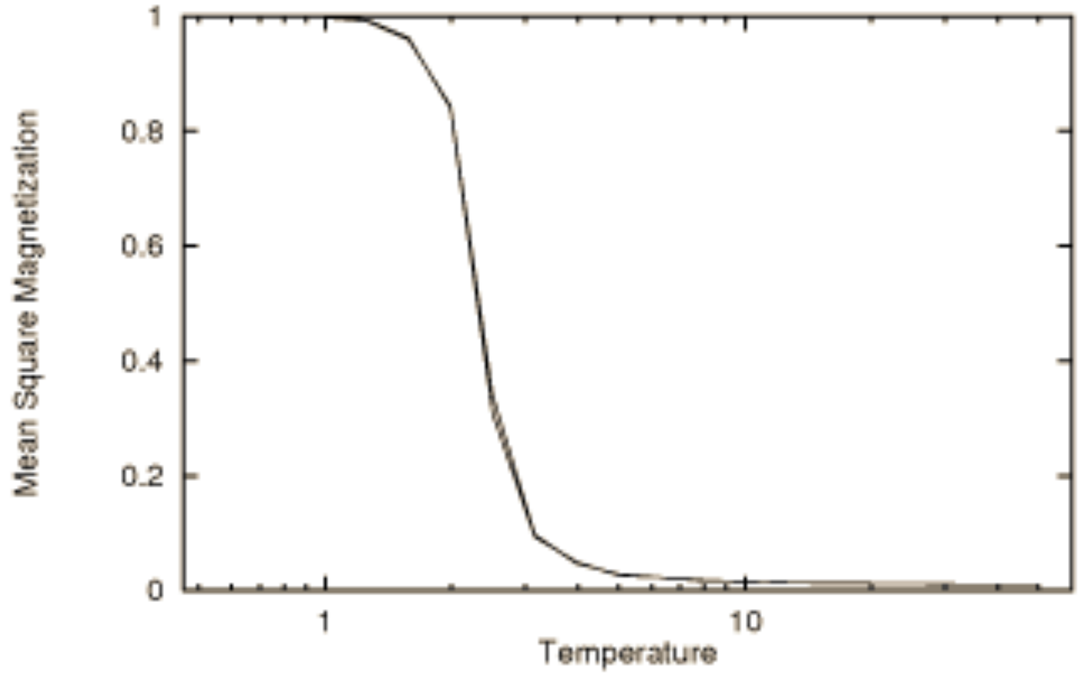
Monte Carlo Simulation of the 2D Ising Model

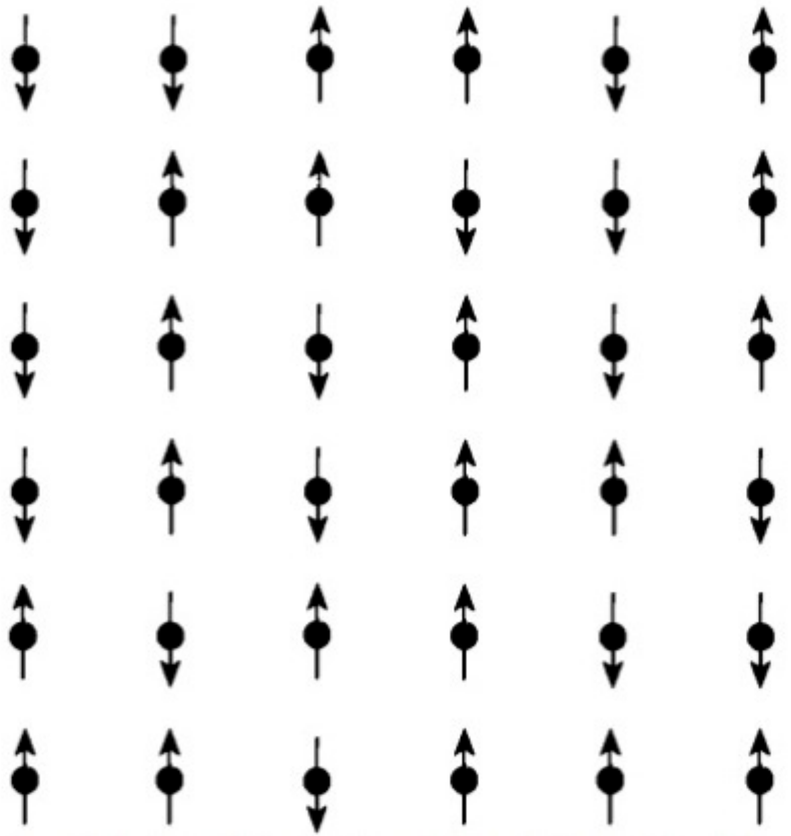
Ferromagnetic materials (Iron, Nickel, etc) exhibit spontaneous magnetization in the absence of an applied magnetic field — but only if $T<T_c$.

This is an example of *"critical phenomena".*



The Ising Model can explain this.

The Ising Model



Consider a 2D array (lattice) of magnetic dipoles that can be in one of two orientations: "*spin-up*" or "*spin-down*"

Each dipole interacts with all of the other dipoles. The interaction between any pair is like

$$E = -J\,\vec{S}_1 \cdot \vec{S}_2$$

Lowest energy if the dipoles align in the same direction.

In general, $E = -\frac{1}{2}\sum_i \sum_{j \neq i} J_{ij} S_i S_j$ where $S_i = \pm 1$

Depends on separation and angle between the dipoles, etc.

The Ising Model

Classical systems obey Boltzmann statistics, so the average (or expectation) value for some observable property of the system will be

$$\langle A \rangle = \frac{\sum_s A_s e^{-\frac{E_s}{k_B T}}}{\sum_s e^{-\frac{E_s}{k_B T}}}$$

where the sum is over all possible states of the system and E_s is the energy of that state.

Each state is a possible configuration of all of the dipoles.

Since each dipole has 2 states, an $N \times N$ lattice has $2^{N \times N}$ possible states.

If N=20, there would be 10^{120} different states!

We can't possibly do these sums!

The Metropolis Algorithm:

Since we can't do the sums directly, lets use random numbers to sample the states. There are two ways we could do this:

1) Randomly pick states (set of all spin values). Compute E_s, e^{-E_s/k_BT}, and all $\langle\rangle$'s we want.

   Problem: could spend a lot of time computing states with very small probabilities

2) Instead, we would like to compute more probable states more often. Choose states based on their Boltzmann factor and weight them equally (*instead of choosing them equally and weighting them by their Boltzmann factor*).

The Metropolis Algorithm:

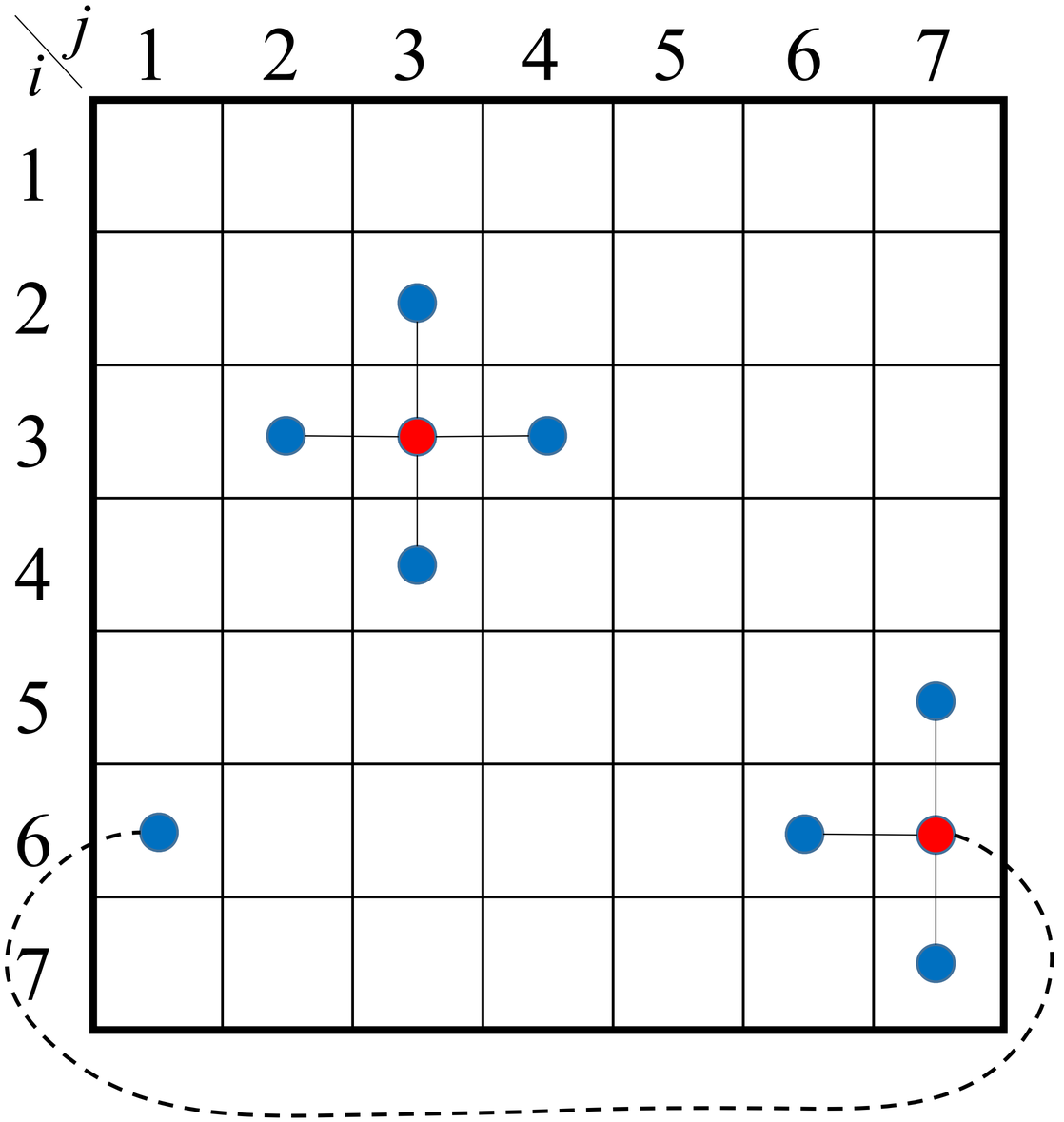
Starting from one possible configuration of the dipoles:

1) Choose 1 dipole at random and change its spin.
2) Compute δE_s, the change in the energy from the original state to the new state.
3) Decide if we will accept this new state:
   a) if $\delta E_s < 0$, accept the energetically more favorable state
   b) if $e^{-\delta E_s / k_B T} >$ a random number in the range $[0,1]$, then accept the new state
      (Note: (b) includes (a))

The Ising Model

- The Ising model assumes only nearest neighbor interactions are significant.
- It also uses *"periodic boundary conditions"* to deal with the edges of the lattice.

Note that the single index designating which spin we are interested in will be replaced with a pair of indices giving the lattice location.

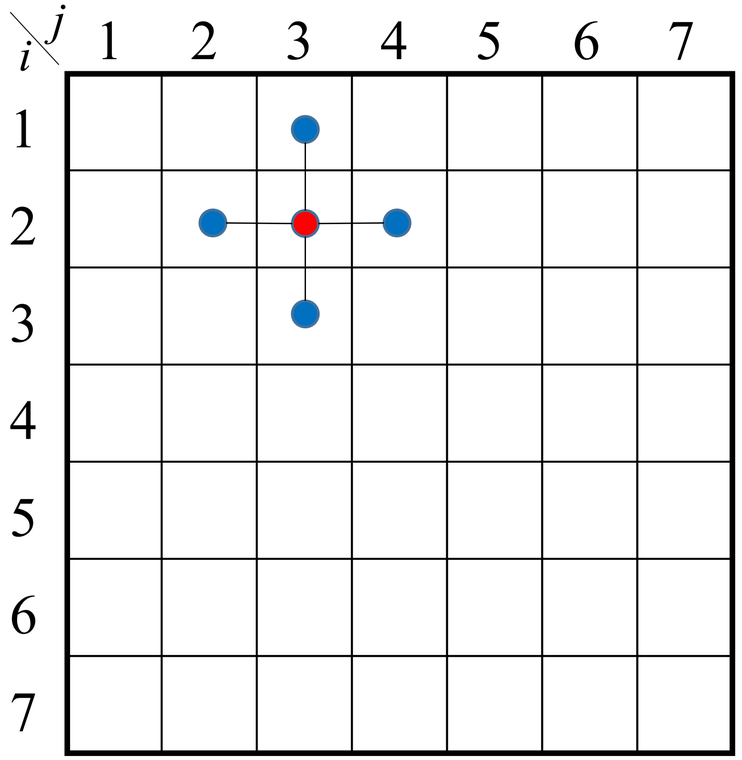


The Ising Model

Energy of the state:

$$E_s = -\frac{1}{2} J \sum_{i,j} S_{ij} \left(S_{i+1,j} + S_{i-1,j} + S_{i,j+1} + S_{i,j-1} \right)$$

$$= -J \sum_{i,j} S_{ij} \left(S_{i+1,j} + S_{i,j+1} \right)$$



If we flip the spin at only 1 location (i,j) then

$$\delta E_s = 2JS_{ij} \left(S_{i+1,j} + S_{i-1,j} + S_{i,j+1} + S_{i,j-1} \right)$$

"old", un-flipped spin value

Magnetization:

$$M_s \equiv \sum_{i,j} S_{ij}$$

The Metropolis Algorithm:
Computing Expectation Values

Given an $N \times N$ lattice:

- Define a *"sweep"* as applying the Metropolis algorithm to $N \times N$ randomly chosen spins in the lattice.
- Make repeated sweeps through the lattice of spins, updating the spins. This simulates what will happen to the system as it evolves in time!
- Sample the system every N_s sweeps, measuring E_s and M_s each time. *(Note: you can easily update both E_s and M_s in each Metropolis step)*
  "Accumulate" $\sum E_s$, $\sum E_s^2$, $\sum M_s$, and $\sum M_s^2$ from each sample.
- After n_s samples:

$$\langle E_s \rangle = \frac{\sum E_s}{n_s}, \quad \langle E_s^2 \rangle = \frac{\sum E_s^2}{n_s}, \quad \ldots$$

The Metropolis Algorithm:
Computing Expectation Values

- Note: We really want to look at $\langle E \rangle / N \times N$, etc. so the values remain close if we change the value of N.
- Caution: Before you collect data for $\langle E \rangle$, etc., let the system *"thermalize".* The expectation value will only be meaningful after the system has come into equilibrium.
- Can assume $J = k_B = 1$.
- The interesting range of temperature is $T \sim 0 — 5$.
- Test your algorithm for small N, then run the algorithm for $N \sim 32$ or 64.

The Ising Model

Expectation Values to compute

- Energy/spin: $\langle E \rangle / N \times N$
- Magnetization/spin: $\langle |M| \rangle / N \times N$
- Specific Heat/spin: $C_V / N \times N$

$$C_V = \frac{\partial \langle E_s \rangle}{\partial T} = \cdots = \frac{1}{k_B T^2} \left[\langle E_s^2 \rangle - \langle E_s \rangle^2 \right]$$

- Magnetic Susceptibility/spin: $\chi / N \times N$

$$\chi = \frac{\partial \langle\, |M| \,\rangle}{\partial T} = \cdots = \frac{1}{k_B T} \left[\langle M^2 \rangle - \langle M \rangle^2 \right]$$