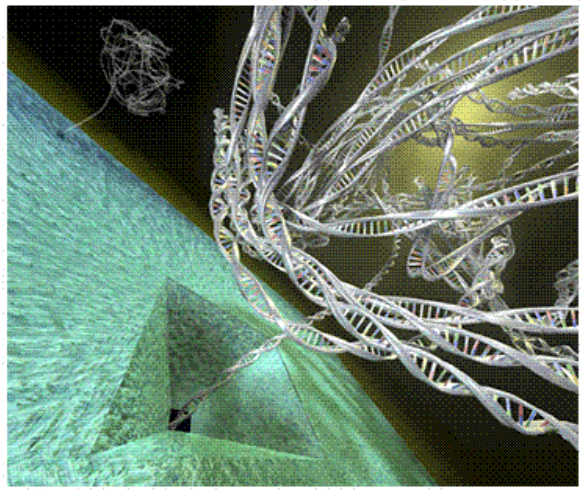
Basics of Monte Carlo simulations

Prof. Gerard T. Barkema
ITP, Utrecht + Instituut-Lorentz, Leiden



- basic MC, explained using Ising model
- Intermezzo I: anomalous dynamics
- advanced MC: cluster algorithms
- Intermezzo II: self-avoiding walks
- Current research: beyond detailed balance

Leuven, June 21+22, 2013

Ising model

- lattice with sites $\mathbf{i = 0 \ldots N - 1}$ (often square or cubic)
- spins $\sigma_\mathbf{i}$ on lattice sites $\mathbf{i = 0 \ldots N - 1}$
- Hamiltonian: $\mathbf{H = -J \, \Sigma_{\langle i,j \rangle} \, \sigma_i \sigma_j}$
  Here, $\langle \rangle$ is notation for nearest-neighbor sites
- Probability of a configuration $\mathbf{S \equiv \{ \, \sigma_i \, \}}$:
  $\mathbf{P(S)} = \frac{\mathbf{1}}{\mathbf{Z}} \exp\left(-\beta \mathbf{E(S)}\right)$
  in which $\beta \equiv (\mathbf{k_b T})^{-1}$ and $\mathbf{E(S)}$ the energy (result of the hamiltonian)
  Large β (low temperature): spins are mostly aligned
  Small β (high temperature): spins are more or less random

Ising model

Goal of Monte Carlo simulations: generate a representative set of states

General approach: Markov process

- **start from some initial state S_0**
- **in state S_i, propose some small change, resulting in S_{i+1}^T**
- **either *accept*, i.e., $S_{i+1} = S_{i+1}^T$, or *reject*, i.e., $S_{i+1} = S_i$, the proposed change.**
- **iterate the last two steps many times**

What is a "small change"?

For Ising model, for instance random spin flip

What should the probability of acceptance be?

long answer...

Detailed Balance

Convergence to some steady state, in which $\mathbf{S_i}$ has a probability $\mathbf{p(S_i)}$

Definitions:

$\mathbf{T}(\mu \to \nu)$: proposition probability from μ to ν

$\mathbf{A}(\mu \to \nu)$: acceptance probability from μ to ν

in the steady state:

$$\sum_{\mu} \mathbf{p}(\mu)\mathbf{T}(\mu \to \nu)\mathbf{A}(\mu \to \nu) = \sum_{\mu} \mathbf{p}(\nu)\mathbf{T}(\nu \to \mu)\mathbf{A}(\nu \to \mu)$$

$$\Leftarrow \forall \mu, \nu :: \mathbf{p}(\mu)\mathbf{T}(\mu \to \nu)\mathbf{A}(\mu \to \nu) = \mathbf{p}(\nu)\mathbf{T}(\nu \to \mu)\mathbf{A}(\nu \to \mu)$$

This is known as: Detailed Balance

Detailed Balance

In our simple example, in a system with N sites:
$\mathbf{T}(\mu \to \nu) = \mathbf{1/N}$ if μ and ν differ in a single spin only. (Else it is zero.)

We want to sample the Boltzmann distribution with

$$\frac{\mathbf{p}(\mu)}{\mathbf{p}(\nu)} = \exp(-\beta(\mathbf{E}_\mu - \mathbf{E}_\nu))$$

Thus, detailed balance demands

$$\frac{\mathbf{A}(\mu \to \nu)}{\mathbf{A}(\nu \to \mu)} = \exp(-\beta(\mathbf{E}_\nu - \mathbf{E}_\mu))$$

Restriction: acceptance probabilities are limited to $[\mathbf{0}, \mathbf{1}]$

Fastest simulations if either $\mathbf{A}(\mu \to \nu)$ or $\mathbf{A}(\nu \to \mu)$ is 1; thus

$$\mathbf{A}(\mu \to \nu) = \mathrm{Min}\,[\mathbf{1}, \exp(-\beta(\mathbf{E}_\mu - \mathbf{E}_\nu))]$$

This is the Metropolis algorithm

Ergodicity

Detailed balance is not enough for correct sampling
[Simple counter-example: take $A(\mu \to \nu) = 0$ for all μ, ν]

Second demand is ergodicity:
With a finite number of steps, any configuration can be reached

In our case:
From any μ to any ν is done by sequentially flipping all different spins

Detailed Balance + Ergodicity $\equiv$ Correct algorithm

i.e., eventually the Boltzmann distribution is sampled

Metropolis simulation of the Ising model

A simulation program thus looks like:

- **initialize the spins (for instance all** up**)**

Then do many iterations:

- **select random site**
- **compute energy difference** $\mathbf{\Delta E}$ **if flipped**
- **if** $\mathbf{\Delta E} < 0$ **or if random number** $< \exp(-\beta \mathbf{\Delta E})$**: flip it**
- **occasionally, measure some observable(s), e.g.** $\mathbf{E}$ **or** $\mathbf{M} = \sum \sigma_{\mathrm{i}}$

E **and** M **keep fluctuating, but their averages converge slowly**

Practical matter 1: Periodic boundary conditions

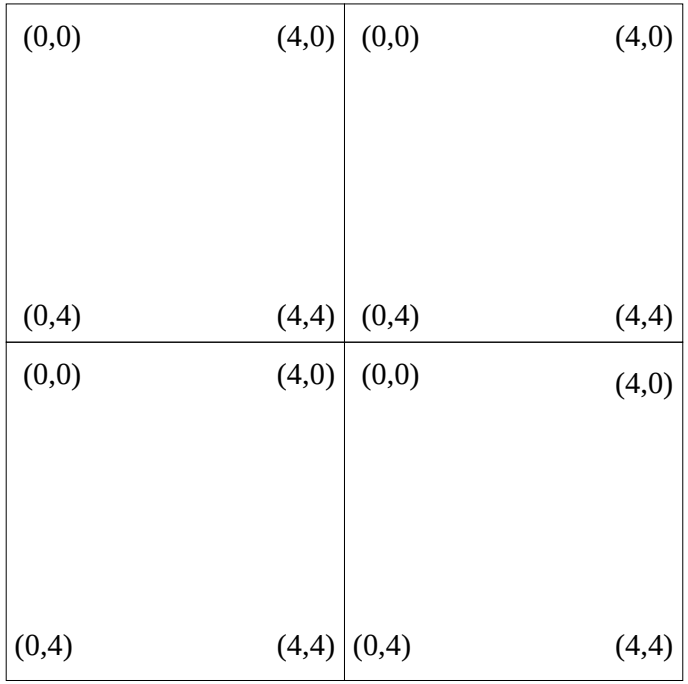
Interest in the thermodynamic limit (i.e., an infinite system)
Outside reach of simulation.
Natural choice: finite system with $L \times L$ sites (with open boundaries).

Better in practice are
periodic boundary conditions:

the neighbors of site (x, y) are
$(x, (y \pm 1) \bmod L)$ and $((x \pm 1) \bmod L, y)$.

In two dimensions, this yields a torus

- No special sites
- finiteness only over distances $\geq L$.

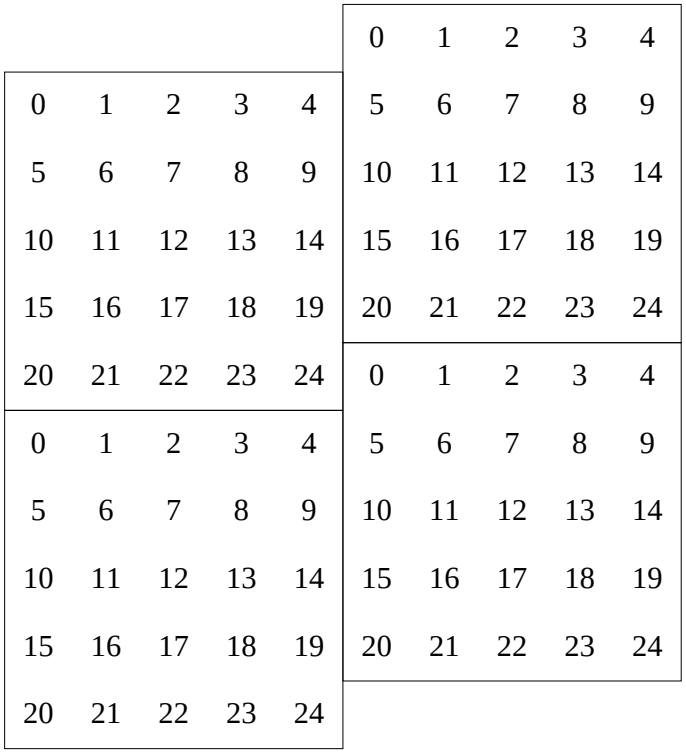


Helical boundary conditions

A slight variation is helical boundary conditions

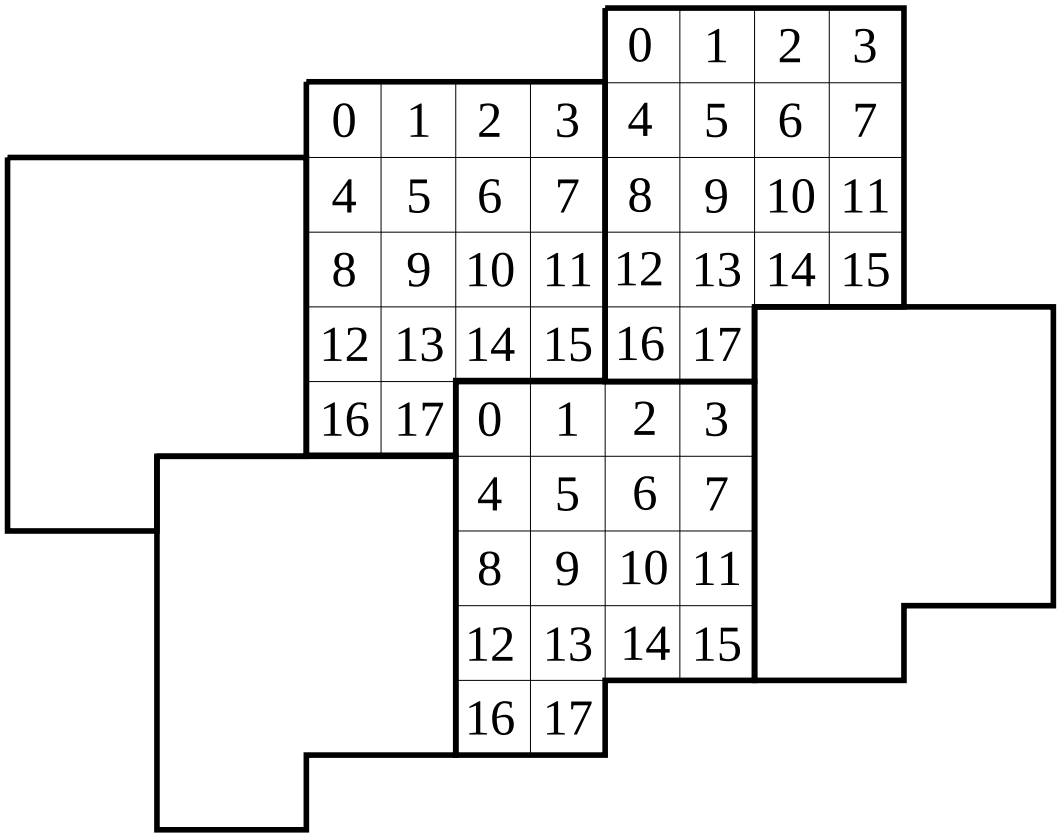
- **enumerate the sites $i = 0 \ldots N$**
- **neighors of site i are $i \pm 1$ and $i \pm L$.**

advantage: working with a single index is often faster



Helical boundary conditions

Number of sites does not need to be a multiple of L



Sometimes, it is computationally advantageous to take $N = 2^k$

energy differences

- **Consider a single proposed flip of σ_k**

- **Acceptance probability depends on resulting change in energy ΔE**

  This is only determined by itself and its neighbor spins:

$$\Delta E = 2J\sigma_k(\sigma_{k+1} + \sigma_{k-1} + \sigma_{k+L} + \sigma_{k-L})$$

  (on a square lattice, leaving out boundary conditions)

$\Rightarrow$ no need to compute the energy before and after!

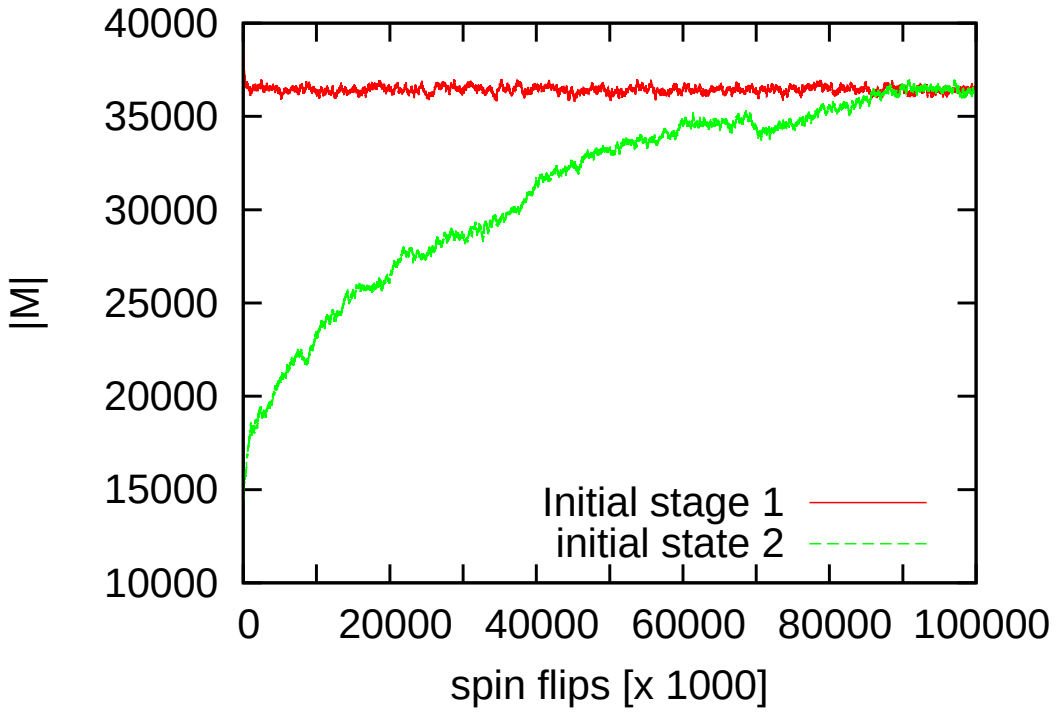
Thermalization

Another key question:

How many spin flips before steady state?

Answer: sometimes surprisingly long

- **Consider some observable, for instance the energy E or magnetization $M \equiv \Sigma_i \sigma_i$**
- **Plot its value vs. time**
- **Once it fluctuates around a fixed value, apparently the system has thermalized**



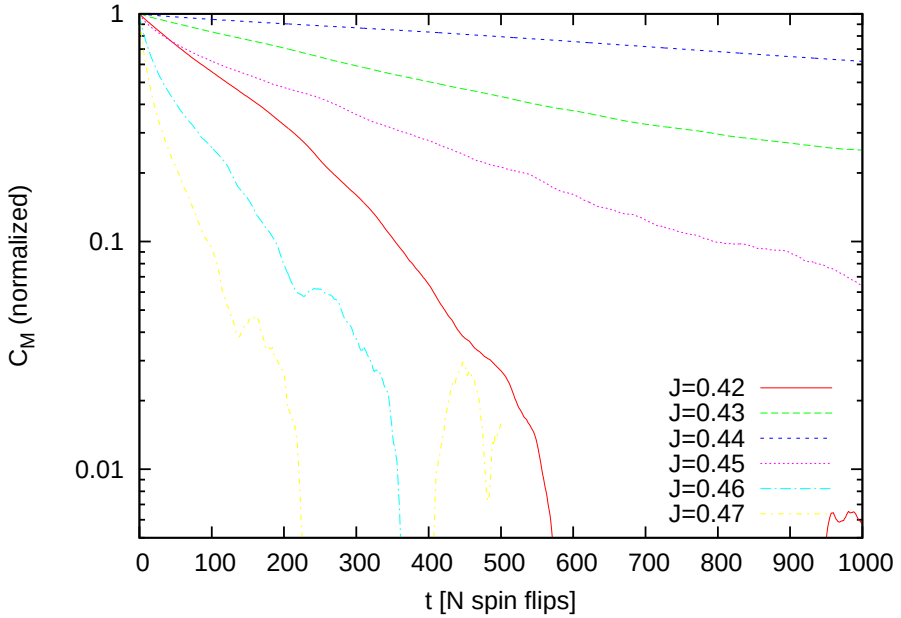
Thermalization

More insight can be obtained from the autocorrelation function:

$$\mathbf{C_M(\Delta t) = \langle (M(t) - \langle M \rangle) \cdot (M(t + \Delta t) - \langle M \rangle) \rangle}$$

(measured in steady state; in practice second half of a long simulation).

- **Often shows exponential decay: $C(\Delta t) \sim \exp(-t/\tau)$, defining the correlation time τ.**
- **Up to τ, surely the simulation is still correlated to the (arbitrarily) initial configuration.**
- **To be safe, thermalization should be at least 2 or 5 times τ.**
- **Knowledge of τ is also useful for statistical errors.**



Thermalization

A crude estimate of τ near the critical temperature:
Magnetic susceptibility

$$\chi = \frac{\beta}{\mathbf{N}} \left(\langle \mathbf{M^2} \rangle - \langle \mathbf{M} \rangle^{\mathbf{2}} \right) \sim \mathbf{L}^{\gamma/\nu}$$

with (in two dimensions) $\gamma = 7/4$ and $\nu = 1$.

- **a single spin changes M by ± 1**
- **if all spin flips are uncorrelated, M shows random-walk behavior with unit steps per accepted spin flip.**
- **$\Delta \mathbf{M^2}$ needs to cover a range of $\mathbf{N} \cdot \mathbf{L}^{\gamma/\nu} \sim \mathbf{L}^{15/4}$**
  $\Rightarrow$ an $\mathbf{L} \times \mathbf{L}$ system has a correlation time of $\sim \mathbf{L}^4$ spin flips

This means $\sim$ 100 million spin flips in a system with 100×100 sites!

Cluster algorithms

Can severe critical slowing down ($\tau \sim L^2$ flips per site or worse) be avoided?

For a long time, the believe was not.
Conclusion: only progress by faster (special-purpose) computers

- **DICE I and II (=Delft Ising Computing Engine)**
- **mTIS (megaspin model of the Tokyo University Ising spin)**
- **other ones in Bell labs, Santa Barbara, and in the Ukraine**

Revolution in 1987!

Swendsen-Wang algorithm.

Two years later, an improvement was made by Ulli Wolff (1989).

A new generation of non-local algorithms, a.k.a. cluster algorithms.

Wolff algorithm

Wolff algorithm:

1. **Randomly select a site k, which will act as the seed for a cluster.**
2. **For each newly added site, list all links to neighboring sites i, obeying:**
   - **i is not yet part of the cluster**
   - $\sigma_{\mathrm{i}} = \sigma_{\mathrm{k}}$

   and each such link has a probability $\mathrm{P}_{\mathrm{add}} = 1 - \exp(-2\beta \mathrm{J})$ to be activated (i.e., site i is added to the cluster (after which also other links need to be considered).
3. **Flip all spins of sites in the cluster.**

Correctness, Wolff algorithm

Correctness derived from Ergodicity + Detailed Balance

Detailed Balance:

$$\forall \mu, \nu :: \mathbf{p}(\mu)\mathbf{T}(\mu \to \nu)\mathbf{A}(\mu \to \nu) = \mathbf{p}(\nu)\mathbf{T}(\nu \to \mu)\mathbf{A}(\nu \to \mu)$$

All proposed cluster flips are accepted, $\Rightarrow \mathbf{A}(\mu \to \nu) = \mathbf{A}(\nu \to \mu) = \mathbf{1}$

Energy difference between μ **and** ν **only due to neighboring spins** $i \in C$ **and** $j \notin C$

$$\mathbf{E}_\mu - \mathbf{E}_\nu = -\mathbf{2J} \sum_{\langle \mathbf{i} \in \mathbf{C}, \mathbf{j} \notin \mathbf{C} \rangle} \sigma_{\mathbf{i}}^{(\nu)} \sigma_{\mathbf{j}}^{(\nu)}$$

some rewriting yields

$$\frac{\mathbf{p}_\mu}{\mathbf{p}_\nu} = \exp\left(-\beta(\mathbf{E}_\mu - \mathbf{E}_\nu)\right) = \prod_{\langle \mathbf{i} \in \mathbf{C}, \mathbf{j} \notin \mathbf{C} \rangle} \exp\left(\mathbf{2}\beta\mathbf{J}\sigma_{\mathbf{i}}^{(\nu)}\sigma_{\mathbf{j}}^{(\nu)}\right)$$

Correctness, Wolff algorithm

Proposition probability $\mathbf{T}(\mu \to \nu)$:

- **probability to select site k: $\frac{\mathbf{1}}{\mathbf{N}}$**
- **Each activated link adds a factor $\mathbf{P}_{\text{add}}$**
- **Each unactivated link between aligned spins adds a factor $\mathbf{1} - \mathbf{P}_{\text{add}} = \exp(-\mathbf{2}\beta\mathbf{J})$)**
- **(Each unactivated links between anti-aligned spins adds a factor 1)**

$$\Rightarrow \mathbf{T}(\mu \to \nu) = \frac{\mathbf{1}}{\mathbf{N}} \cdot \prod_{\langle i \in C, j \in C \rangle} (\mathbf{P}_{\text{add}}) \cdot \prod_{\substack{\langle i \in C, j \notin C \rangle \\ \sigma_i = \sigma_j}} (\mathbf{1} - \mathbf{P}_{\text{add}})$$

$\mathbf{T}(\nu \to \mu)$ differs in the third factor since for $\langle \mathbf{i} \in \mathbf{C}, \mathbf{j} \notin \mathbf{C} \rangle$, $\sigma_{\mathbf{i}}^{(\mu)} \sigma_{\mathbf{j}}^{(\mu)} = -\sigma_{\mathbf{i}}^{(\nu)} \sigma_{\mathbf{j}}^{(\nu)}$
Consequence:

$$\frac{\mathbf{T}(\mu \to \nu)}{\mathbf{T}(\nu \to \mu)} = \prod_{\langle \mathbf{i} \in \mathbf{C}, \mathbf{j} \notin \mathbf{C} \rangle} \exp\left(-\mathbf{2}\beta\mathbf{J}\sigma_{\mathbf{i}}^{(\nu)}\sigma_{\mathbf{j}}^{(\nu)}\right)$$

[Given $\sigma_{\mathbf{i}} = \sigma_{\mathbf{j}}$, we used $\mathbf{1} - \mathbf{P}_{\text{add}} = \exp\left(-\mathbf{2}\beta\mathbf{J}\sigma_{\mathbf{i}}\sigma_{\mathbf{j}}\right)$]

Correctness, Wolff algorithm

Putting pieces together:

$$\mathbf{A}(\mu \to \nu) = \mathbf{A}(\nu \to \mu) = \mathbf{1}$$

$$\frac{\mathbf{p}_\mu}{\mathbf{p}_\nu} = \exp\left(-\beta(\mathbf{E}_\mu - \mathbf{E}_\nu)\right) = \prod_{\langle \mathbf{i}\in\mathbf{C},\mathbf{j}\notin\mathbf{C}\rangle} \exp\left(\mathbf{2}\beta\mathbf{J}\sigma_\mathbf{i}^{(\nu)}\sigma_\mathbf{j}^{(\nu)}\right)$$

$$\frac{\mathbf{T}(\mu \to \nu)}{\mathbf{T}(\nu \to \mu)} = \prod_{\langle \mathbf{i}\in\mathbf{C},\mathbf{j}\notin\mathbf{C}\rangle} \exp\left(-\mathbf{2}\beta\mathbf{J}\sigma_\mathbf{i}^{(\nu)}\sigma_\mathbf{j}^{(\nu)}\right)$$

yields Detailed Balance:

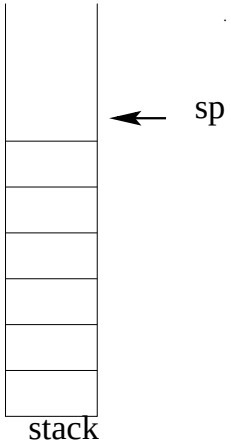
$$\forall \mu, \nu :: \mathbf{p}(\mu)\mathbf{T}(\mu \to \nu)\mathbf{A}(\mu \to \nu) = \mathbf{p}(\nu)\mathbf{T}(\nu \to \mu)\mathbf{A}(\nu \to \mu)$$

Ergodicity: individual spin flips are possible, and can result in any configuration

Combined, this provides a proof of correctness!

Implementation, Wolff algorithm

Useful data concept: 'stack'



Two basic operations:

- **push (on stack)**
- **pop (from stack)**

Implementation in C:

stack[sp++]=a;	↔	**push a**
a=stack[−−sp];	↔	**pop a**
if (sp==0)	↔	**test if stack is empty**
sp=0;	↔	**make stack empty**

Implementation, Wolff algorithm

```
sp=0;
seed=random * N;
oldspin=spin[seed];
newspin= -oldspin;
stack[sp++]=seed;
spin[seed]=newspin;
while (sp) {
  k=stack[--sp];
  for (i=0;i<4;i++) {
    if (spin[nb[k][i]]==oldspin) {
      if (random < P_add) {
        stack[sp++]=nb[k][i];
        spin[nb[k][i]]=newspin;
      }
    }
  }
}
```

make stack empty

push cluster seed

while (stack not empty), pop site and deal with neighbors

Alternatively, one can write a recursive program

Swendsen-Wang algorithm

Pre-dates Wolff algorithm by ∼2 years

Difference with Wolff algorithm:

- **cover entire lattice with clusters**
- **assign a random spin value to each cluster**

The SW algorithm alternates between a spin representation and a cluster representation

In the cluster (a.k.a. Fortuin-Kasteleyn) representation, spin-spin correlations are easily determined

Computer power is spent on, on average, smaller clusters

This leads to slightly more critical slowing-down

Analysis, critical exponents

Scaling relations describe the system's behavior near its critical point

Definition of reduced temperature: $\mathbf{t \equiv \frac{T - T_c}{T_c}}$

1. **correlation length** $\xi \sim |\mathbf{t}|^{-\nu}$
2. **magnetic susceptibility** $\chi \sim |\mathbf{t}|^{-\gamma}$
3. **specific heat** $\mathbf{c} \sim |\mathbf{t}|^{-\alpha}$
4. **magnetization** $\mathbf{m} \sim (-\mathbf{t})^{\beta}$

These relations can be combined, for instance (1) + (2): $\chi \sim \xi^{\gamma/\nu}$

On a finite lattice with size L, ξ **cannot exceed** $\mathbf{L} \Rightarrow \chi = \xi^{\gamma/\nu} \cdot \chi_0(\mathbf{L}/\xi)$

$\chi_0(\mathbf{x}) \to \text{const}$ **if** $\mathbf{x} \gg \mathbf{1}$
$\chi_0(\mathbf{x}) \sim \mathbf{x}^{\gamma/\nu}$ **if** $\mathbf{x} \to \mathbf{0}$

Some rewriting then yields: $\mathbf{L^{-\gamma/\nu}\chi = \tilde{\chi}\left(L^{1/\nu}t\right)}$

Analysis, critical exponents

from previous sheet: $\mathbf{L^{-\gamma/\nu}\chi = \tilde{\chi}\left(L^{1/\nu}t\right)}$

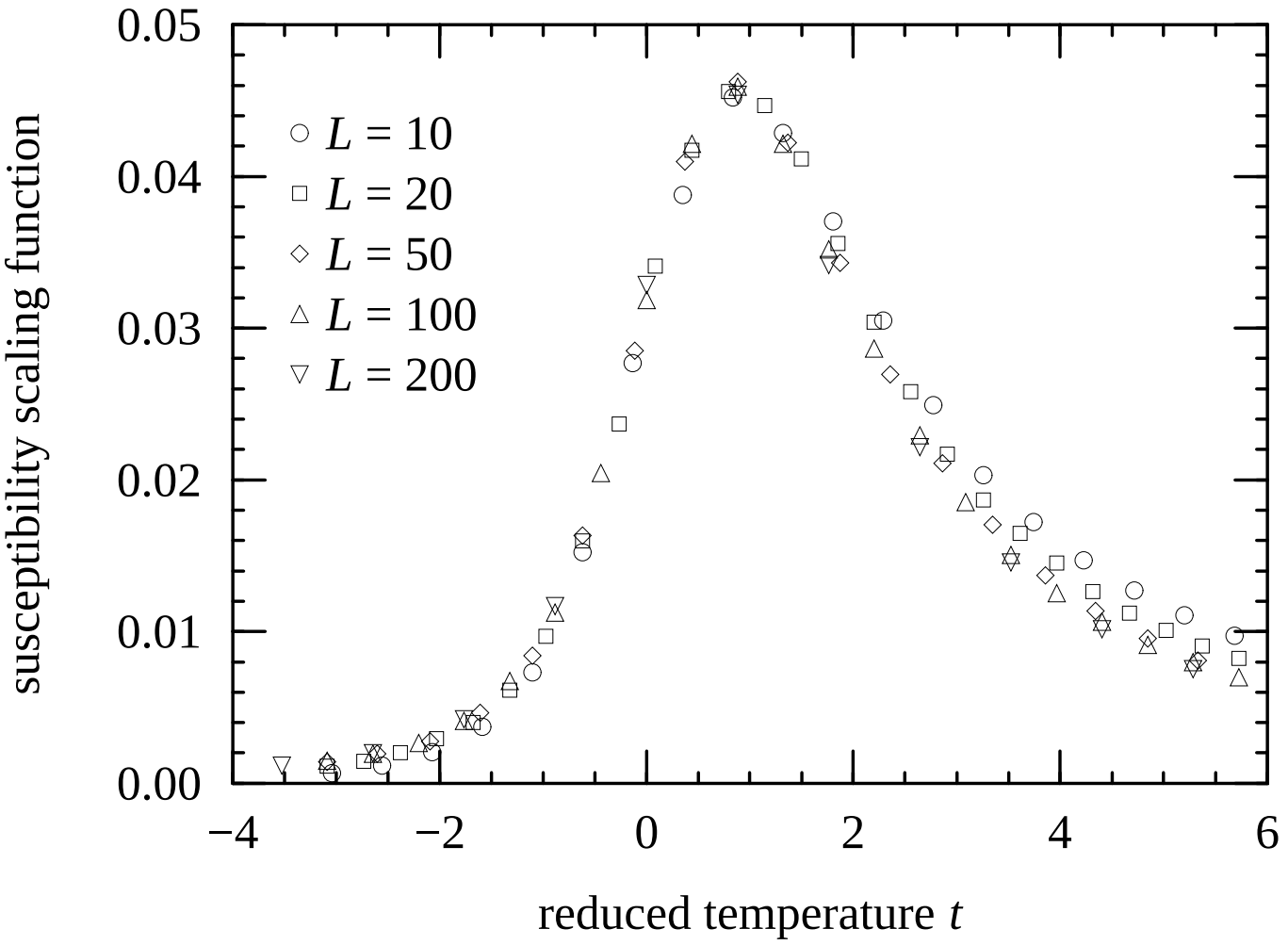
What does this mean?

- measure χ as a function of T in simulations with various sizes L
- if $L^{-\gamma/\nu}\chi$ is plotted as a function of $L^{1/\nu}t$, then these curves should collapse
- This only works with the proper choice of T_c, γ and ν
- $\Rightarrow$ vary your estimates of T_c, γ and ν until a data collapse results
- Indirectly, this measures these!

A similar procedure can be used to determine other exponents

'Data collapse' is a much more general technique, with wide application!

Data collapse



Rejection-free implementation

Standard Metropolis approach:

- **propose each spin flip with unit rate**
- **accept if with probability** $P_{\rm acc} = {\rm Min}\left[1, \exp\left(-\beta \Delta E\right)\right]$

Same dynamics results if:

- **propose each spin flip with its rate** ${\rm Min}\left[1, \exp\left(-\beta \Delta E\right)\right]$
- **then always accept it**

In practice:

- **maintain lists** $\mathbf{Q_i}$ **with spins with equal number of aligned neighbors**
- **select a specific list k with probability** $\mathbf{P_k}||\mathbf{Q_k}||/\mathbf{R}$ **with** $\mathbf{R} = \Sigma_{\mathbf{i}}(\mathbf{P_i}||\mathbf{Q_i}||)$
- **flip a randomly selected spin from that list** $\mathbf{Q_k}$
- **increment the time scale with** $1/\mathbf{R}$
- **update the bookkeeping**

Beyond 'Detailed Balance'

Ballistic motion samples the Boltzmann distribution, violating 'Detailed Balance'

Ballistic motion can explore phase space much faster than diffusion

Is there a general framework for MC simulations without 'Detailed Balance'?

A first step:

- **Increase the phase space with some discretized velocity $\mathrm{v} = \pm 1$**
- **In a rejection-free algorithm, keep separate lists for up- and down-spins with total rate R_+, resp. R_-**
- **If $\mathrm{v} = 1$: if $(\mathrm{R}_- < \mathrm{R}_+)$ revert the velocity with probability $\mathrm{R}_-/\mathrm{R}_+$ else select a down-spin and flip it**
- **If $\mathrm{v} = -1$: if $(\mathrm{R}_+ < \mathrm{R}_-)$ revert the velocity with probability $\mathrm{R}_+/\mathrm{R}_-$ else select an up-spin and flip it**

The idea is to satisfy 'generalized detailed balance':

$$\mathbf{R}(\mu \overset{v=1}{\rightarrow} \nu) = \mathbf{R}(\nu \overset{v=-1}{\rightarrow} \mu)$$

Beyond 'Detailed Balance'

- The resulting algorithm seems to be correct
- At short times, the magnetization fluctuations are significantly enhanced: $\langle (M_t - M_0)^2 \rangle \sim t^2$
- However, at longer times, the initial gain is lost; $\tau \sim L^z$ persists (same z)

- Similar scheme can be implemented in Wolff algorithm
- Result: smaller clusters are flipped, but again z seems unaffected

Current status:

- Correct MC algorithms can be designed which violate detailed balance
- There is yet no example with a significantly higher computational efficiency

Closing remarks

For all of you:

- **Writing Monte Carlo simulation programs is not hard (you can do it!)**
- **Make sure that you thermalize before measuring**
- **Often sophisticated data analysis yields more than from brute force**

For the future computational physicists:

- **Monte Carlo is a very versatile, easy to use simulation technique**
- **It is intellectually challenging to think of new, efficient algorithms**
- **Help is appreciated in finding examples where DB-violation pays off**

Thank you for your attention!