The Ising Model

Today we will switch topics and discuss one of the most studied models in statistical physics the **Ising Model**

- Some applications:
  - Magnetism (the original application)
  - Liquid-gas transition
  - Binary alloys (can be generalized to multiple components)
- Onsager solved the 2D square lattice (1D is easy!)
- Used to develop *renormalization group theory* of phase transitions in 1970's.
- Critical slowing down and "cluster methods".

Figures from Landau and Binder, MC Simulations in Statistical Physics, 2000 (LB)

Why should you care?

Why should you care?

1. Prototypical model of phase transitions

Why should you care?

1. Prototypical model of phase transitions

2. Map to many other systems (gas, etc.)

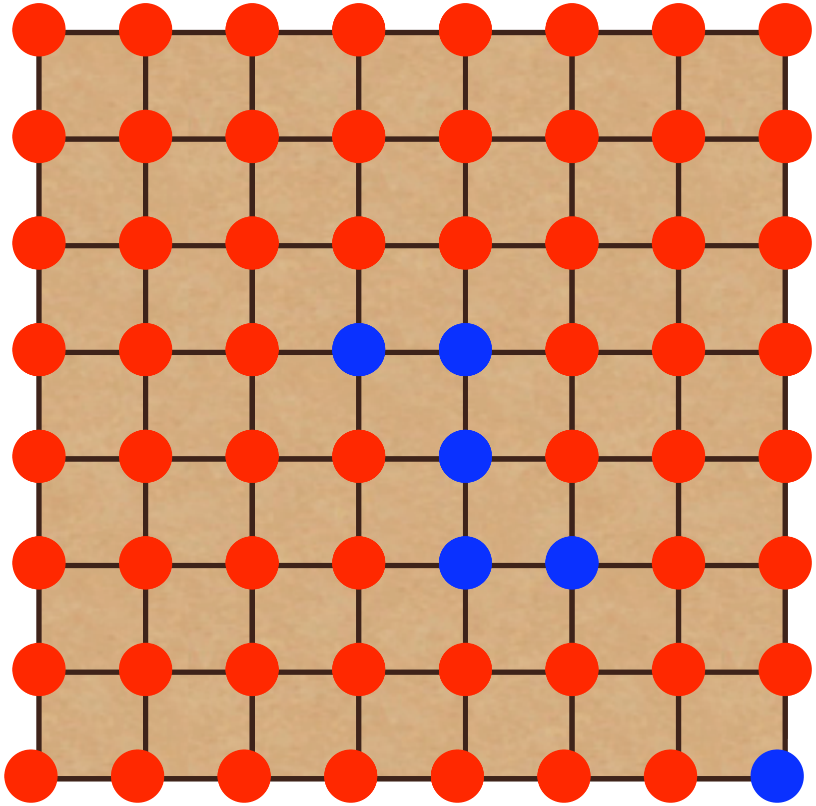
Why should you care?

1. Prototypical model of phase transitions

2. Map to many other systems (gas, etc.)

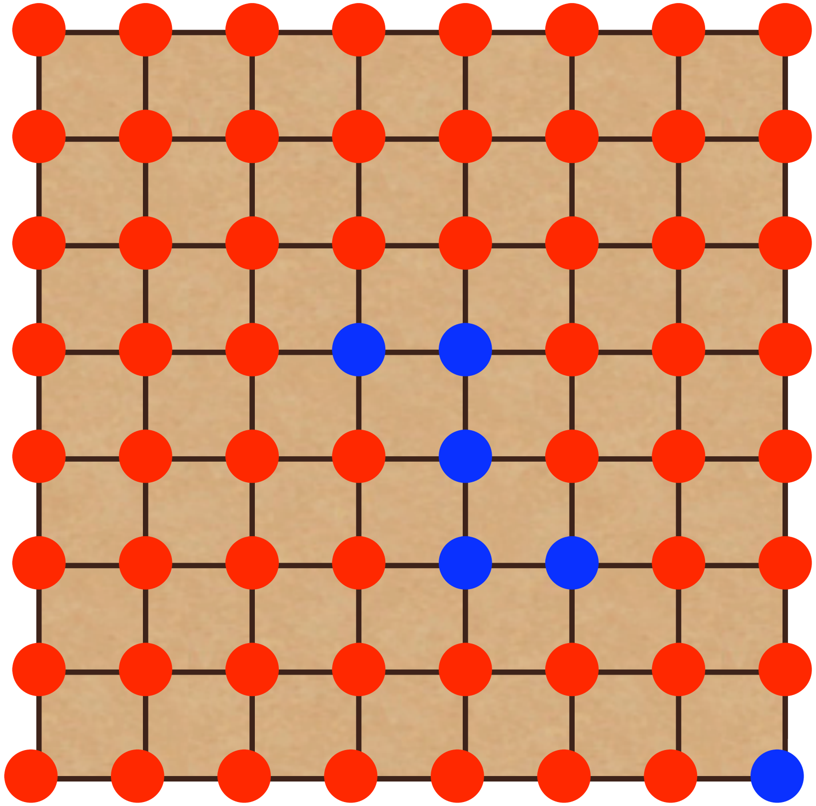
3. Techniques critical to other simulations.

The Model



$$H = -J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

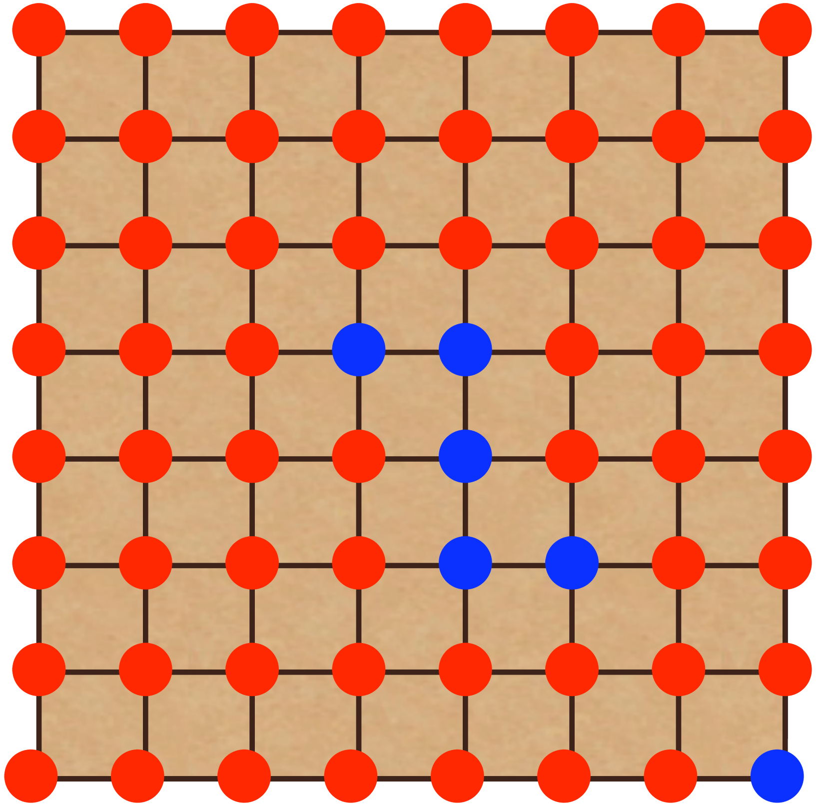
The Model



- Each lattice site has a single spin variable: $s_i = \pm 1$.

$$H = -J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

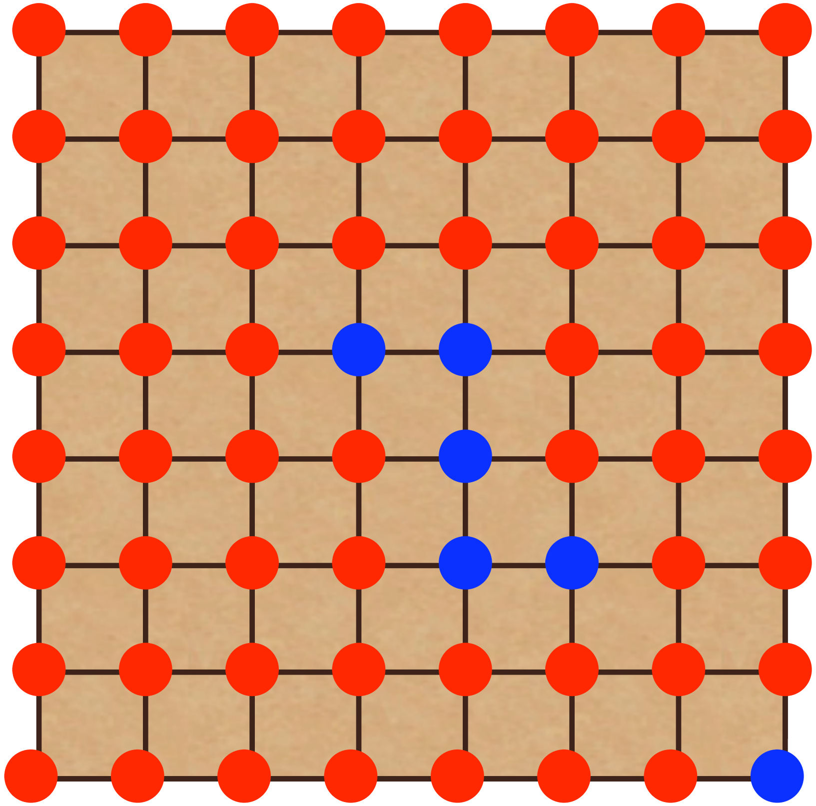
The Model



- Each lattice site has a single spin variable: $s_i = \pm 1$.
- with magnetic field **h**

$$H = -J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

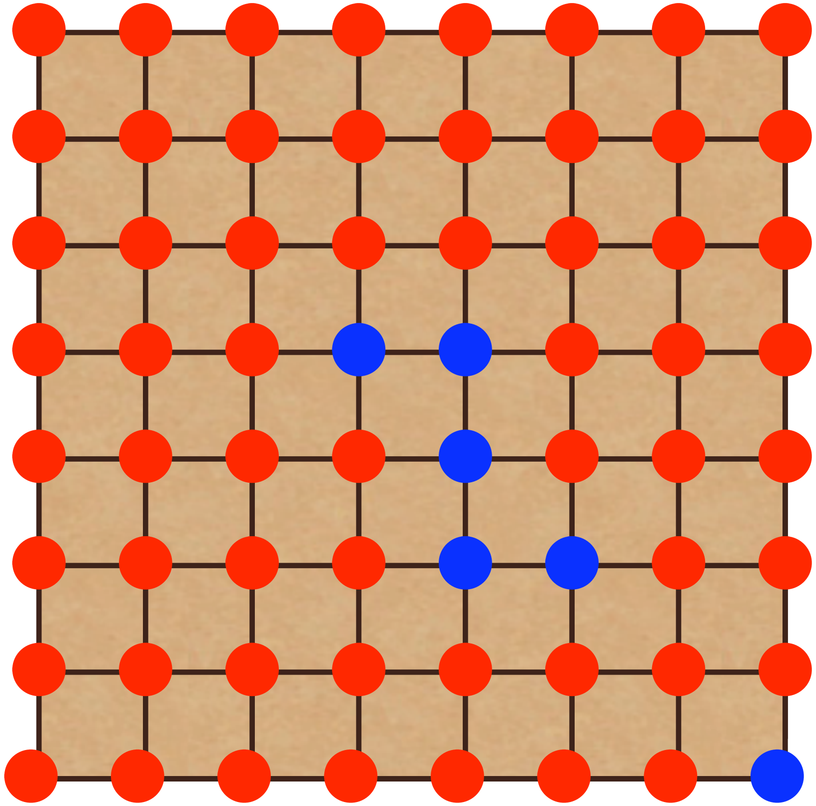
The Model



- Each lattice site has a single spin variable: $s_i = \pm 1$.
- with magnetic field **h**
- J is the nearest neighbors (i,j) coupling:
  - If J>0, ferromagnetic
  - If J<0, antiferromagnetic

$$H = -J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

The Model

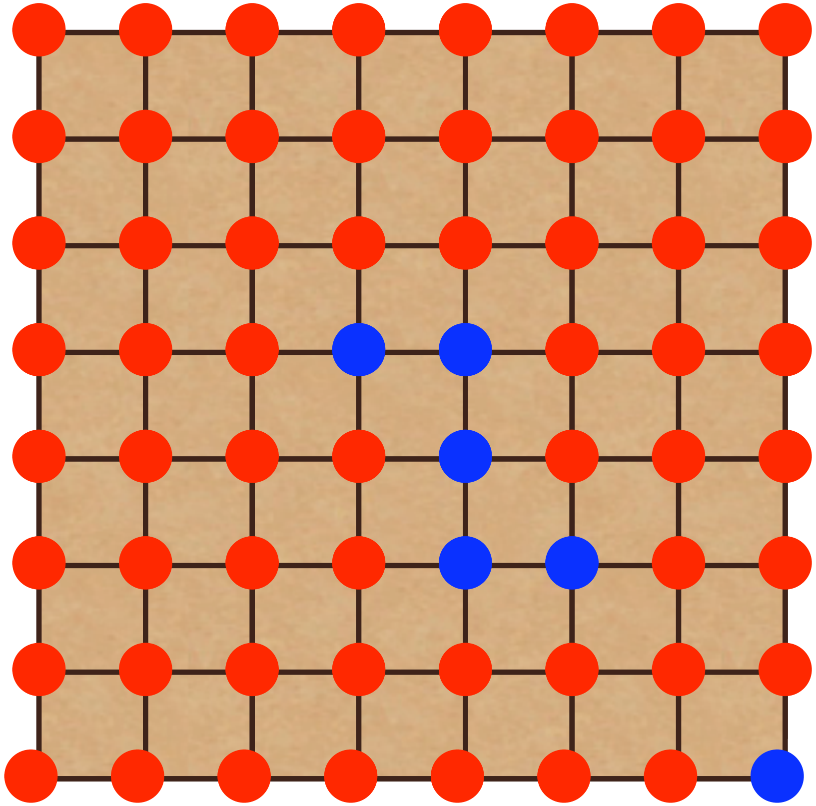


- Each lattice site has a single spin variable: $s_i = \pm 1$.
- with magnetic field **h**
- J is the nearest neighbors (i,j) coupling:
  - If J>0, ferromagnetic
  - If J<0, antiferromagnetic

•Picture of spins at the critical temperature T_c. (Note the connected (percolated) clusters.)

$$H = -J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

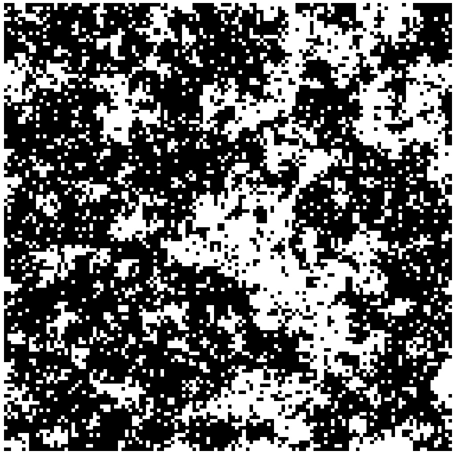
The Model



- Each lattice site has a single spin variable: $s_i = \pm 1$.
- with magnetic field **h**
- J is the nearest neighbors (i,j) coupling:
  - If J>0, ferromagnetic
  - If J<0, antiferromagnetic

•Picture of spins at the critical temperature T_c. (Note the connected (percolated) clusters.)

$$H = -J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$



Competition between Parameters

Competition between Parameters

$$H = J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

Competition between Parameters

h	J	T
Wants all red spins	Wants like color neighbors	Wants to randomize spins.

$$H = J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

Competition between Parameters

- **High-T phase:** spins are random (uncorrelated).

h	J	T
Wants all red spins	Wants like color neighbors	Wants to randomize spins.

$$H = J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

Competition between Parameters

- **High-T phase:** spins are random (uncorrelated).
- **T > T_c phase near T_c:** spins are random but correlated: magnetic short-range (local) order.

h	J	T
Wants all red spins	Wants like color neighbors	Wants to randomize spins.

$$H = J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

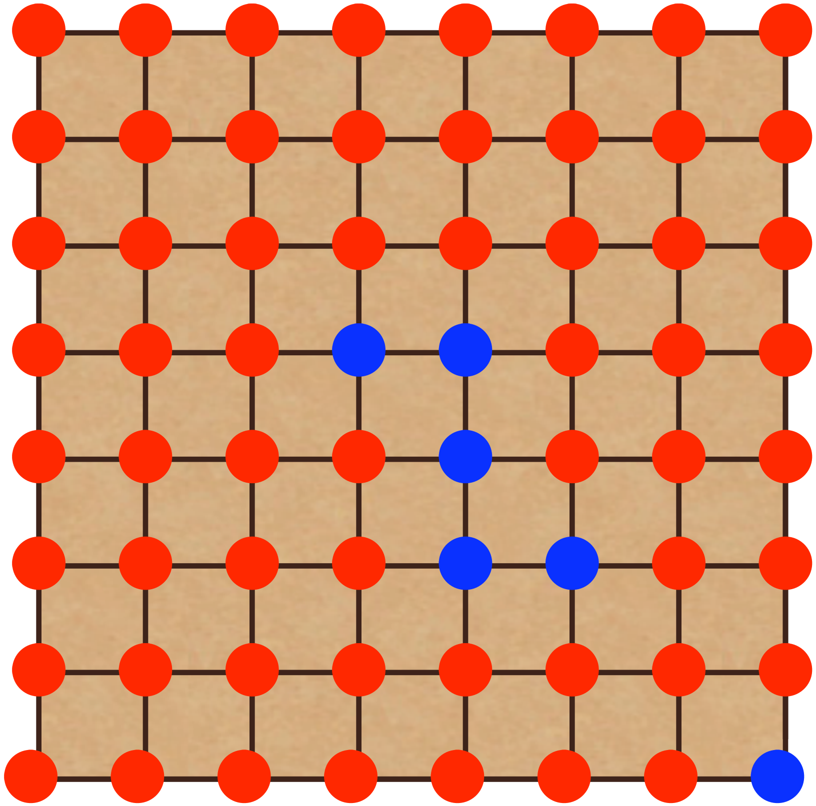
Competition between Parameters

- **High-T phase:** spins are random (uncorrelated).
- **T > T_c phase near T_c:** spins are random but correlated: magnetic short-range (local) order.
- **Low-T (T~0) phase:** spins are aligned (fully correlated).

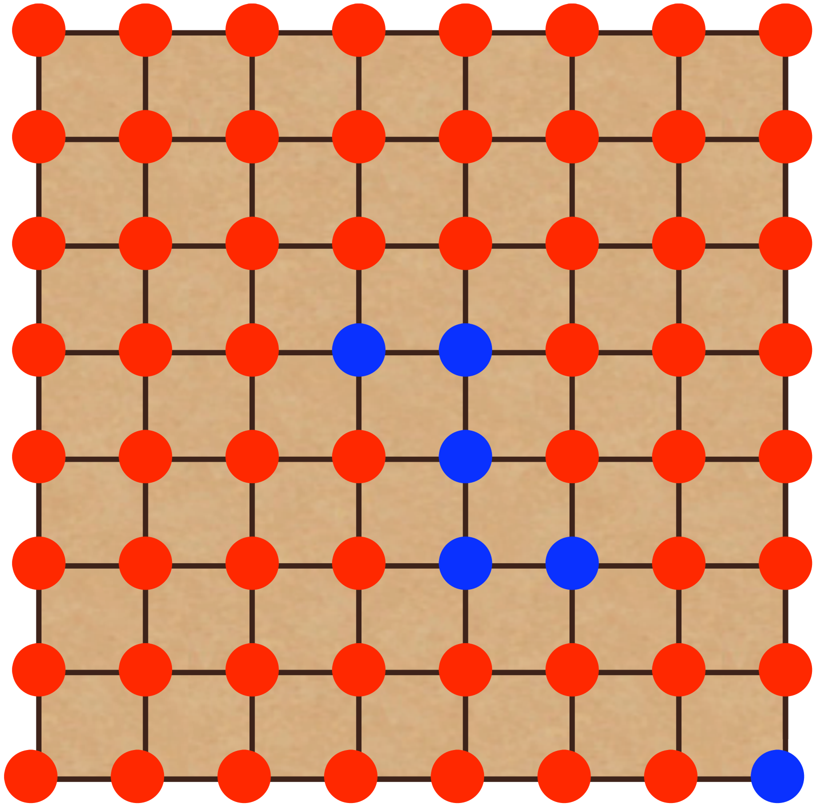
h	J	T
Wants all red spins	Wants like color neighbors	Wants to randomize spins.

$$H = J \sum_{\langle i,j \rangle} s_i \cdot s_j + \sum_i h \cdot s_i$$

What do you measure



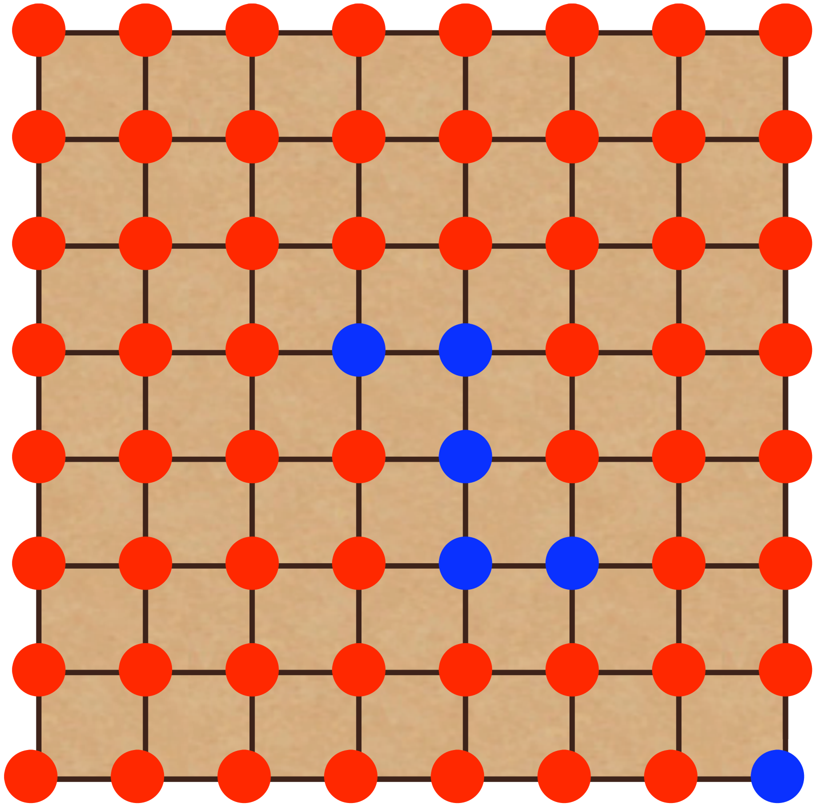
What do you measure



- Magnetization: 52

$$m(r) = \langle s(r) \rangle$$

What do you measure



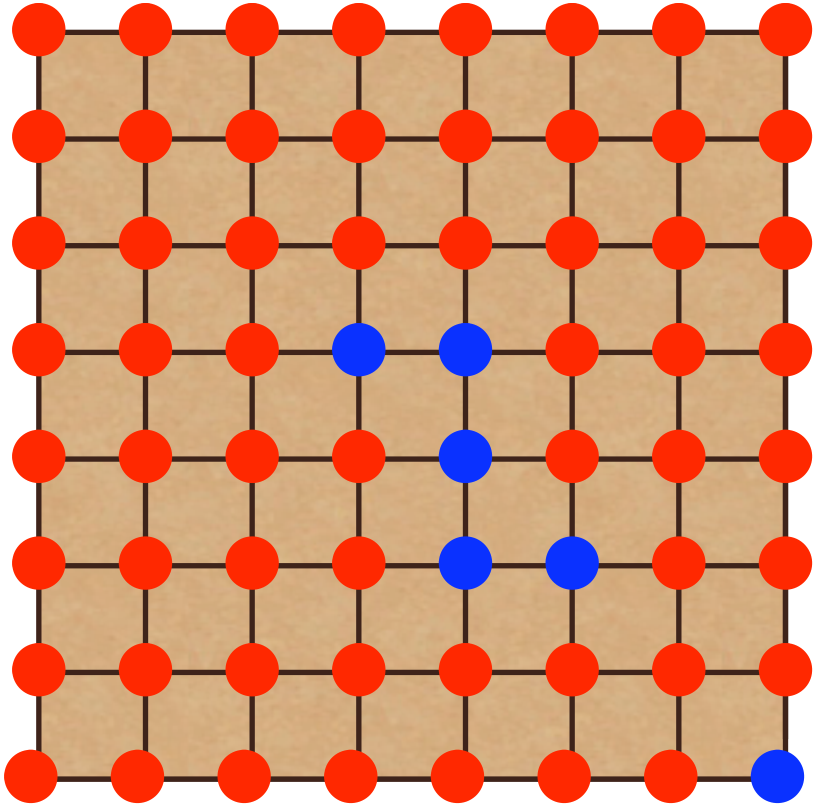
- Magnetization: 52

$$m(r) = \langle s(r) \rangle$$

- Magnetization squared (why?)

$$m^2(r) = \langle s^2(r) \rangle$$

What do you measure



- Magnetization: 52

$$m(r) = \langle s(r) \rangle$$

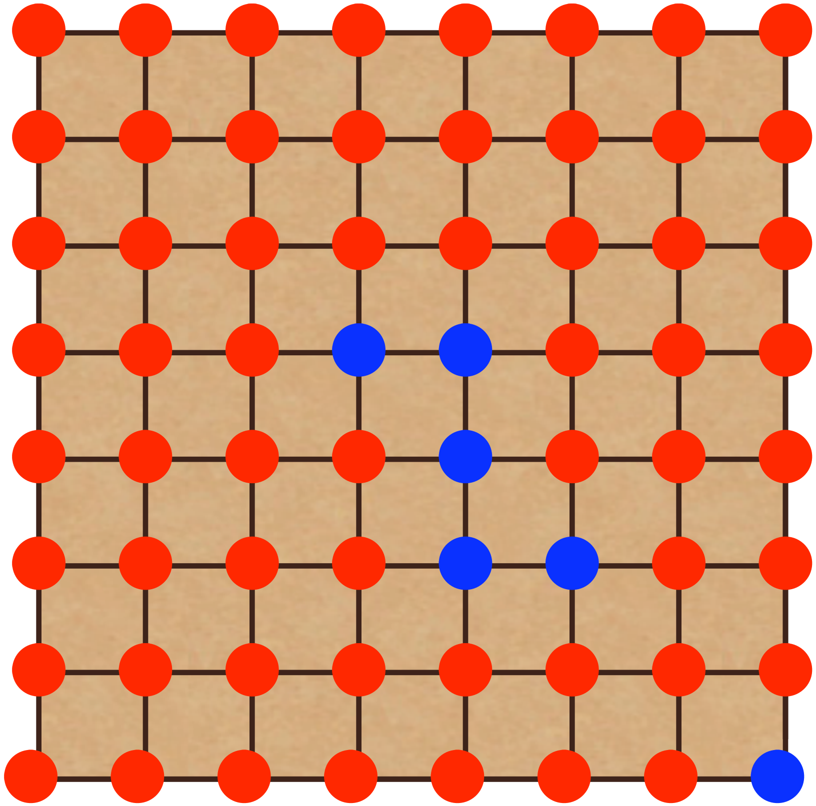
- Magnetization squared (why?)

$$m^2(r) = \langle s^2(r) \rangle$$

- Magnetic Susceptibility

$$\frac{dm(r)}{dh(r')}\Big|_{h\to 0} = \beta\chi(r - r')$$

What do you measure



- Magnetization: 52

$$m(r) = \langle s(r) \rangle$$

- Magnetization squared (why?)

$$m^2(r) = \langle s^2(r) \rangle$$

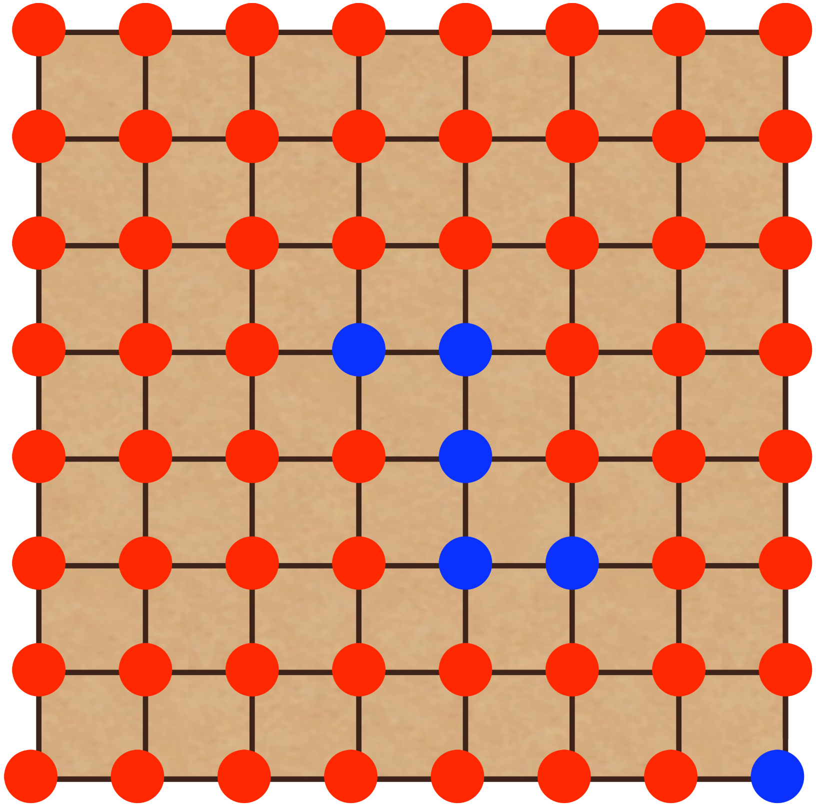
- Magnetic Susceptibility

$$\frac{dm(r)}{dh(r')}|_{h \to 0} = \beta \chi(r - r')$$

- Correlation functions

$$\chi(r - r') = \langle s(r) s(r') \rangle - \langle s(r) \rangle \langle s(r') \rangle$$

What do you measure



- Magnetization: 52

$$m(r) = \langle s(r) \rangle$$

- Magnetization squared (why?)

$$m^2(r) = \langle s^2(r) \rangle$$

- Magnetic Susceptibility

$$\frac{dm(r)}{dh(r')}\Big|_{h \to 0} = \beta \chi(r - r')$$

- Correlation functions

$$\chi(r - r') = \langle s(r) s(r') \rangle - \langle s(r) \rangle \langle s(r') \rangle$$

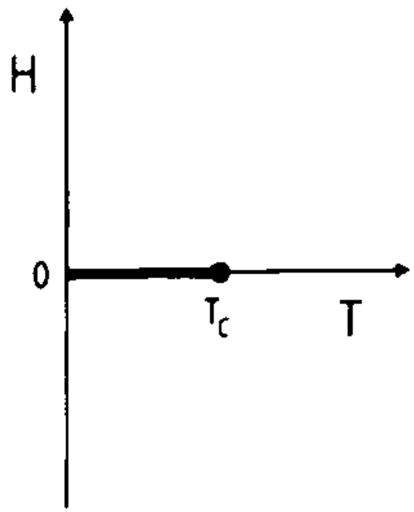
- Dynamics?

Phase Transitions

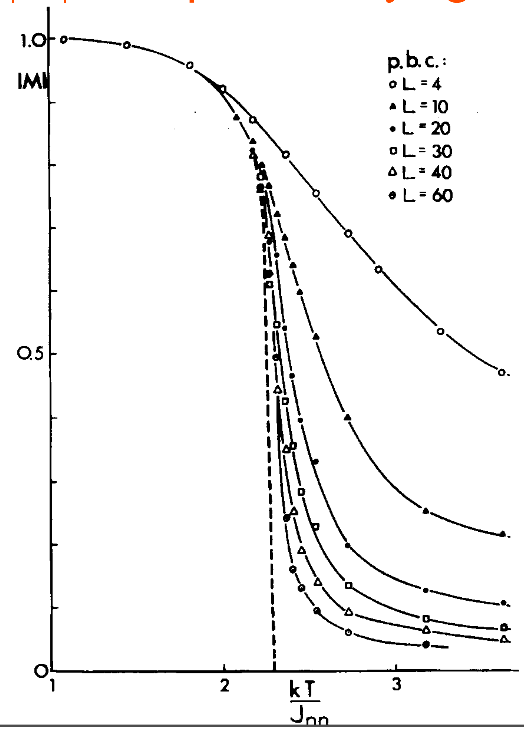
Phase Transitions

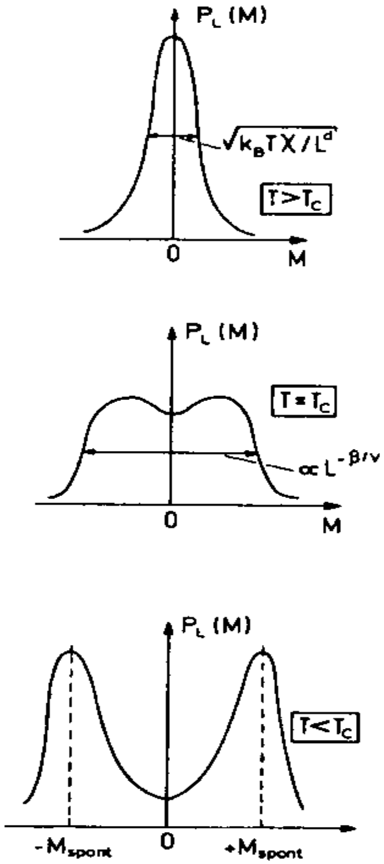
- Concepts and understanding are universal.

  Apply to all phase transitions of similar type.
- Order parameter is *average* magnetization
- Let's understand phases:
  - Change T (H=0)
  - Bigger => sharper transition
  - Second order



|M| vs. 1/βJ for varying L





Second Order Phase Transitions

- Critical Point! and Universal Scaling!
- In ordered phase, spin is correlated over long distance.
- At critical point, fluctuations of all scales.

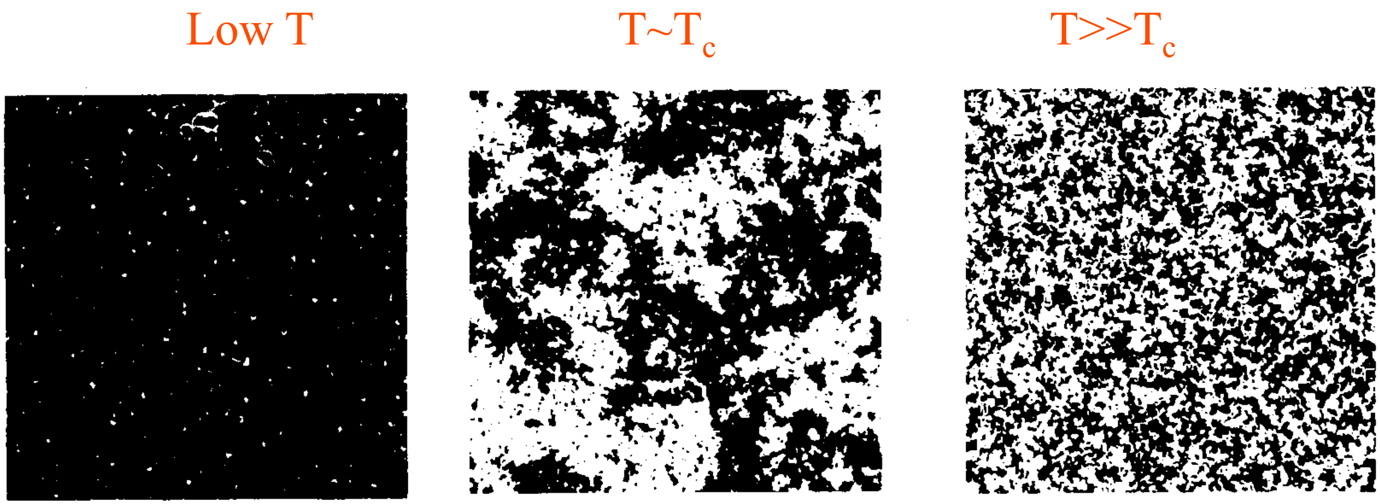


Fig. 4.1 Typical spin configurations for the two-dimensional Ising square lattice: (left) $T \ll T_c$; (center) $T \sim T_c$; (right) $T \gg T_c$.

Magnetization Scaling depends on T:

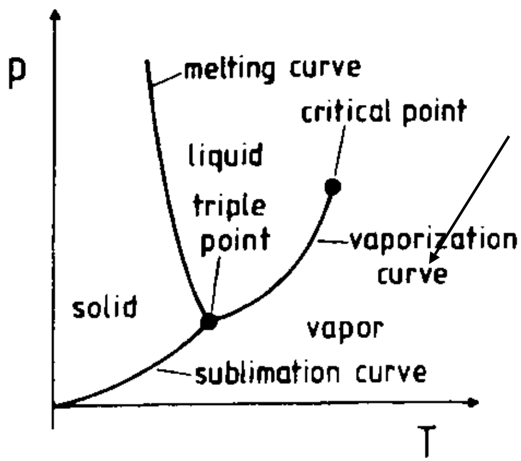
$M \propto (T_c - T)^{\beta}$ for $T < T_c$

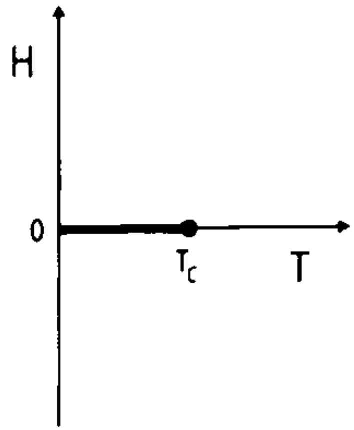
$\beta = 0.125$ for D=2.

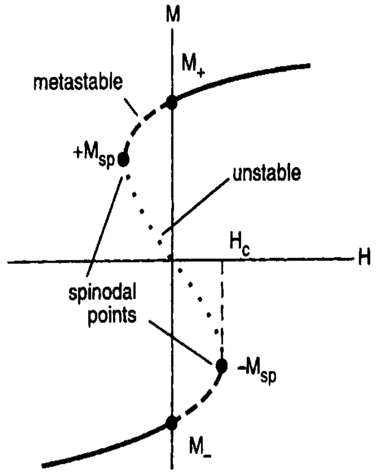
$\beta = 0.325$ for D=3.

1'st order phase diagrams

- A ***first-order transition*** (where there is a discontinuous jump in **M**) occurs as H passes through zero for $T<T_c$.
- Similar to **LJ phase diagram**. Magnetic field=pressure.







Mapping to Other Models

Mapping liquid-gas to Ising

- For ***liquid-gas*** transition let n(r) be the density at lattice site r and have two values n(r)=(0,1).

$$E = \sum_{(i,j)} v_{ij} n_i n_j + \mu \sum_i n_i$$

- Let's map this into the Ising model spin variables:

$$s = 2n - 1 \quad \text{or} \quad n = \frac{1}{2}(s+1)$$

$$H = \frac{v}{4} \sum_{(i,j)} s_i s_j + \frac{(v+\mu)}{2} \sum_i s_i + c$$

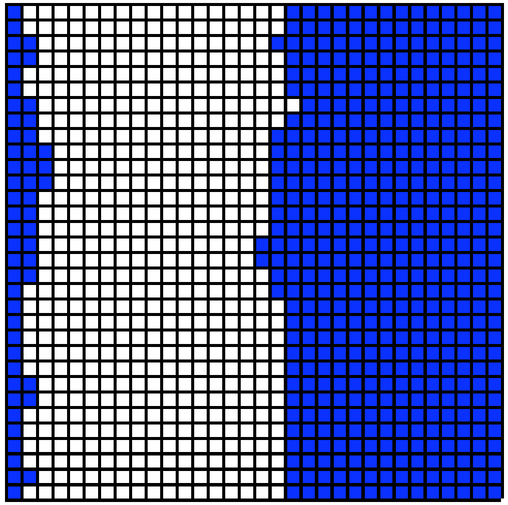
$$J = -v/4$$

$$h = -(v+\mu)/2$$

$$M = \frac{1}{N} \sum_i s_i \qquad \langle \mathrm{n} \rangle = \frac{1}{N} \sum_i n_i = \frac{1}{2}(M+1)$$

Surfaces/Boundary Conditions

- By quenching quickly we may catch a “trapped” surface.
- Topological excitation.
- You can see steps, etc.



- Can use *twisted boundary conditions* to study a liquid-gas surface without worrying about it disappearing.
- Just put -J along one plane (side). Antiferromagnetic interaction along one plane.

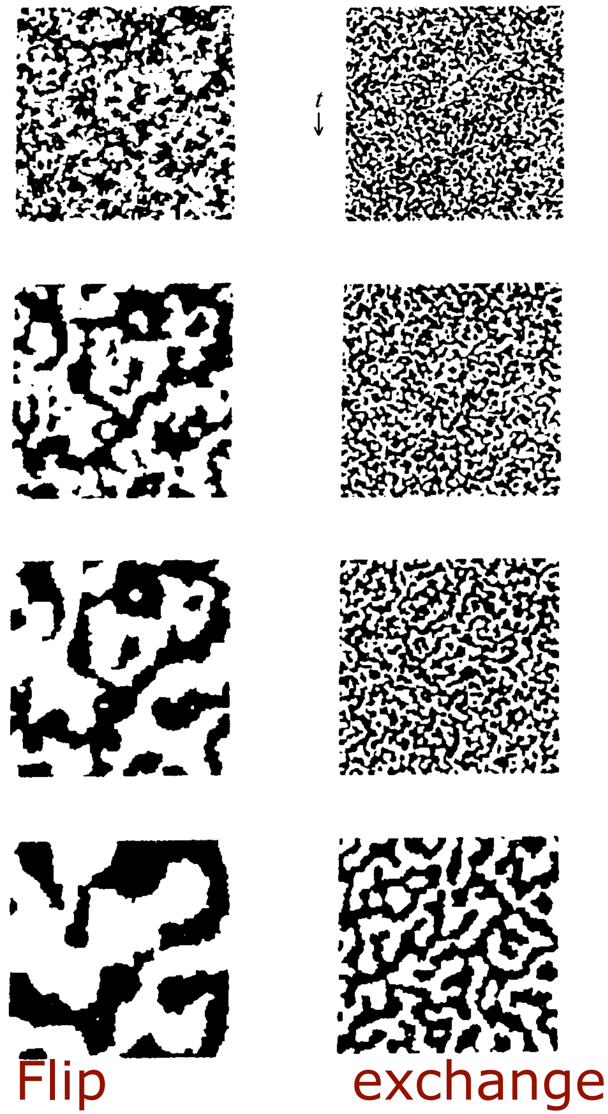
$$H = -\sum_{(i,j)} J_{ij} s_i s_j$$

$$J_{ij} = \begin{cases} J & i \neq 0 \\ -J & i = 0 \end{cases}$$

Spinoidal decomposition

Suppose spin flips only locally.

- Model for phase separation such as a binary "alloy" (oil and vinegar).
- Dynamics depends on whether the spin is conserved
  - Spin flip (left)
  - Spin exchange (right). conserves particle number
- Transition appears through a coarsening of the separation.
- Becomes slower and slower as the transition proceeds.
  - Critical Slowing down.



$T=0.6T_c$

Simulations

Simulations

- "Naive Metropolis"
- Heat Bath
- Kawaski Dyamics
- Kinetic Monte Carlo

- Cluster Moves
  - Wolff
  - Swedsen
- Worm Algorithms

Being pedantic

MC does integrals of the form $I(x) = \frac{1}{S}\int p(x)O(x)dx$

where $S = \int p(x)dx$

We want $M^2(x) = \frac{1}{Z}\int \exp(-\beta J \sum_{\langle i,j\rangle} s_i s_j) \sum_i s_i^2 di$

where $Z = \int \exp(-\beta J \sum_{\langle i,j\rangle} s_i s_j) di$

So... we need to sample a configuration of spins with probability $\exp(-\beta J \sum_{\langle i,j\rangle} s_i s_j)$ using monte carlo

Being pedantic

MC does integrals of the form $I(x) = \frac{1}{S}\int p(x) O(x) dx$

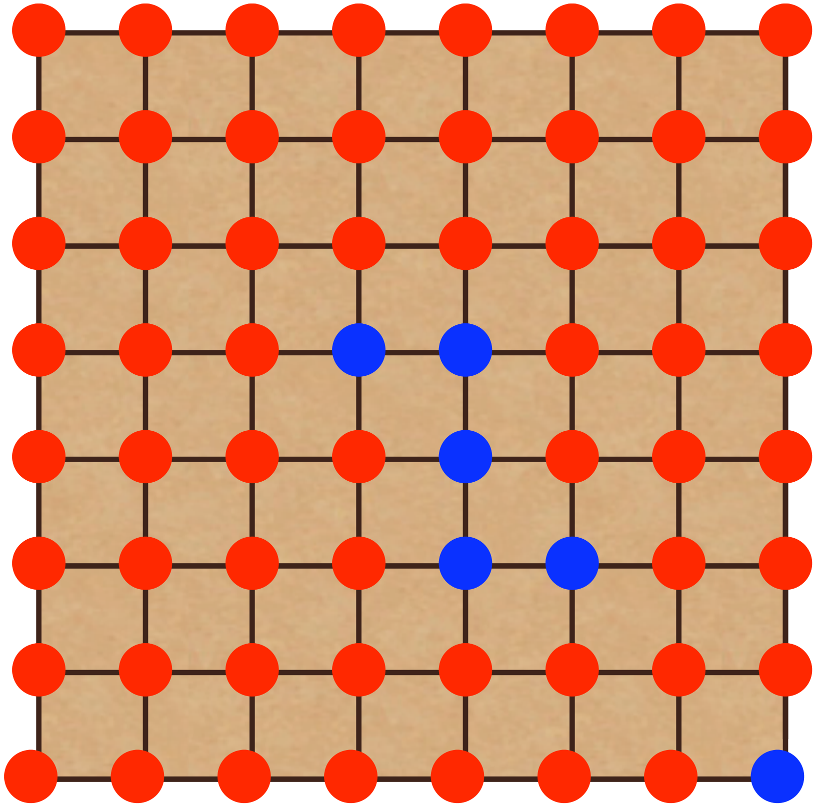
where $S = \int p(x) dx$

We want $M^2(x) = \frac{1}{Z}\int \exp(-\beta J \sum_{\langle i,j\rangle} s_i s_j) \sum_i s_i^2 di$

where $Z = \int \exp(-\beta J \sum_{\langle i,j\rangle} s_i s_j) di$

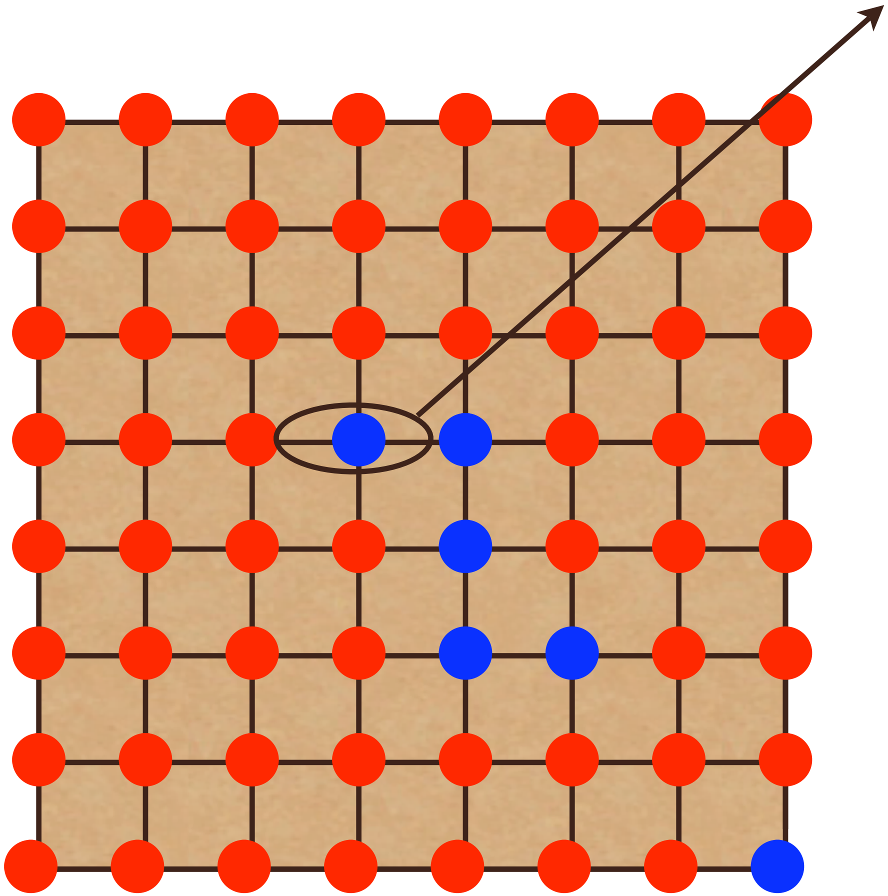
So... we need to sample a configuration of spins with probability $\exp(-\beta J \sum_{\langle i,j\rangle} s_i s_j)$ using monte carlo

Naive Metropolis



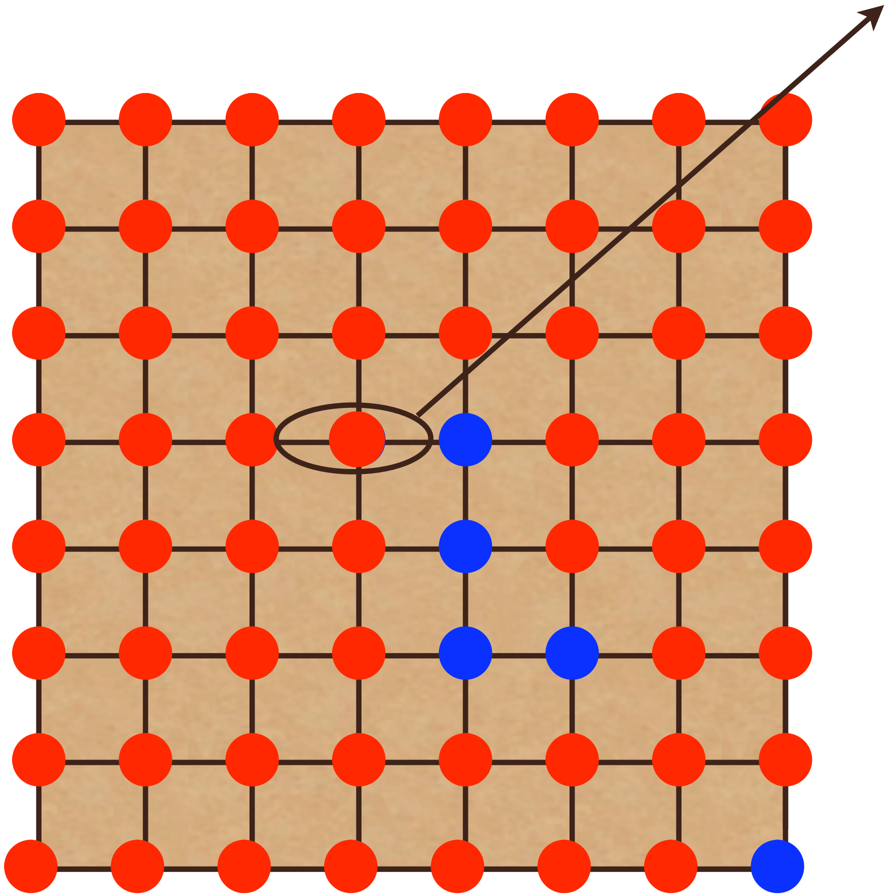
17

Naive Metropolis



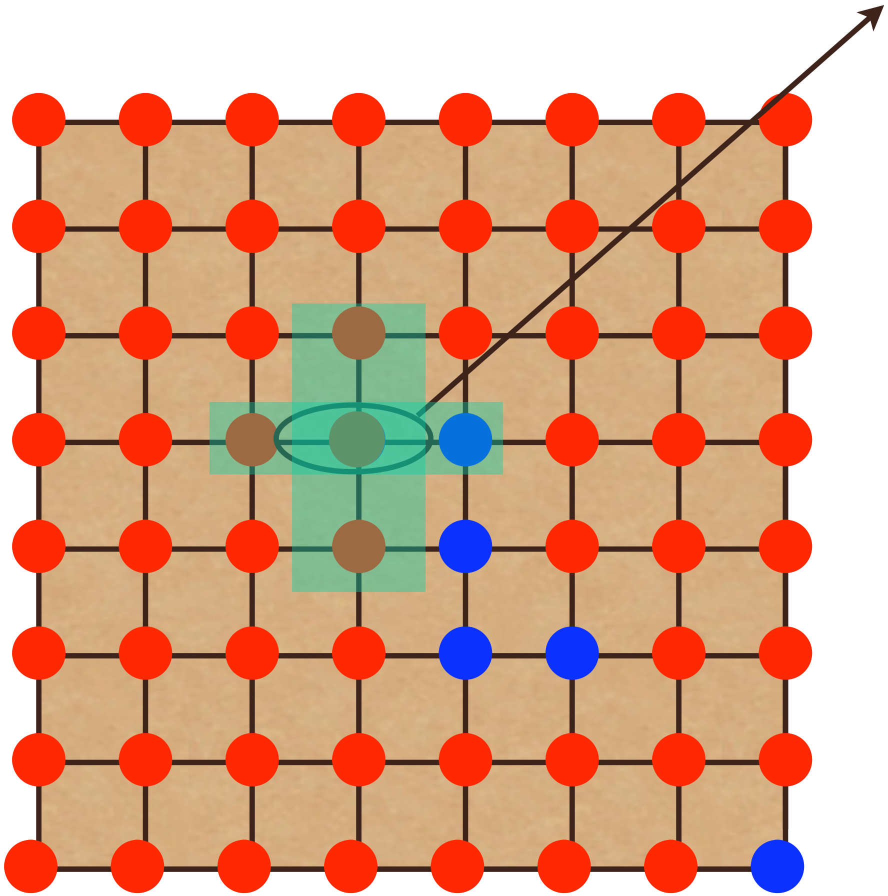
- Choose a spin at random

Naive Metropolis



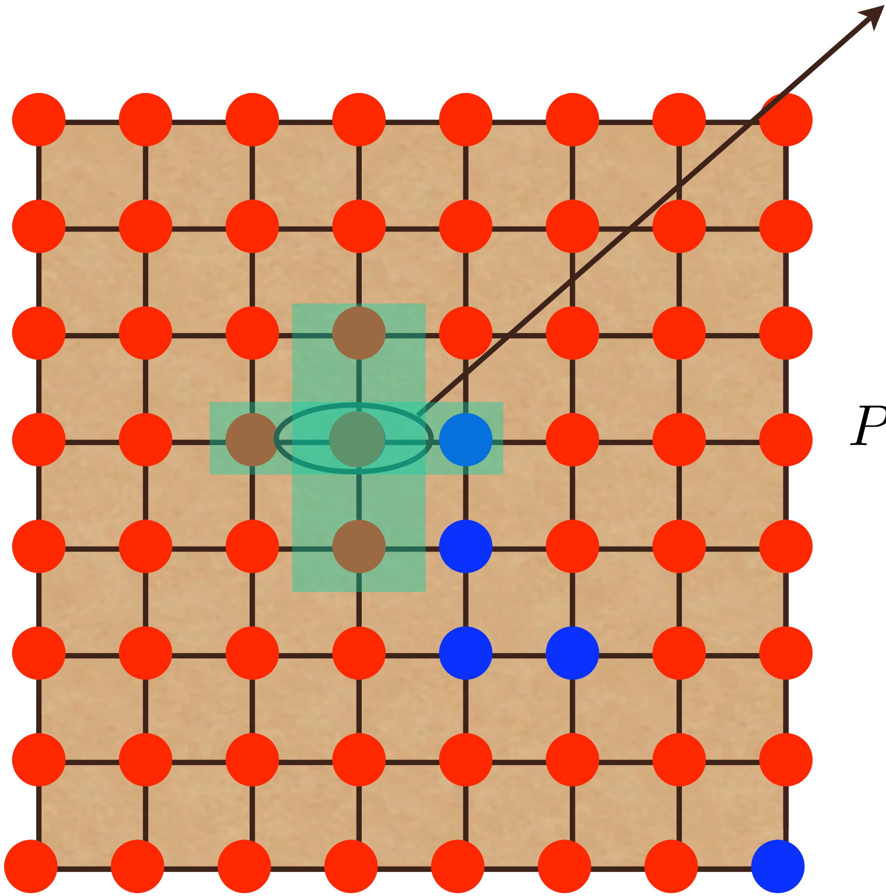
- Choose a spin at random
- Flip this spin

Naive Metropolis



- Choose a spin at random
- Flip this spin
- Calculate $\Delta E = J \sum_{\langle i,j \rangle} s_i s_j$

Naive Metropolis

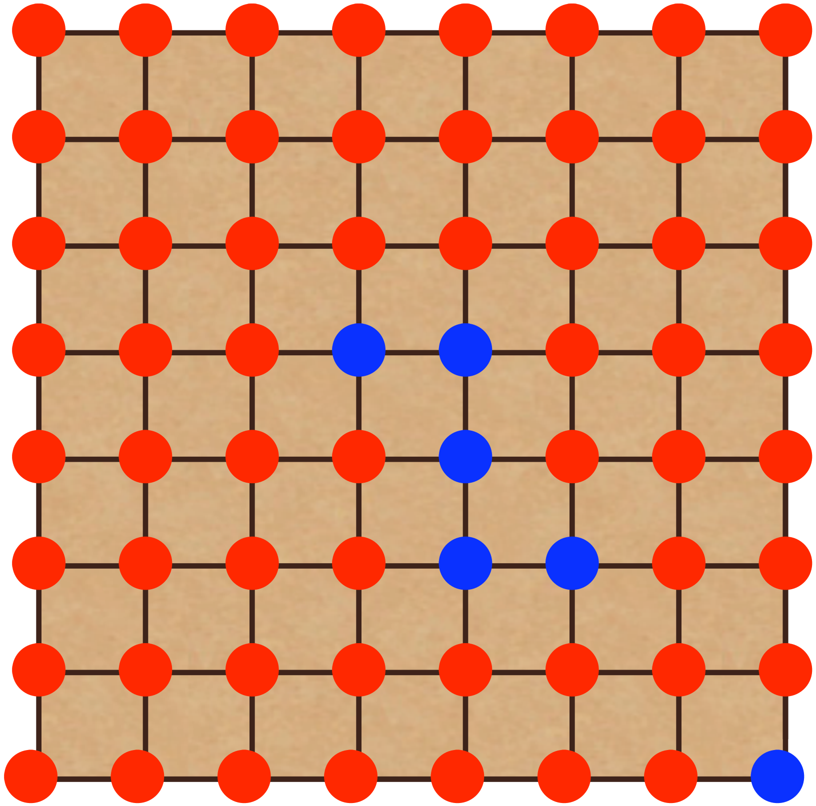


- Choose a spin at random
- Flip this spin
- Calculate $\Delta E = J \sum_{\langle i,j \rangle} s_i s_j$
- Accept with probability

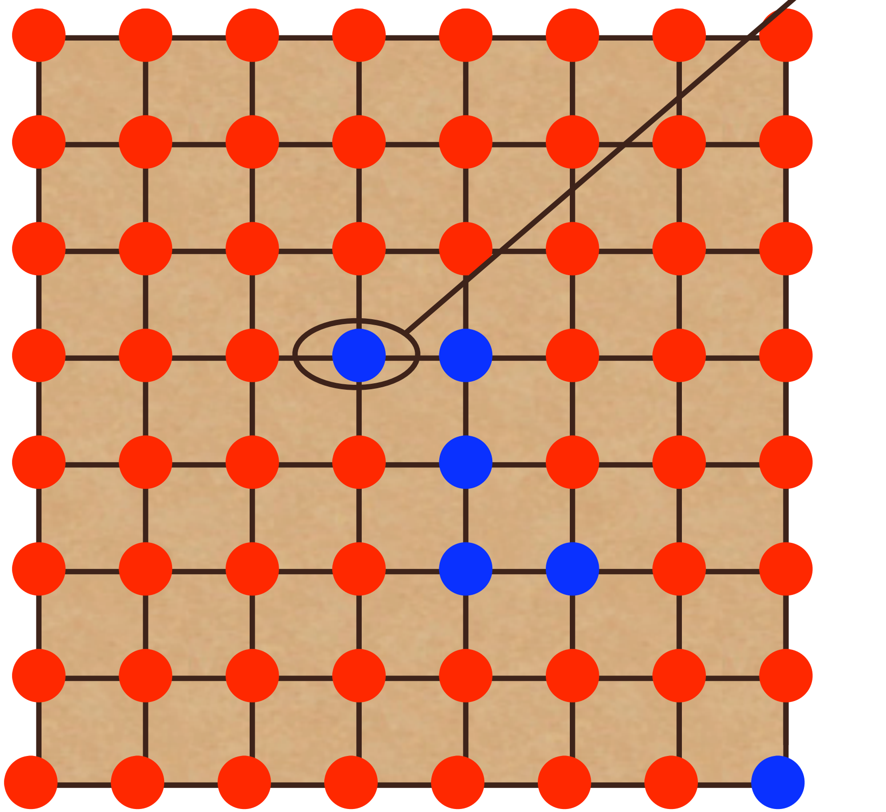
$$P_{\text{accept}} = min\left\{1, \exp(-\beta \Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

MSE Illinois   Physics Illinois

Heat Bath

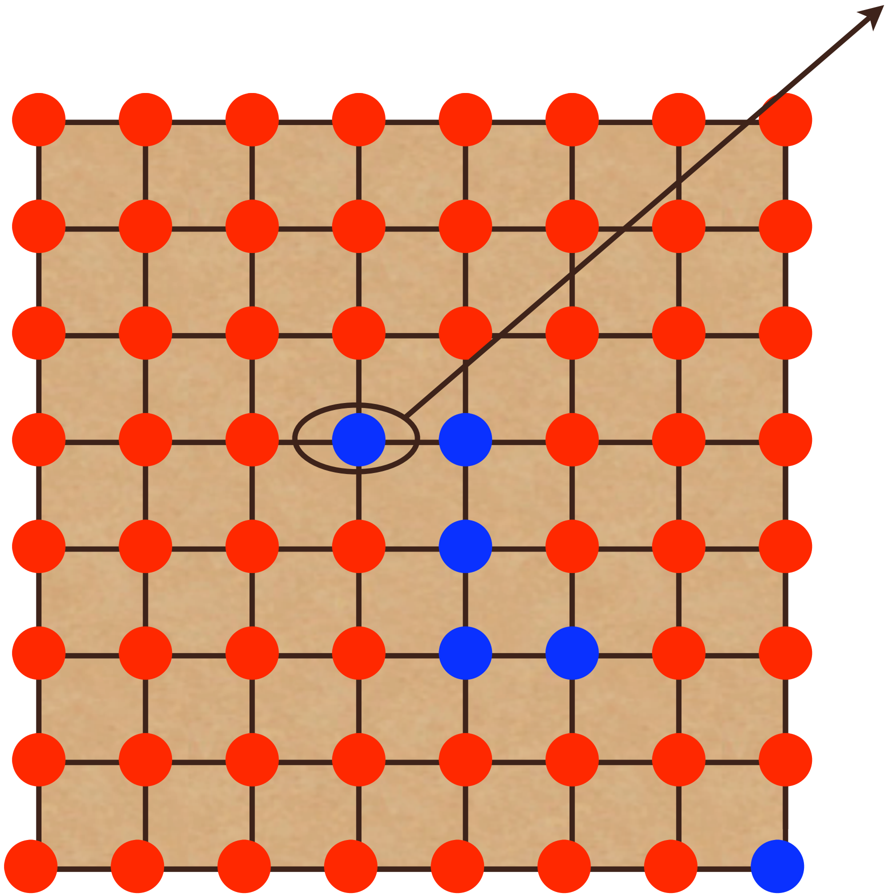


Heat Bath



- Choose a spin at random

Heat Bath

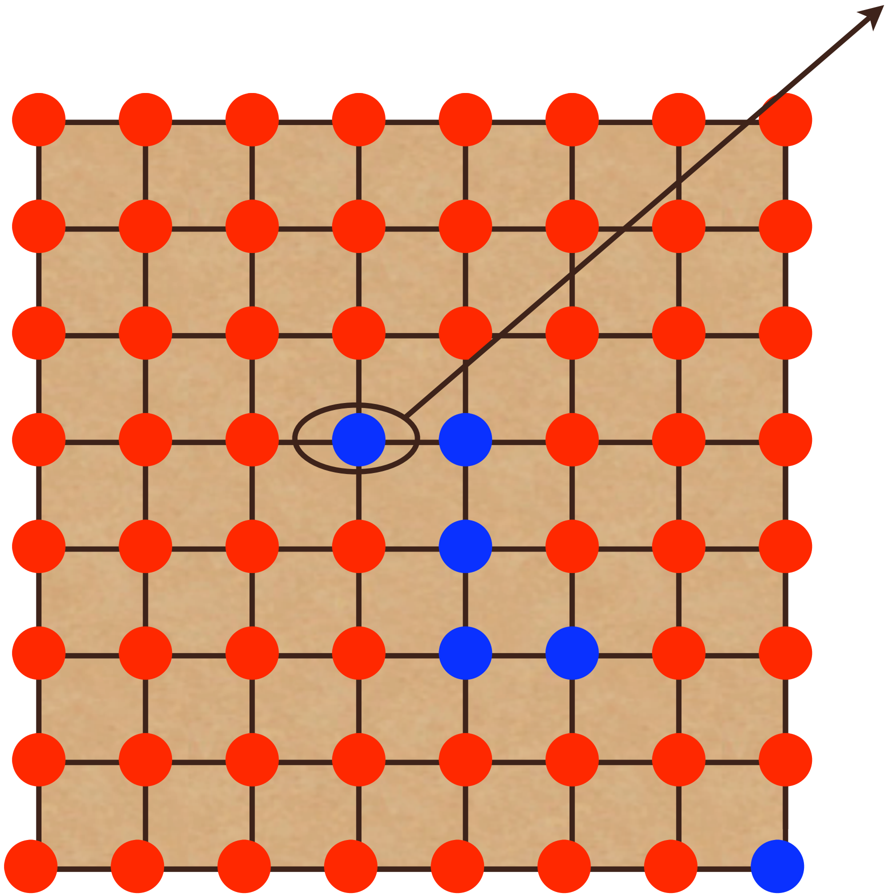


- Choose a spin at random
- Define:

$$p_r = \exp(-\beta J s_r(s_r + s_r + s_b + s_r))$$

$$p_b = \exp(-\beta J s_b(s_r + s_r + s_b + s_r))$$

Heat Bath



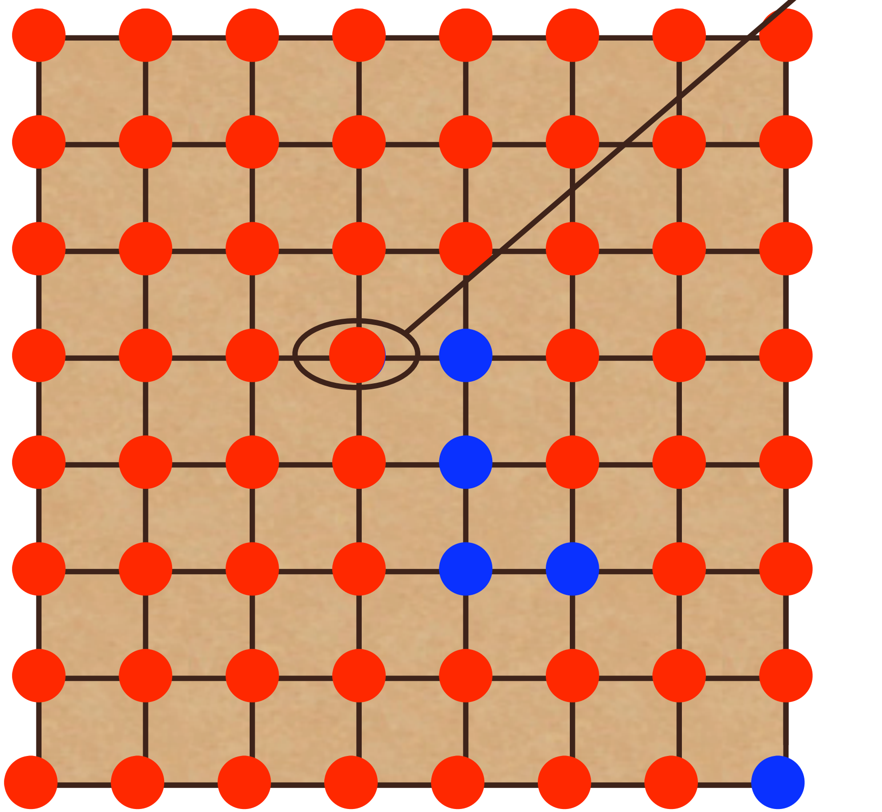
- Choose a spin at random
- Define:

$$p_r = \exp(-\beta J s_r (s_r + s_r + s_b + s_r))$$

$$p_b = \exp(-\beta J s_b (s_r + s_r + s_b + s_r))$$

- Make this spin

Heat Bath



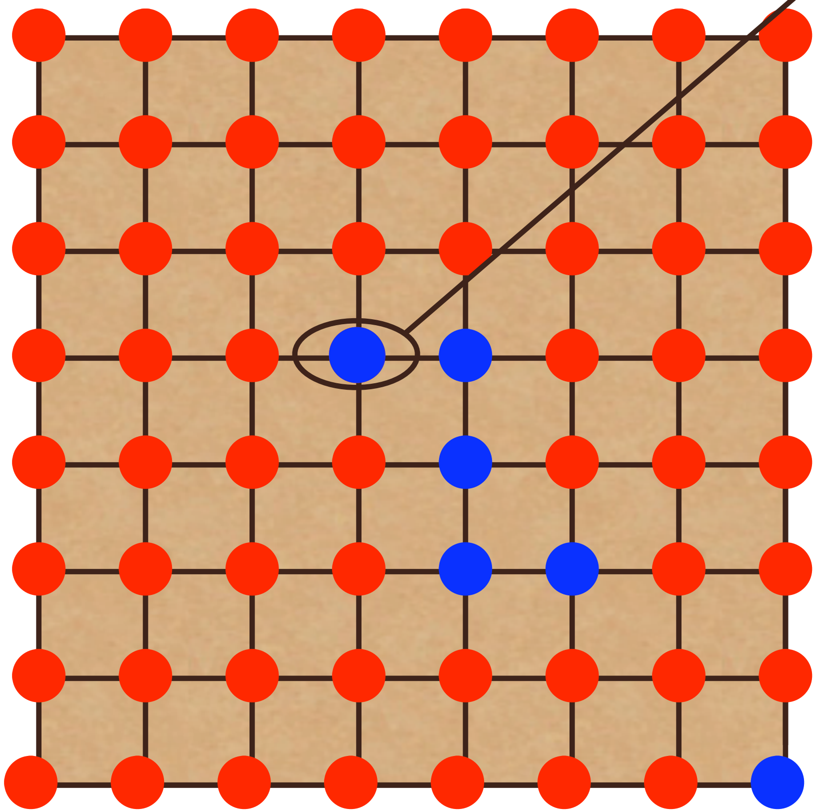
- Choose a spin at random
- Define:

$$p_r = \exp(-\beta J s_r(s_r + s_r + s_b + s_r))$$

$$p_b = \exp(-\beta J s_b(s_r + s_r + s_b + s_r))$$

- Make this spin
  - red: Probability $\frac{p_r}{p_r + p_b}$

Heat Bath



- Choose a spin at random
- Define:

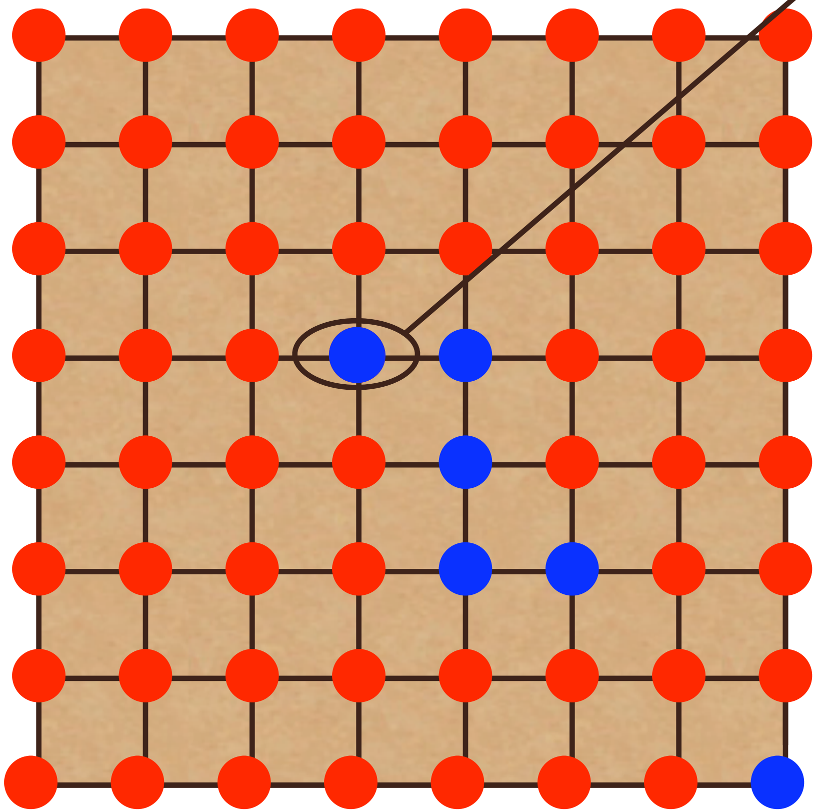
$$p_r = \exp(-\beta J s_r(s_r + s_r + s_b + s_r))$$

$$p_b = \exp(-\beta J s_b(s_r + s_r + s_b + s_r))$$

- Make this spin
  - red: Probability $\dfrac{p_r}{p_r + p_b}$
  - blue: Probability $\dfrac{p_b}{p_r + p_b}$

MSE Illinois | Physics Illinois

Heat Bath



- Choose a spin at random
- Define:

$$p_r = \exp(-\beta J s_r(s_r + s_r + s_b + s_r))$$

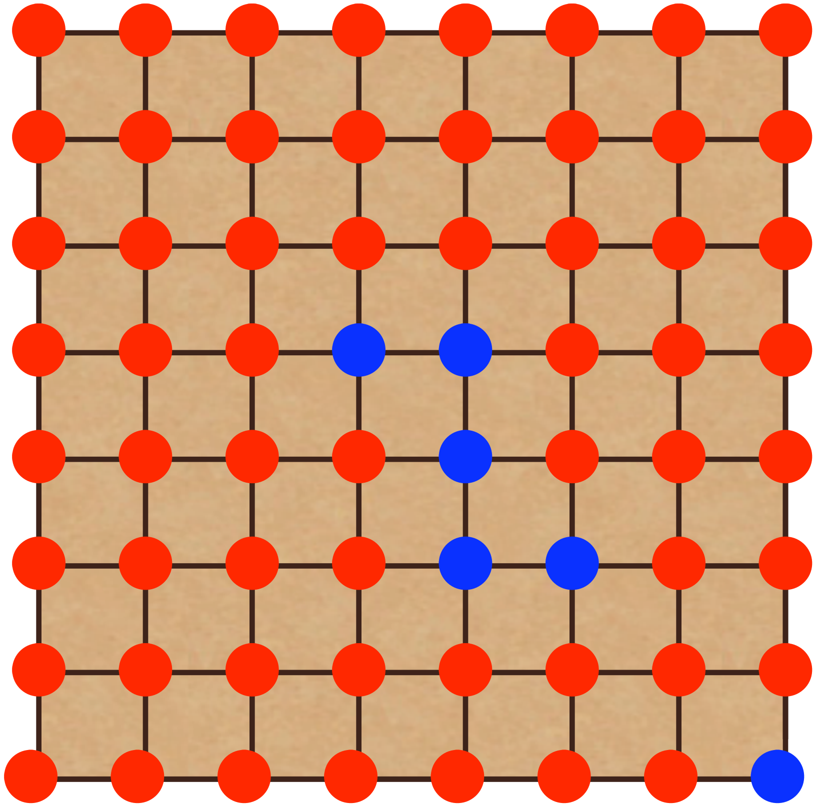
$$p_b = \exp(-\beta J s_b(s_r + s_r + s_b + s_r))$$

- Make this spin
  - red: Probability $\dfrac{p_r}{p_r + p_b}$
  - blue: Probability $\dfrac{p_b}{p_r + p_b}$
- Accept with probability

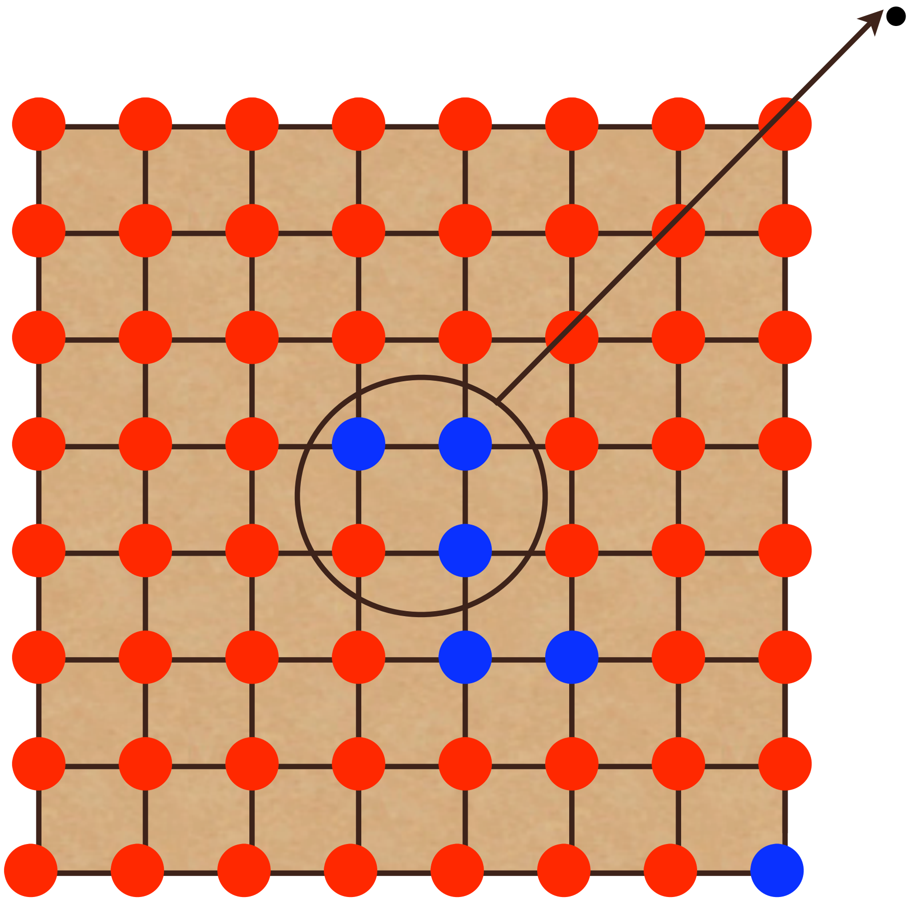
$$P_{\text{accept}} = min\left\{1, \exp(-\beta\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

MSE at Illinois | Physics Illinois

Heat Bath on Steroids

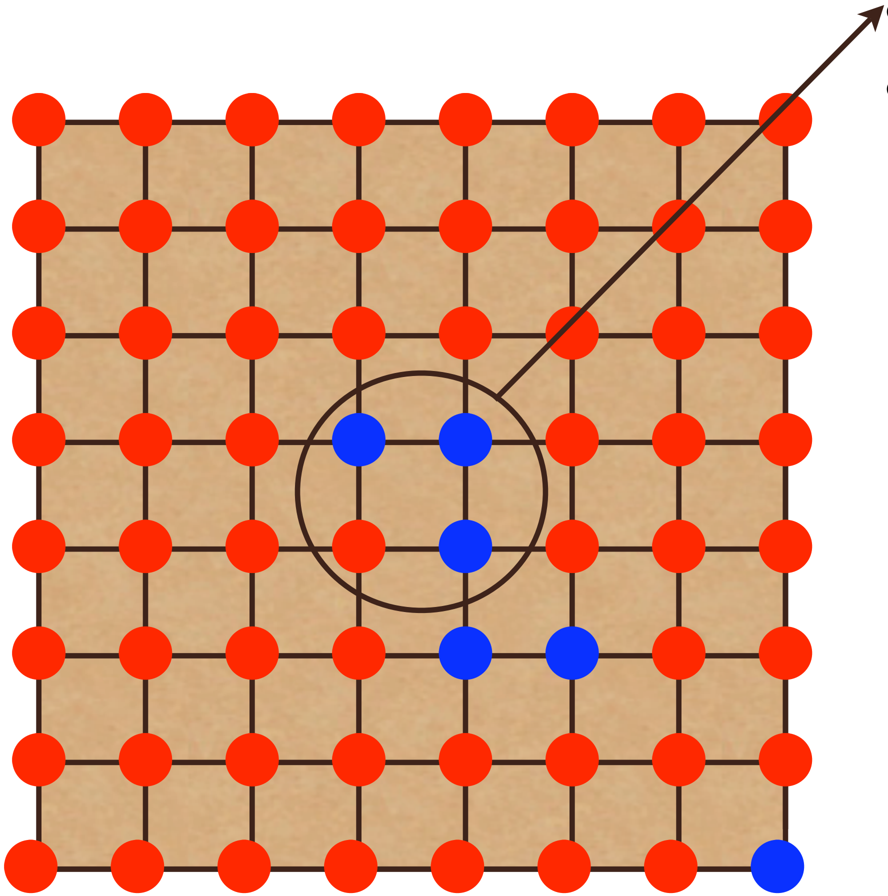


Heat Bath on Steroids



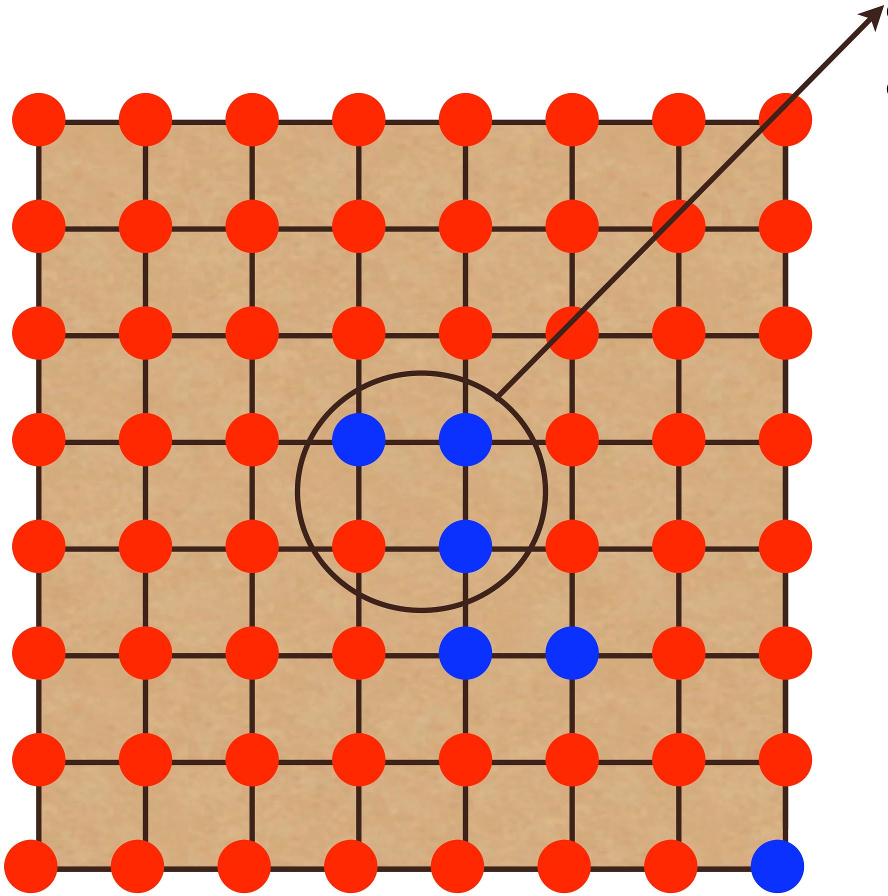
- Choose group of spins at random

Heat Bath on Steroids



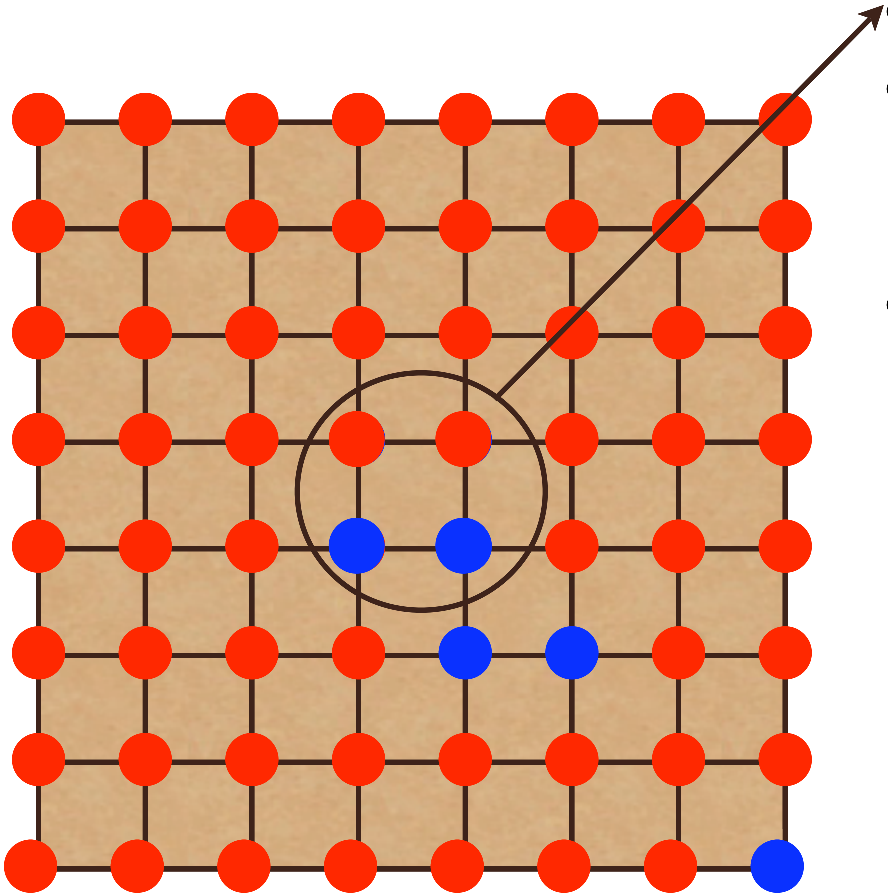
- Choose group of spins at random
- Define p_{rrrr} and Z as per heat bath algorithm

Heat Bath on Steroids



- Choose group of spins at random
- Define p_{rrrr} and Z as per heat bath algorithm

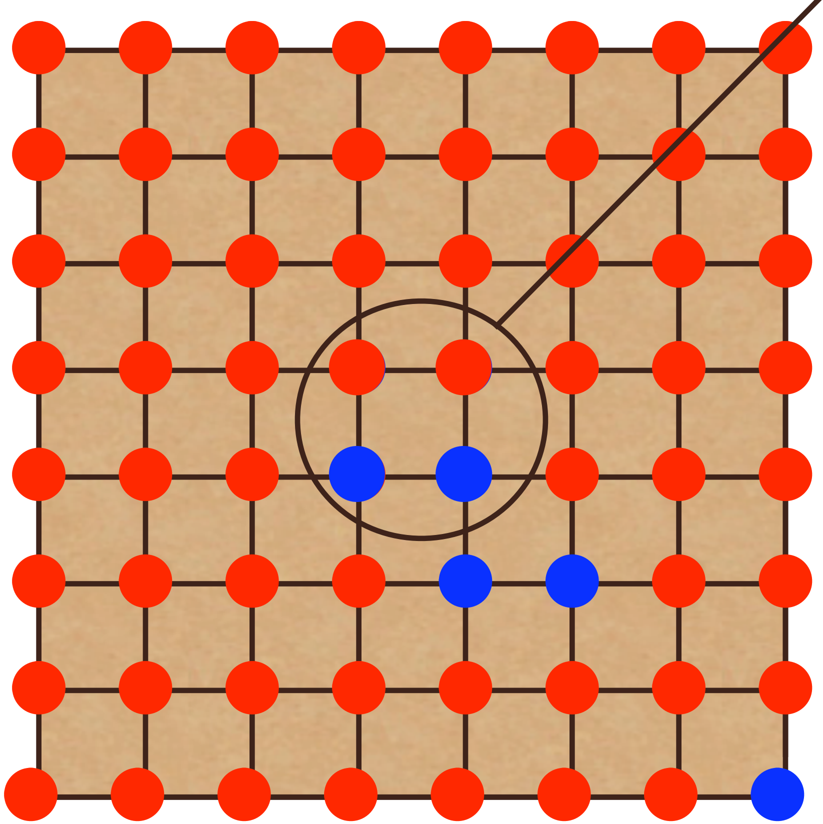
Heat Bath on Steroids



- Choose group of spins at random
- Define p_{rrrr} and Z as per heat bath algorithm

- Make this spin
  - rrrr: p_{rrrr} /Z
  - rrrb: p_{rrrb} /Z
  - rrbb: p_{rrbb} /Z
  - rbrr: p_{rbrr} /Z
  - …

Heat Bath on Steroids



- Choose group of spins at random
- Define p_{rrrr} and Z as per heat bath algorithm

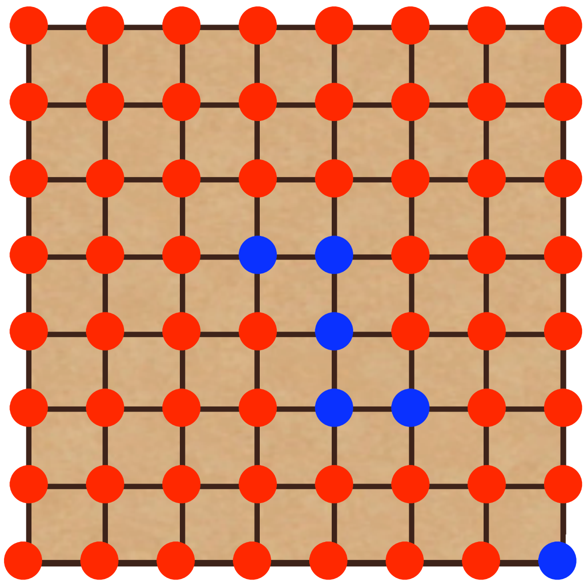
- Make this spin
  - rrrr: p_{rrrr} /Z
  - rrrb: p_{rrrb} /Z
  - rrbb: p_{rrbb} /Z
  - rbrr: p_{rbrr} /Z
  - ...
- Accept with probability

$$P_{\text{accept}} = min\left\{1, \frac{\pi_{\text{new}}}{\pi_{\text{old}}} \frac{T(n \to o)}{T(o \to n)}\right\}$$

MSE at Illinois — Physics Illinois

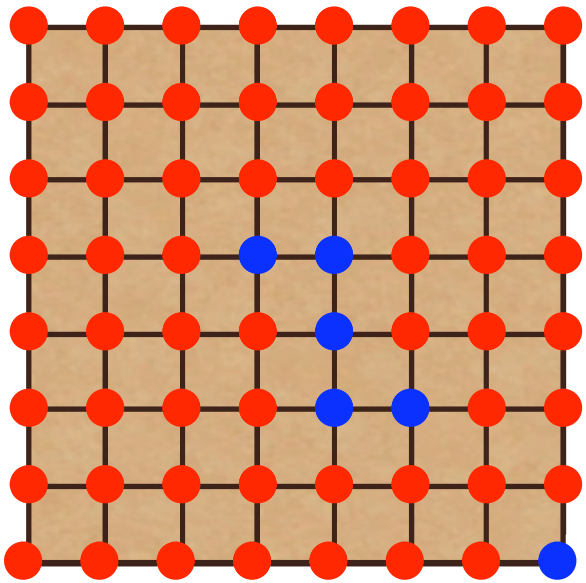
Atomic Scale SIMULATION MSE485/PHY466/CSE485

Kawasaki Dynamics



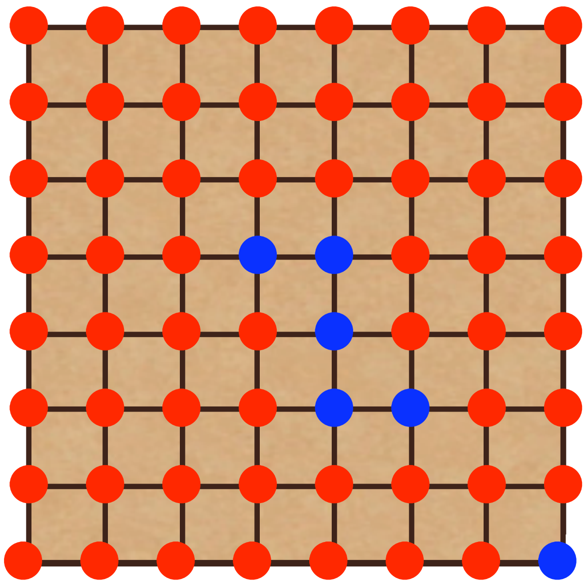
Kawasaki Dynamics

- Simulating a constant spin system



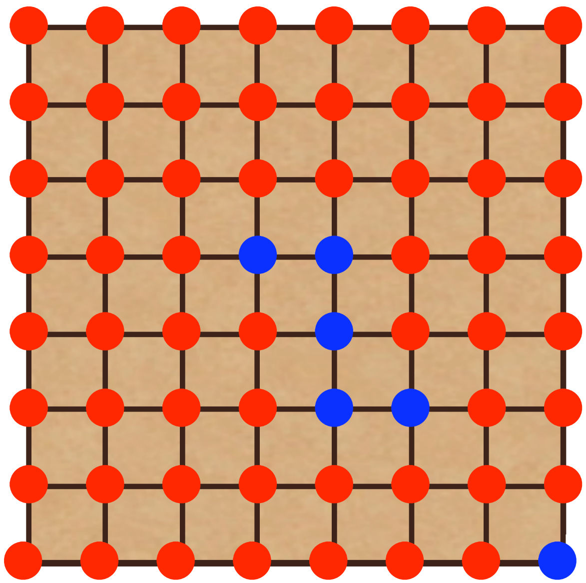
Kawasaki Dynamics

- Simulating a constant spin system
  - spinoidal decomposition



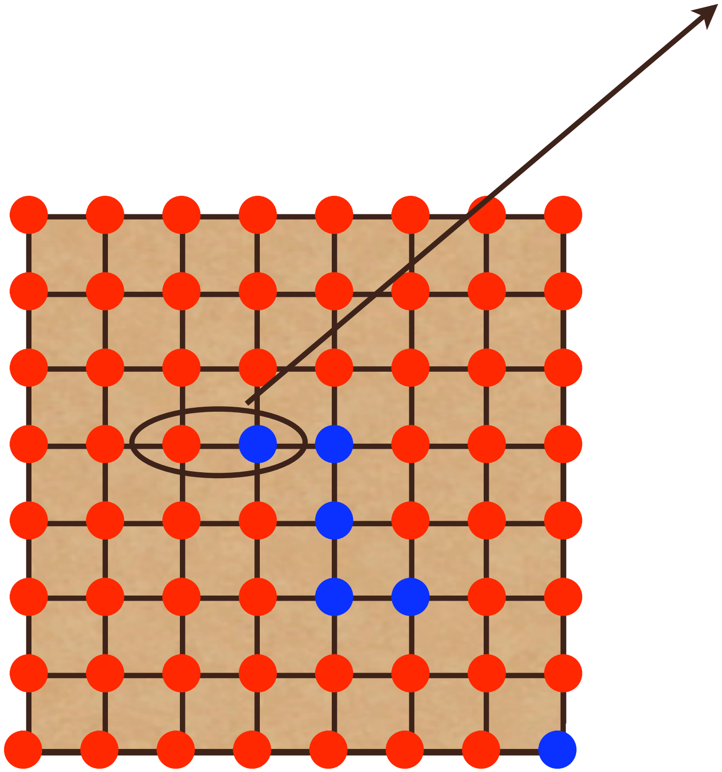
Kawasaki Dynamics

- Simulating a constant spin system
  - spinoidal decomposition
- Previous algorithms don't work!



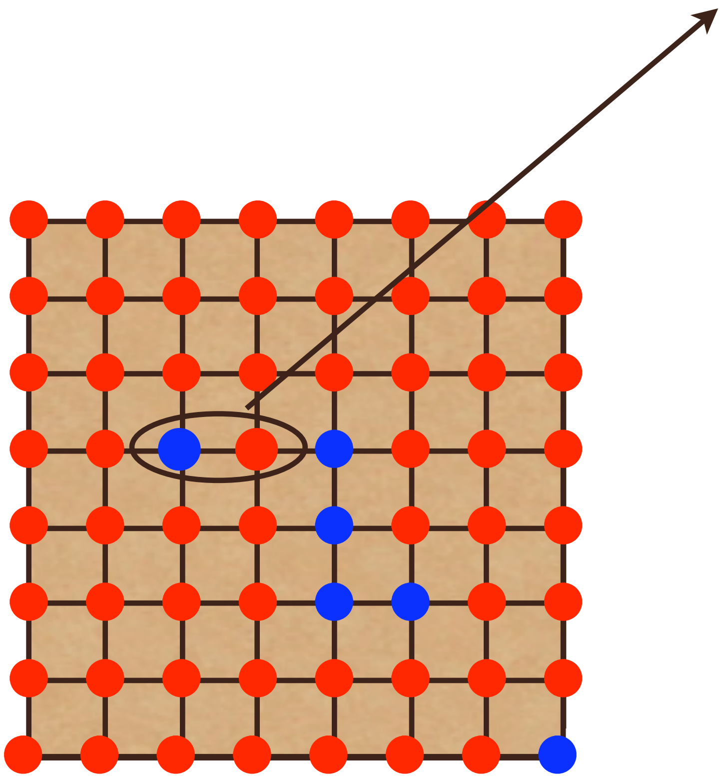
Kawasaki Dynamics

- Simulating a constant spin system
  - spinoidal decomposition
- Previous algorithms don't work!
- Select a random spin and it's neighbor



Kawasaki Dynamics

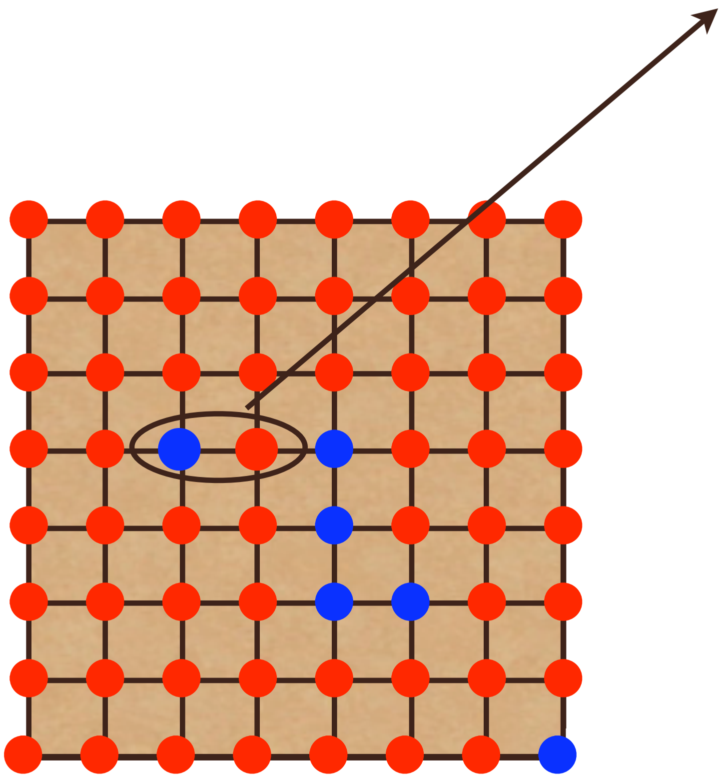
- Simulating a constant spin system
  - spinoidal decomposition
- Previous algorithms don’t work!
- Select a random spin and it’s neighbor
- Exchange spins



Kawasaki Dynamics

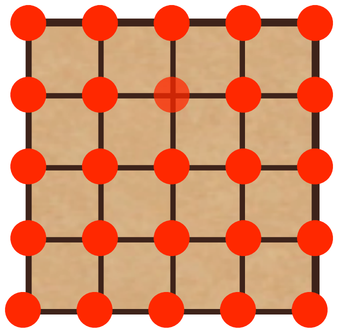
- Simulating a constant spin system
  - spinoidal decomposition
- Previous algorithms don't work!
- Select a random spin and it's neighbor
- Exchange spins
- Accept or reject

$$P_{\rm accept} = min\left\{1, \frac{\pi_{\rm new}}{\pi_{\rm old}} \frac{T(n \to o)}{T(o \to n)}\right\}$$



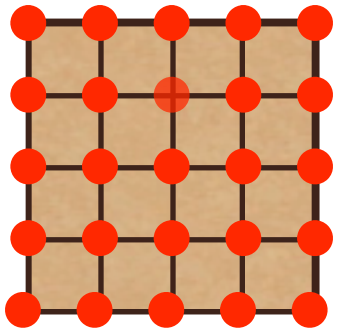
20

Kinetic Monte Carlo



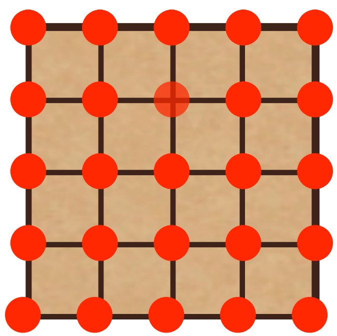
Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods



Kinetic Monte Carlo

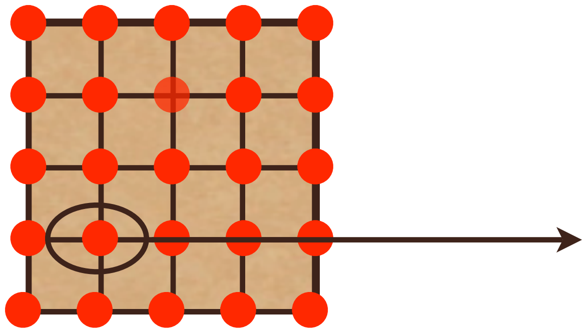
- Not as “rigorous” a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.



MSE485/PHY466/CSE485
Atomic Scale SIMULATION

Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.
- In metropolis, each site would flip with $p_i = \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right)$



Flip w/ $P_b = e^{-10}$

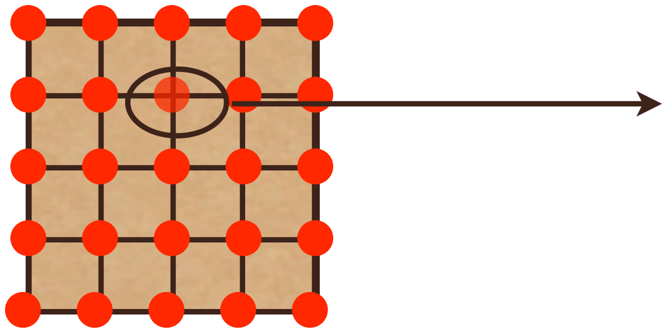
Reject

Time passed:

1

Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.
- In metropolis, each site would flip with $p_i = \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right)$



Flip w/ $P_b = e^{-10}$

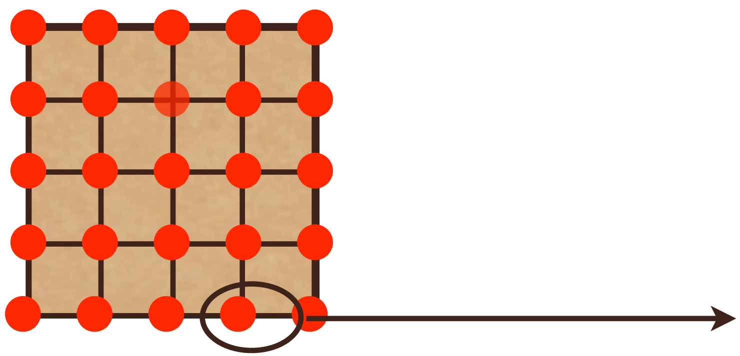
Reject

Time passed:

2

Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.
- In metropolis, each site would flip with $p_i = \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right)$



Time passed:

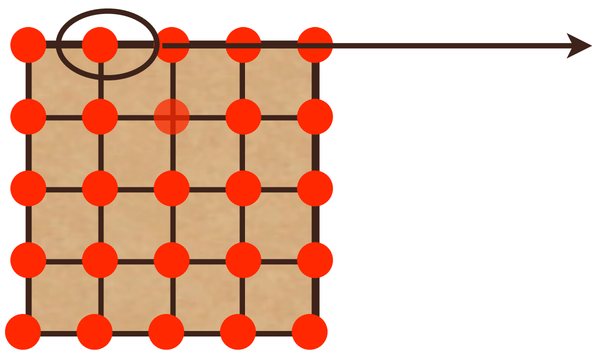
...

Flip w/ $P_b = e^{-10}$

Reject

Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.
- In metropolis, each site would flip with $p_i = \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right)$



Flip w/ $P_b = e^{-10}$

Reject

Time passed:

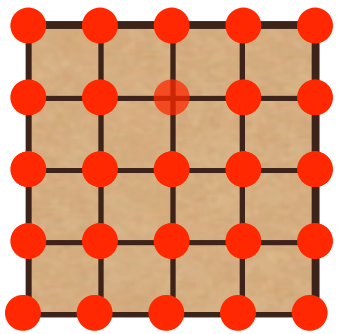
22026

Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.
- In metropolis, each site would flip with $p_i = \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right)$
- In KMC, choose a site i to flip with probability $\frac{p_i}{\sum_i p_i}$

  and assume that $-\frac{\ln U}{\sum_i p_i}$ time has passed.

$$P_1 = \frac{e^{-10}}{25e^{-10}} = \frac{1}{25}$$



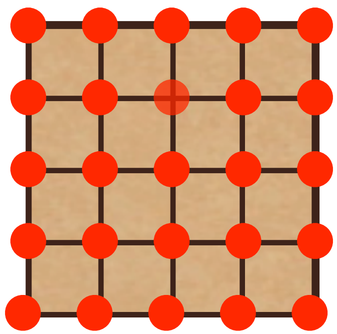
Time passed:

344

Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.
- In metropolis, each site would flip with $p_i = \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right)$
- In KMC, choose a site i to flip with probability $\frac{p_i}{\sum_i p_i}$ and assume that $-\frac{\ln U}{\sum_i p_i}$ time has passed.
- Trick is doing this fast (N-Fold way, etc)

$$P_1 = \frac{e^{-10}}{25e^{-10}} = \frac{1}{25}$$



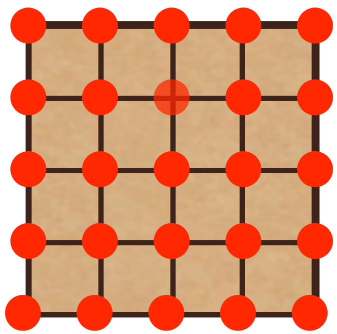
Time passed:

344

Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.
- In metropolis, each site would flip with $p_i = \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right)$
- In KMC, choose a site i to flip with probability $\frac{p_i}{\sum_i p_i}$ and assume that $-\frac{\ln U}{\sum_i p_i}$ time has passed.
- Trick is doing this fast (N-Fold way, etc)
- Useful for gas adsorbtion, etc.

$$P_1 = \frac{e^{-10}}{25e^{-10}} = \frac{1}{25}$$



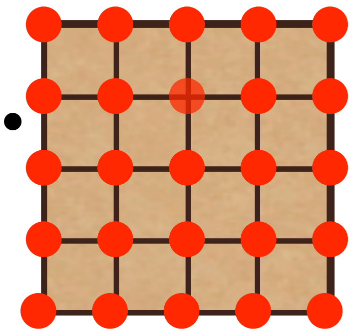
Time passed:

344

Kinetic Monte Carlo

- Not as "rigorous" a footing as other methods
- Really low temperature, things evolve slowly since no one wants to flip.
- In metropolis, each site would flip with $p_i = \exp\left(-\beta \sum_{\langle i,j \rangle} s_i s_j\right)$
- In KMC, choose a site i to flip with probability $\frac{p_i}{\sum_i p_i}$
  and assume that $-\frac{\ln U}{\sum_i p_i}$ time has passed.
- Trick is doing this fast (N-Fold way, etc)
- Useful for gas adsorbtion, etc.

$$P_1 = \frac{e^{-10}}{25e^{-10}} = \frac{1}{25}$$

- 

Time passed:

344

Speeding things up: Constants

Lots of tricks to make it run faster.

Metropolis importance sampling Monte Carlo scheme

(1) Choose an initial state
(2) Choose a site i
(3) Calculate the energy change ΔE which results if the spin at site i is overturned
(4) Generate a random number r such that $0 < r < 1$
(5) If $r < \exp(-\Delta E / k_B T)$, flip the spin
(6) Go the next site and go to (3)

Calculating the magnetization on the fly!

Playing games with bit operations

Speeding things up: Constants

Lots of tricks to make it run faster.

Metropolis importance sampling Monte Carlo scheme

(1) Choose an initial state
(2) Choose a site i
(3) Calculate the energy change ΔE which results if the spin at site i is overturned
(4) Generate a random number r such that $0 < r < 1$
(5) If $r < \exp(-\Delta E/k_B T)$, flip the spin
(6) Go the next site and go to (3)

Calculating the magnetization on the fly!

Playing games with bit operations

Speeding things up: Constants

Lots of tricks to make it run faster.

Metropolis importance sampling Monte Carlo scheme

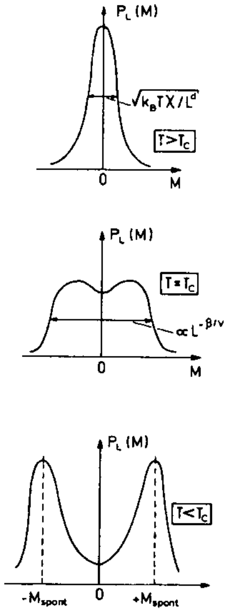
(1) Choose an initial state
(2) Choose a site i
(3) Calculate the energy change ΔE which results if the spin at site i is overturned
(4) Generate a random number r such that $0 < r < 1$
(5) If $r < \exp(-\Delta E/k_B T)$, flip the spin
(6) Go the next site and go to (3)

Calculating the magnetization on the fly!

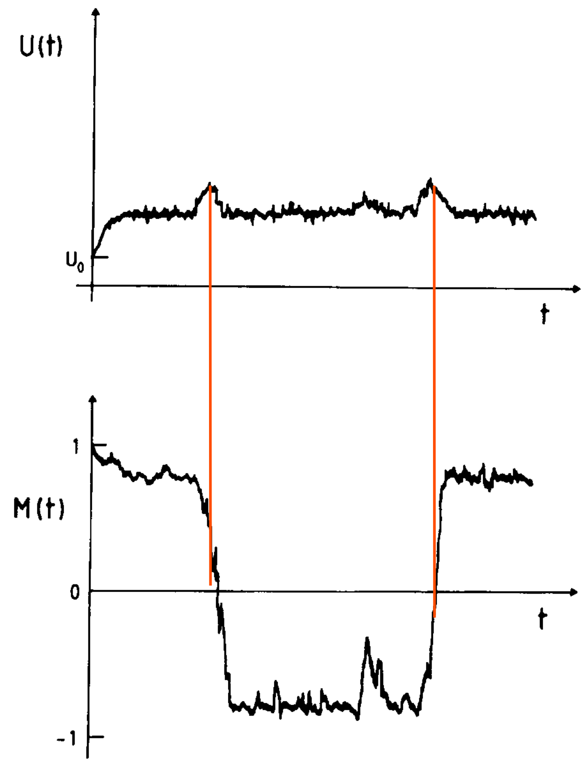
Playing games with bit operations

Critical slowing down

- Near the transition dynamics gets very slow if you use any local update method.
- The larger the system the less likely it is that the system can flip over.



Monte Carlo of a zero-field Ising Lattice
U vs. time and M vs. time.



Metropolis importance sampling Monte Carlo scheme

(1) Choose an initial state

(2) Choose a site i

(3) Calculate the energy change ΔE which results if the spin at site i is overturned

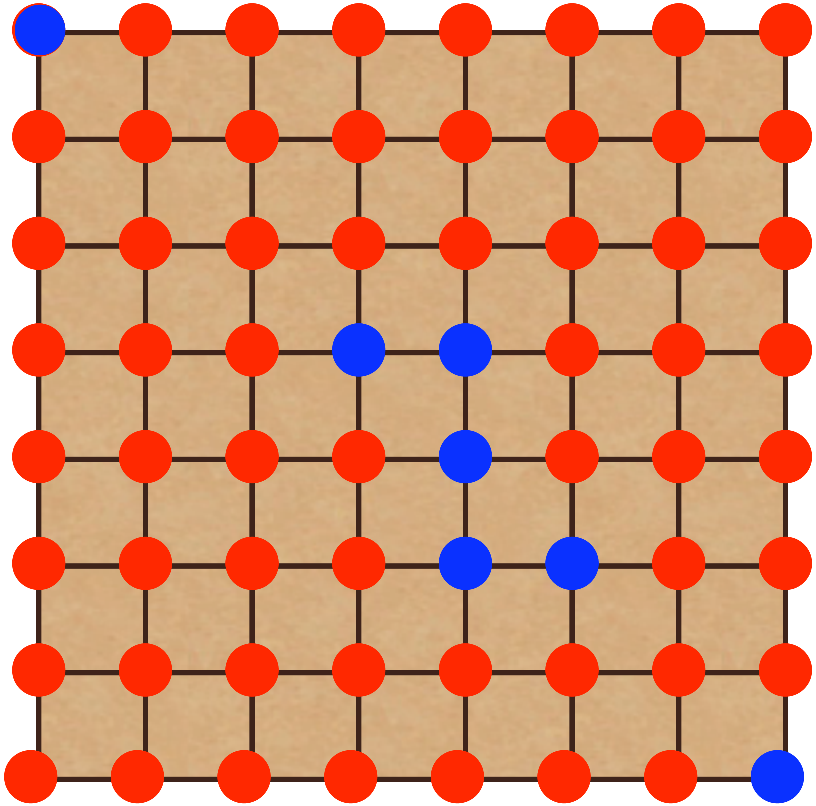
(4) Generate a random number r such that $0 < r < 1$

(5) If $r < \exp(-\Delta E/k_B T)$, flip the spin

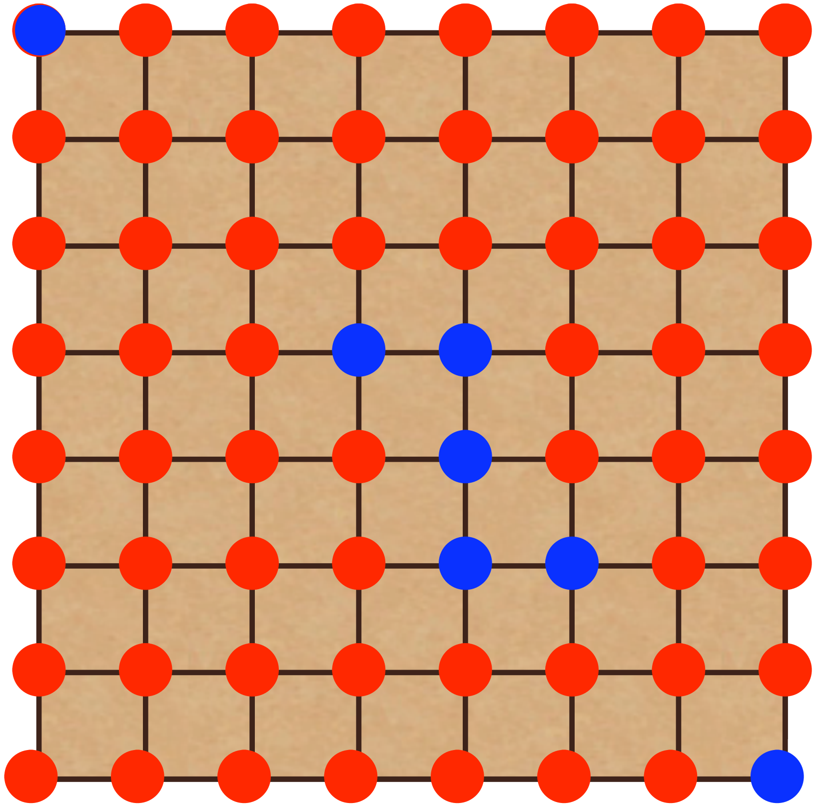
(6) Go the next site and go to (3)

Cluster Algorithms

Swendsen Algorithm



Cluster Algorithms

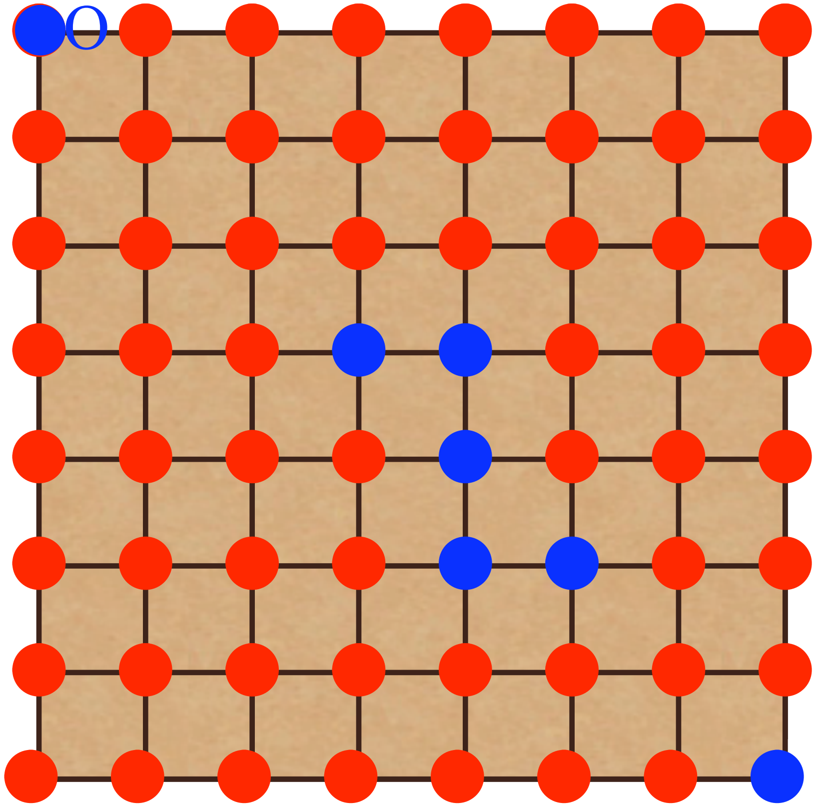


Swendsen Algorithm

- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms



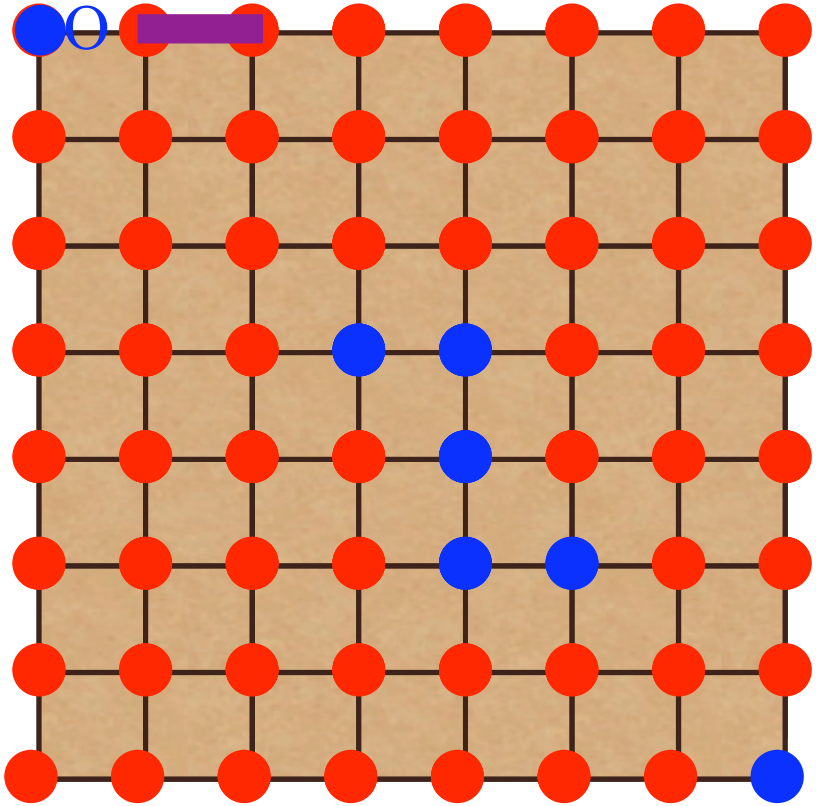
Swendsen Algorithm

- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

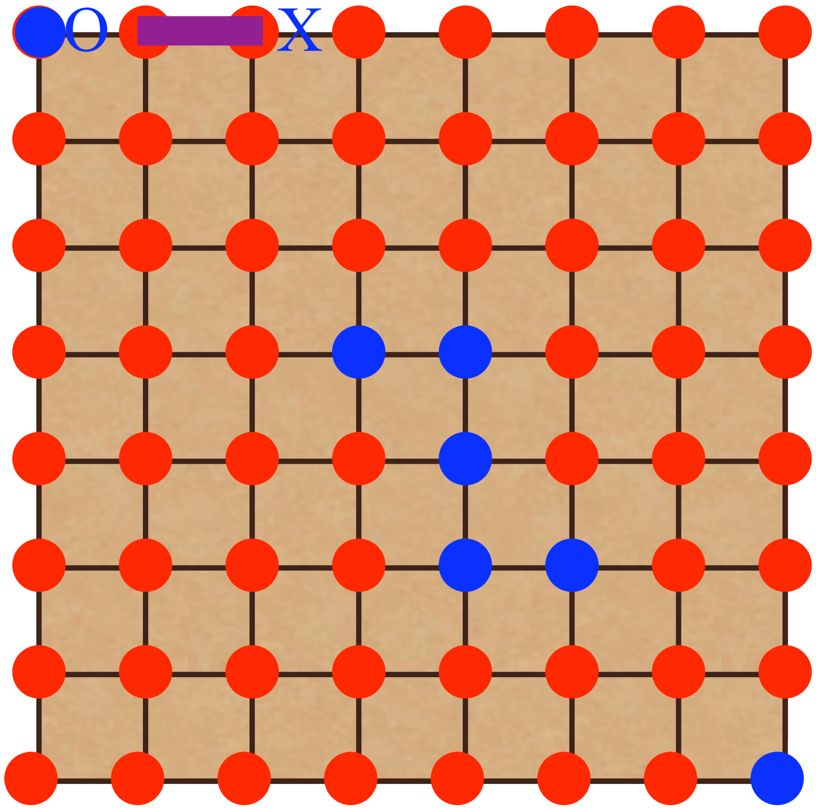
Swendsen Algorithm



- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

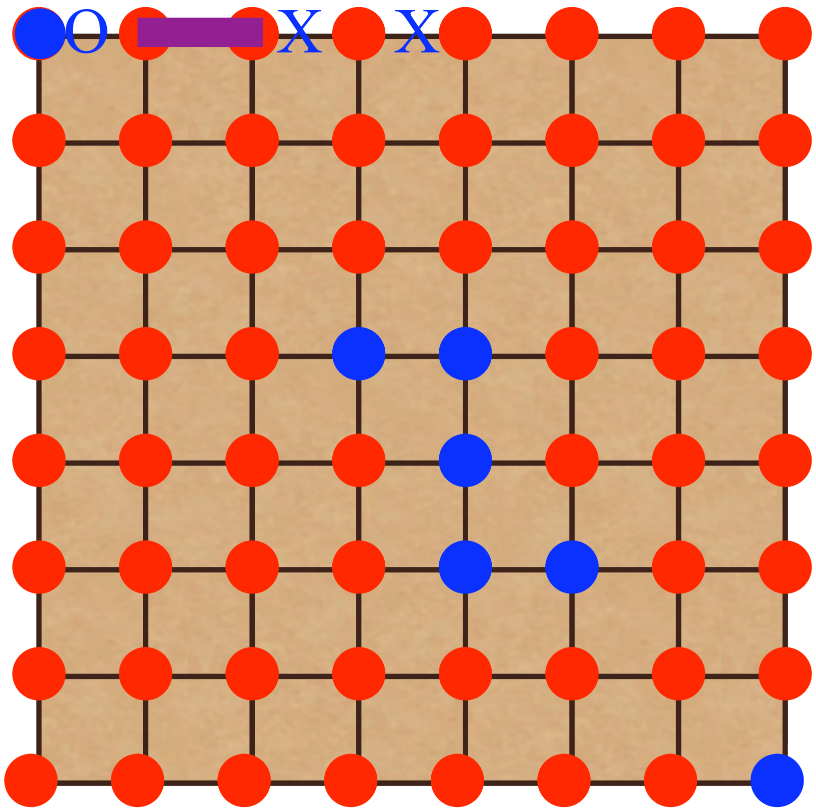


Swendsen Algorithm

- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

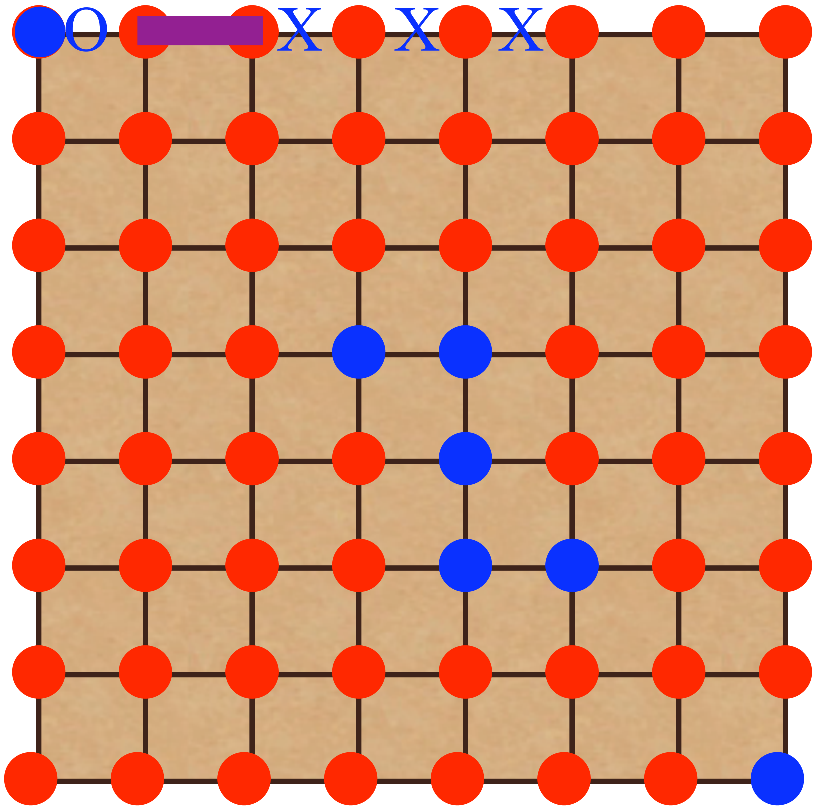


Swendsen Algorithm

- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms



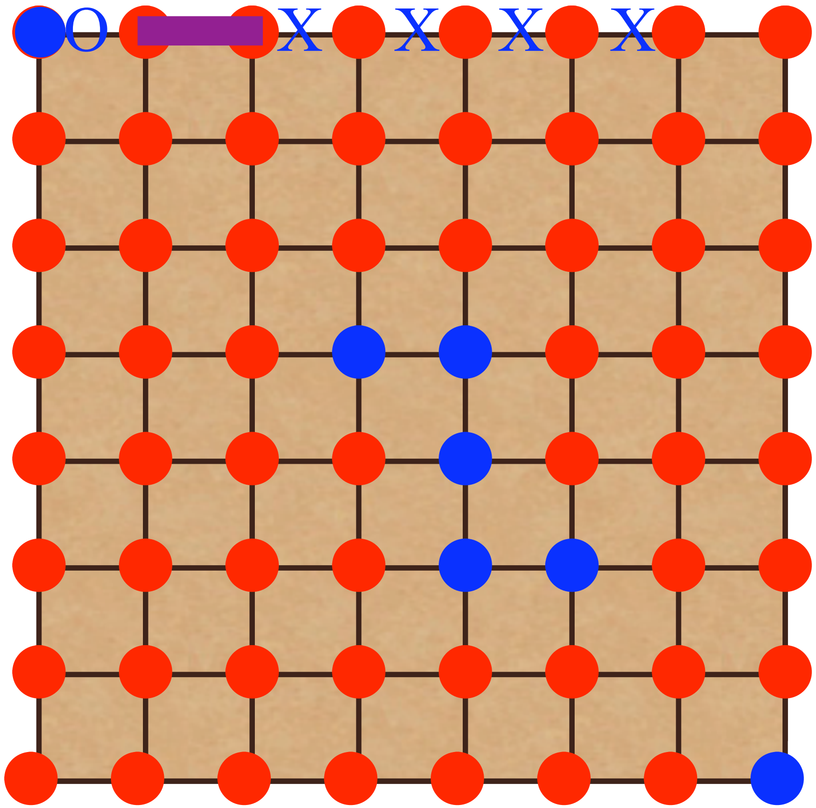
Swendsen Algorithm

- For each same-spin bond:
"Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

Swendsen Algorithm

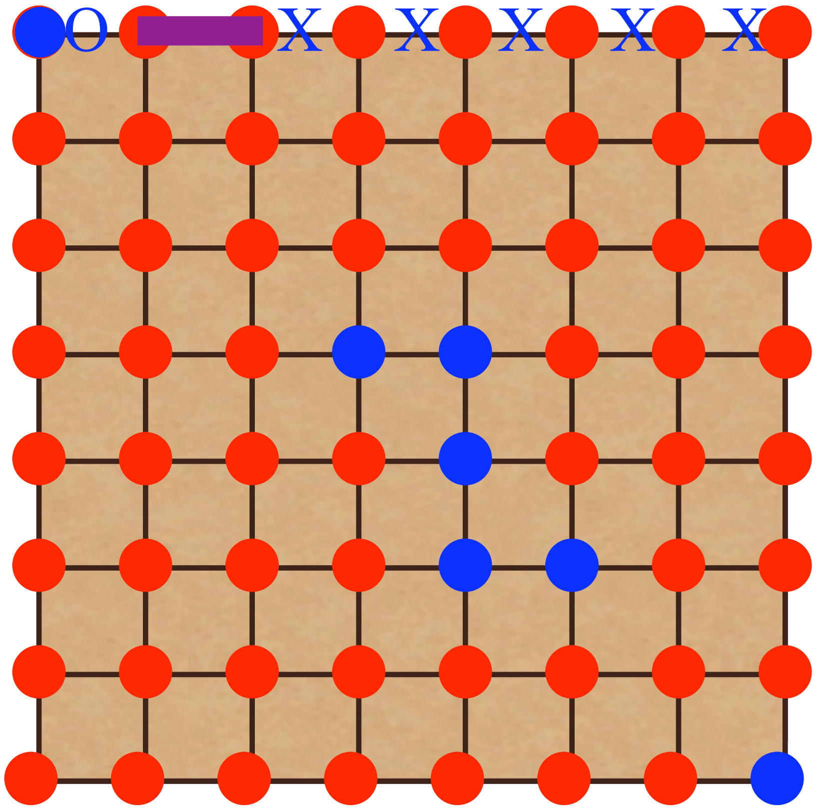


- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

Swendsen Algorithm

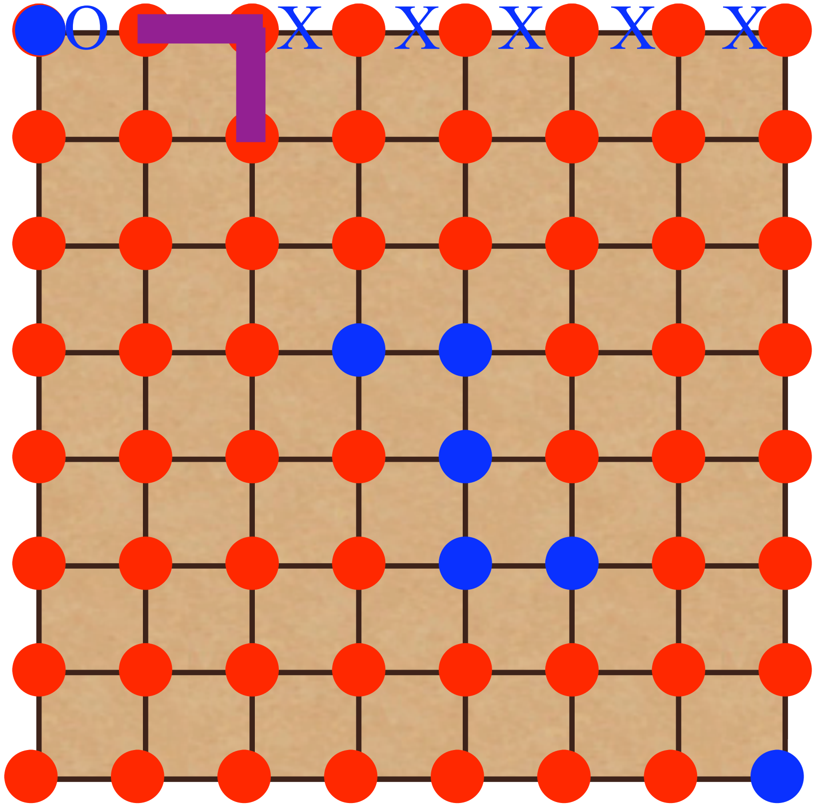


- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

Swendsen Algorithm

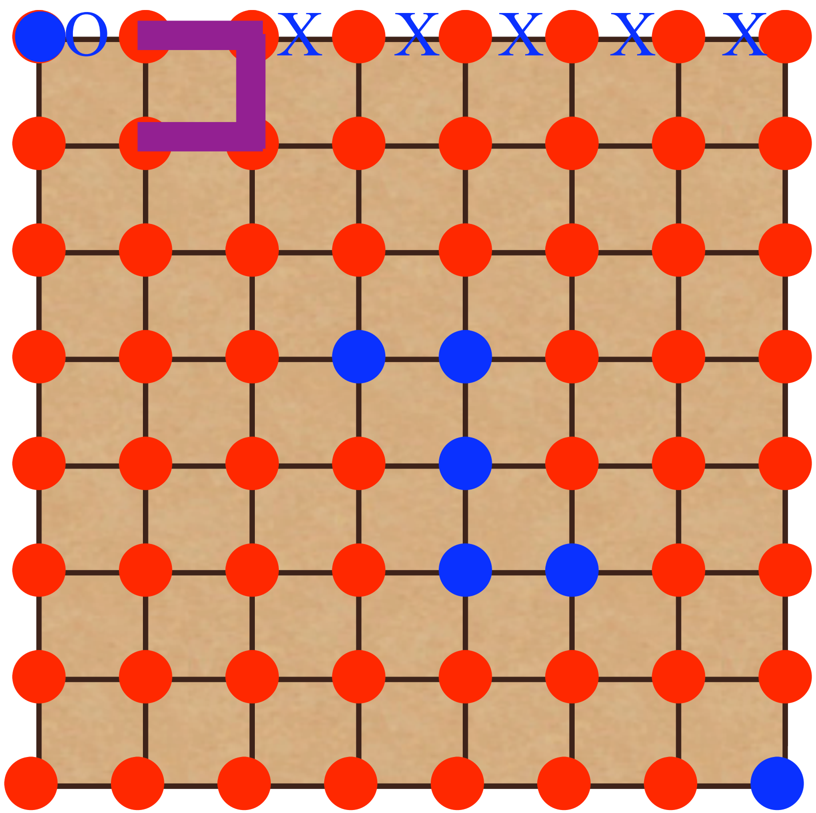


- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

Swendsen Algorithm

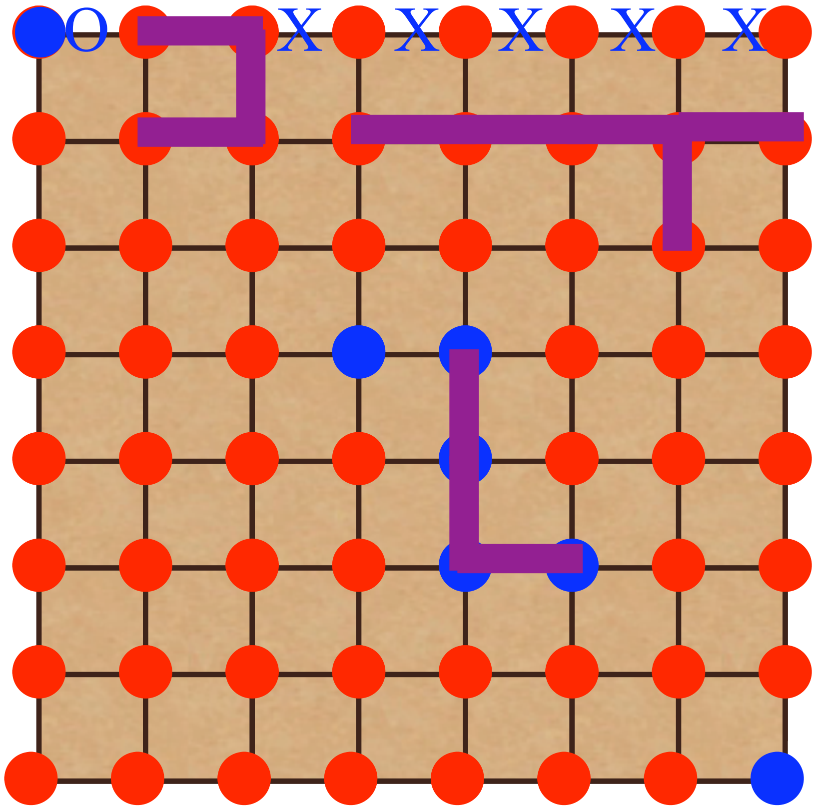


- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

Swendsen Algorithm

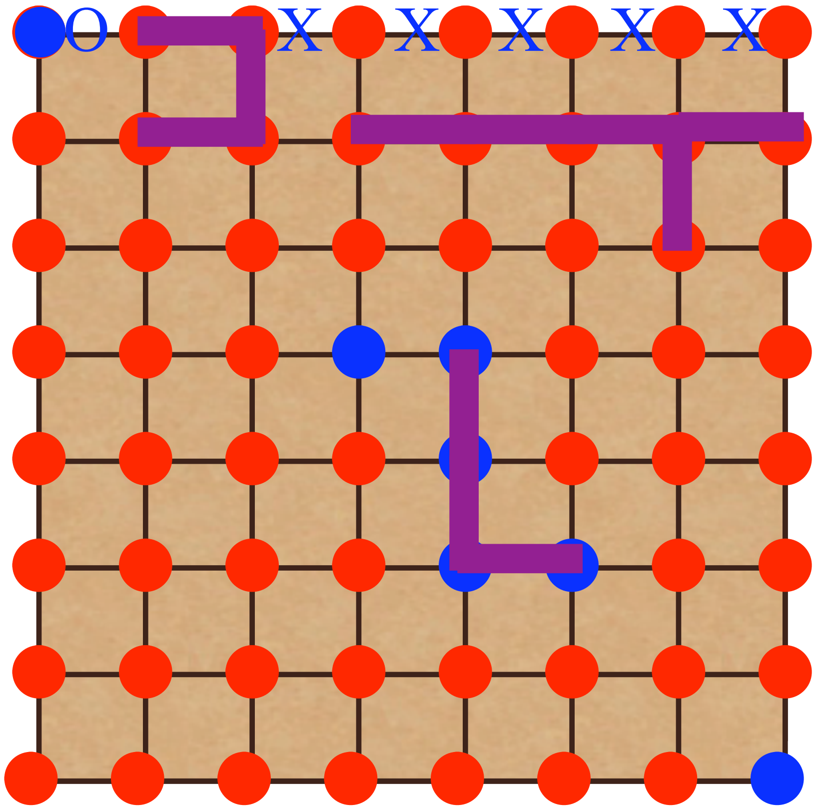


- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms

Swendsen Algorithm

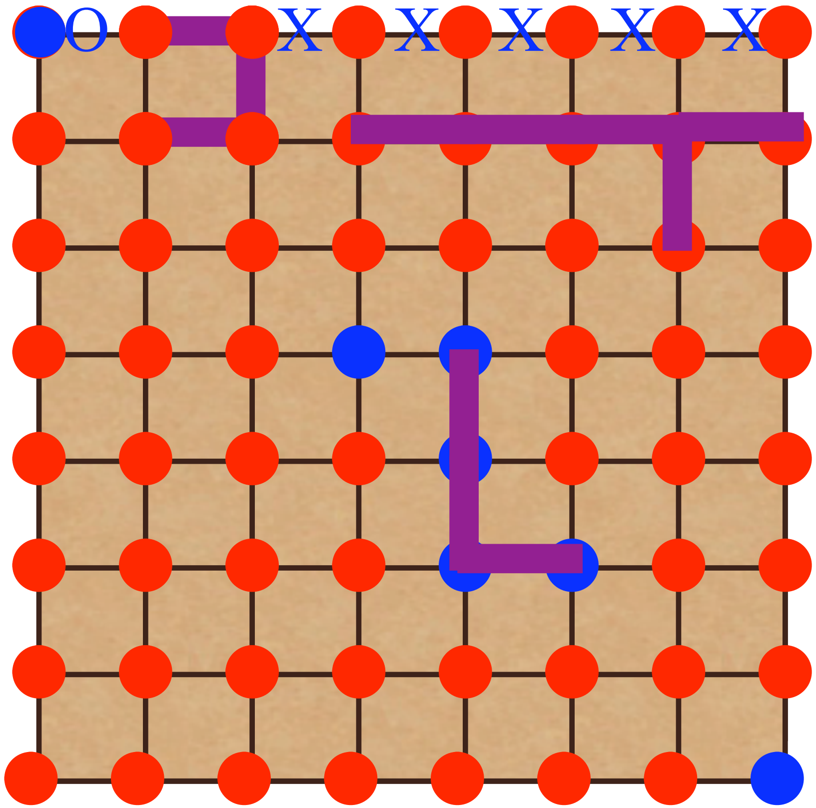


- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

- For each cluster, assign it a random color.

Cluster Algorithms



Swendsen Algorithm

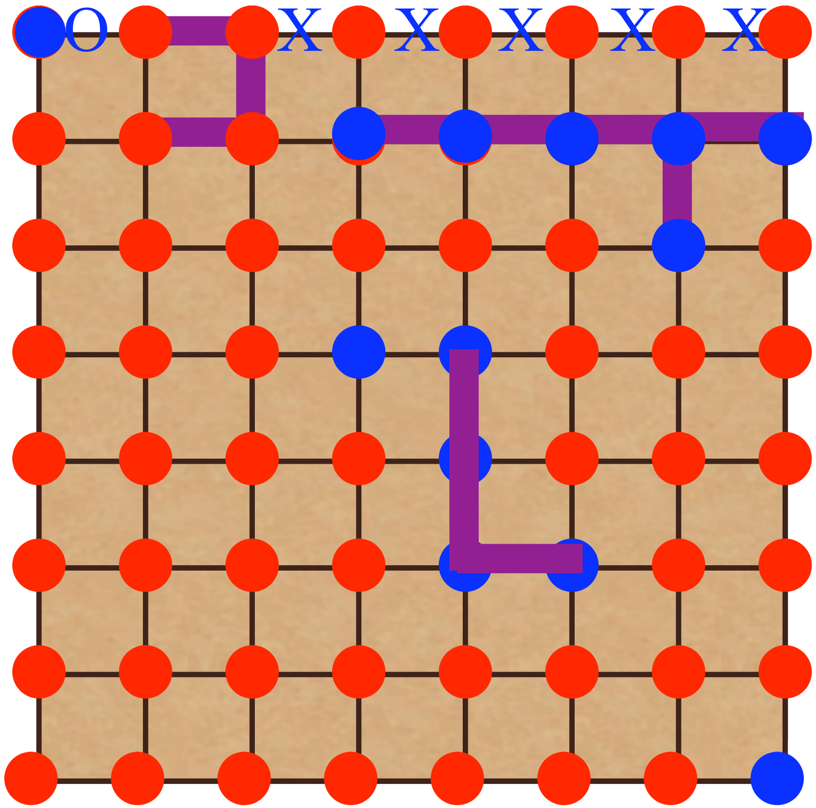
- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

- For each cluster, assign it a random color.

Cluster Algorithms

Swendsen Algorithm

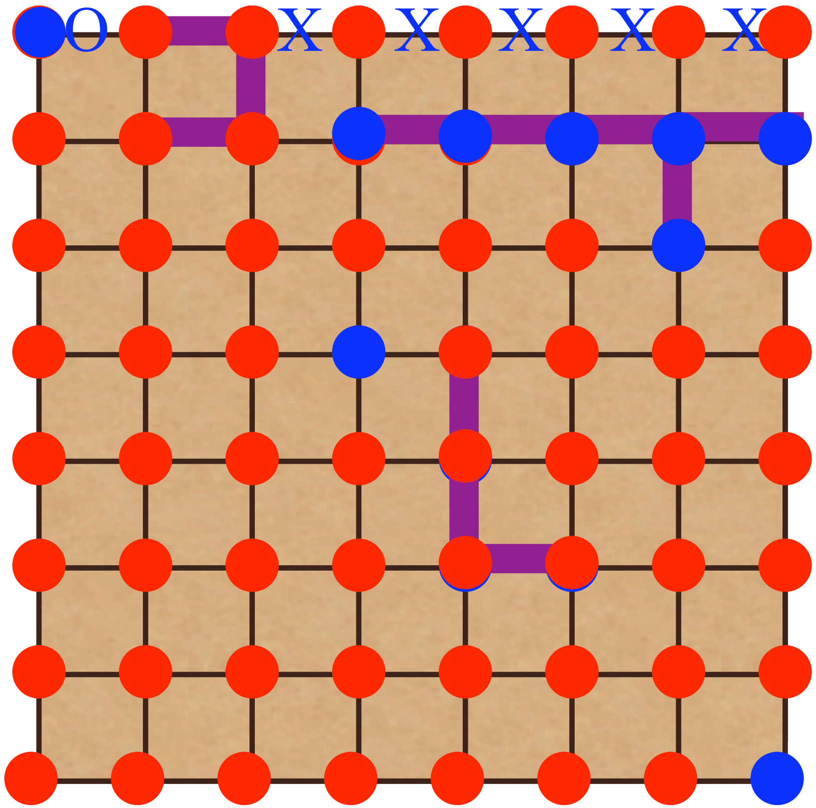


- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

- For each cluster, assign it a random color.

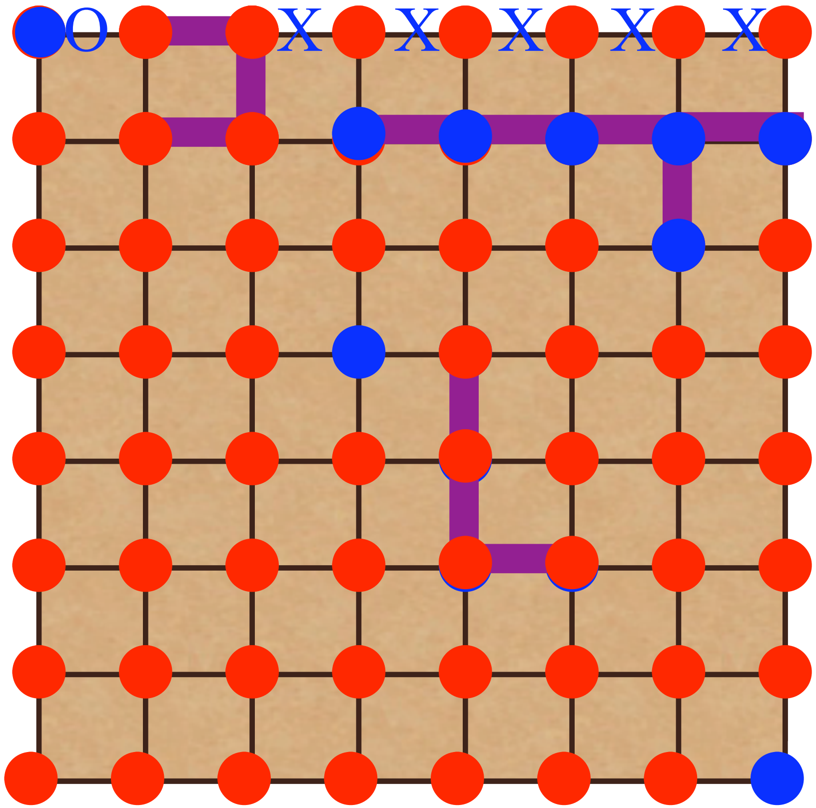
Cluster Algorithms



Swendsen Algorithm

- For each same-spin bond: "Turn the bond purple" with
$$p_i = 1 - \exp(-2\beta J)$$
- For each cluster, assign it a random color.

Cluster Algorithms



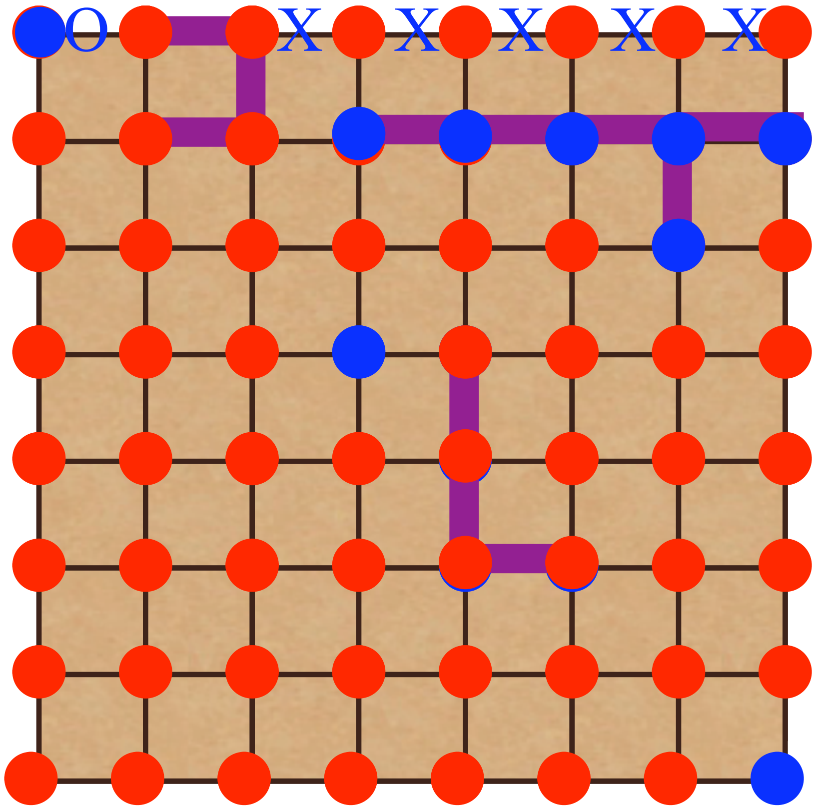
Swendsen Algorithm

- For each same-spin bond: "Turn the bond purple" with

$$p_i = 1 - \exp(-2\beta J)$$

- For each cluster, assign it a random color.
- Accept with probability

Cluster Algorithms



Swendsen Algorithm

- For each same-spin bond: "Turn the bond purple" with

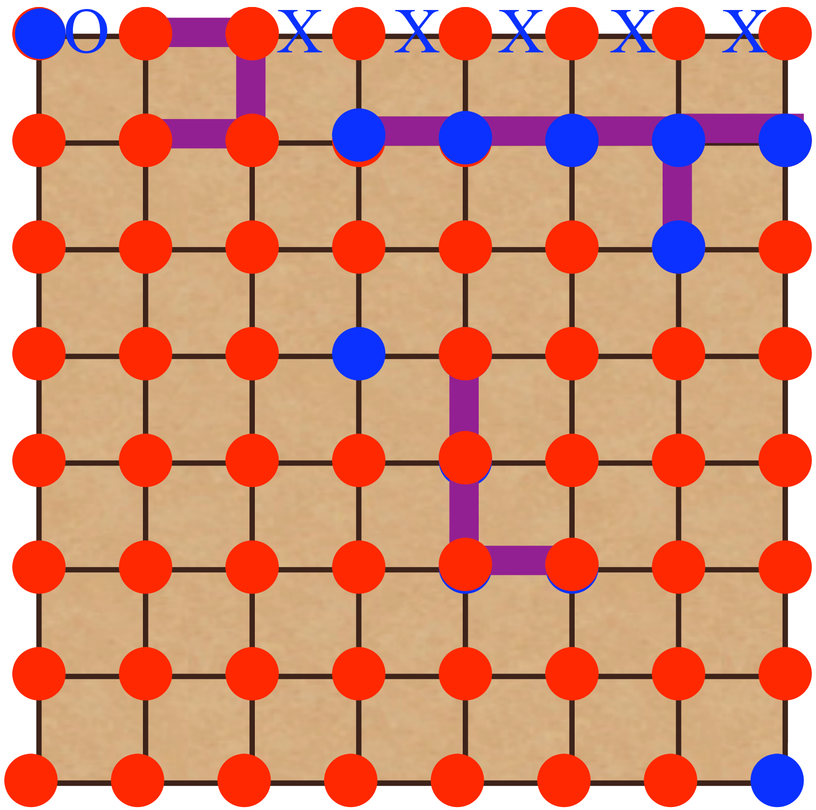
$$p_i = 1 - \exp(-2\beta J)$$

- For each cluster, assign it a random color.

- Accept with probability

$$P_{\mathrm{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

Cluster Algorithms



Swendsen Algorithm

- For each same-spin bond: "Turn the bond purple" with

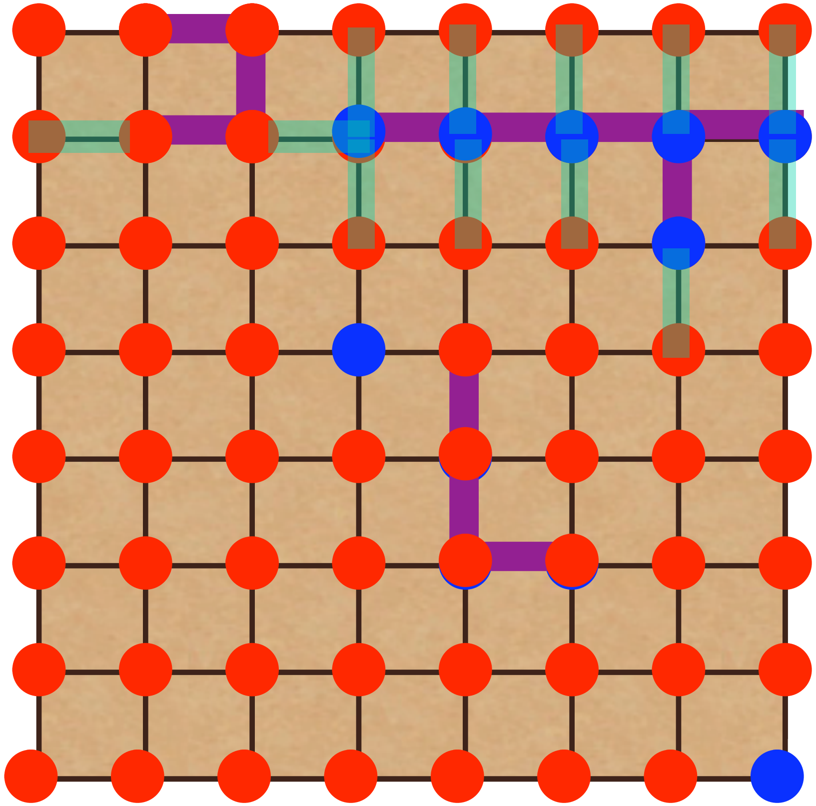
$$p_i = 1 - \exp(-2\beta J)$$

- For each cluster, assign it a random color.

- Accept with probability

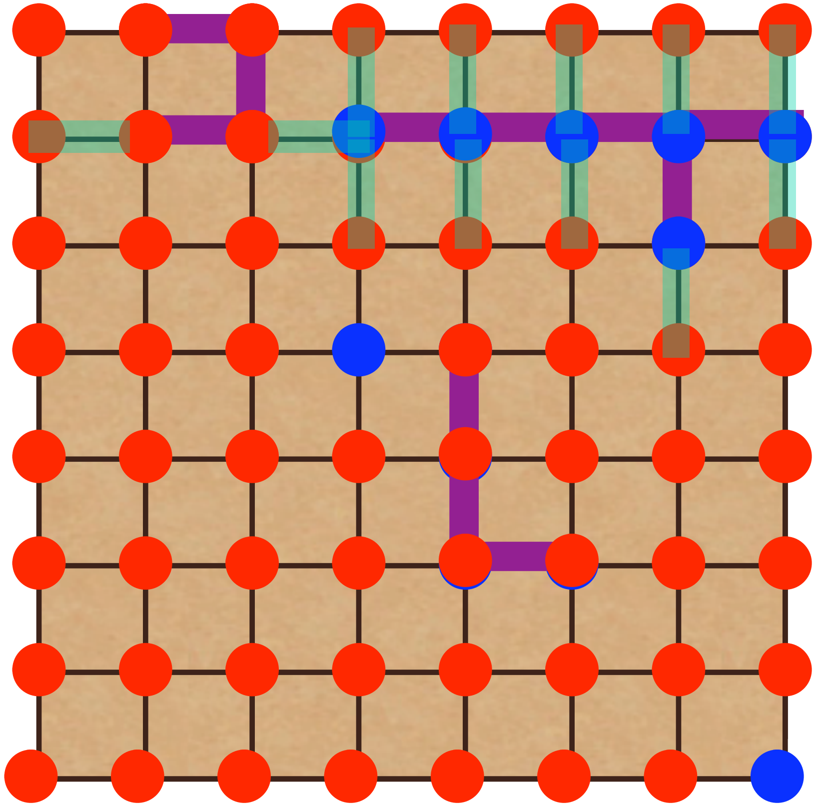
$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

- Can this work if h>0?



$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

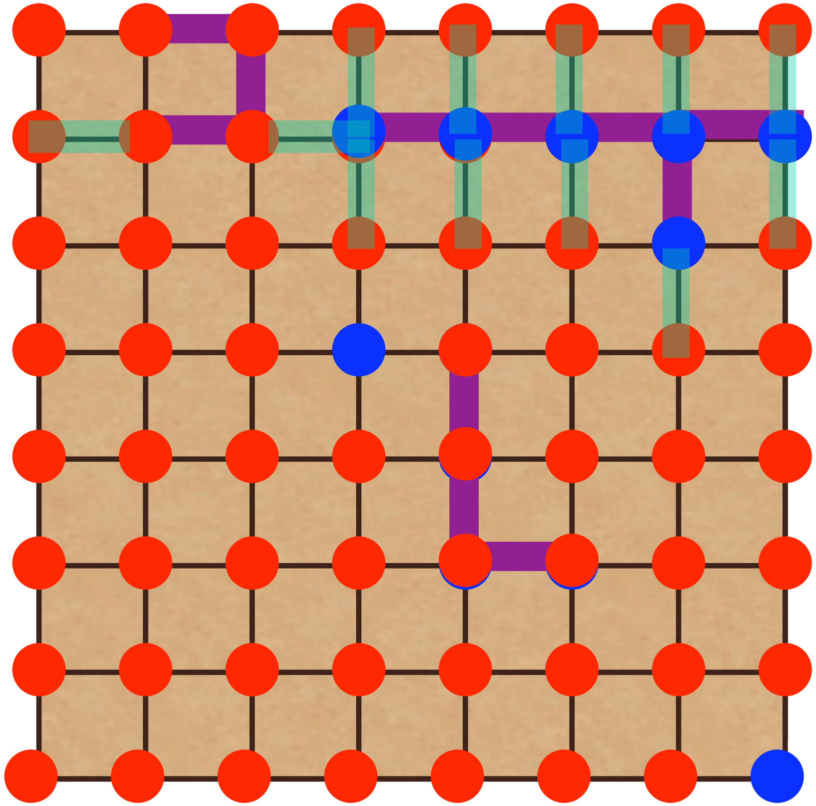
$$p_i = 1 - \exp(-2\beta J)$$



4 cases:

$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

$$p_i = 1 - \exp(-2\beta J)$$

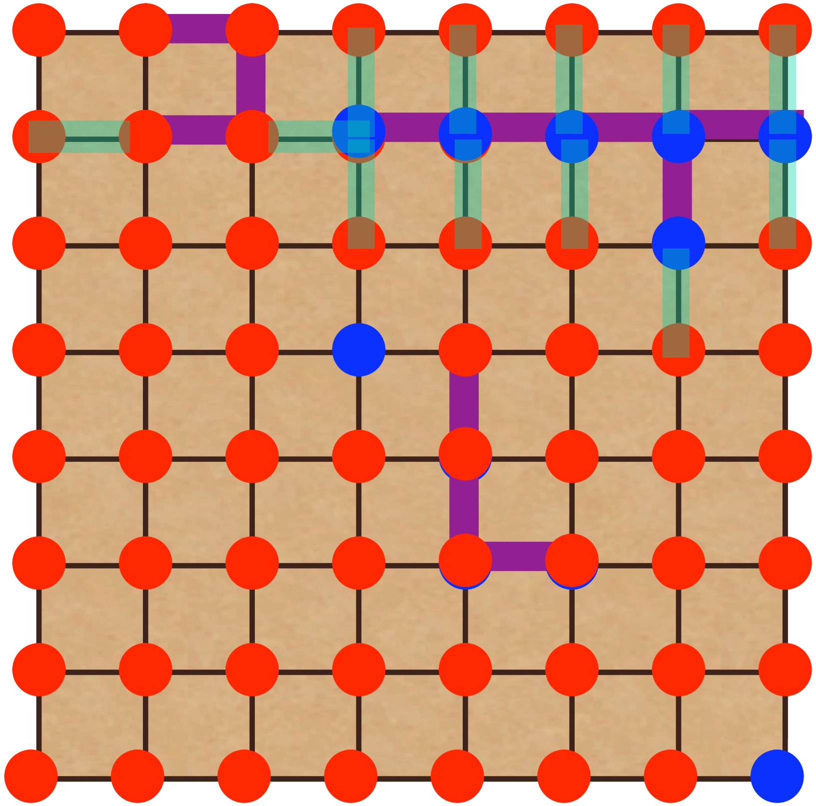


4 cases:

1. We colored the bond purple!

$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

$$p_i = 1 - \exp(-2\beta J)$$



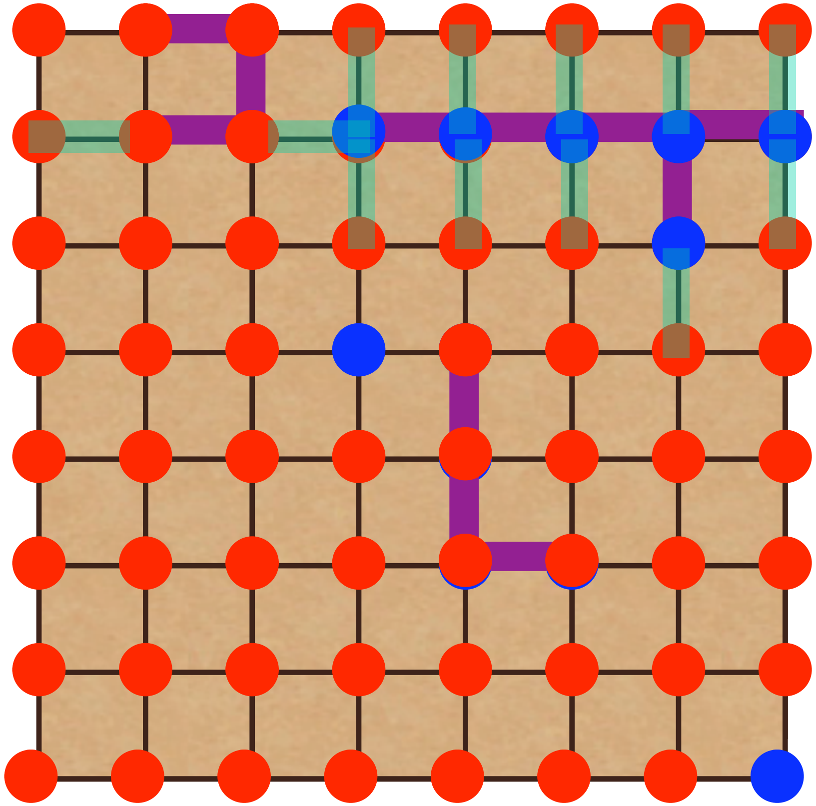
4 cases:

1. We colored the bond purple!

2. Bond starts and ends both the same (or different)

$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n\to o)}{T(o\to n)}\right\}$$

$$p_i = 1 - \exp(-2\beta J)$$



4 cases:

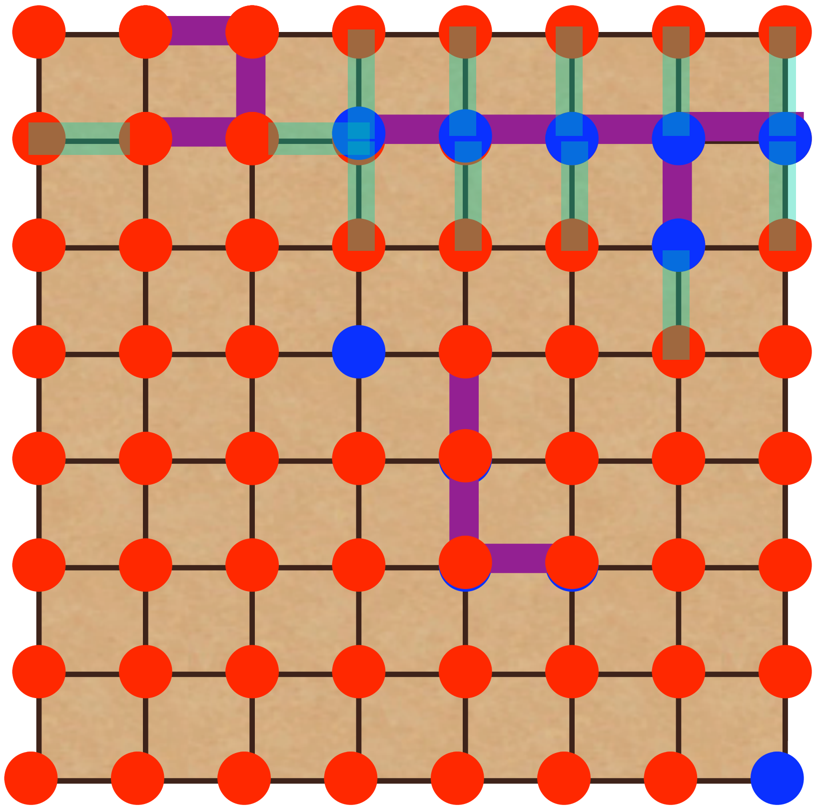
1. We colored the bond purple!

2. Bond starts and ends both the same (or different)

3. old: diff color, new: same color

$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

$$p_i = 1 - \exp(-2\beta J)$$



4 cases:

1. We colored the bond purple!

2. Bond starts and ends both the same (or different)

3. old: diff color, new: same color

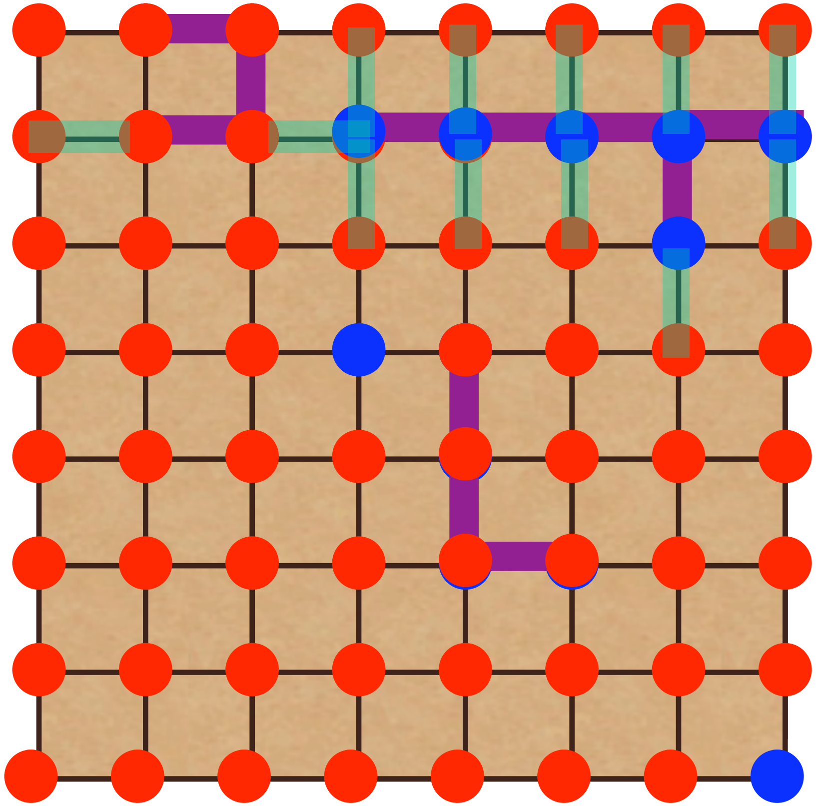
$$T(n \to o) = \exp(-2\beta J)$$

$$T(o \to n) = 1$$

$$\exp(-\Delta E) = \exp(2\beta J)$$

$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

$$p_i = 1 - \exp(-2\beta J)$$



4 cases:

1. We colored the bond purple!

2. Bond starts and ends both the same (or different)

3. old: diff color, new: same color

$$T(n \rightarrow o) = \exp(-2\beta J)$$

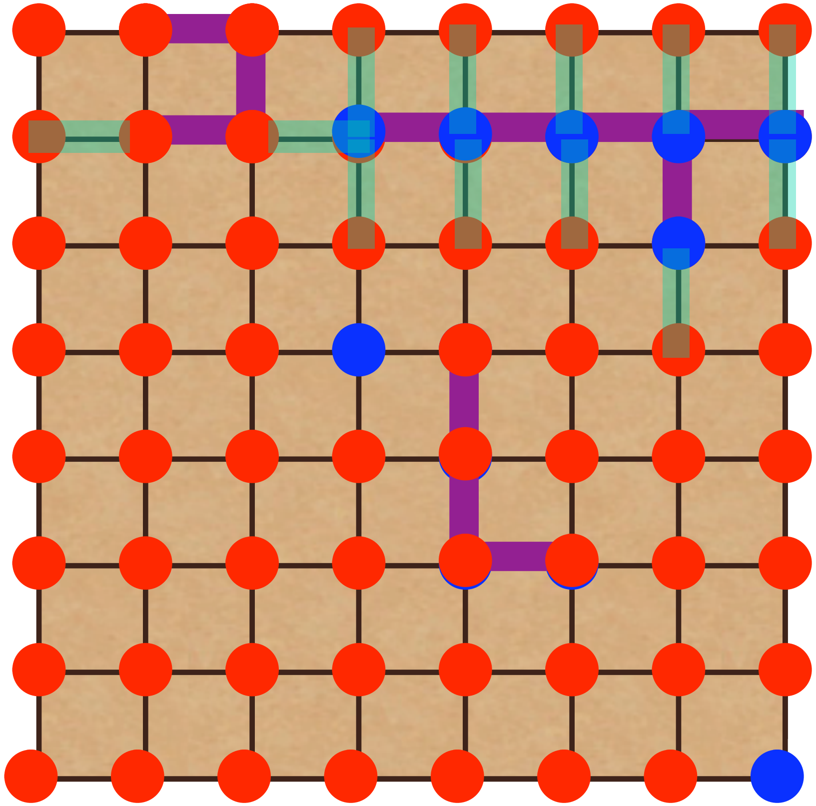
$$T(o \rightarrow n) = 1$$

$$\exp(-\Delta E) = \exp(2\beta J)$$

4. new: diff color, old: same color

$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \rightarrow o)}{T(o \rightarrow n)}\right\}$$

$$p_i = 1 - \exp(-2\beta J)$$



4 cases:

1. We colored the bond purple!

2. Bond starts and ends both the same (or different)

3. old: diff color, new: same color

$$T(n \to o) = \exp(-2\beta J)$$

$$T(o \to n) = 1$$

$$\exp(-\Delta E) = \exp(2\beta J)$$

$$P_{\text{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

$$p_i = 1 - \exp(-2\beta J)$$

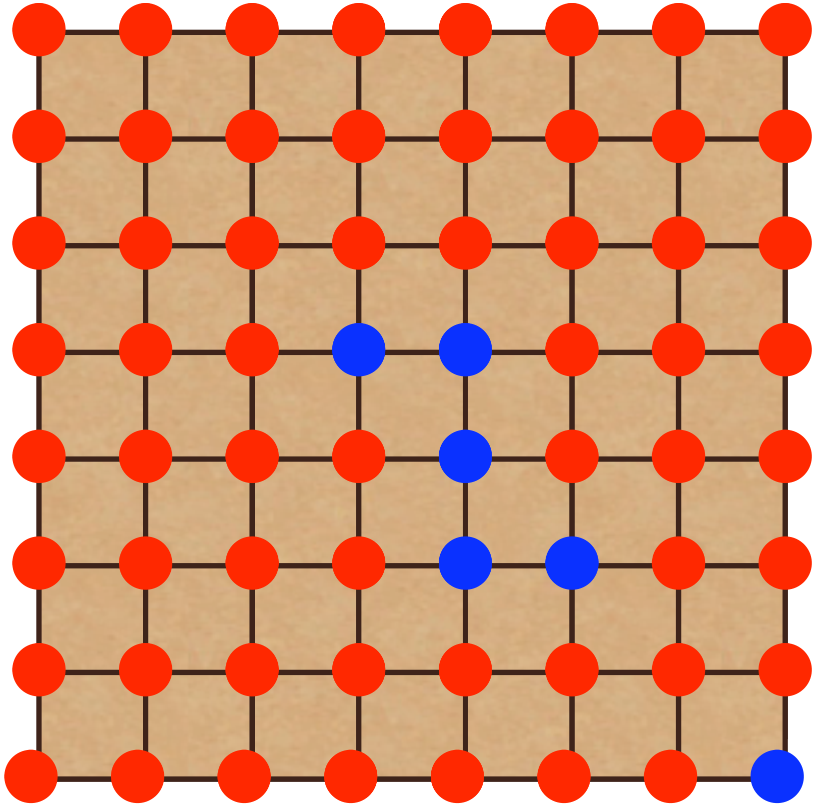
4. new: diff color, old: same color

$$T(o \to n) = \exp(-2\beta J)$$

$$T(n \to o) = 1$$

$$\exp(-\Delta E) = \exp(-2\beta J)$$

Cluster Algorithms

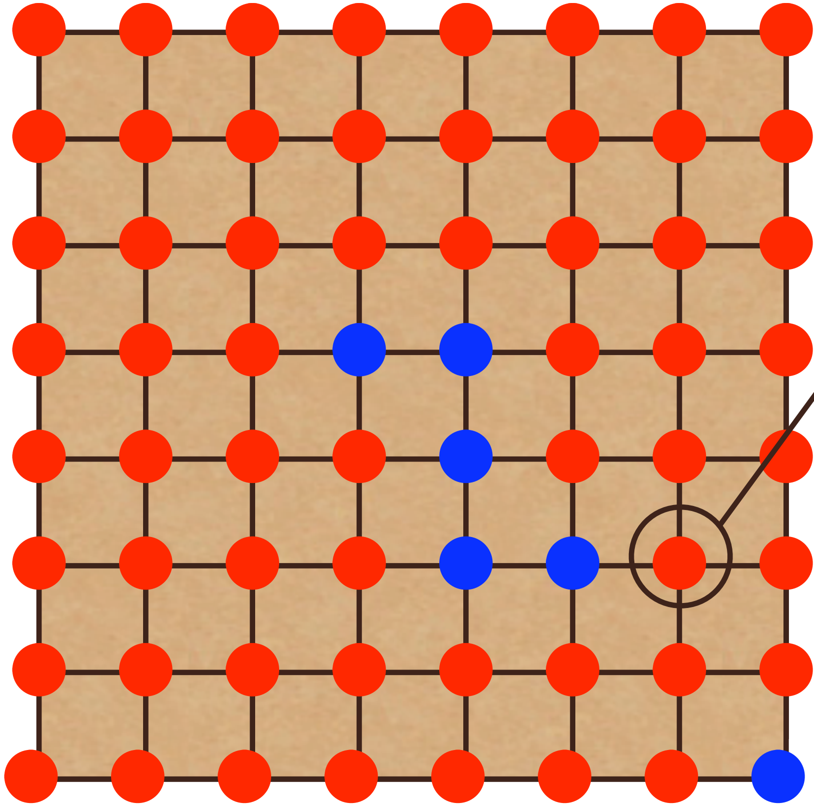


Wolff Algorithm

$$p_i = 1 - \exp(-2\beta J)$$

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms



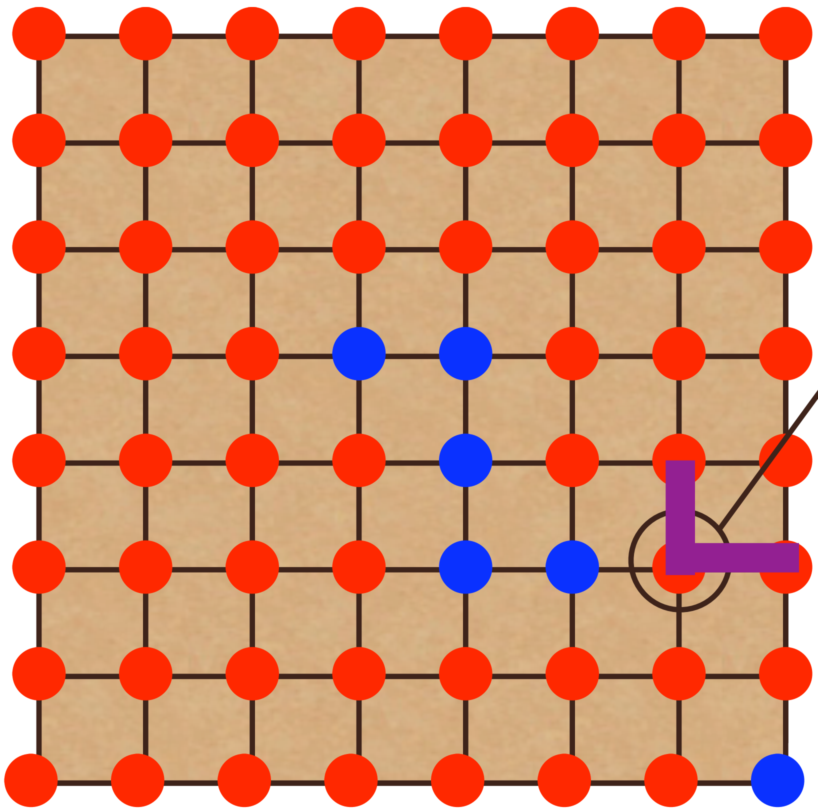
Wolff Algorithm

- Choose a spin at random

$$p_i = 1 - \exp(-2\beta J)$$

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms



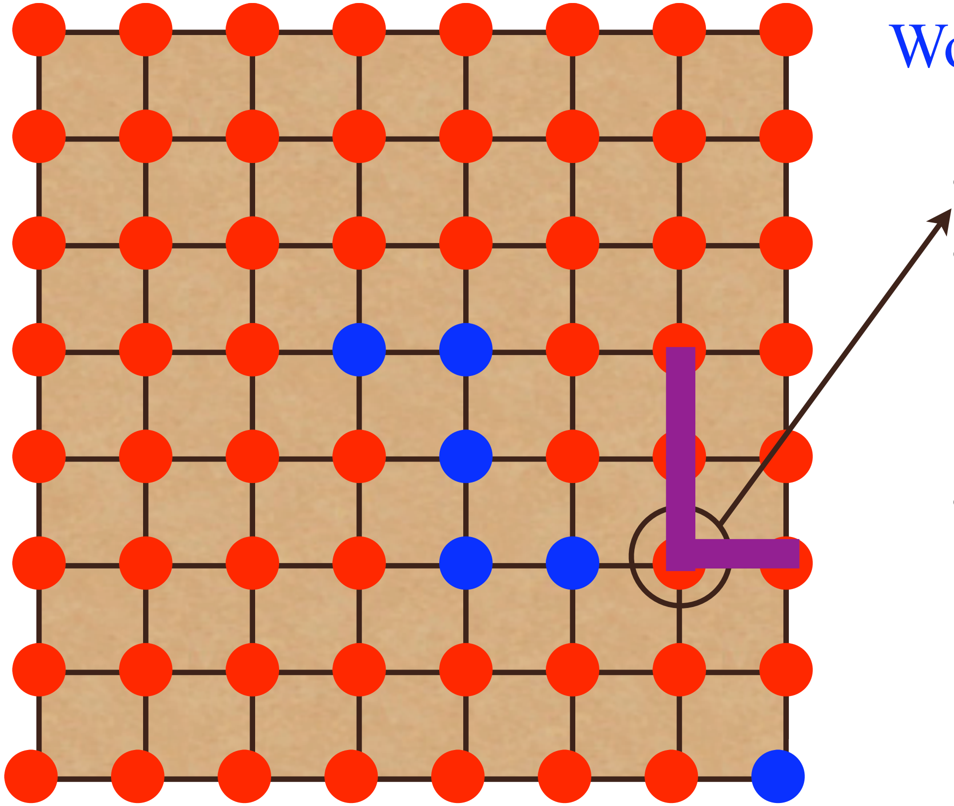
Wolff Algorithm

- Choose a spin at random
- Look at its same spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms



Wolff Algorithm

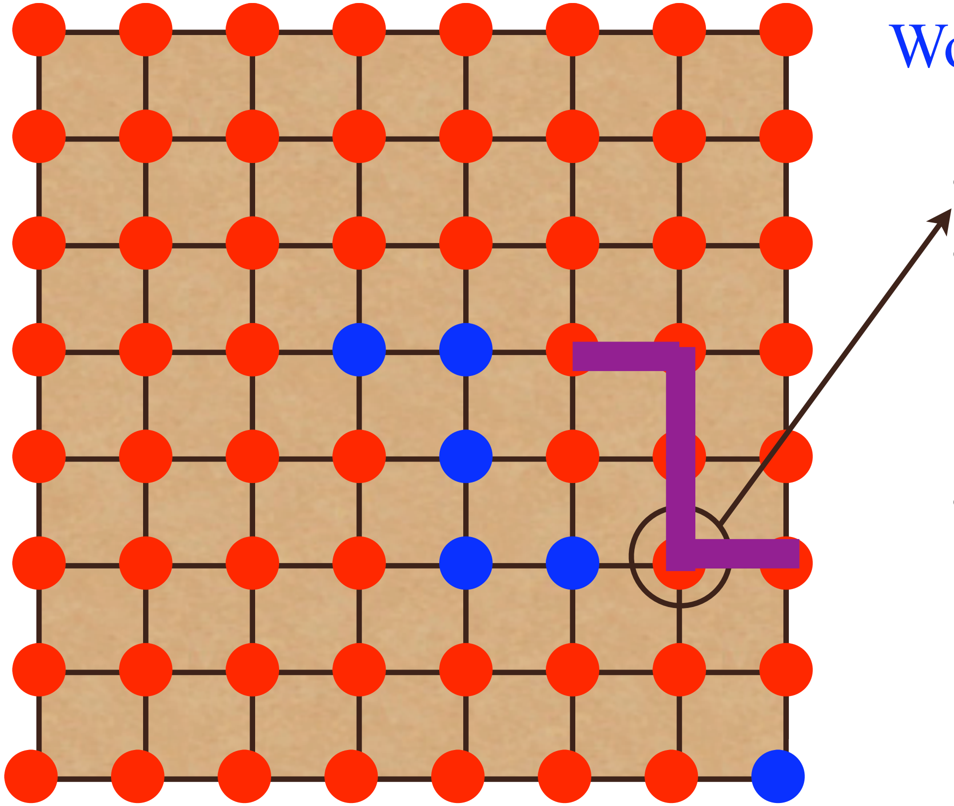
- Choose a spin at random
- Look at its same spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

- Look at their same-spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms



Wolff Algorithm

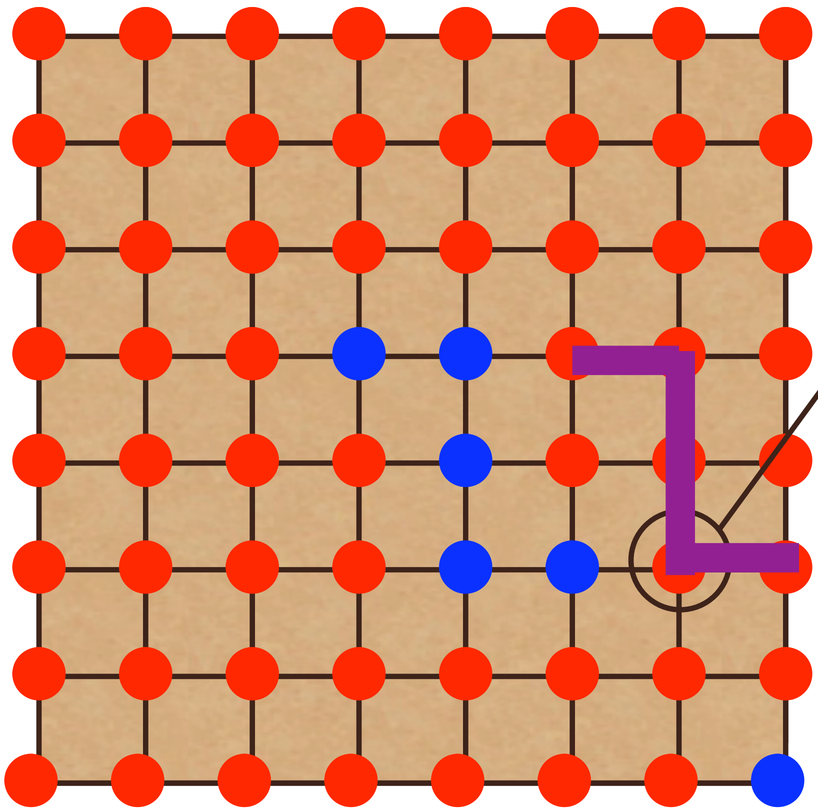
- Choose a spin at random
- Look at its same spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

- Look at their same-spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

Cluster Algorithms



Wolff Algorithm

- Choose a spin at random
- Look at its same spin neighbors and include them with probability

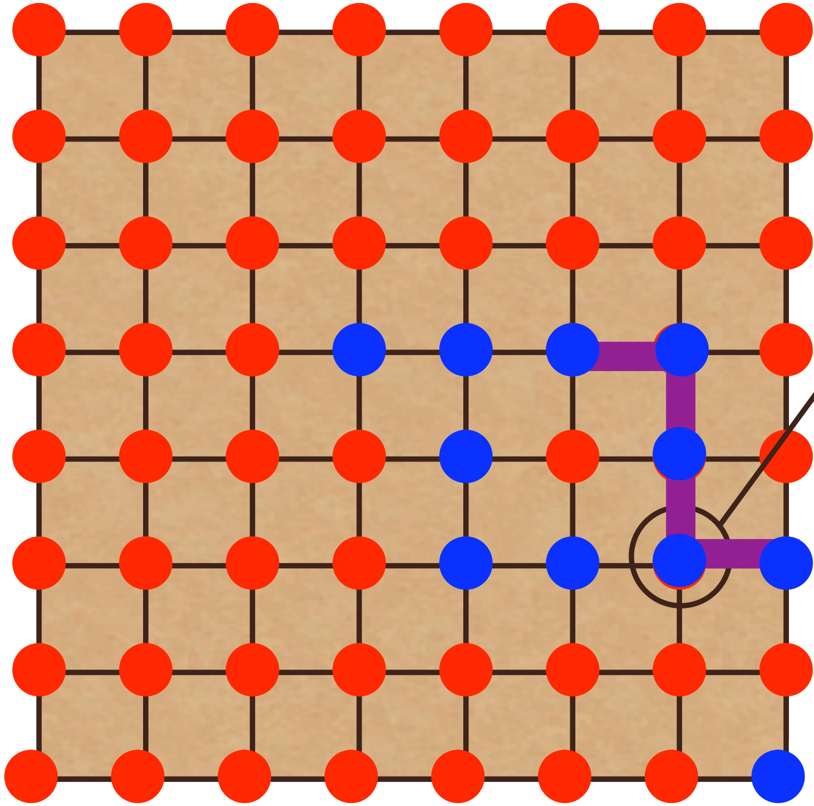
$$p_i = 1 - \exp(-2\beta J)$$

- Look at their same-spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

- Flip the cluster to opposite spin.

Cluster Algorithms



Wolff Algorithm

- Choose a spin at random
- Look at its same spin neighbors and include them with probability

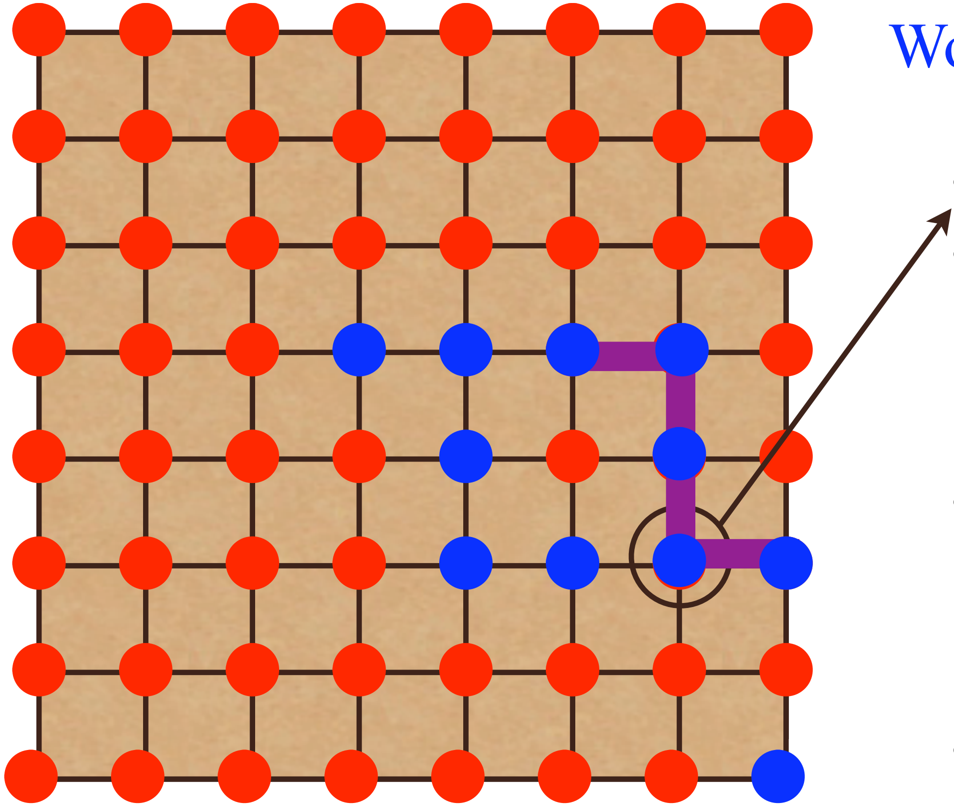
$$p_i = 1 - \exp(-2\beta J)$$

- Look at their same-spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

- Flip the cluster to opposite spin.

Cluster Algorithms



Wolff Algorithm

- Choose a spin at random
- Look at its same spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

- Look at their same-spin neighbors and include them with probability

$$p_i = 1 - \exp(-2\beta J)$$

- Flip the cluster to opposite spin.
- Accept with probability

$$P_{\mathrm{accept}} = min\left\{1, \exp(-\Delta E)\frac{T(n \to o)}{T(o \to n)}\right\}$$

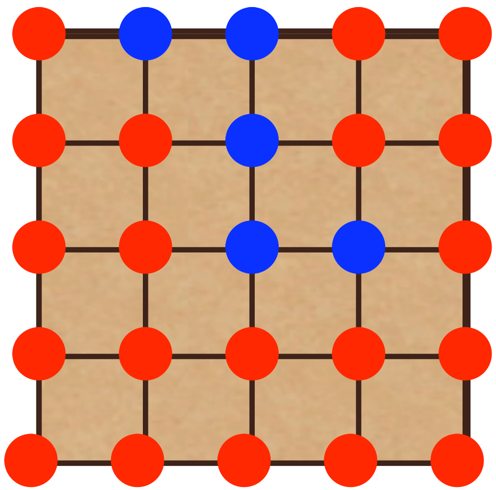
The story so far

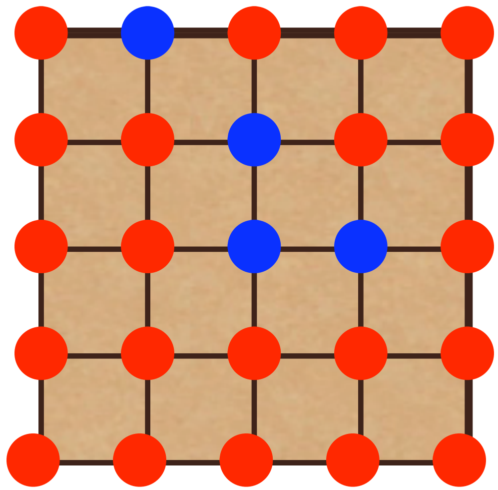
We need to calculate things like:

$$Z = \sum_{\{s\}} \exp(-K s_1 s_2) \exp(-K s_1 s_3) \exp(-K s_2 s_3)$$

and

$$\langle M^2 \rangle = \frac{1}{Z} \sum_{\{s\}} (s_1 + s_2 + s_3)^2 \exp(-K s_1 s_2) \exp(-K s_1 s_3) \exp(-K s_2 s_3)$$





Need Mapping to p and O

Need Mapping to p and O

Monte Carlo:

$$I(x) = \frac{1}{S}\sum p(x)O(x)dx$$

$$S = \sum p(x)dx$$

Need Mapping to p and O

Monte Carlo:

$$I(x) = \frac{1}{S} \sum p(x) O(x) dx \qquad S = \sum p(x) dx$$

Want:

$$M^2(x) = \frac{1}{Z} \sum_{s \in \{-1,1\}^{n \times n}} \exp(-\beta J \sum_{\langle i,j \rangle} s_i s_j) (\sum_i s_i)^2 di$$

$$Z = \int \exp(-\beta J \sum_{\langle i,j \rangle} s_i s_j) di$$

Worm Algorithm

Expand the exponential

What we need is a different representation!

Have: $\sum_{s \in \{-1,1\}^{n \times n}} \exp(-K s_1 s_2) \exp(-K s_1 s_3) \exp(-K s_2 s_3) (\sum_i s_i)^2$

Expand the exponential

What we need is a different representation!

Have: $\sum_{s\in\{-1,1\}^{n\times n}} \exp(-Ks_1s_2)\exp(-Ks_1s_3)\exp(-Ks_2s_3)(\sum_i s_i)^2$

Expand the exponential

What we need is a different representation!

$$\text{Have: } \sum_{s\in\{-1,1\}^{n\times n}} \exp(-Ks_1s_2)\exp(-Ks_1s_3)\exp(-Ks_2s_3)(\sum_i s_i)^2$$

$$(1 - Ks_1s_2 + \frac{K^2}{2!}(s_1s_2)^2 - \frac{K^3}{3!}(s_1s_2)^3 + \ldots)$$

Expand the exponential

What we need is a different representation!

Have: $\sum_{s\in\{-1,1\}^{n\times n}} \exp(-Ks_1s_2)\exp(-Ks_1s_3)\exp(-Ks_2s_3)(\sum_i s_i)^2$

$$(1 - Ks_1s_2 + \frac{K^2}{2!}(s_1s_2)^2 - \frac{K^3}{3!}(s_1s_2)^3 + \ldots)$$

$$(1 + \frac{K^2}{2!} + \ldots) - (s_1s_2)(K + \frac{K^3}{3!} + \frac{K^5}{5!}\ldots)$$

MSE485/PHY466/CSE485
Atomic Scale SIMULATION

Expand the exponential

What we need is a different representation!

Have: $\sum_{s\in\{-1,1\}^{n\times n}} \exp(-Ks_1s_2)\exp(-Ks_1s_3)\exp(-Ks_2s_3)(\sum_i s_i)^2$

$$(1 - Ks_1s_2 + \frac{K^2}{2!}(s_1s_2)^2 - \frac{K^3}{3!}(s_1s_2)^3 + \ldots)$$

$$(1 + \frac{K^2}{2!} + \ldots) - (s_1s_2)(K + \frac{K^3}{3!} + \frac{K^5}{5!}\ldots)$$

$$\cosh(K) + (s_1s_2)\sinh(K) =$$

$$\cosh(K)(1 + s_1s_2\tanh(K))$$

Sum over spins -> sum over bonds

$$\sum_{\{s\}} \exp(-K s_1 s_2) \exp(-K s_1 s_3) \exp(-K s_2 s_3) (\sum_i s_i)^2$$

Last slide:

$\cosh(K)(1 + s_1 s_2 \tanh(K))$

Sum over spins -> sum over bonds

$$\sum_{\{s\}} \exp(-K s_1 s_2) \exp(-K s_1 s_3) \exp(-K s_2 s_3) (\sum_i s_i)^2$$

Last slide:

$\cosh(K)(1 + s_1 s_2 \tanh(K))$

$$\cosh(K)^{2^n} \sum_{\{s\}} (1 + s_1 s_2 \tanh(K))(1 + s_1 s_3 \tanh(K))(1 + s_2 s_3 \tanh(K))$$

Sum over spins -> sum over bonds

$$\sum_{\{s\}} \exp(-K s_1 s_2) \exp(-K s_1 s_3) \exp(-K s_2 s_3) (\sum_i s_i)^2$$

Last slide:

$\cosh(K)(1 + s_1 s_2 \tanh(K))$

$$\cosh(K)^{2^n} \sum_{\{s\}} (1 + s_1 s_2 \tanh(K))(1 + s_1 s_3 \tanh(K))(1 + s_2 s_3 \tanh(K))$$

Expand the sum!

$$\sum_{s\in\{-1,1\}^{n\times n}} (s_1 s_2)(s_1 s_3) [\tanh(K)]^2 (\sum_i s_i)^2$$

Sum over spins -> sum over bonds

$$\sum_{\{s\}} \exp(-K s_1 s_2) \exp(-K s_1 s_3) \exp(-K s_2 s_3) (\sum_i s_i)^2$$

Last slide:

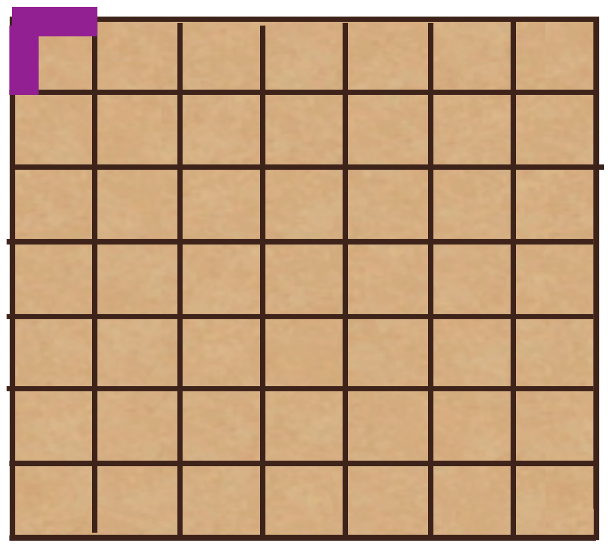
$\cosh(K)(1 + s_1 s_2 \tanh(K))$

$$\cosh(K)^{2^n} \sum_{\{s\}} (1 + s_1 s_2 \tanh(K))(1 + s_1 s_3 \tanh(K))(1 + s_2 s_3 \tanh(K))$$

Expand the sum!

Represent visually

$$\sum_{s \in \{-1,1\}^{n \times n}} (s_1 s_2)(s_1 s_3) [\tanh(K)]^2 (\sum_i s_i)^2$$



Add bonds as Monte Carlo variables!

Expand the square

$$\sum_{\text{terms}} \sum_{s\in\{-1,1\}^{n\times n}} (s_1 s_2)(s_1 s_3)\left[\tanh(K)\right]^2 \left(\sum_i s_i\right)^2$$

Expand the square

$$\sum_{\text{terms}} \sum_{s\in\{-1,1\}^{n\times n}} (s_1 s_2)(s_1 s_3)\left[\tanh(K)\right]^2 \left(\sum_i s_i\right)^2$$

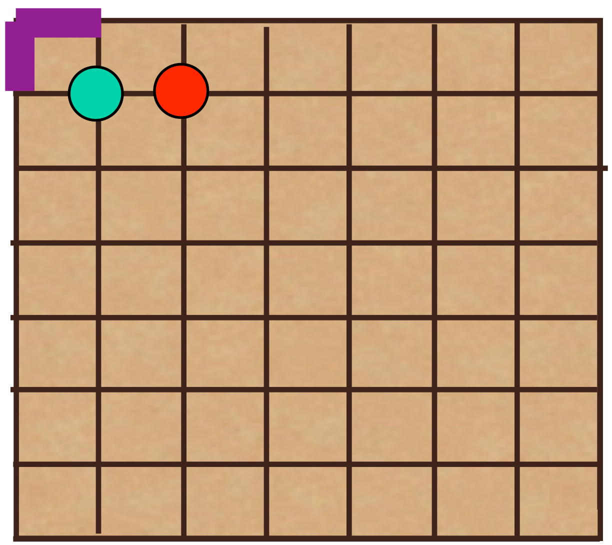
$$\sum_{h,t} \sum_{\text{terms}} \sum_{s\in\{-1,1\}^{n}} (s_1 s_2)(s_1 s_3)\left[\tanh(K)\right]^2 (s_h s_t)$$

Expand the square

$$\sum_{\text{terms}} \sum_{s\in\{-1,1\}^{n\times n}} (s_1 s_2)(s_1 s_3)\,[\tanh(K)]^2 (\sum_i s_i)^2$$

$$\sum_{h,t} \sum_{\text{terms}} \sum_{s\in\{-1,1\}^{n}} (s_1 s_2)(s_1 s_3)\,[\tanh(K)]^2\,(s_h s_t)$$

Represent visually

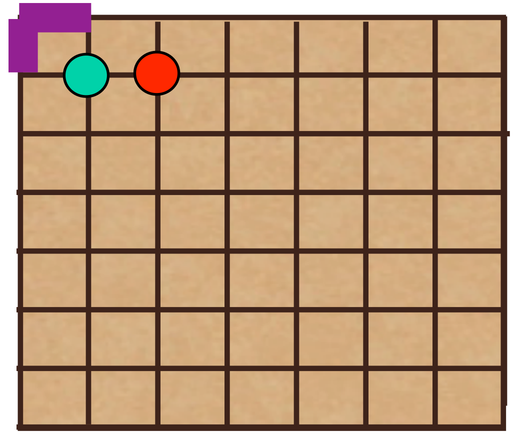


Add two monte carlo variables, the head and tail.

More abstractly ...

Let $\hat{B}$ be the set of all bonds

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} \exp(-K s_u s_v)\left(\sum_i s_i\right)^2$$

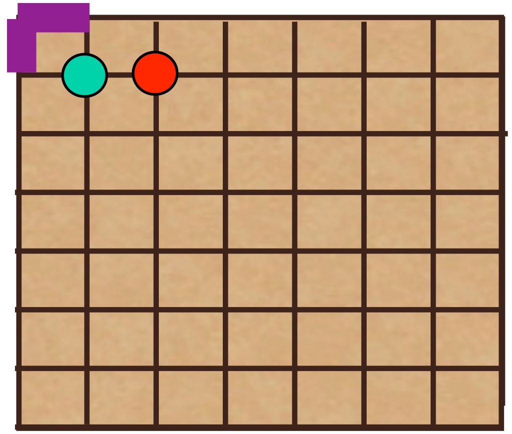


More abstractly ...

Let $\hat{B}$ be the set of all bonds

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} \exp(-K s_u s_v)(\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} (1 + s_u s_v \tanh(K))(\sum_i s_i)^2$$



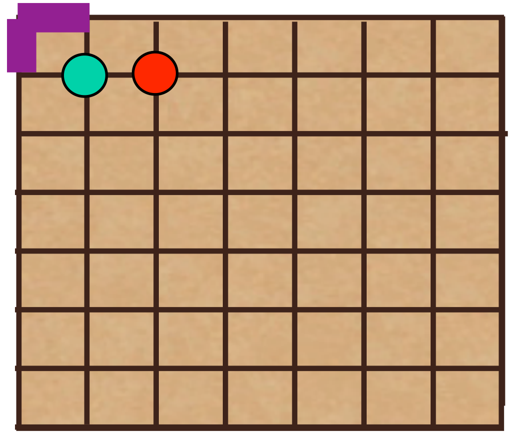
More abstractly …

Let $\hat{B}$ be the set of all bonds

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} \exp(-K s_u s_v)(\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} (1 + s_u s_v \tanh(K))(\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \sum_{B\subset\hat{B}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v (\sum_i s_i)^2$$



MSE at Illinois | Physics Illinois | Atomic Scale SIMULATION MSE485/PHY466/CSE485

More abstractly ...

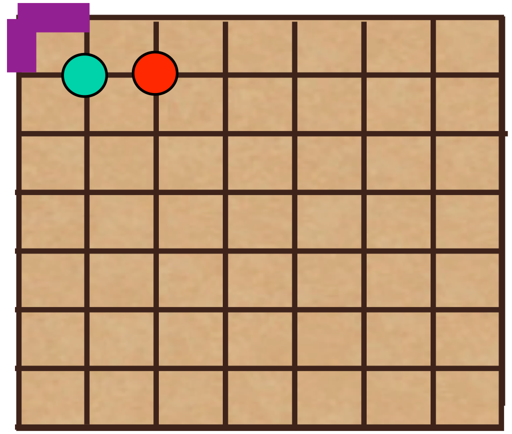
Let $\hat{B}$ be the set of all bonds

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} \exp(-K s_u s_v) (\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} (1 + s_u s_v \tanh(K)) (\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \sum_{B\subset\hat{B}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v (\sum_i s_i)^2$$

$$\sum_{B\subset\hat{B}} \sum_{s\in\{-1,1\}^{n\times n}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v (\sum_i s_i)^2$$



More abstractly ...

Let $\hat{B}$ be the set of all bonds

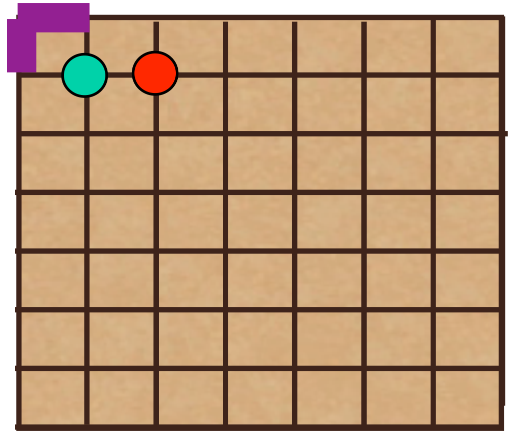
$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} \exp(-K s_u s_v)(\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} (1 + s_u s_v \tanh(K))(\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \sum_{B\subset\hat{B}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v(\sum_i s_i)^2$$

$$\sum_{B\subset\hat{B}} \sum_{s\in\{-1,1\}^{n\times n}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v(\sum_i s_i)^2$$

$$\sum_{h,t} \sum_{B\subset\hat{B}} \sum_{s\in\{-1,1\}^{n\times n}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v(s_h s_t)$$



MSE Illinois Physics Illinois

More abstractly ...

Let $\hat{B}$ be the set of all bonds

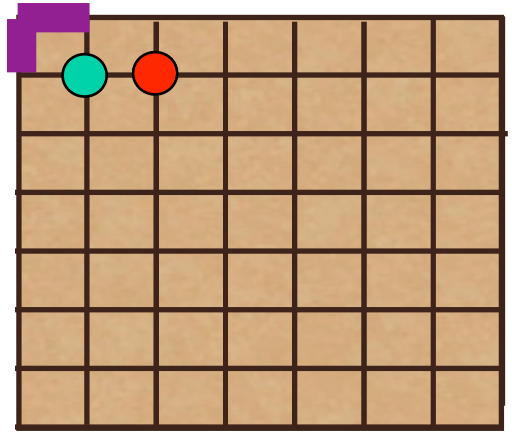
$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} \exp(-K s_u s_v) (\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in\hat{B}} (1 + s_u s_v \tanh(K)) (\sum_i s_i)^2$$

$$\sum_{s\in\{-1,1\}^{n\times n}} \sum_{B\subset\hat{B}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v (\sum_i s_i)^2$$

$$\sum_{B\subset\hat{B}} \sum_{s\in\{-1,1\}^{n\times n}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v (\sum_i s_i)^2$$

$$\sum_{h,t} \sum_{B\subset\hat{B}} \sum_{s\in\{-1,1\}^{n\times n}} \tanh(K)^{|B|} \prod_{uv\in B} s_u s_v (s_h s_t)$$



Monte Carlo:

$$I(x) = \frac{1}{S}\sum p(x) O(x) dx \qquad S = \sum p(x) dx$$

Monte Carlo:

$$I(x) = \frac{1}{S}\sum p(x) O(x) dx \qquad S = \sum p(x) dx$$

Want:

$$M^2 = \frac{1}{Z} \sum_{t,h,B\subset\hat{B}} \left[\tanh(K)^{|B|} \sum_{s\in\{0,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_t s_h) \right]$$

$$Z = \sum_{t,h,B\subset\hat{B}} \tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v$$

$$\hat{M}^2 = \frac{M^2 Z}{S}$$

Monte Carlo:

$$I(x) = \frac{1}{S}\sum p(x) O(x) dx \qquad S = \sum p(x) dx$$

Want:

$$M^2 = \frac{1}{Z}\sum_{t,h,B\subset\hat{B}}\left[\tanh(K)^{|B|}\sum_{s\in\{0,1\}^{n\times n}}\prod_{uv\in B} s_u s_v (s_t s_h)\right]$$

$$Z = \sum_{t,h,B\subset\hat{B}}\tanh(K)^{|B|}\sum_{s\in\{-1,1\}^{n\times n}}\prod_{uv\in B} s_u s_v$$

Can Have:

$$\hat{M}^2 = \frac{1}{S}\sum_{h,t,B\subset\hat{B}}\left[\tanh(K)^{|B|}\sum_{s\in\{-1,1\}^{n\times n}}\prod_{uv\in B}(s_u s_v)(s_h s_t)\right] \qquad 1$$

$$S = \sum_{h,t,B\in\hat{B}}\tanh(K)^{|B|}\sum_{s\in\{-1,1\}^{n\times n}}\prod_{uv\in B} s_u s_v (s_h s_t)$$

$$\hat{M}^2 = \frac{M^2 Z}{S}$$

Getting Z/S

$$\hat{M}^2 = \frac{M^2 Z}{S}$$

Getting Z/S

$$\hat{M}^2 = \frac{M^2 Z}{S}$$

Monte Carlo:

$$I(x) = \frac{1}{S} \sum p(x) O(x) dx$$

$$S = \sum p(x) dx$$

Getting Z/S

$$\hat{M}^2 = \frac{M^2 Z}{S}$$

Monte Carlo:

$$I(x) = \frac{1}{S}\sum p(x) O(x) dx \qquad S = \sum p(x) dx$$

Want:

$$\frac{Z}{S} = \frac{1}{S}\sum_{B\in\hat{B}}\left[\tanh(K)^{|B|}\sum_{s}\prod_{uv\in B} s_u s_v\right] \qquad S = \sum_{h,t,B\in\hat{B}} \tanh^{|B|}\sum_{s}\prod_{i\in B} s_u s_v (s_h s_t)$$

Getting Z/S

$$\hat{M}^2 = \frac{M^2 Z}{S}$$

Monte Carlo:

$$I(x) = \frac{1}{S}\sum p(x) O(x) dx \qquad S = \sum p(x) dx$$

Want:

$$\frac{Z}{S} = \frac{1}{S}\sum_{B\in\hat{B}}\left[\tanh(K)^{|B|}\sum_s \prod_{uv\in B} s_u s_v\right] \qquad S = \sum_{h,t,B\in\hat{B}} \tanh^{|B|}\sum_s \prod_{i\in B} s_u s_v (s_h s_t)$$

$$\frac{Z}{S} = \frac{1}{S}\sum_{h,t,B\in\hat{B}}\left[\tanh(K)^{|B|}\sum_s \prod_{uv\in B} s_u s_v (s_h s_t)\right] \frac{\tanh(K)^{|B|}\sum_s \prod_{uv\in B} s_u s_v \delta_{h,t}}{\tanh(K)^{|B|}\sum_s \prod_{uv\in B} s_u s_v (s_h s_t)}$$

Getting Z/S

$$\hat{M}^2 = \frac{M^2 Z}{S}$$

Monte Carlo:

$$I(x) = \frac{1}{S}\sum p(x)O(x)dx \qquad S = \sum p(x)dx$$

Want:

$$\frac{Z}{S} = \frac{1}{S}\sum_{B\in\hat{B}}\left[\tanh(K)^{|B|}\sum_s \prod_{uv\in B} s_u s_v\right] \qquad S = \sum_{h,t,B\in\hat{B}} \tanh^{|B|}\sum_s \prod_{i\in B} s_u s_v (s_h s_t)$$

$$\frac{Z}{S} = \frac{1}{S}\sum_{h,t,B\in\hat{B}}\left[\tanh(K)^{|B|}\sum_s \prod_{uv\in B} s_u s_v (s_h s_t)\right] \frac{\tanh(K)^{|B|}\sum_s \prod_{uv\in B} s_u s_v \delta_{h,t}}{\tanh(K)^{|B|}\sum_s \prod_{uv\in B} s_u s_v (s_h s_t)}$$

Getting M

- Calculate $\hat{M}^2 = \frac{M^2 Z}{S}$ and $\frac{Z}{S}$

- Still need to evaluate:

  - $\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n \times n}} \prod_{uv \in B} s_u s_v \delta_{h,t}$

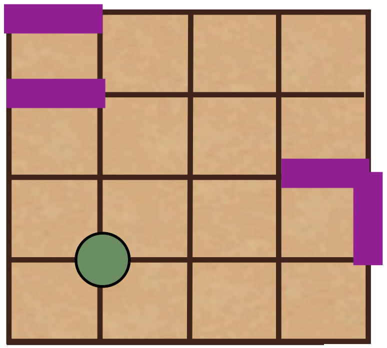
  - $\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n \times n}} \prod_{uv \in B} s_u s_v (s_h s_t)$

$$\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n \times n}} \prod_{uv \in B} s_u s_v \delta_{h,t}$$

37

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

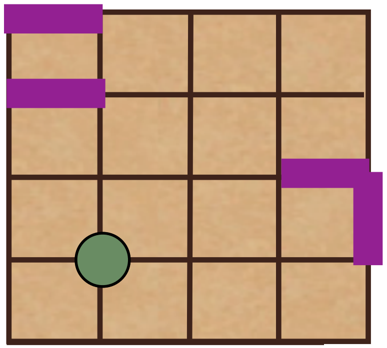
$$\sum_{\{s\}} (s_1 s_2)(s_6 s_7)(s_{14} s_{15})(s_{15} s_{20}) \left[\tanh(K)\right]^4 \delta_{h,t}$$



$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

$$\sum_{\{s\}} (s_1 s_2)(s_6 s_7)(s_{14} s_{15})(s_{15} s_{20}) \left[\tanh(K)\right]^4 \delta_{h,t}$$

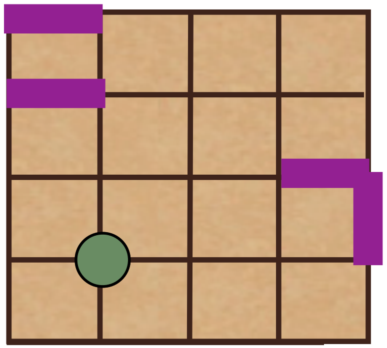
1



$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

$$\sum_{\{s\}} (s_1 s_2)(s_6 s_7)(s_{14} s_{15})(s_{15} s_{20}) \left[\tanh(K)\right]^4 \delta_{h,t}$$

1



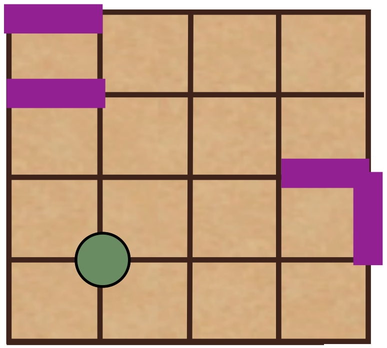
$$(1) * \sum_{\{s-s_1\}} (s_2 s_6 s_7 s_{14} s_{20}) +$$

$$(-1) * \sum_{\{s-s_1\}} (s_2 s_6 s_7 s_{14} s_{20})$$

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

1

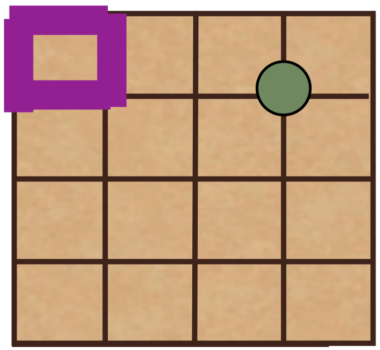
$$\sum_{\{s\}} (s_1 s_2)(s_6 s_7)(s_{14} s_{15})(s_{15} s_{20}) \left[\tanh(K)\right]^4 \delta_{h,t}$$



$$(1) * \sum_{\{s-s_1\}} (s_2 s_6 s_7 s_{14} s_{20}) +$$

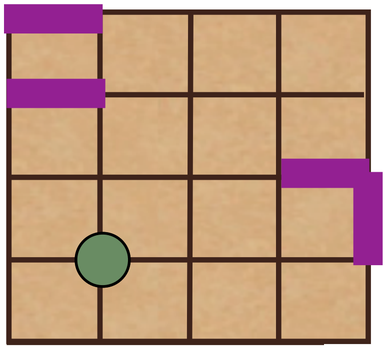
$$(-1) * \sum_{\{s-s_1\}} (s_2 s_6 s_7 s_{14} s_{20})$$

$$\sum_{\{s\}} (s_1 s_2)(s_2 s_7)(s_7 s_6)(s_6 s_1) \left[\tanh(K)\right]^4$$



$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

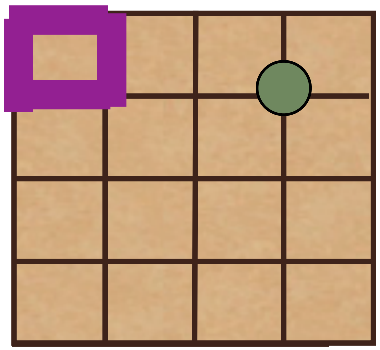
$$\sum_{\{s\}} (s_1 s_2)(s_6 s_7)(s_{14} \overbrace{s_{15})(s_{15}}^{1} s_{20}) \left[\tanh(K)\right]^4 \delta_{h,t}$$



$$(1) * \sum_{\{s-s_1\}} (s_2 s_6 s_7 s_{14} s_{20}) +$$

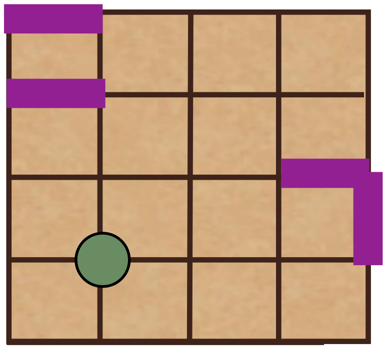
$$(-1) * \sum_{\{s-s_1\}} (s_2 s_6 s_7 s_{14} s_{20})$$

$$\sum_{\{s\}} (s_1 \overbrace{s_2)(s_2}^{1} \overbrace{s_7)(s_7}^{1} \overbrace{s_6)(s_6}^{1} s_1) \left[\tanh(K)\right]^4$$



$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

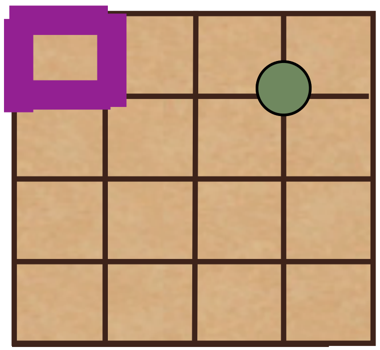
$$\sum_{\{s\}} (s_1 s_2)(s_6 s_7)(s_{14} \overbrace{s_{15})(s_{15}}^{1} s_{20}) \, [\tanh(K)]^4 \, \delta_{h,t}$$



$$(1) * \sum_{\{s-s_1\}} (s_2 s_6 s_7 s_{14} s_{20}) +$$

$$(-1) * \sum_{\{s-s_1\}} (s_2 s_6 s_7 s_{14} s_{20})$$

$$\sum_{\{s\}} (s_1 \overbrace{s_2)(s_2}^{1} \overbrace{s_7)(s_7}^{1} \overbrace{s_6)(s_6}^{1} s_1) \, [\tanh(K)]^4$$

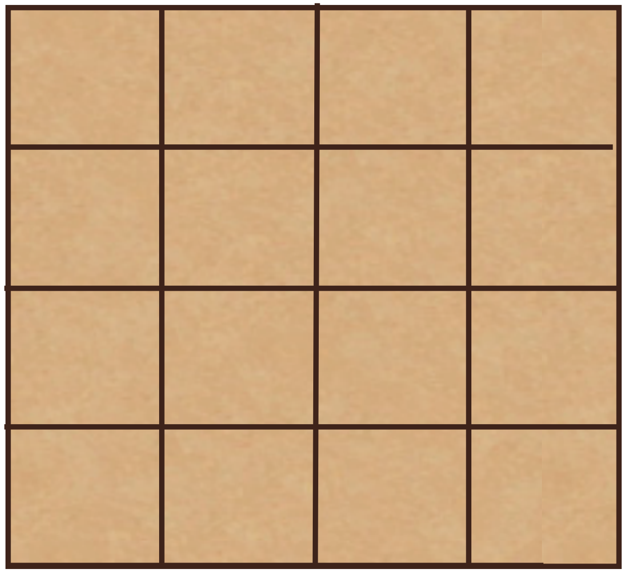


Conclusion:

"Even s" contribute tanh(K)$^{\#\text{ of bonds}}$

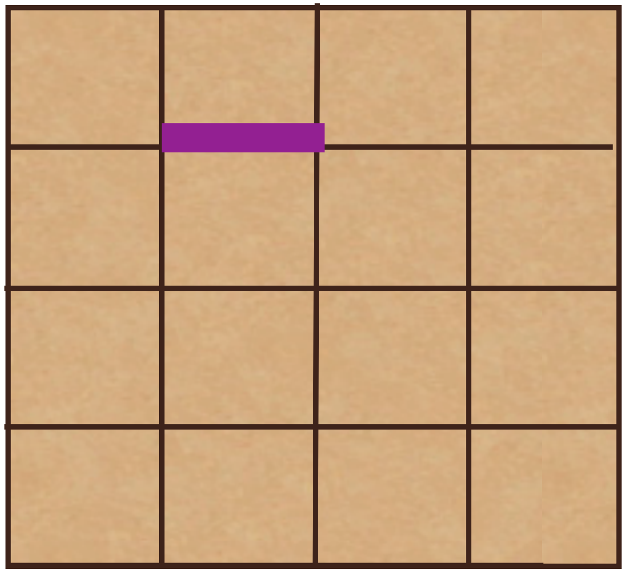
$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

Even s's



$$\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n \times n}} \prod_{uv \in B} s_u s_v \delta_{h,t}$$

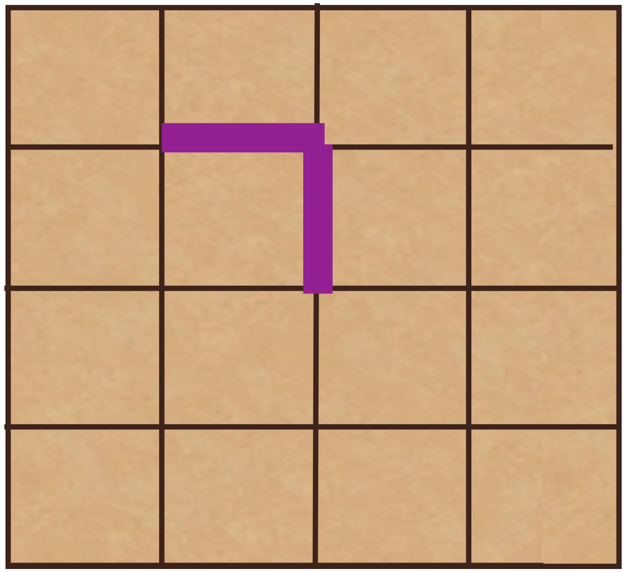
Even s's



(s7s8)

$$\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n \times n}} \prod_{uv \in B} s_u s_v \delta_{h,t}$$

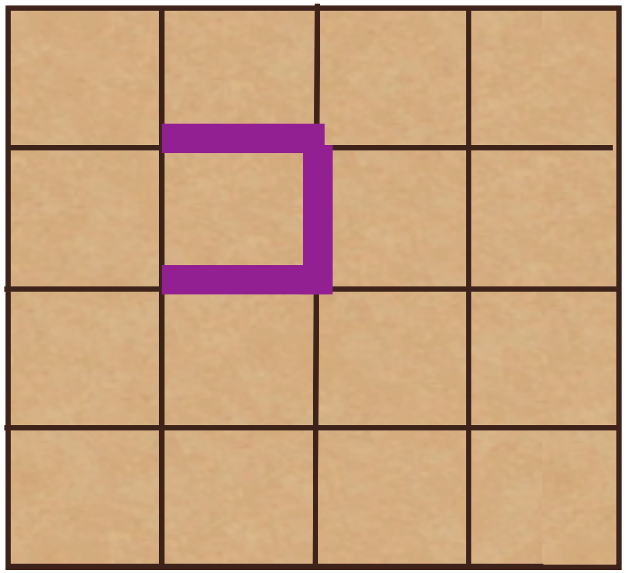
Even s's



(s7s8)(s8s13)

$$\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n \times n}} \prod_{uv \in B} s_u s_v \delta_{h,t}$$

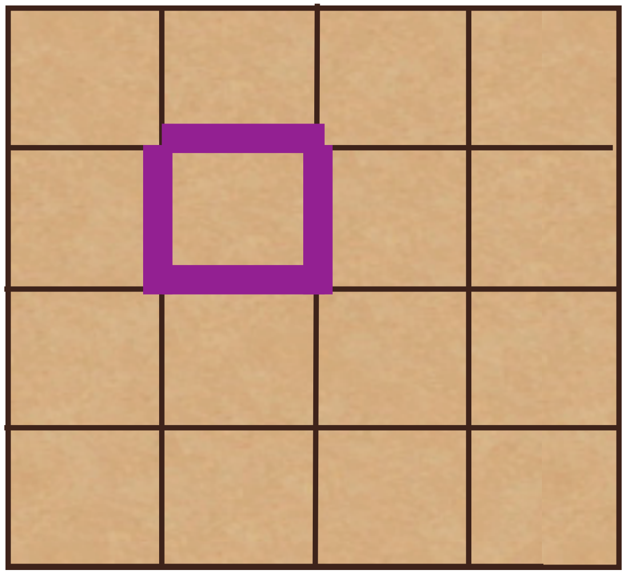
Even s's



(s7s8)(s8s13)(s13s12)

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

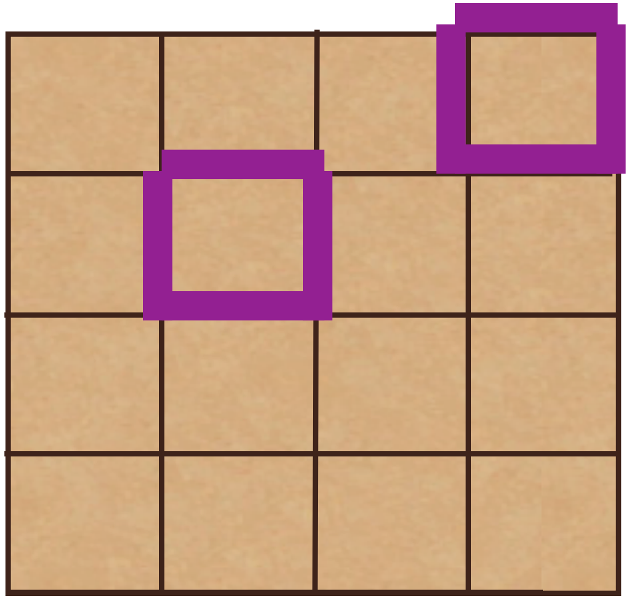
Even s's



(s7s8)(s8s13)(s13s12)(s12s7)

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v \delta_{h,t}$$

Even s's

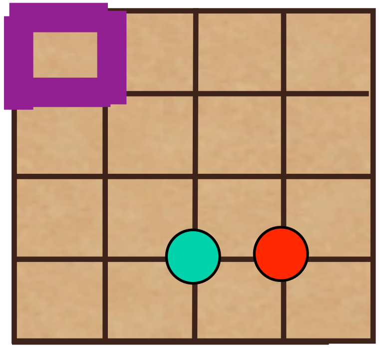


(s7s8)(s8s13)(s13s12)(s12s7)

(s7s8)(s8s13)(s13s12)(s12s7) (s4s5)(s5s10)(s10s9)(s9s5)

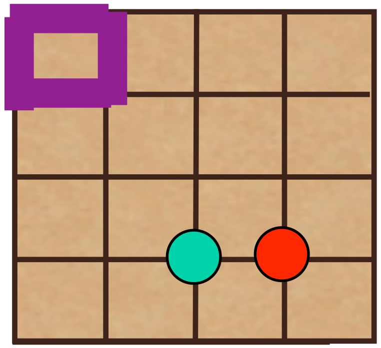
$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

$$\sum_{\{s\}} (s_1 s_2)(s_2 s_7)(s_7 s_6)(s_6 s_1) \left[\tanh(K)\right]^4 s_h s_t$$



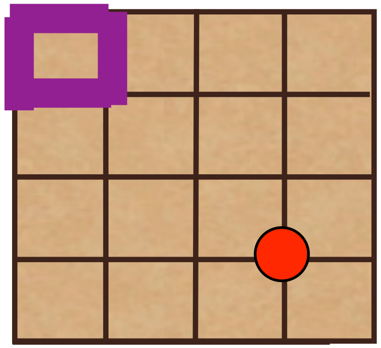
$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

$$\sum_{\{s\}} (s_1 \overset{1}{\overbrace{s_2)(s_2}} \overset{1}{\overbrace{s_7)(s_7}} \overset{1}{\overbrace{s_6)(s_6}} s_1) \, [\tanh(K)]^4 \, s_h s_t$$



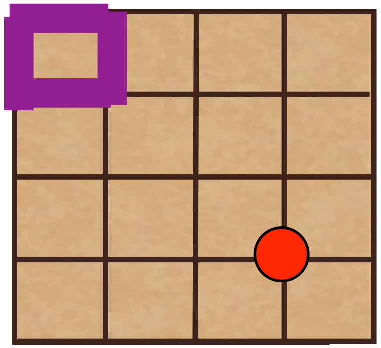
$$\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n \times n}} \prod_{uv \in B} s_u s_v (s_h s_t)$$

$$\sum_{\{s\}} (s_1 \overbrace{s_2)(s_2}^{1} \overbrace{s_7)(s_7}^{1} \overbrace{s_6)(s_6}^{1} s_1) \left[\tanh(K)\right]^4 s_h s_t$$



$$\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n \times n}} \prod_{uv \in B} s_u s_v (s_h s_t)$$

$$\sum_{\{s\}} (s_1 \underbrace{s_2)(s_2}_{1} \underbrace{s_7)(s_7}_{1} \underbrace{s_6)(s_6}_{1} s_1) \left[\tanh(K)\right]^4 s_h s_t$$

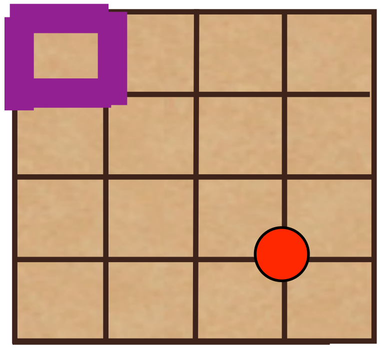


Even s: hd must equal tl

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

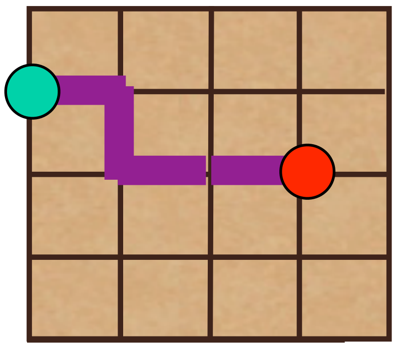
1 1 1

$$\sum_{\{s\}} (s_1 s_2)(s_2 s_7)(s_7 s_6)(s_6 s_1)\,[\tanh(K)]^4\, s_h s_t$$



Even s: hd must equal tl

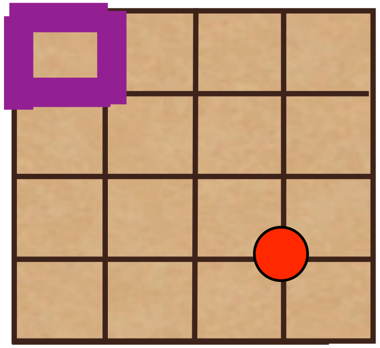
$$\sum_{\{s\}} (s_6 s_7)(s_7 s_{12})(s_{12} s_{13} s_{13} s_{14})\,[\tanh(K)]^4\,(s_6 s_{14})$$



$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

1 1 1

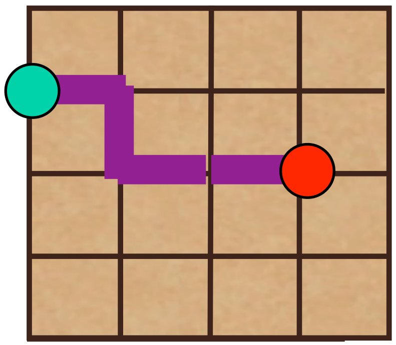
$$\sum_{\{s\}} (s_1 s_2)(s_2 s_7)(s_7 s_6)(s_6 s_1)\ [\tanh(K)]^4\ s_h s_t$$



Even s: hd must equal tl

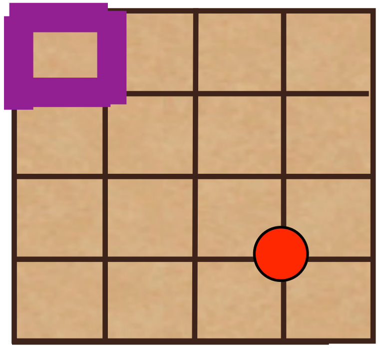
1 1 1

$$\sum_{\{s\}} (s_6 s_7)(s_7 s_{12})(s_{12} s_{13} s_{13} s_{14})\ [\tanh(K)]^4\ (s_6 s_{14})$$



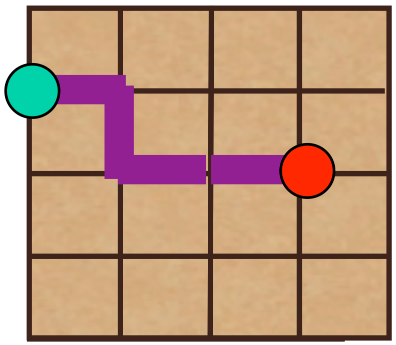
$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

$$\sum_{\{s\}} (s_1 \overbrace{s_2)(s_2}^{1} \overbrace{s_7)(s_7}^{1} \overbrace{s_6)(s_6}^{1} s_1)\,[\tanh(K)]^4\, s_h s_t$$



Even s: hd must equal tl

$$\sum_{\{s\}} (s_6 \overbrace{s_7)(s_7}^{1} \overbrace{s_{12})(s_{12}}^{1} \overbrace{s_{13} s_{13}}^{1} s_{14})\,[\tanh(K)]^4\,(s_6 s_{14})$$

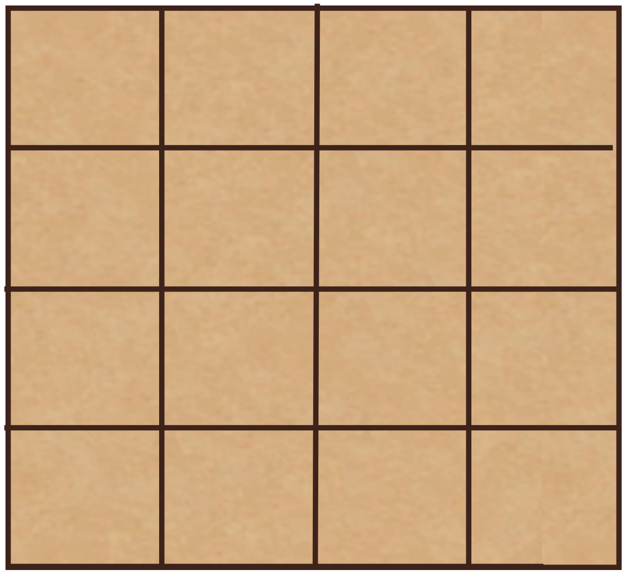


2 odd: hd and tl must cover them

> 2 odd: 0

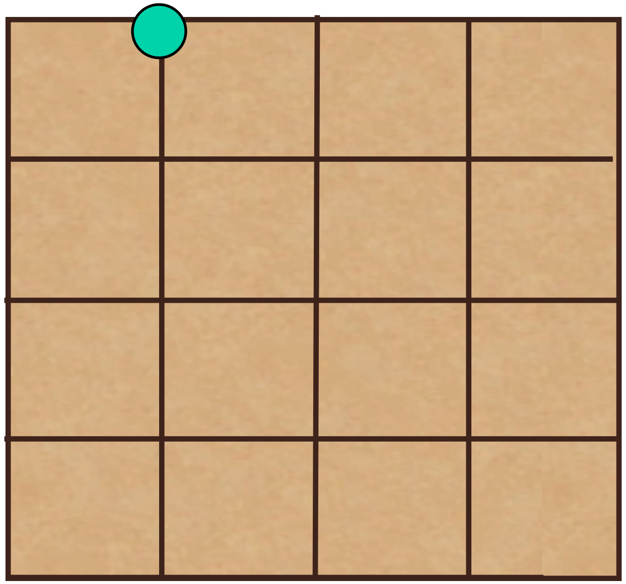
$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

Even s's + <2 odd



$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

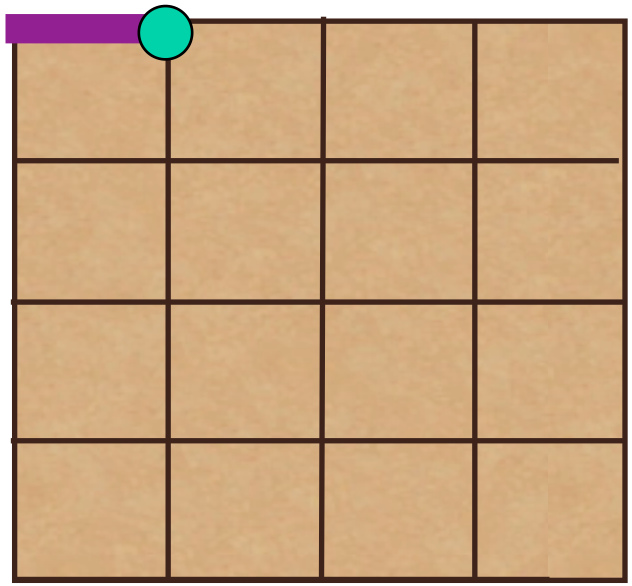
Even s's + <2 odd



s2

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

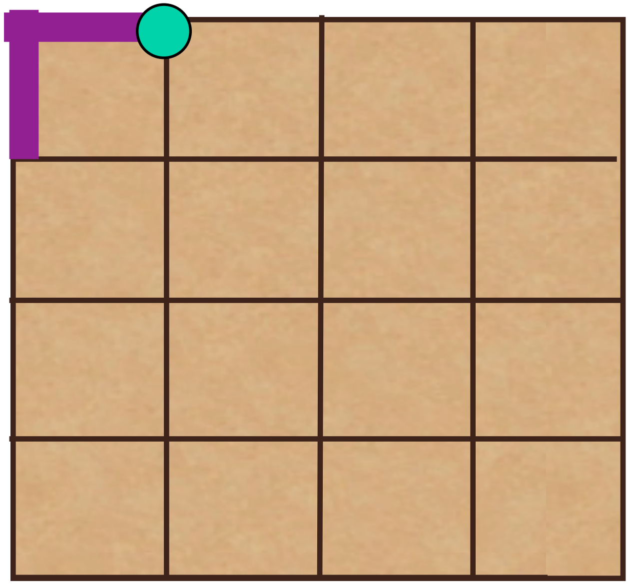
Even s's + <2 odd



s2(s2s1)

$$\tanh(K)^{|B|} \sum_{s \in \{-1,1\}^{n\times n}} \prod_{uv \in B} s_u s_v (s_h s_t)$$

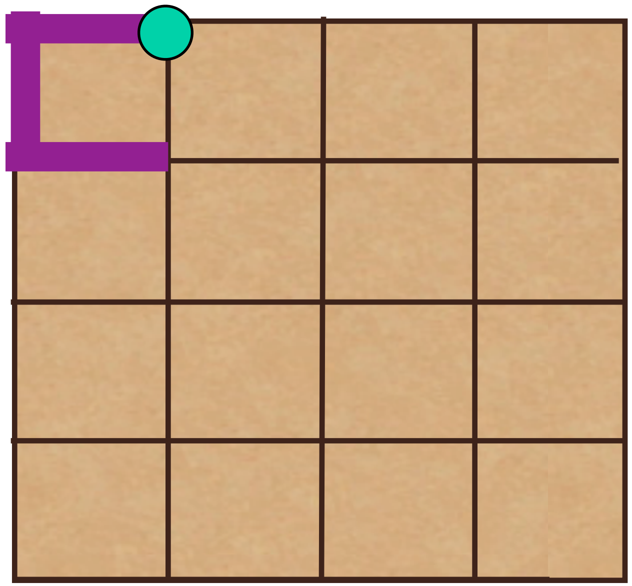
Even s's + <2 odd



s2(s2s1)(s1s6)

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

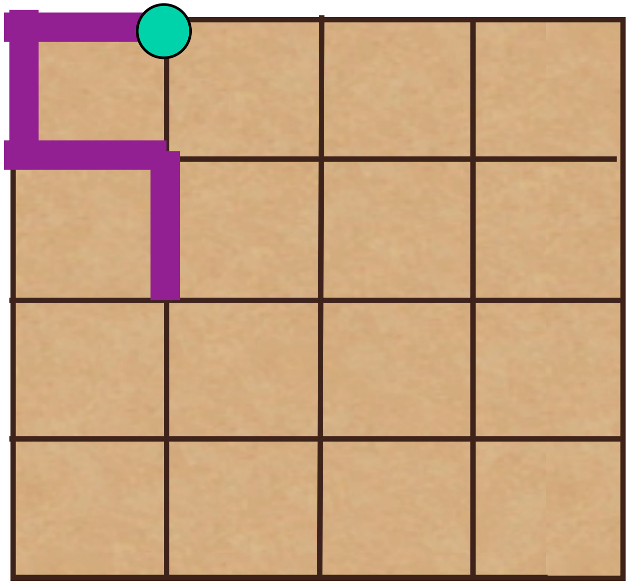
Even s's + <2 odd



s2(s2s1)(s1s6)(s6s7)

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

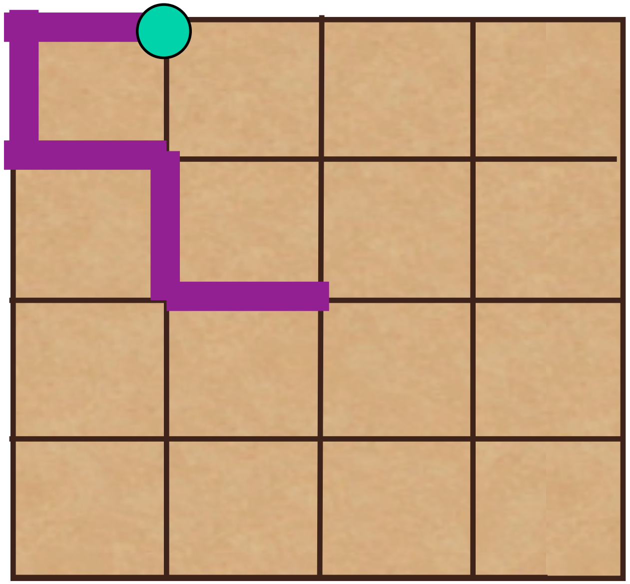
Even s's + <2 odd



s2(s2s1)(s1s6)(s6s7)(s7s12)

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

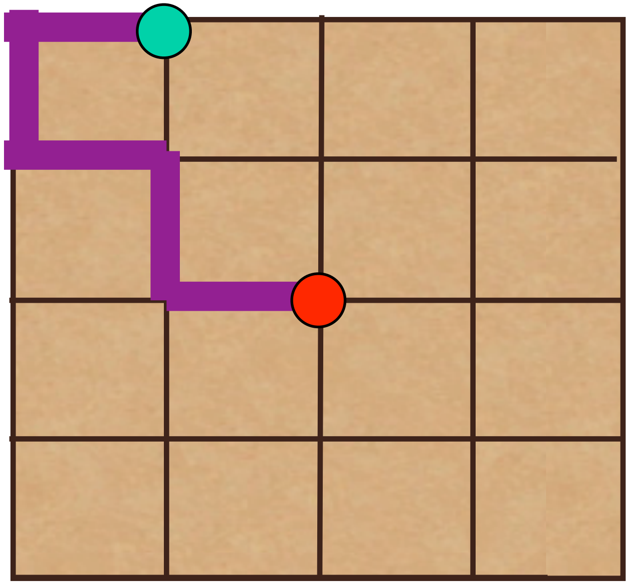
Even s's + <2 odd



s2(s2s1)(s1s6)(s6s7)(s7s12)(s12s13)

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

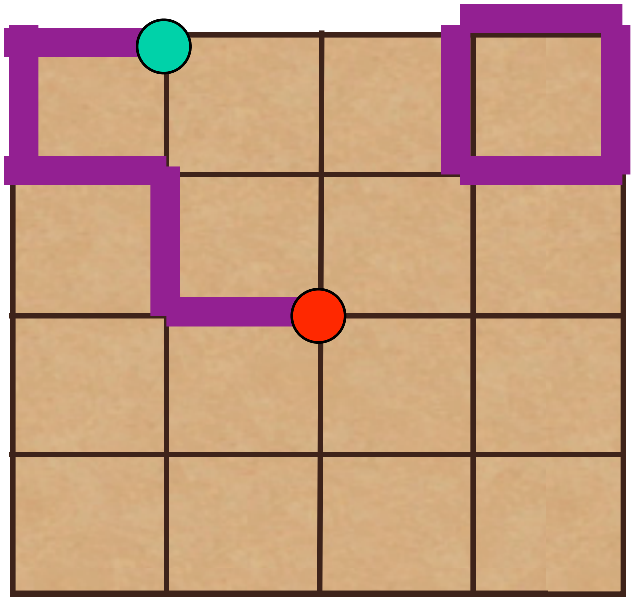
Even s's + <2 odd



s2(s2s1)(s1s6)(s6s7)(s7s12)(s12s13) s13

$$\tanh(K)^{|B|} \sum_{s\in\{-1,1\}^{n\times n}} \prod_{uv\in B} s_u s_v (s_h s_t)$$

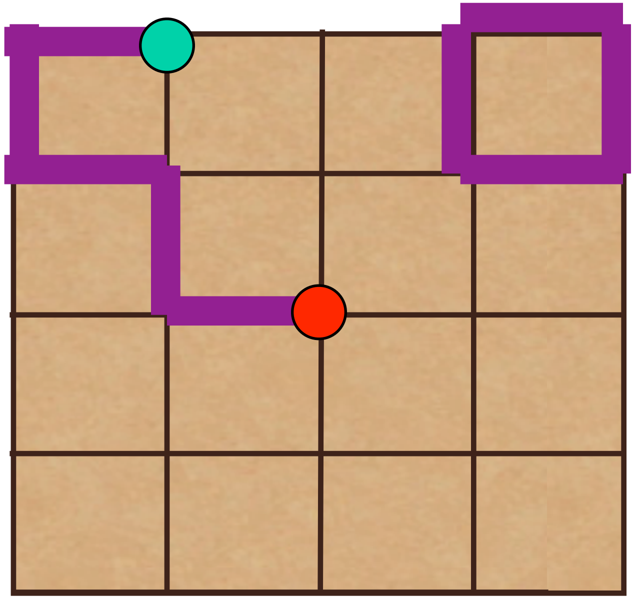
Even s's + <2 odd



s2(s2s1)(s1s6)(s6s7)(s7s12)(s12s13) s13

(s7s8)(s8s13)(s13s12)(s12s7) (s4s5)(s5s10)(s10s9)(s9s5)

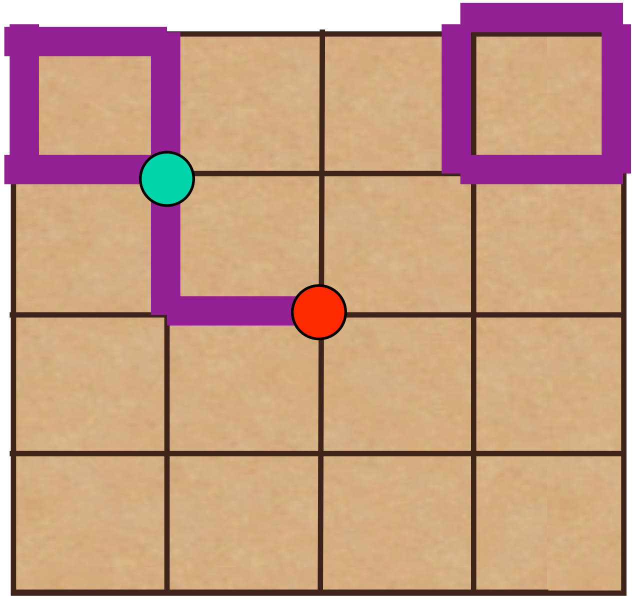
Doing Worm Monte Carlo



We now need an algorithm that does monte carlo over these configurations!

- Choose a direction for the head.
- If the head "adds" a bond:
  - Accept w/ Pr $\min\left(1, \tanh(K)\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head "removes" a bond:
  - Accept w/ Pr $\min\left(1, \frac{1}{\tanh(K)}\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head == tail:
  - mv hd and tl to new spot
  - increment Z
- increment M2

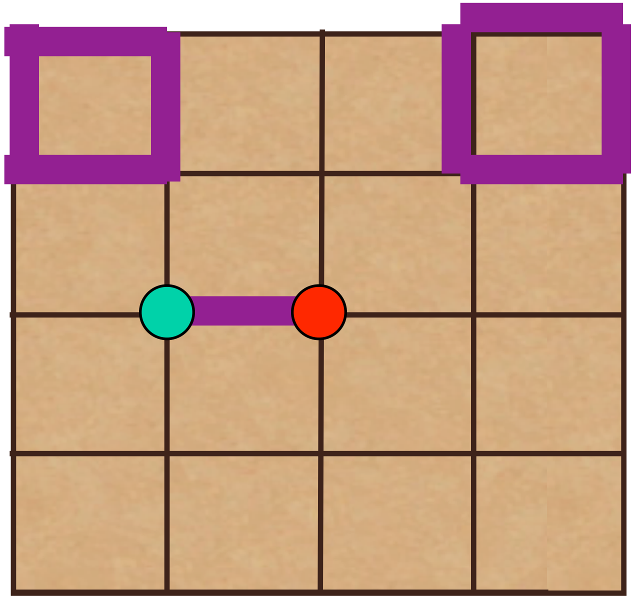
Doing Worm Monte Carlo



We now need an algorithm that does monte carlo over these configurations!

- Choose a direction for the head.
- If the head "adds" a bond:
  - Accept w/ Pr $\min\left(1, \tanh(K)\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head "removes" a bond:
  - Accept w/ Pr $\min\left(1, \frac{1}{\tanh(K)}\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head == tail:
  - mv hd and tl to new spot
  - increment Z
- increment M2

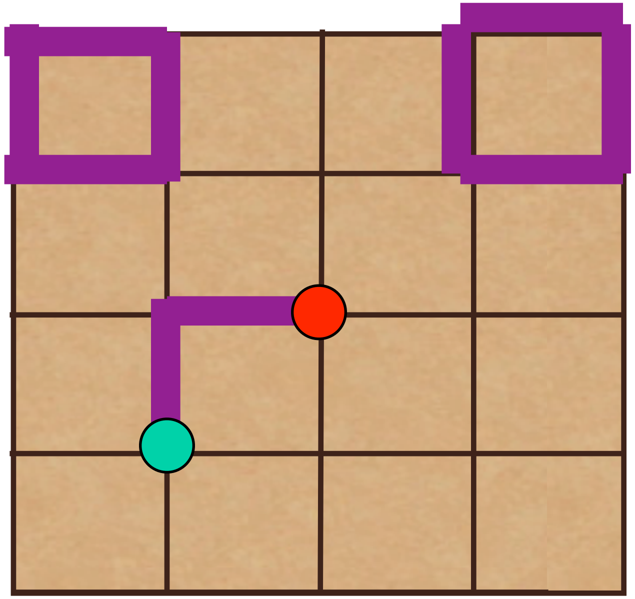
Doing Worm Monte Carlo



We now need an algorithm that does monte carlo over these configurations!

- Choose a direction for the head.
- If the head "adds" a bond:
  - Accept w/ Pr $\min\left(1, \tanh(K)\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head "removes" a bond:
  - Accept w/ Pr $\min\left(1, \frac{1}{\tanh(K)}\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head == tail:
  - mv hd and tl to new spot
  - increment Z
- increment M2

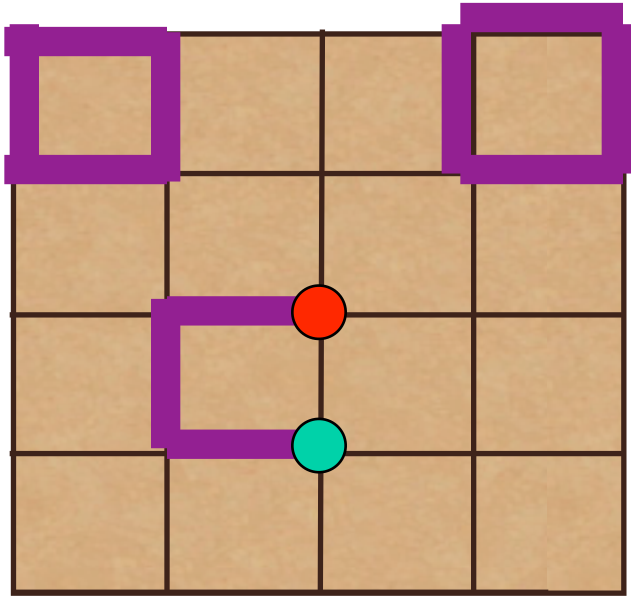
Doing Worm Monte Carlo



We now need an algorithm that does monte carlo over these configurations!

- Choose a direction for the head.
- If the head "adds" a bond:
  - Accept w/ Pr $\min\left(1, \tanh(K)\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head "removes" a bond:
  - Accept w/ Pr $\min\left(1, \frac{1}{\tanh(K)}\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head == tail:
  - mv hd and tl to new spot
  - increment Z
- increment M2

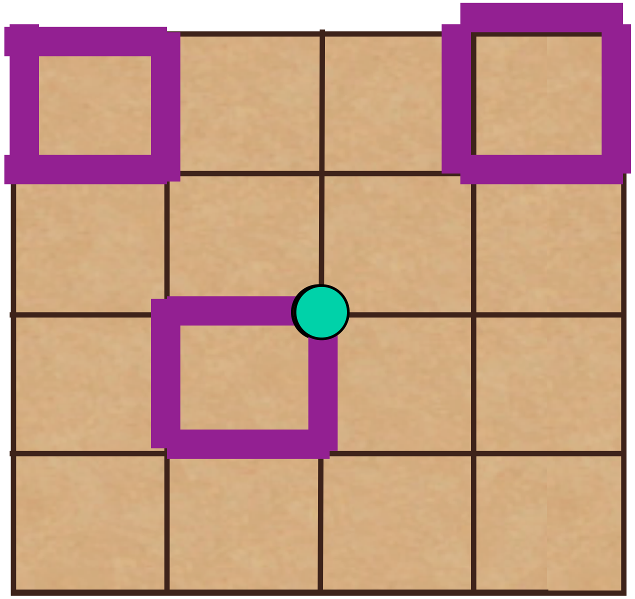
Doing Worm Monte Carlo



We now need an algorithm that does monte carlo over these configurations!

- Choose a direction for the head.
- If the head "adds" a bond:
  - Accept w/ Pr $\min\left(1, \tanh(K)\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head "removes" a bond:
  - Accept w/ Pr $\min\left(1, \frac{1}{\tanh(K)}\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head == tail:
  - mv hd and tl to new spot
  - increment Z
- increment M2

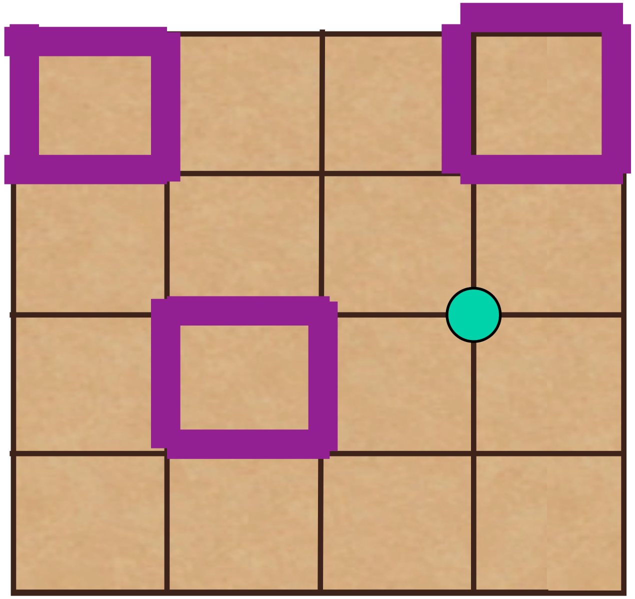
Doing Worm Monte Carlo



We now need an algorithm that does monte carlo over these configurations!

- Choose a direction for the head.
- If the head "adds" a bond:
  - Accept w/ Pr $\min\left(1, \tanh(K)\frac{T(n\to o)}{T(o\to n)}\right)$
- If the head "removes" a bond:
  - Accept w/ Pr $\min\left(1, \frac{1}{\tanh(K)}\frac{T(n\to o)}{T(o\to n)}\right)$
- If the head == tail:
  - mv hd and tl to new spot
  - increment Z
- increment M2

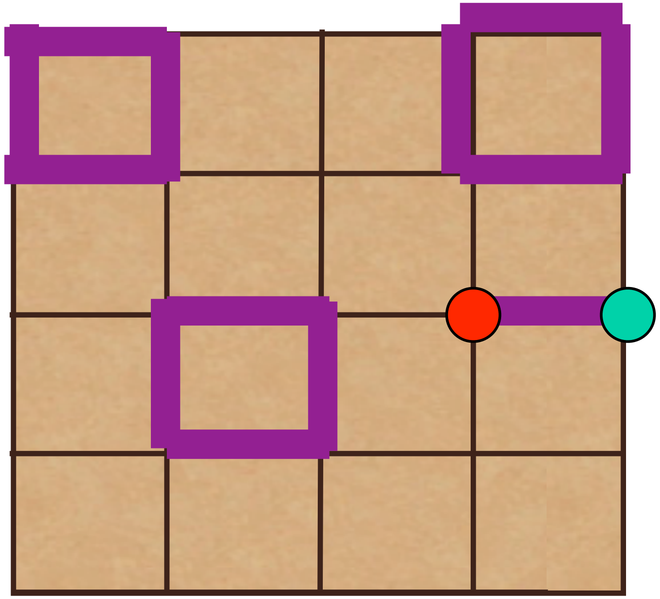
Doing Worm Monte Carlo



We now need an algorithm that does monte carlo over these configurations!

- Choose a direction for the head.
- If the head "adds" a bond:
  - Accept w/ Pr $\min\left(1, \tanh(K)\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head "removes" a bond:
  - Accept w/ Pr $\min\left(1, \frac{1}{\tanh(K)}\frac{T(n \to o)}{T(o \to n)}\right)$
- If the head == tail:
  - mv hd and tl to new spot
  - increment Z
- increment M2

Doing Worm Monte Carlo



We now need an algorithm that does monte carlo over these configurations!

- Choose a direction for the head.
- If the head "adds" a bond:
  - Accept w/ Pr $\min\left(1, \tanh(K)\frac{T(n\to o)}{T(o\to n)}\right)$
- If the head "removes" a bond:
  - Accept w/ Pr $\min\left(1, \frac{1}{\tanh(K)}\frac{T(n\to o)}{T(o\to n)}\right)$
- If the head == tail:
  - mv hd and tl to new spot
  - increment Z
- increment M2