Jakia Chowdhury

Worm Algorithm

Ising

→ $E = -J\sum S_i S_j$, $P_{accept} = \min[1, e^{-\Delta E/T}]$

→ Limitation: Critical slowing down near T_c

Worm has something like bond

→ Sampling efficiently using → measure observables like M, E

Closed loops: Z (partition func) $= 2^N \sum \tanh(K)^{\sum n_b}$ (n_b → 0,1; represents bond; loop length?)

Open loops: G (Corr func) $g(i-m) = \langle S_i S_m \rangle$

worm

no dangling end (each site much have even # of bonds)

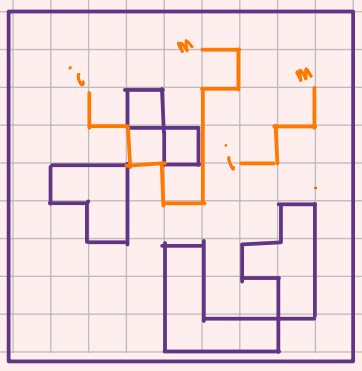
dangling ends i and m

Local moves:

- Add/remove bonds → extend/shrink worm
- Move ends → shift I/M to adjacent sites
- When I = M the worm vanishes

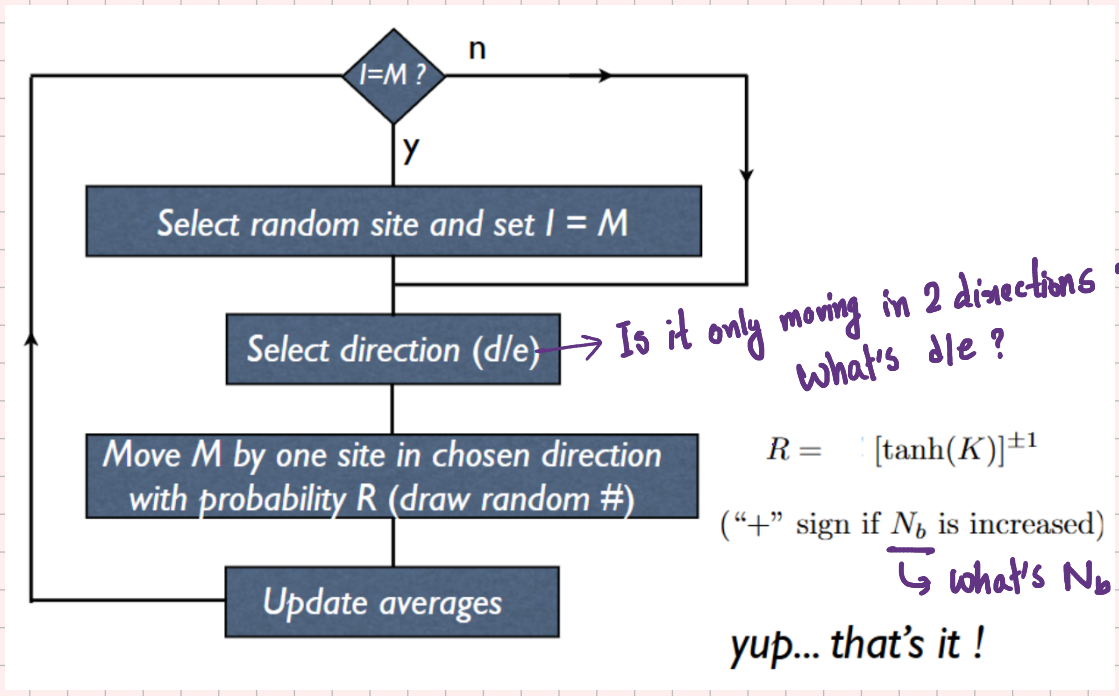
Why we care?

☆ No critical slow down → Samples loops instead of individual spins



— Closed loops
— Open loops

Algorithms:



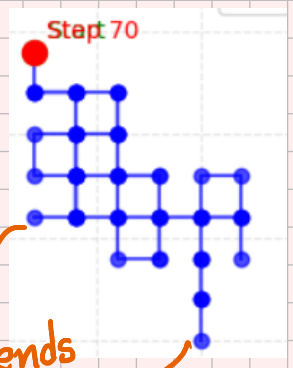
→ Is it only moving in 2 directions ? what's d/e ?

↳ what's N_b ?

* Non-reversible walk can prevent situations like the following. Are the following cases ok for closed loops ?

↳ My guess is no since double invoking bonds and dangling ends are not allowed.

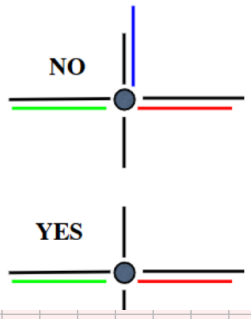
From Owen's code:



dangling ends

were invoked more than twice

Closed loops ? *What* loops ?



Number of times each site occurs in the product must be **even**

Sum of all occupied bonds involving that site must be **even**

(no bond can be invoked *twice*)

Consequently, bonds connecting sites *necessarily* must form closed loops (not necessarily connected)

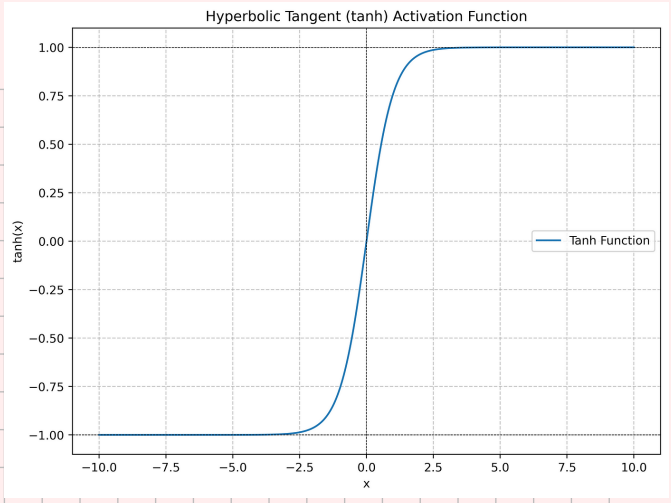
This def of closed loops is consistent w/ our def of loops in random walk section. YAY!

so the solution is just erasing the bond

* with every step the worm takes, if there existed a bond before, it gets erased w/ p=1. Otherwise a bond is created w/ prob R.

$$K = \frac{J}{T} = \frac{1}{T}$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \in (-1, 1)$$



Bond Probability:

Bond → n_b : $n_b = 1$ → Bond Present

$n_b = 0$ → Bond Absent

Bond weight $w = (\tanh k)^{n_b}$

Bond present : $W = (\tanh k)^{1} = \tanh k$

Bond Absent : $W = (\tanh k)^{0} = 1$

Bond creating : $\Delta n_b = +1$

removing : $\Delta n_b = -1$

Probability $P_{flip} = \min(1, (\tanh k)^{\Delta n_b})$

Bond creating prob, $P = (\tanh k)^{1} = \tanh k$

removing prob $p = \min(1, \underbrace{(\underbrace{\tanh k}_{<1})^{-1}}_{>1}) = 1$

So, if there is a bond it will always get removed

if there is no bond there may or may not be new bond created.

(★)

At $T < T_c$: $\tanh k \xleftarrow{\text{large}} \tanh 1/T \approx 1$: dense bond

At $T > T_c$: $\tanh k \approx k = 1/T$: sparse bond

(k: small)

At $T \approx T_c$: $\tanh\left(\frac{1}{2.269}\right) = \tanh(0.44)$: Balanced

Auto-correlation ($G(r)$)

$G(r) = \langle S_i \cdot S_j \rangle$

But we are not sampling all sites which is $O(N^2)$.

Instead we are using open and closed loops which is $O(1)$.

$$G(r) = \frac{1}{Z} \sum_{\text{confi.}} S_i S_j e^{-E/T} = \frac{\text{weight of open paths from } i \text{ to } j}{\text{weight of the closed loops}}$$

$$= \frac{G_n[r]}{G_0 \times \text{degeneracy}[r]}$$

# of distinct pairs that have dis r

e.g $r=1$: 4 ↔

$r=\sqrt{2}$: 4 ✕

$r=2$: 8

For $J>1$ (Ferromagnetic) $G(r) \in [0,1]$

How does this work?

Flipped to $n_b = 1 \Rightarrow S_i \cdot S_j = 1$ (spin aligned)

$n_b = 0 \Rightarrow S_i \cdot S_j = -1$ (spin misaligned)

(dictated by the prob fun)

From ⍟, at low T: more bonds → more connectivity

⍟ → the worm can reach there better.

$G(r)$ calculates how frequently worms can reach that far.

That's 2-pt corr at distance r.

So, at low T, $G(r) \to m^2$

at high T, $G(r) \to 0$

⍟ How does worm's movement depend on the bonds?

Why does low T ⟺ more bonds ⟺ reach far ⟺ ↑ $G(r)$

```python
def move_worm(x, y, bonds, beta, L, stats):
    """Perform one worm move with detailed balance"""
    while True:  # Keep trying until move is accepted
        dir_index = random.randint(0, 3)
        dx, dy = DIRECTIONS[dir_index]
        x_new, y_new = (x + dx) % L, (y + dy) % L
        opp_dir = get_opposite_dir(dir_index)

        # Determine if we're adding or removing a bond
        bond_change = 1 if bonds[x, y, dir_index] == 0 else -1


        # Bond acceptance
        if bond_change == 1:
            p_accept = math.tanh(beta)
            stats['attempts'] += 1
        else:
            p_accept = 1.0  # Always accept bond deletion

        if random.random() < p_accept:
            bonds[x, y, dir_index] ^= 1  # Flip current bond
            bonds[x_new, y_new, opp_dir] ^= 1  # Flip reciprocal bond
            if bond_change == 1:
                stats['acceptances'] += 1
            return x_new, y_new
```

New position is accepted only if bond is created or erased.

That means worms can move only by creating or erasing bonds.

$T \downarrow \Rightarrow p \uparrow$

→ create more bonds

→ existing bonds to erase.

At high T, there aren't that many options, so the worms close faster

$\Rightarrow$ worm lenght $\downarrow$ $\Rightarrow$ $G(r)$ $\downarrow$

Energy Calculation

From the slide, $\langle E \rangle = -J \tanh(K) \left[dN + \langle N_b \rangle / \sinh^2(K) \right]$ (†)

- $\tanh(K)$: occupasion prob
- dN: # total bonds, $4L^2/2$ (dim = 2, # total sites = L^2)
- $\langle N_b \rangle$: average # of occupied bonds
- $\sinh^2(K)$: (*)

(*) $1 + \tanh(K) = \tanh(K)(1 + \coth^2(K)) = \tanh(K) / \sinh^2(K)$

Manipulating the equation, $\langle E \rangle = -J \left[N_{occupied} - N_{empty} \tanh(\beta J) \right]$

$= -J \left[\langle N_b \rangle - (2N - \langle N_b \rangle) \tanh(K) \right]$

$= -J \left[\langle N_b \rangle (1 + \tanh(K)) - 2N \tanh(K) \right]$

$= -J \tanh(K) \left[\langle N_b \rangle \frac{1 + \tanh(K)}{\tanh(K)} - 2N \right]$

$= -J \tanh(K) \left[\langle N_b \rangle / \sinh^2(K) - dN \right]$ [same as (†)]

?