LATTICE MONTE CARLO: the Ising model

Ising Model [Two dimensions, square lattice]

- Spins $S_\alpha = \pm 1$ $\qquad \alpha = [i, j]$
- Bonds $J_{\alpha\beta}$ connect to four nearest neighbors

$$J_{\alpha\beta} = \begin{cases} 1 & |\alpha - \beta| = 1 \\ 0 & \text{otherwise} \end{cases}$$

- External field H favors spins pointing along H

Energy of S_α

$$\mathcal{H}(S_\alpha) = -\sum_{\alpha\beta} J_{\alpha\beta} S_\alpha S_\beta - \sum_\alpha H S_\alpha$$

"Hamiltonian" — Energy lower for spins parallel, parallel to H

Magnet: $\vec{S}$ magnetic spin, H external field, $J_{\alpha\beta}$ = ferromagnetic exchange interactions

Crystal: $\vec{S} = \begin{cases} +1 \text{ atom} \\ -1 \text{ vacancy} \end{cases}$, $J_{\alpha\beta}$ = bond strength, $H \propto$ chemical potential

Liquid/Gas Mixture: $S = \begin{cases} +1 \text{ atom} \\ -1 \text{ vacuum} \end{cases}$, $H \propto$ pressure

"Lattice Gas" → Excellent model for critical endpoint

- Equilibrium at temperature T:
  Want spin configurations to occur with a Boltzmann distribution

$$P(S_\alpha) \propto \exp(-\mathcal{H}(S_\alpha)/kT)$$

$$P(S_\alpha) = e^{-\mathcal{H}(S_\alpha)/kT} / Z \qquad Z = \sum_{\{S_\alpha\}} e^{-\mathcal{H}(S_\alpha)/kT}$$

- Dynamics [sometimes irrelevant]

  Want diffusion rates
  nucleation rates
  coarsening, island diffusion, ...

Algorithm #1: Heat-Bath Monte Carlo
Equilibrate One Spin at a Time

- Pick random spin $[i, j]$
- Find E for $\uparrow$, $\downarrow$ state

$$E_\pm = \left(\sum_{nn} S_\beta + H\right)(\pm 1) = \mp\left(\sum_{nn} S_\beta + H\right)$$

Neighbors up

$\sum_{nn} S_\beta$	Neighbors up
-4	0
-2	1
0	2
2	3
4	4

$$S_\beta = 2 \cdot (\text{Neighbors up} - 2)$$

- Set spin to +1 with probability $\dfrac{e^{-E_+/T}}{e^{-E_+/T} + e^{-E_-/T}} = \dfrac{1}{1 + e^{\Delta E/T}}$

otherwise to −1

Vary T, find approximate T_c!

HOW TO EQUILIBRATE

Monte-Carlo Move takes a current spin configuration A to another B

$\Gamma(A \to B)$ = rate given in A to go to B

Heat Bath: A and B differ by one flipped spin s_α

$\Gamma(A \to B) = e^{-\beta E_+}/(e^{-\beta E_+} + e^{-\beta E_-})$ if $s_\alpha^A = -1$, $s_\alpha^B = +1$

Three Rules for Equilibration • ($\frac{1}{N}$ chance of picking 1 / N choices / sweep)

(1) No Memory [Markovian]

Current rate from A → B depends only on A, B, not on how it got to A

- Not necessary, but violations can give weird & wrong results

(2) Can Get Everywhere [Ergodicity]

Enough moves to allow all states to be reached

(3) Detailed Balance

$$\frac{\Gamma(A \to B)}{\Gamma(B \to A)} = e^{-(E_B - E_A)/kT}$$

In equilibrium, net fluxes balance

$$\underbrace{P(A)}_{e^{-E_A/T}}\,\Gamma(A \to B) = \underbrace{P(B)}_{e^{-E_B/T}}\,\Gamma(B \to A)$$

Heat Bath MC?

(1), (2) obvious [usually so §]

Detailed Balance: $e^{-E_A/T}\Gamma(A\to B) = e^{-E_B/T}\Gamma(B\to A)$

if A, B differ by more than one spin $0 = 0$

" " same $e^{-E_A/T}\Gamma(A\to A) = \ldots$

" " differ by spin α, $s_\alpha^A = 1$

$E_B - E_A =$

$= 2\sum_{nn}(s_\beta + H) = E_+ - E_-$

All other bonds unchanged

$$\frac{\Gamma(A\to B)}{\Gamma(B\to A)} = \frac{e^{-E_+/kT}/(e^{-E_+/T}+e^{-E_-/T})}{e^{-E_-/T}/(e^{-E_+/T}+e^{-E_-/T})} = e^{-(E_+ - E_-)/T}$$

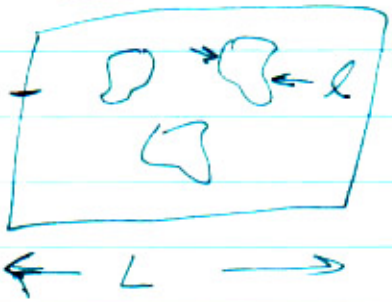
Metropolis: If $E_A > E_B$ $\Gamma(A\to B) = 1$

$\Gamma(B\to A) = e^{-(E_B - E_A)/T}$

slightly more efficient.

Cluster Moves: The Wolff Algorithm

- Random configuration (high T), quench to small T
  - Coarsening $\ell(t) \sim \sqrt{t}$ t = time



Equilibration Time

$t_{equil} \sim L^2$ sweeps

$\sim L^4$ computer time

- Explore critical point: slowly cool until near T_c [try width = height = 500]
  - Up, down spin regions clump, $\xi \sim (T - T_c)^{-\nu}$
  - Sluggish: long time for spins to correlate
  - $t_{equil} \sim |T - T_c|^{-z\nu} \sim \xi^z$

time for "information" to diffuse a length ξ

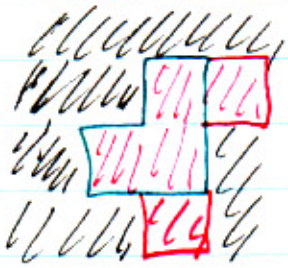
How to speed things up?

Consider flipping a cluster of spins as a unit:

Cluster: Before & After

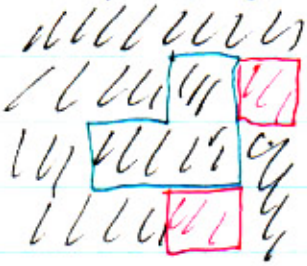
- Assume all spins start pointing the same direction, say up.

A



$n_\uparrow = 2$, $n_\downarrow = 6$

B



Let $n_\uparrow$ be the # of bonds to up-spins on the perimeter of the cluster; $n_\downarrow$ = # bonds to ↓ spins.

$$E_B - E_A = E_B^{\text{Perimeter}} - E_A^{\text{Perimeter}}$$

$$E_B^{\text{Perimeter}} = +n_\uparrow J - n_\downarrow J = +2J - 6J = -4J$$

$$E_A^{\text{Perimeter}} = -n_\uparrow J + n_\downarrow J = -2J + 6J = +4J$$

$$\boxed{E_B - E_A = 2(n_\uparrow - n_\downarrow) J.}$$

Wolff Algorithm

(oldSpin)

- Pick a random spin: remember it's direction
- Flip it
- For each neighboring spin, if it is in the same direction as the original, {with probability p} flip it, call (FlipNeighbors)

FlipNeighbors → Recursive

Wolff cluster moves are ergodic & Markov.

Detailed Balance?

$$\Gamma_{A\to B} = \sum_{\substack{\text{spins } \alpha \\ \text{in cluster}}} \left(\begin{array}{c}\text{Prob starting} \\ \text{from } \alpha \text{ makes it} \\ \text{internally}\end{array}\right) (1-p)^{n_\uparrow}$$

$n_\uparrow$ = # of times cluster chose not to grow

$$\Gamma_{B\to A} = \sum_{\alpha \in C} \left(P_\alpha\left(\begin{array}{c}\text{makes} \\ \text{internal}\end{array}\right)\right) (1-p)^{n_\downarrow}$$

Need $$\frac{\Gamma_{A\to B}}{\Gamma_{B\to A}} = \frac{(1-p)^{n_\uparrow}}{(1-p)^{n_\downarrow}} = e^{-(E_B - E_A)/T} = e^{-2(n_\uparrow - n_\downarrow)J/T}$$

$$(1-p)^{n_\uparrow - n_\downarrow} = \left(e^{-2J/T}\right)^{(n_\uparrow - n_\downarrow)}$$

$$\boxed{p = 1 - e^{-2/T}}$$

2/28/99 (MC 8)
Mat Sim

- New, generic class Wolff Dynamics, derived from Spin Dynamics strategy

- Look at SpinDynamics.h
  - Wolff inherits all member functions
  - Wolff can access all protected variables (not private)
  - Wolff can redefine any virtual member function

- Change constructor for Wolff.h
  - Takes argument SpinLattice *S (as does SpinDynamics)
  - Call SpinDynamics(S) constructor first

WolfDynamics::WolffDynamics(SpinLattice *s) : SpinDynamics(s)

- Add protected member variables

double p [storing $1-e^{-2/T}$]
int oldSpin

- Redefine SetTemperature to also set p
  if T=0 or 2/T > 300, set p=1
  to avoid divide by 0, exp overflow

- Redefine Sweep
  - store oldSpin
  - call FlipNeighbors on random, flipped spin
  - return deltaTime = spinsFlipped / width·height

2/28/99 (LMC 8)
Mat Sim

- Add protected member function
  FlipNeighbors (i, j)
  - Recursion on four neighbors
  - Return total # of spins flipped

- Change IsingSimulation to create WolffDynamics
  - include WolffDynamics.h.

- Stack overflow?
  Project ▷ Settings ▷ Link ▷ (Category Menu) Output
  ▷ Reserve 10,000,000 bytes

Accellerating Monte Carlo with Realistic Dynamics

* Continuous Time — Bortz, Kalos, Lebowitz '74 Mal
  Kinetic MC [Measure energy barriers to motion] Voter

Set $T \leq 1$, Run Heat Bath:
- Coarsening — up & down regions slowly separate

Run Wolff: huge cluster flip → equilibrium
Coarsening bypassed

Glassy behavior, Island Decay, Binary Alloys, spinodal decomposition,

Suppose we want to study coarsening? Faster?

Electromigration & voids, Nucleation & growth, Diffusion,

At low temperatures, only the few spins near boundaries between ↑ and ↓ have any hope of flipping. Tons of wasted effort picking happy spins and asking them to flip. Keep lists of active sites?

BKL: flip a spin every time!
First figure out relative probabilities for all the spins to flip. Then choose the pair with the right probabilities!

Net rate for spins to flip

$$Q = \sum_{\text{environments } m} n_m P_m$$

P_m: Probability/time for spin flip

Heat Bath: $\frac{1}{1+e^{\Delta E/T}}$

n_m: # of spins in environment m

$n_m P_m / Q$ = Prob. next spin to flip has environment j.

- Pick a random R; $\Delta t = -\ln(R)/Q$; if $t_{elapsed} + \Delta t > t_{max}$ return t_{max}
- Pick a random number R between 0 and Q
- Find $Q_k = \sum_{m=1}^{k} n_m P_m$ with $Q_{k-1} \le R < Q_k$
- Pick a random integer l between 0 and $n_k - 1$, inclusive.
- Update all the lists
- Flip spin # l ($0 \le l \le n_k - 1$) among list of spins with environment k.
- $\Delta t = -\ln R / Q$ ** Note: weight states by Δt in equilibrium

(A) Not easy: lots of lists, bookkeeping.
C: 3 days → 3 months for debugging

(B) Beautiful example of vector class (C++ supplied)
Handles allocation, lengths, ...

(C) Matt & I: 1:30 - 7:00 for debugging
We'll create lists, provide routine for updating

683 LMC
2/5/98

(3)
LMC 17

Code in [LMC Assign 2]

Continuous Time, derived from Spin Dynamics

probFlip[s][nUp] $\leftrightarrow$ $P_j = \frac{1}{1+e^{+\Delta E/T}}$, set up

LOC[s][nUp] = vector of (i,j) Points with spin $S = 2s-1$ and nUp up-spin neighbors

LOC[s][nUp].size() = n_{snn} [Convenient!]

LOOK[i][j] = Position of S_{ij} on vector LOC[s_{ij}][nUp_{ij}]

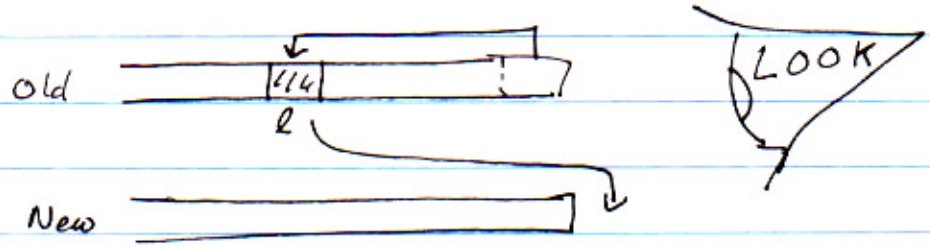
Overload Sweep():

- Compute total rate Q
- Pick random R, Find environment s, nUp
- Pick l, store old Spin
  Point toFlip = LOC[s][nUp][l]
- Loop over 4 neighbors: $dNUp_{nbr}$ = new S_{ij} − old S_{ij} = dS_{ij}

Point Neighbor; Neighbor.i = toFlip.i + l; Neighbor.j = ...
  Fix LOOK LOK[neighbor, dn, ds=0]
- Fix LOOK LOK[toFlip, dn=0, ds]
- Flip spin S_{ij}
- Return Δt should return here! when $\Delta t = \Delta t_{tot}$!

What does FixLookLOC do?

- Finds sOld, sNew, nUpOld, nUpNew
- Appends given point l to LOC[sNew][nUpNew]. using LOC[I][].push_back( )
- Moves end of LOC[sOld][nUpOld] to where "l" was
  Changes LOOK for end-point
- LOC[ ][ ].pop-back() to shorten list
  Changes LOOK for point "l"



setupLOOKLOC

What does [illegible] do to make LOC & LOOK at initialization?

- Runs over [i][j]
- Pushes each site onto right LOC[][].push_back
- Sets LOOK.

Curie Law

At high temperatures, independent spins

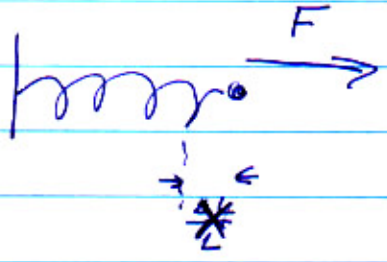
$$M(H,T) \approx N_{spins}\left(\frac{(+1)e^{H/T}+(-1)e^{-H/T}}{e^{H/T}+e^{-H/T}}\right)$$

$$= N_{spins}\left(\frac{e^{H/T}-e^{-H/T}}{e^{H/T}+e^{-H/T}}\right)$$

$$= N_{spins}\left(\frac{2H/T}{2}\right) = H\left(\frac{N_{spins}}{T}\right)$$

$$\chi = \frac{\partial M}{\partial H} = \frac{N_{spins}}{T}$$

Susceptibility and Fluctuations:



$F = k(x-x_0)$ $U(x) = \frac{1}{2}k(x-x_0)^2$

Equipartition: $\langle U\rangle = \frac{1}{2}k_B T$ (1 DOF)

$\frac{1}{2}k\langle(x-x_0)^2\rangle = \frac{1}{2}T$

$1/k = \frac{\langle(x-x_0)^2\rangle}{T}$

Magnet: $M - \bar{M}(T) = \chi(T)\,H$

$1/k$ $\quad$ F

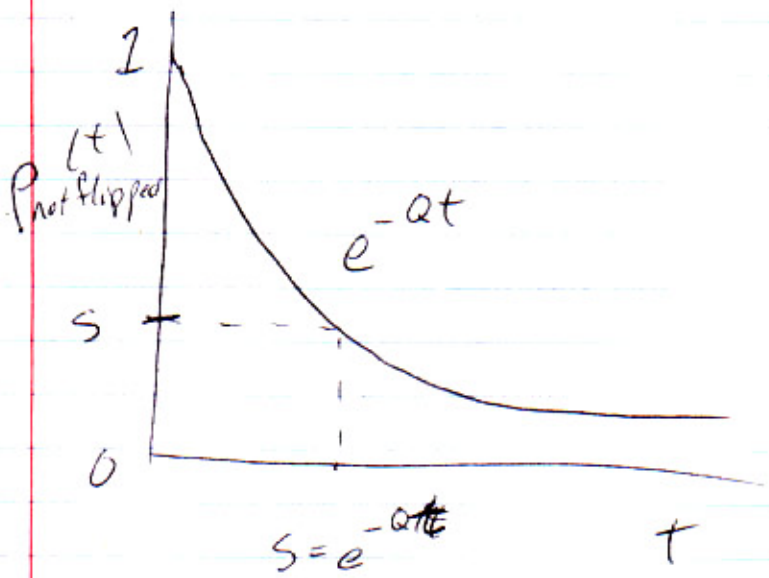
$$\chi(T) = \frac{\langle(M-\bar{M})^2\rangle}{T}$$

Q = rate of spins flipping (sec^{-1})

$p_{\text{not flipped}}(t)$ = Prob no spins have flipped by time t

$$\frac{dp}{dt} = -Qp \quad \text{(obvious?)}$$

$$p(t) = e^{-Qt}$$

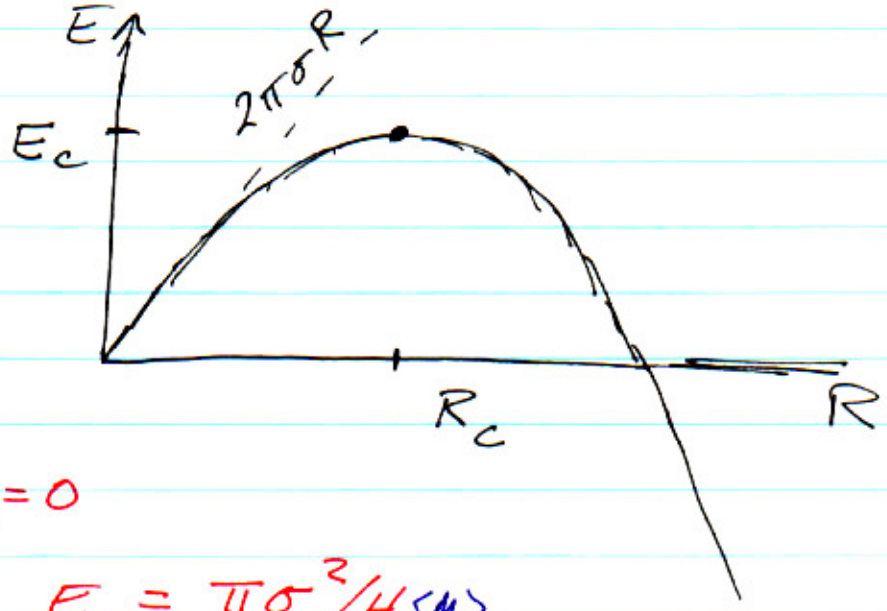


$$t = -(\log s)/Q$$

$$s \in (0, 1]$$

[Pretend critical droplet is a circle]

Critical Nucleus



Energy =

$2\pi\sigma R - H\pi R^2\langle M\rangle$

$2\pi\sigma - 2H\pi R_c = 0$

$R_c = \sigma / H\langle M\rangle \qquad E_c = \pi\sigma^2 / H\langle M\rangle$

Time to nucleate critical droplet

$$\simeq \underbrace{(\text{Prefactor})}_{\text{Many wiggly drops near top}} \underbrace{e^{-E_c/T}}_{\text{Probability of reaching top of barrier}} / \underbrace{L^2}_{\#\text{ of nucleation sites in system}}$$

- In CMDDoc::OnParamChange, call sim->SetTime(0) to restart clock
- Plot time vs. H for T=1; does it grow like the prediction?

Forward Transform

$$\tilde{S}_\ell = \sum_{j=0}^{W-1} S_j \, e^{\frac{-2\pi i \, k \cdot \ell}{W}}$$

Power Spectrum (X-ray intensity ...)

$$\tilde{C}_\ell = |\tilde{S}_\ell|^2 = \tilde{S}_\ell^* \tilde{S}_\ell$$

$$= \sum_{j=0}^{W-1} \sum_{j'=0}^{W-1} S_j \, e^{+\frac{2\pi i \, j \cdot \ell}{W}} \, S_{j'} \, e^{-\frac{2\pi i \, j' \cdot \ell}{W}}$$

$$= \sum_{j=0}^{W-1} \sum_{j'=0}^{W-1} S_j S_{j'} \, e^{-\frac{2\pi i}{W} \ell (j'-j)}$$

Inverse Transform

$$C_r = \frac{1}{W} \sum_{\ell=0}^{W-1} C_\ell \, e^{\frac{2\pi i}{W} \ell \cdot r}$$

$$= \frac{1}{W} \sum_j \sum_{j'} S_j S_{j'} \underbrace{\sum_\ell e^{-\frac{2\pi i}{W} \ell (j'-j+r)}}$$

$= W$ if $j'-j-r = 0$ [or W]
$= 0$ otherwise

$$= \sum_j S_j S_{j+r}$$

$$= W \cdot \langle S_j S_{j+r} \rangle$$