51, 51, and 53. The patterns are easier to see on the smaller grid.



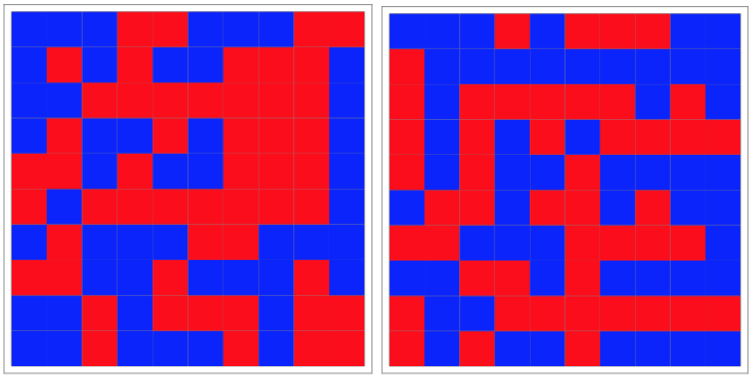
Plots of magnetization versus time for the 100x100 (left) and 10x10 (right) grids shown in the snapshots above. The magnetization is small, but fluctuating quickly.

The Wolff Algorithm

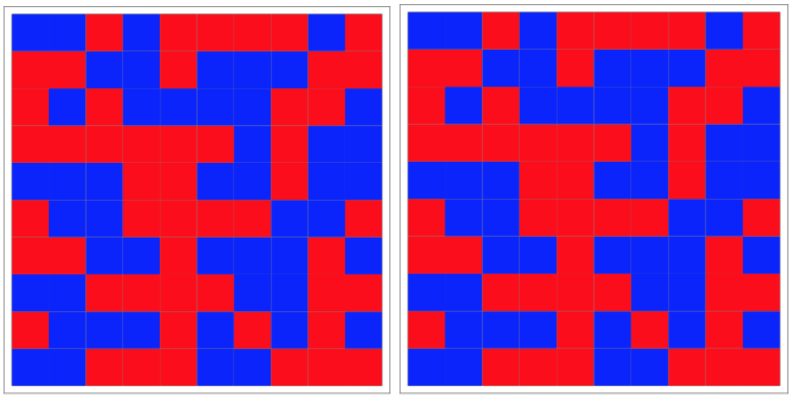
Our initial results from our Wolff algorithm code are a bit baffling. At temperatures where we would expect the grid to form one or two domains (kT=1), we do not see domains form. At very low temperatures, our grid appears static. This indicates a possible problem with the code, though it is also possible that it simply takes a very long time for the algorithm to run.



Initial (left) and final (right) states for a 10x10 grid at kT=1 iterated 10000 times. While there is a larger red region in the final state, we do not see clear domains which we expect based on the Metropolis algorithm.



Initial (left) and final (right) states for a 10x10 grid at kT=2 iterated 10000 times.



Initial (left) and final (right) states for a 10x10 grid at kT=0.05 iterated 10000 times. The initial and final states are identical.

Implementation and Analysis of Wolff Algorithm:

The Wolff algorithm was invented by Ulli Wolff in 1989 (wikipedia.com). What makes this algorithm different from the metropolis method is that it updates multiple particles at a time. For a given particle, the algorithm is as follows:

```
Add current particle to list (flipList) of particles to flip
For each currentParticle in flipList
    for each neighbor
        if neighbor is not in visitedList
            if sign of neighbor is same as currentParticle
                if $\text{prob}(0, 1) < 1 - \exp(-\Delta H/kT)$
                    add neighbor to flipList
                else
                    add neighbor to visitedList
            else
                add neighbor to visitedList
    add currentParticle to visitedList
Flip all values in flipList
```