Simulations in Statistical Physics

Course for MSc physics students

János Kertész

Lecture 5

Monte Carlo (MC) for thermal systems: Importance sampling

How to calculate *R*(*p,L*) for percolation?
Generate samples with occupation probability *p,* measure if they percolate, take the ratio of percolating / all samples.

A stupid way to do it would be:
Occupy $n < L^d$ sites at random. Assign to it the probability $P_n = p^n (1-p)^{L^d - n}$. Calculate $R(p,L) \approx \sum_{n=0}^{L^d} \binom{L^d}{n} P_n (N_n^{|} / N_n)$ where $N_n^{|}$ means the number of percolating configurations out of N_n generated configurations with *n* occupied sites.
However, in for large *L* the binomial distribution is very sharp, meaning that except of the neighborhood of $n = L^d p$ the contribution is VERY small.
Fortunately, it is easy to generate configurations with the above probability (the usual way) – then simple average leads to $R \approx N_p^{|} / N_p$

In physics a system is defined by its Hamiltonian. E.g.,

$$\mathcal{H}_{\text{Ising}} = -J\sum_{<i,j>} s_i s_j - h\sum_i s_i \qquad \text{Ising}: s_i = \pm 1$$

$$\mathcal{H}_{\text{XY}} = -J_{XY}\sum_{<i,j>} \mathbf{s}_i \mathbf{s}_j - \mathbf{h}_{XY}\sum_i \mathbf{s}_i \qquad \text{XY}: \mathbf{s}_i \text{ planar unit vector, } \mathbf{h} \text{ in plane}$$

$$\mathcal{H}_{H} = -J_{H}\sum_{<i,j>} \mathbf{s}_i \mathbf{s}_j - \mathbf{h}\sum_i \mathbf{s}_i \qquad \text{Heisenberg}: \mathbf{s}_i \text{ 3d unit vector}$$

$$\mathcal{H}_{\text{Potts}} = -J_{H}\sum_{<i,j>} \delta_{s_i,s_j} - \sum_i \delta_{h,s_j} \qquad \text{Potts}: s_i = 1,2,3...q$$

$$\mathcal{H} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m} + \sum_{i<j=2}^{N} U(\mathbf{r}_i - \mathbf{r}_j) \qquad \text{Atomic liquid (e.g., Ar)}$$

...

Frequently asked question: Thermal average of a dynamic quantity $\overline{A} = \int A(q,p) P^{eq}(q,p) d^d\mathbf{q} d^d\mathbf{p}$ where $P^{eq}(q,p) = e^{-\beta\mathcal{H}(q,p)} / Z$

If *A* depends only on the energy: $\overline{A} = \int A(E)\omega(E)P^{eq}(E)dE$

In normal systems ω is a rapidly increasing function → $\omega(E)P^{eq}(E)$ is a very sharply peaked function.

Importance sampling is needed

Generate **configurations with the equilibrium probability**. If this task is solved, the thermal averages are simple averages:

$$\overline{A} \approx \frac{1}{N}\sum_{i=1}^{N} A_i$$

Problem: How to generate the configurations with the appropriate probability?

Solution: Metropolis (Metropoli-Rosenbluth-Rosenbluth-Teller-Teller=MR²T²) algorithm.

We generate a sequence of configurations using a Markov chain, i.e., a configuration is generated from the previous one with a given transition probability. The corresponding master equation is (Q is a general state variable):

$$\frac{\partial P(Q,t)}{\partial t} = \sum_{Q'} W(Q' \to Q) P(Q',t) - \sum_{Q'} W(Q \to Q') P(Q,t)$$

How to choose $W(Q' \to Q)$ in order to have $\lim_{t\to\infty} P(Q,t) = P^{eq}(Q)$
Physics gives the answer: detailed balance!

$$P^{eq}(Q)W(Q \to Q') = P^{eq}(Q')W(Q' \to Q)$$

If the transition probability per unit time fulfills detailed balance the asymptotic distribution is the equilibrium one.
Note that detailed balance fixes only the ratio of *W*-s:

$$\frac{W(Q \to Q')}{W(Q' \to Q)} = \frac{P^{eq}(Q')}{P^{eq}(Q)} = e^{-\beta\Delta E} \qquad \text{where} \quad \Delta E = \mathcal{H}(Q') - \mathcal{H}(Q)$$

leaving much freedom in the particular choice of *W.*
Two frequently used forms are:

$$W(Q \to Q') \begin{cases} e^{-\beta\Delta E} & \text{if } \Delta E > 0 \\ 1 & \text{otherwise} \end{cases}$$

Metropolis

$$W(Q \to Q') = \frac{e^{-\beta\Delta E}}{1 + e^{-\beta\Delta E}}$$

symmetric

Clearly P^{eq} is a solution if W fulfills detailed balance.
Is it stable? Do $P(Q,t)$-s converge to it?

Let us take a large number of independent MC runs for the same problem. We consider two states: Q and Q'. We denote the number of runs in state Q by N, those in state Q' by N'.
Without loss of generality we assume $\Delta E > 0$
The number of these runs will change due to the transitions between the considered states (we take Metropolis transition probabilities):

$$N(Q \to Q') = NW(Q \to Q') = Ne^{-\beta\Delta E}$$

$$N(Q' \to Q) = N'W(Q' \to Q) = N'$$

$$\Delta N = N(Q \to Q') - N(Q' \to Q) = N\left\{\frac{e^{-\beta E(Q')}}{e^{-\beta E(Q)}} - \frac{N'}{N}\right\}$$

This is a typical negative feedback: If the ration corresponds to the canonical weights, nothing happens, otherwise the correction is in the proper direction.

It is not enough to require detailed balance!

Ergodicity

The Markov chain should be able to visit all possible states (as we assume for the physical system ergodicity).

Having the space of configurations, MC starts with a proper definition of elementary MC steps, which enable an ergodic walk in that space. Fix the temperature. Then – after defining the BC and the initial conditions – a MC simulation consists of the following steps.

i) Choose elementary step $Q \rightarrow$ Q'
ii) Calculate ΔE
iii) Calculate $W(Q \rightarrow Q')$
iv) Generate a random number $r \in (0,1)$
v) If $r < W(Q \rightarrow Q')$ the new state is Q', otherwise it remains Q
vi) Count time increment
vii) Do the necessary measurements (! Relaxation time)
viii) go to i) until max # of steps

The sequence of configurations generated by this MC method does not reflect the real time evolution of the system, however, it can be considered as a „MC time”. This time is measured in „Monte-Carlo steps per site” in order to enable the comparison of the „time”-dependence of systems with different sizes. In some cases this „time” has to do with real time, e.g., in diffusive processes.

As „time” is defined by the sequence of configurations generated from each other the reasoning about characteristic times between measurements applies.

Two examples:

i) Simple atomic liquid

The Hamiltonian is the sum of the kinetic and the potential energy. The velocity distribution in a classical system is given by the Maxwell distribution – it is enough to deal with the potential energy. A possible form is the so-called Lennard-Jones pair potential:

$$U(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$ with two parameters.

Elementary step: Pick a particle and move it to a new position.

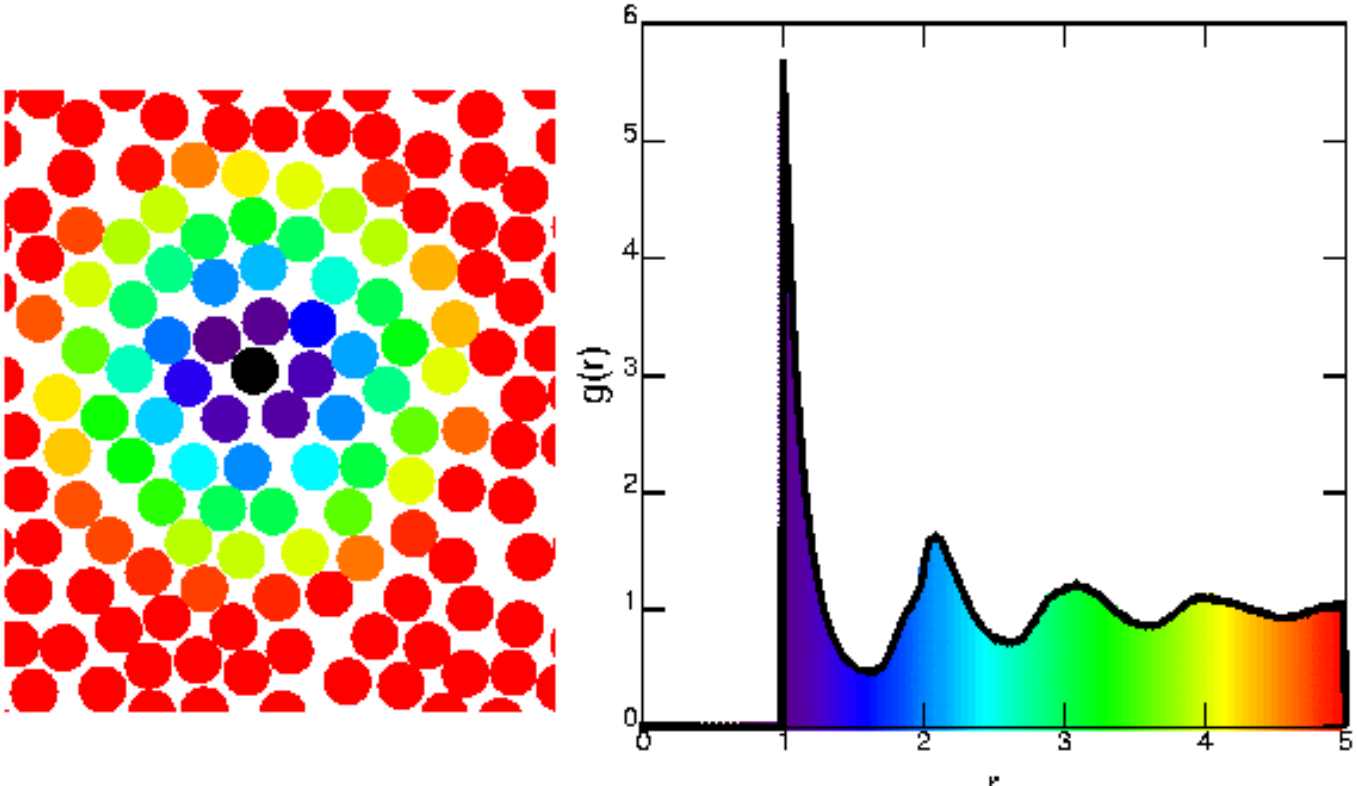
This is clearly ergodic. It is more efficient if we give larger weights to short jumps.

What shall we look for? No dynamics! But

a) Pressure virial theorem

$$P = \frac{1}{3V}\left\langle \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i} - \sum_{i,j=1;i\neq j}^{N} U'(r_{ij}) r_{ij} \right\rangle_T$$

b) Pair correlation function *g*(*r*)



E(*N* = # atoms in the ring of width *dr* and radius *r* around the 0|

∃ atom at 0) = $4\pi r^2 n g(r) dr$, which describes short r. order

The static structure factor of scattering experiments is related to it:

$S(k) = 1 + n\int [g(r) - 1] e^{i\mathbf{kr}} d\mathbf{r}$ measured, e.g., by X-ray scattering

http://stp.clarku.edu/simulations/lj/mc/index.html

ii) Ising model

$$\mathcal{H}_{\text{Ising}} = -J \sum_{<i,j>} s_i s_j - h \sum_i s_i$$

Ergodic elementary step: Pick a spin at random and flip it.
Calculate the energy before (E_i) and after (E_f) the flip.

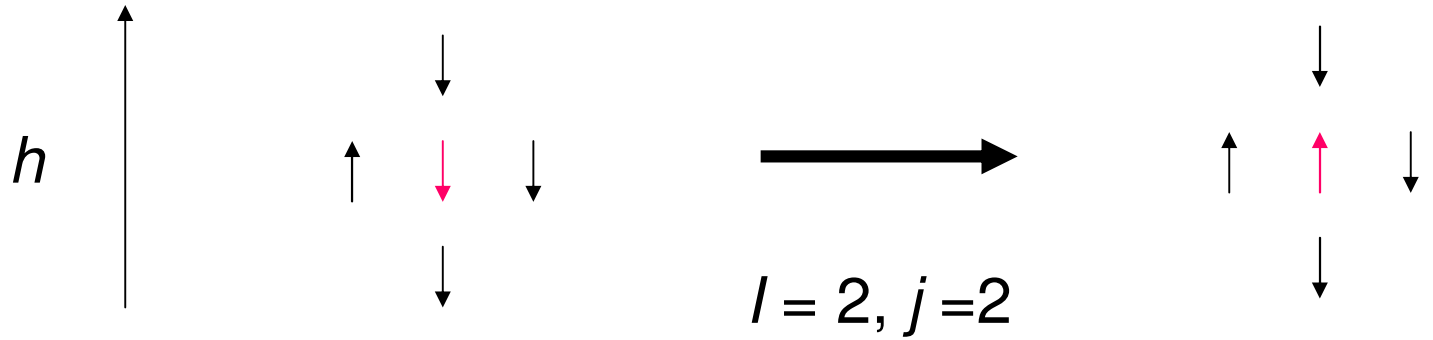
$$\Delta E = E_f - E_i$$

Note that due to the local interaction there are only few possible values of ΔE depending on the number of (anti) parallel neighboring sping and the existence of the external field.
Let us take the square n.n. lattice and $h = 0$.

# of ↑↓ spins	0	1	2	3	4
$\Delta E/J$	8	4	0	-4	-8
$\ln[w(Q \to Q')]$	-8β	-4β	1	1	1

If $h \neq 0$, we have 10 different values instead of 5, because the direction of the spin relative to h has to be considered

Thus we can store the possible values of the transition probabilities in an array of 2 variables w(*i,j*), where *i* -1 is the number of antiparallel spins around the one to be flipped and $j = (s+1)/2+1$, where *s* is the value of that spin ($h \geq 0$)



To be measured:

a) Magnetization (updated when flipping by ± 2)

b) Energy (updated when flipping by ΔE)

Monte Carlo technique is generally used in statistical physics, field theory and material science.

E.g., the study of two component alloys can be mapped onto an Ising model. We start from the energy of the lattice gas model:

$$E = -\mu \sum_j n_j - \epsilon \sum_j \sum_k n_j n_k,$$

Here n_i is 0 or 1 (occupation), μ is the chemical potential and ε the interaction energy. The partition function is

$$\Xi = \sum_{n_1=0}^{1} \sum_{n_2=0}^{1} \cdots \sum_{n_N=0}^{1} e^{\beta\mu \sum_{j=1}^{N} n_j + \beta\epsilon \sum_j \sum_k n_j n_k}.$$

Which is equivalent to an Ising partition function with $J=\varepsilon/4$ and h corresponds to μ (plus a constant). In an alloy, $n = 1$ means A, $n = 0$ means B atoms.

Special techniques for the Ising model

At the critical point not only the correlation length diverges but the relaxation time too.

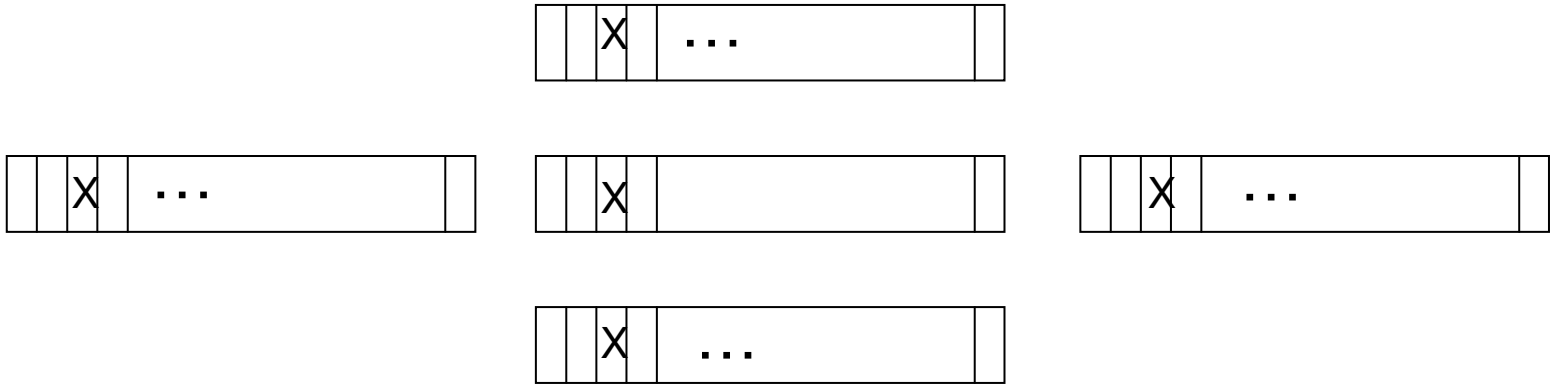
Qualitative explanation: Since spin flips are random, the time τ needed to eliminate/create a regime of size ξ is $\tau \sim \xi^2$ (random walk argument). In fact, 2 is the mean field exponent, in reality it can be different: *z.* (*z* is usually close to 2.)

In a finite size sample the CPU time needed to equilibrate the system is therefore: $t_{\mathrm{CPU}} \sim L^{d+z}$

Efficient algorithms are needed!

Poor men's parallel comuting: **Multi spin coding.**

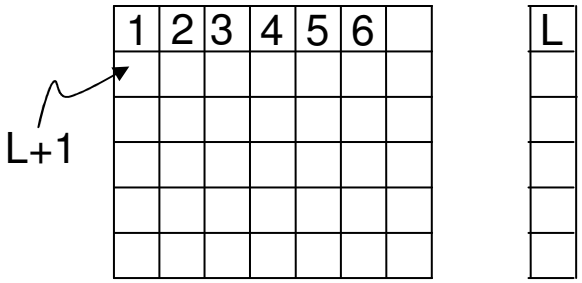
The information about a spin is binary, we can assign 1 to up and 0 to down spins. In principle, we could put 32 spins into a single 32-bit word. While this would save a lot of memory, the handling of the single spins becomes tedious.
We want to handle all the spins in a word simultaneously. Then (taking the square lattice as an example) the following problem arises:



The spins have to be put into the words such that the X-s represent neighbors in the lattice. Then simple XOR indicates the antiparallel spin pairs in a pair of words.

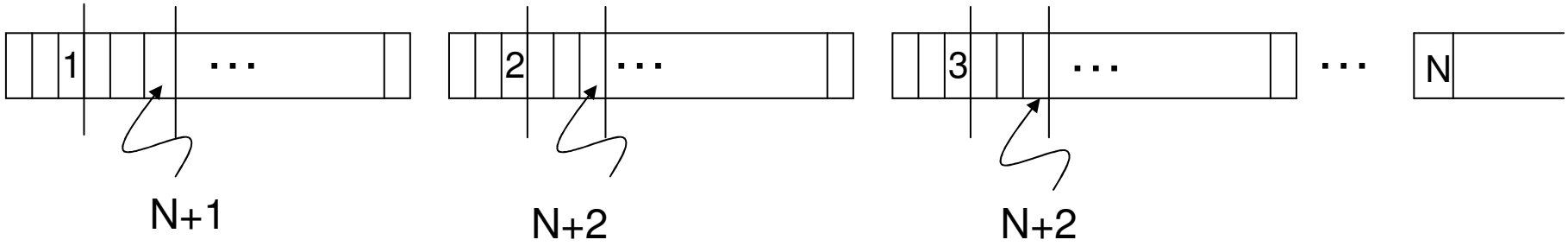
Another problem is that we need the sum of antiparallel neighbors, which can be up to 4. I.e., 3 bits are needed to store the information about the energy of a spin.

These two problems can be solved simultaneously.
We put a spin only into every 3-d bit. The words are filled up in a special way ensuring the proper neighborhods.



$N = L/10$ words/row used in

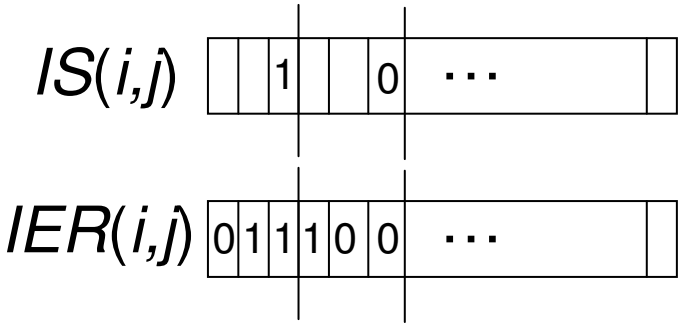
$IS(i,j)$, $i = 1,\ldots L/10$; $j = 1,\ldots L$



We define an array *IER*(*i,j*) for storing the number of antiarallel spins for the 10 spins in *IS*(*i,j*).

$$IER(i,j)=\mathrm{XOR}(IS(i,j),IS(i\text{-}1,j)+\mathrm{XOR}(IS(i,j),IS(i+1,j))+$$
$$\mathrm{XOR}(IS(i,j),IS(i,j\text{-}1))+\mathrm{XOR}(IS(i,j),IS(i,j+1))$$

Now we have for an index pair *i,j* the words *IS* and *IER*



So far we could handle 10 spins in parallel. The MC decision has to be made individually: Shift both the spin under consideration and the corresponding *IER* value to the right end of the words, mask out the *IER* value, calculate the transition probability and flip the spin with a negation if necessary. Special care needed at the end of the words and at BC-s!

The sequence of updates is deterministic, but this does not influence the equilibrium properties. Moreover, the value of the exponent *z* remains unaltered, thus we can only influence the prefactor in the relationship $t_{\mathrm{CPU}} \sim L^{d+z}$ (The gain is about a factor of 3-4.)

How to influence the exponent *z*? Physics helps.
Large *z* due to local („diffusive”) dynamics.
What if we flipped groups of spins together?

Cluster algorithms