Pre-class work

Here is some Python code implementing the Ising model. Use this code to run the experiments described below on a 20 x 20 grid and record your observations.

Create a Google document and record your work and all exercises. **Make sure the Google document is shared** so that it can be assessed, and **be ready to paste a link to your document into a class poll**.

Task 1

From the study guide, we know that the Boltzmann distribution tells us what the probability is of observing a state at equilibrium. At low temperatures, we expect the states where all cells have the same sign (either all + or all –) to be much more probable than any other state. So if we start from a random initial state, we should drop into one of those two states eventually.

Test whether this indeed happens.

- Randomly initialize the simulation, set the temperature to 1 (that is 1 Kelvin, which is close to absolute 0 — a theoretically unobtainable temperature), and wait for the simulation state to stabilize (this might take up to 100,000 steps).
- Measure the average magnetization of the final state.
- Repeat this simulation a number of times to get a probability distribution over the average magnetization. Theoretically, the distribution should be evenly split between +1 and –1, but you will find that the simulation often gets stuck in a suboptimal state with a different average magnetization.
- Paste your histogram into your Google doc.
- Explain what would happen to the acceptance probability of the update step if the temperature is set to 0. This motivates why we always set the temperature at least slightly greater than 0.

Task 2

- Start from a random initial state, as before, but rather than setting the temperature to 1 K straight away, start from T = 2000 K and decrease it by 1 K after every 100 update steps until you reach a value of T = 1 K. Wait for the simulation state to stabilize.
- What is the distribution over average magnetization now? (You can get even better results by decreasing the temperature more slowly, for example, 1 K after every 400 steps.)

Information related to Task 2

This procedure of gradually reducing the model temperature until it converges to a stable state is called simulated annealing. This is yet another algorithm inspired by a real physical process.

- Annealing is used to make various metals less brittle so that they can be shaped more easily. By heating the metal and then letting it cool slowly its atoms tend to better align in a stable lattice. (Some optional short videos on annealing metal: [1] [2].)
- Simulated (and real) annealing works because it is easier for the simulation to move between very different states at high temperatures, but difficult at low temperatures. Imagine how long it would take for tiny random fluctuations to take the state all the way from completely positive to completely negative at low temperature — even though these two states are equally probable according to the Boltzmann distribution.
- So we increase the temperature temporarily to allow the state to change a lot and then reduce the temperature back to where we want it to allow the state to stabilize. There is a difference between reducing the temperature rapidly or slowly. Doing it very rapidly can land you in suboptimal states (which are not from the target distribution). This is a real physical phenomenon, and you observed it in Task 1, but it an undesirable trait in Metropolis-Hastings sampling.

Simulated annealing is an example of a general algorithm used to move between states of an MCMC simulation more efficiently. In class, we look at the Wolff cluster algorithm, which is even more efficient but is a specialized algorithm for generating samples from the Ising model.