

Stochastic simulation methods

or: Monte-Carlo methods

course homepage: <http://www.phys.ubbcluj.ro/~zneda/mc.html>

Zoltán Nédá

UBB-Cluj, Romania

zneda@phys.ubbcluj.ro

- lecture notes
- computer codes
- research projects
- papers to discuss
- important announcements
- interesting links on the web

Main objective of the course:

To give a **PRACTICAL** introduction to Monte Carlo methods in physics. To study modern problems in the field. Start research....

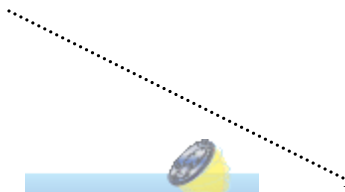
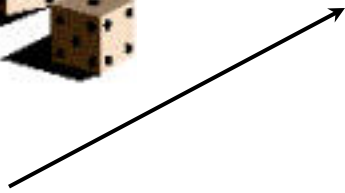
Syllabus

- About Monte Carlo methods
- Random number generators
- Elements of Statistical Physics, Stochastic Processes and Critical Phenomena
- Brownian dynamics
- Monte Carlo methods
- The Ising model
- Metropolis and Glauber dynamics MC for the Ising model
- The BKL algorithm (grain growth, kinetic Monte Carlo methods)
- Cluster algorithms (Swensen and Wang algorithm)
- The histogram Monte Carlo method
- Microcanonical Monte Carlo
- Quantum Monte Carlo methods
- Frustrated systems, spin-glasses
- Application of MC methods in wetting, crack formation, deposition of atoms on surfaces, grain growth , random networks.....
- Discussion and presentation of research projects

What are the Monte Carlo methods?



Computer simulation methods:

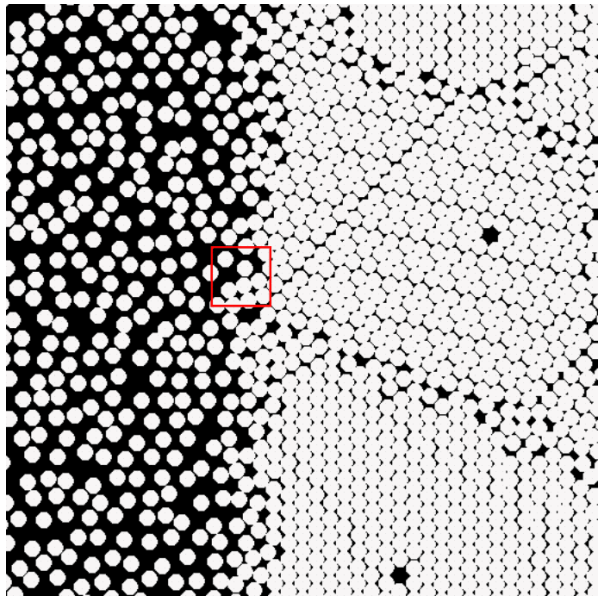


- **Molecular dynamics** (deterministic simulations, based on the integration of the equation of motion)
- **Monte Carlo methods** (Stochastic simulation techniques, where the random number generation plays a crucial role)
- **Cellular automata** (approach to a given phenomena discretized on a lattice, with deterministic or stochastic update rules)

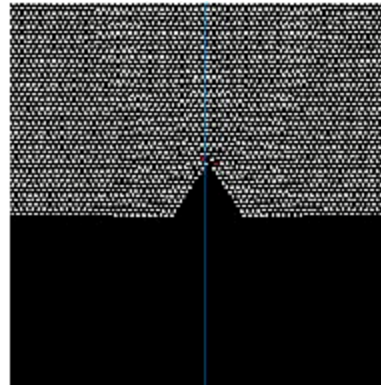
- In general we speak about **Monte Carlo simulation methods** whenever the use of the random numbers are crucial in the algorithm!

- **Monte Carlo techniques are widely used in problems from:** statistical physics, soft condensed matter physics, material science, many-body problems, complex systems, fluid mechanics, biophysics, econo-physics, nonlinear phenomena, particle physics, heavy-ion physics, surface physics, neuroscience etc....

Example: Molecular dynamics simulations



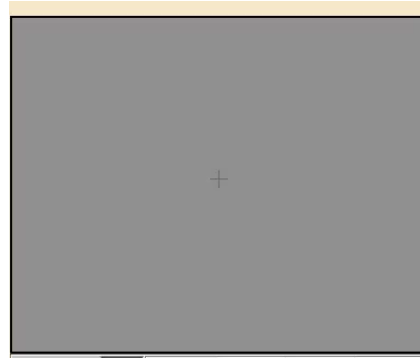
Drying nanosphere suspension on a substrate



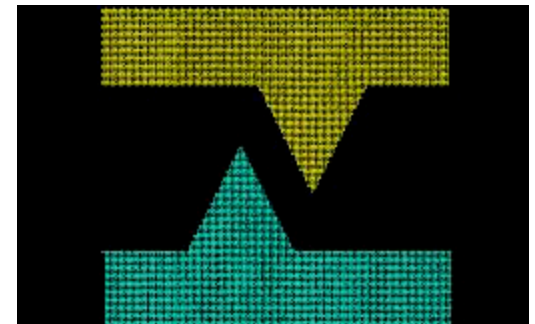
Simulation of crack propagation (atomistic level)



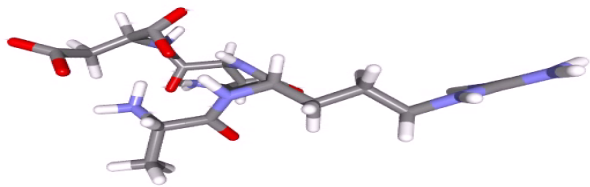
Simulation of crack propagation (large-scale)



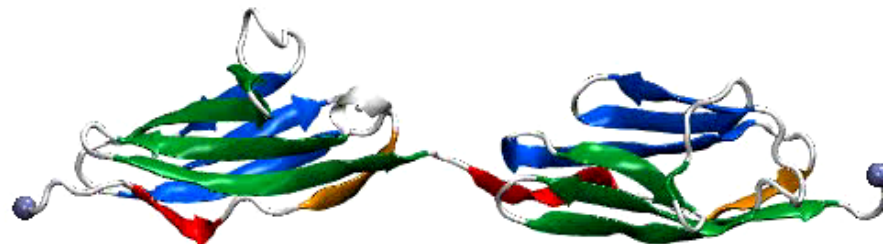
colliding elastic balls



colliding microasperities

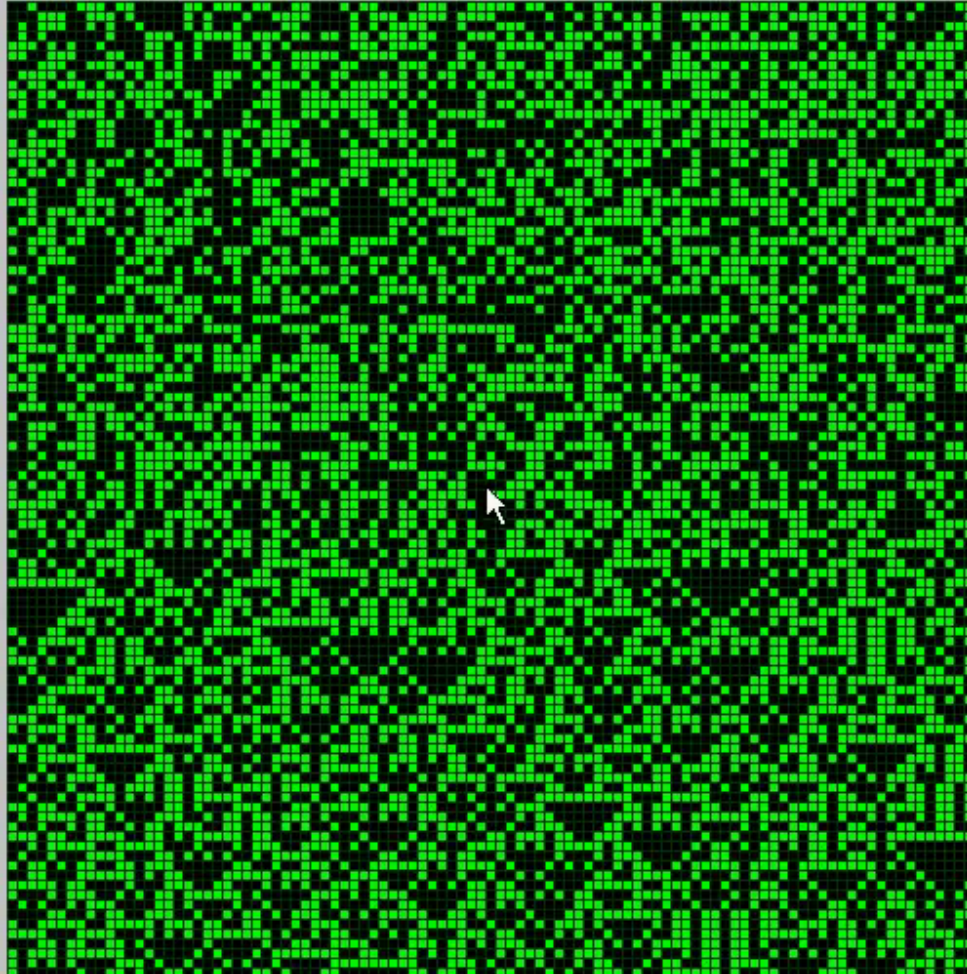


vibrational dynamics of a molecule

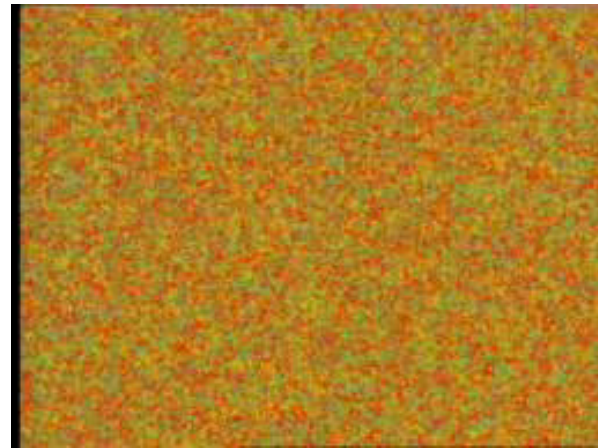


Unfolding a protein

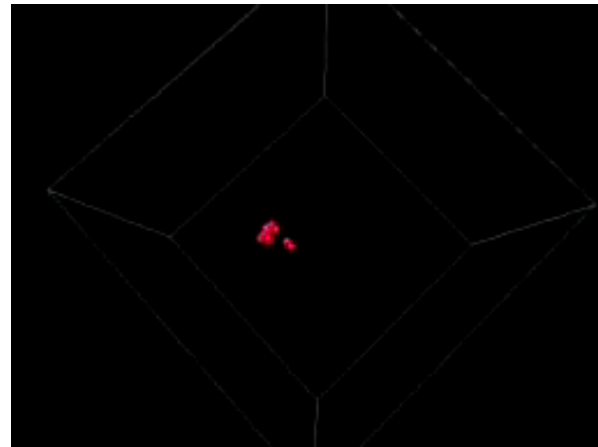
Example: Cellular automata simulation



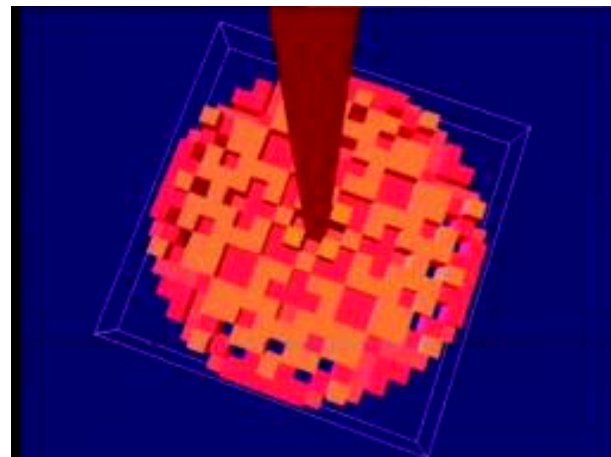
A modulo 2 cellular automata (1D+time evolution)



Cellular automata exhibiting self-organization and spiral waves



3D cellular automata (forest fire model)



3D sandpile cellular automata



DAYTONA
STATE COLLEGE

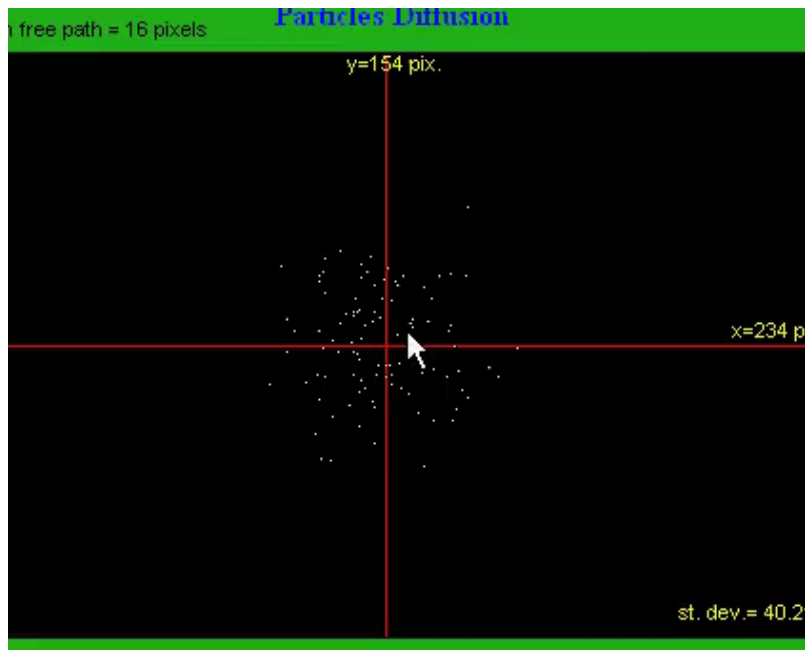
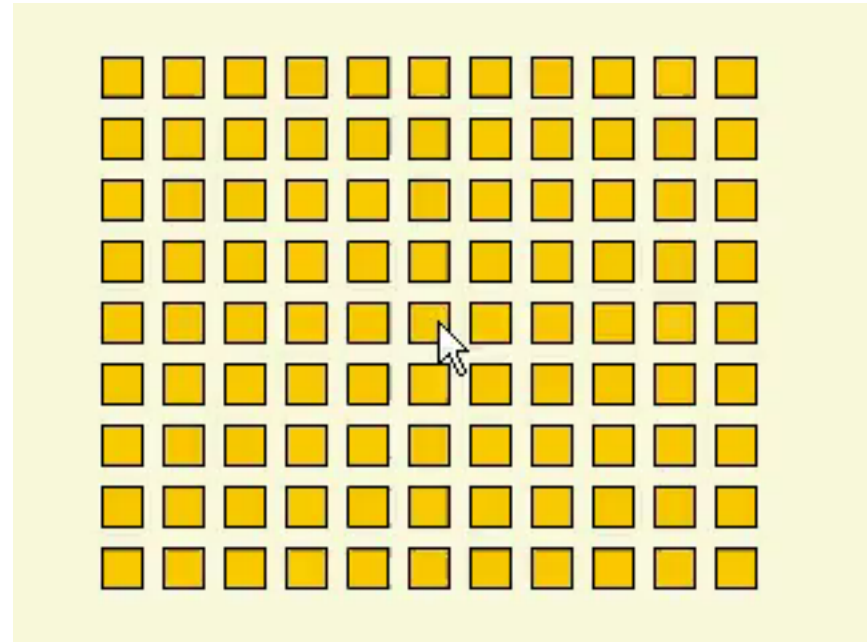
Computer Simulation



Dr. Ron Eaglin

Some well-known problems where MC methods are useful

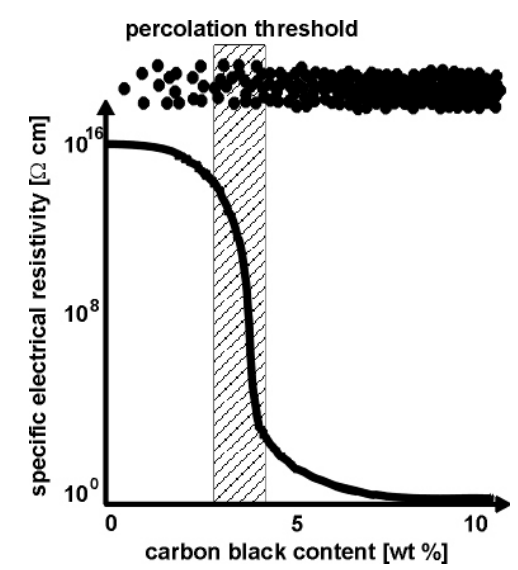
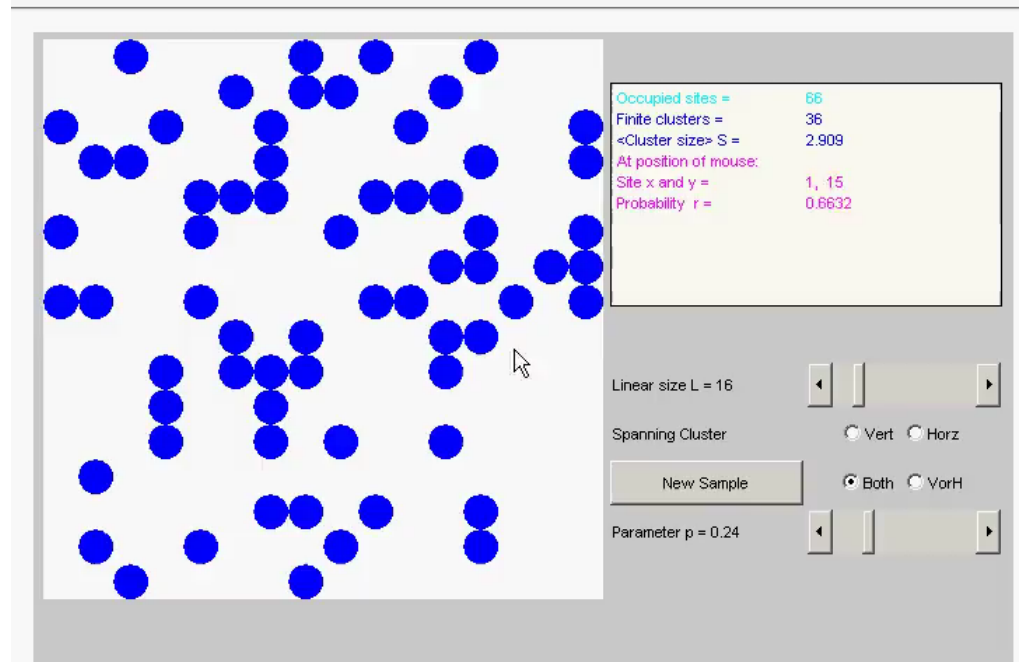
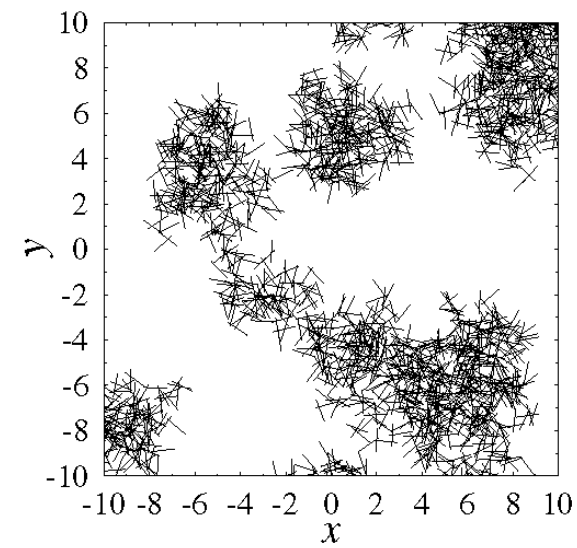
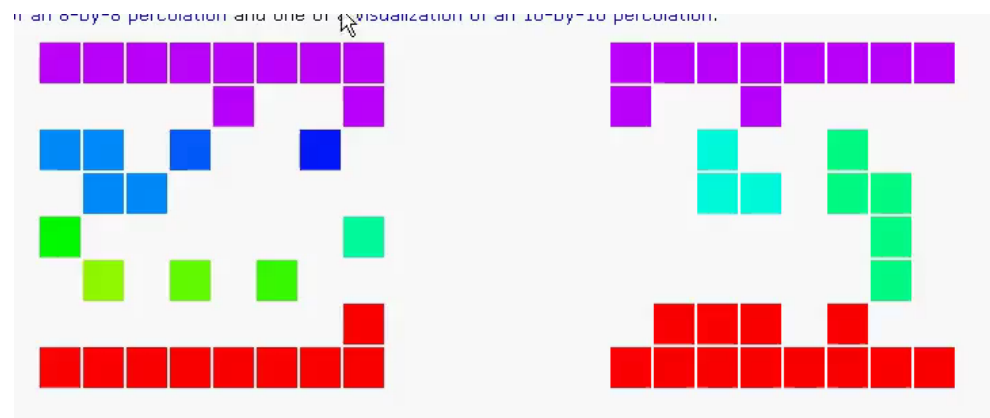
1 .Random walk on a lattice



$$\langle r^2 \rangle \sim N$$

Some well-known problems where MC methods are useful

2. Percolation problems:

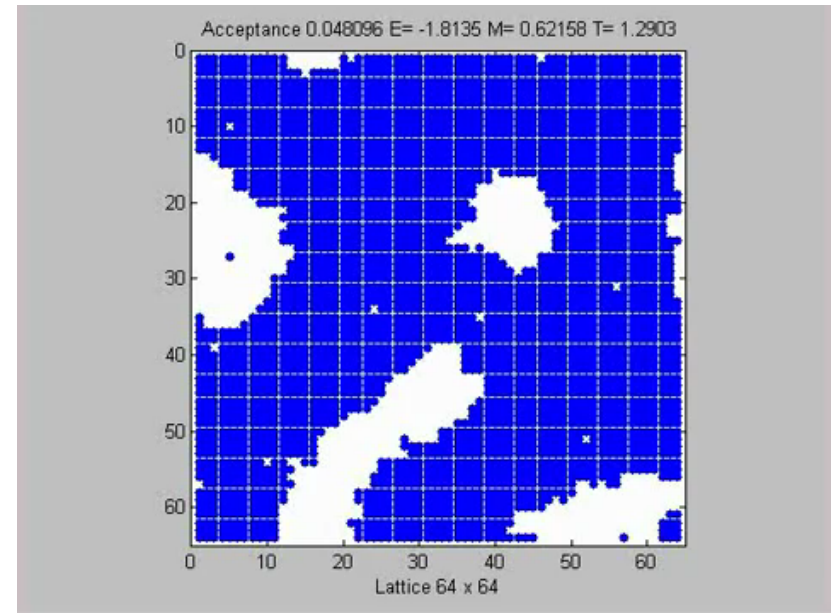
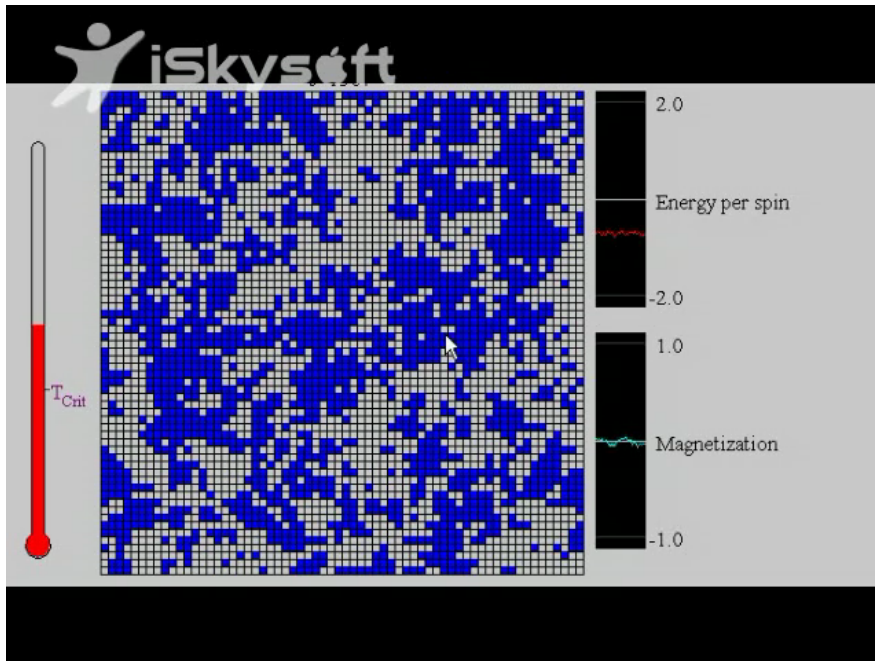
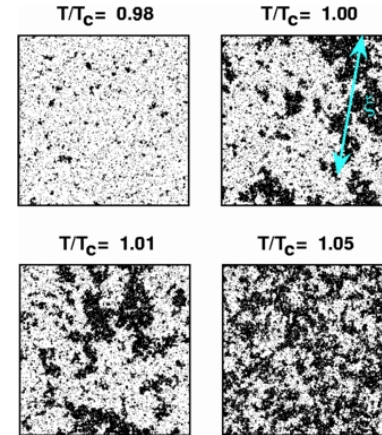
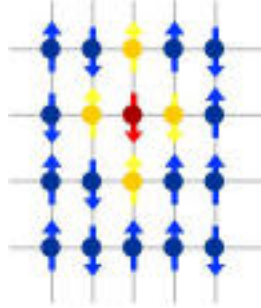


percolation on a two-dimensional square lattice. The picture on the left represents an LxL lattice. Occupied sites are shown

Some well-known problems where MC methods are useful

3.The Ising model:

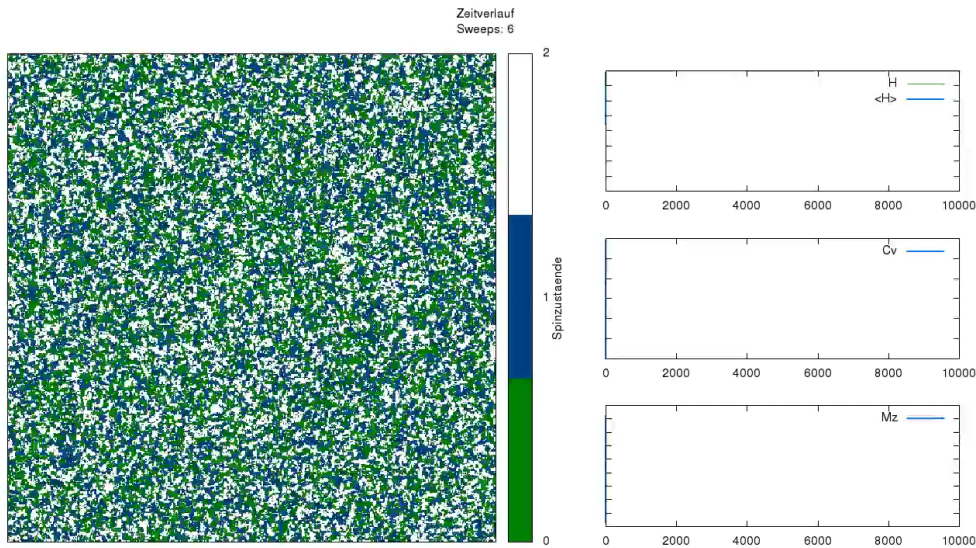
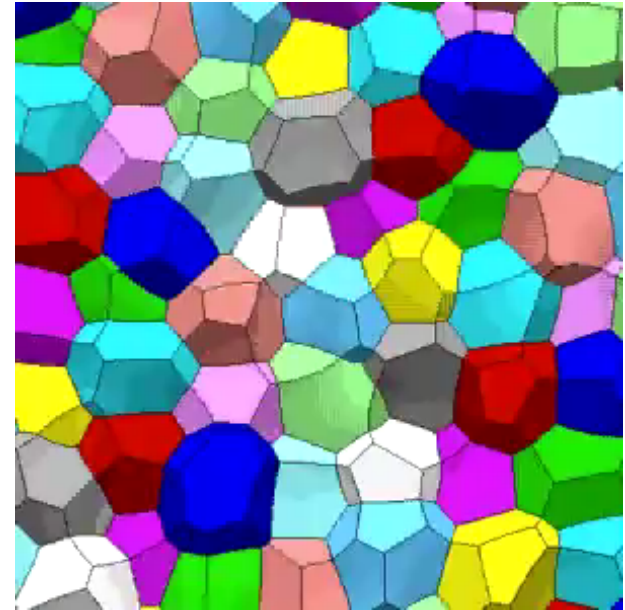
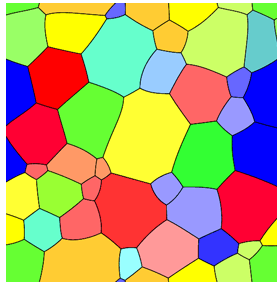
$$H = -J \sum_{\langle i,j \rangle} S_i S_j - \mu B \sum_{\{i\}} S_i$$
$$S_i = \pm 1$$



Some well-known problems where MC methods are useful

4. The p-state Potts model

$$H = -p \sum_{\langle i,j \rangle} J_{ij} \delta_{\sigma(i)\sigma(j)}$$

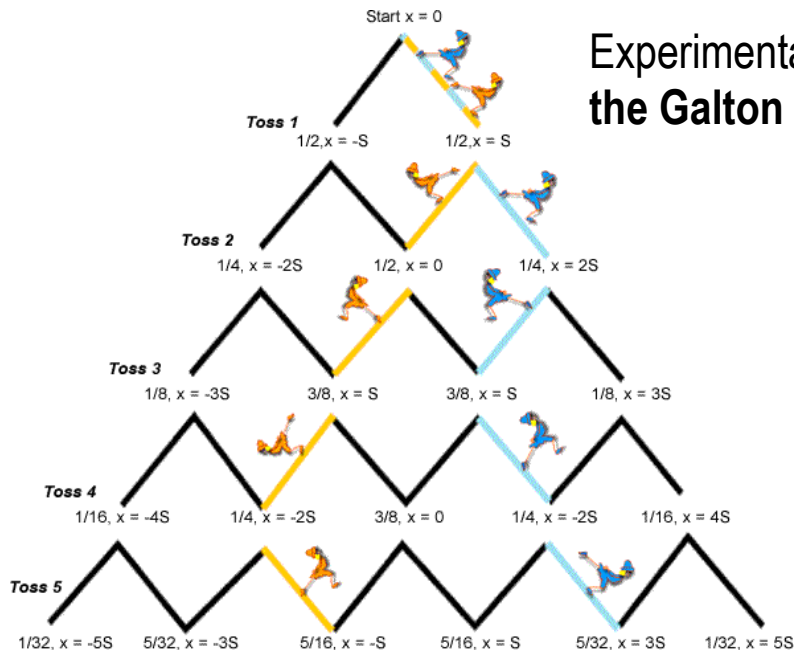
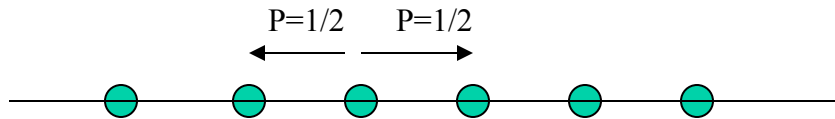


Domain growth (coarse-graining) in the Potts model (low temperature)

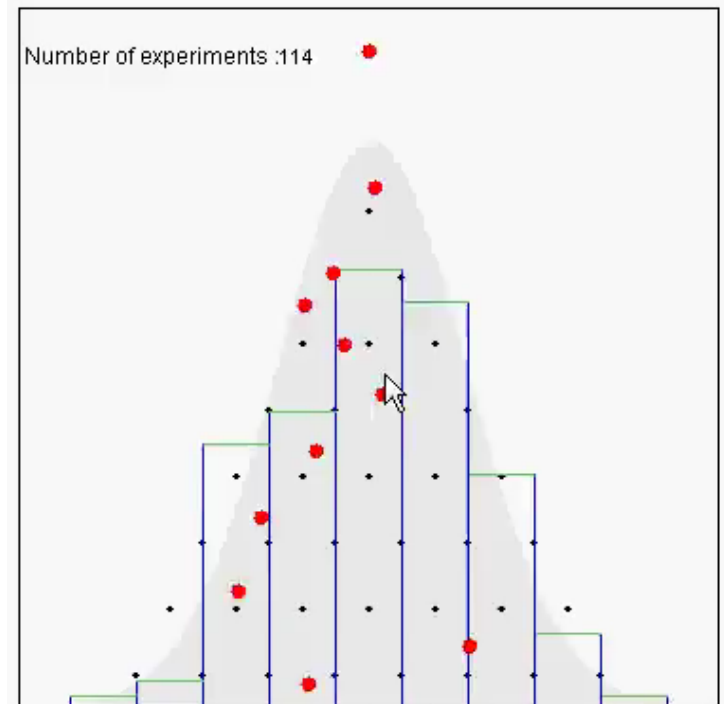
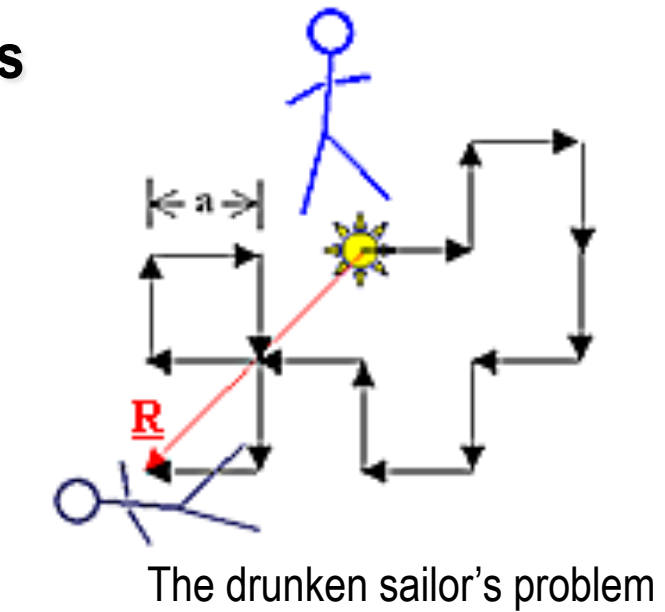
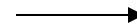
Feeling what Monte Carlo simulation means

- 1. Studying the random walk -

- A basic model in natural sciences
(Brownian motion, fluctuations, diffusion etc...)
- We consider first the simple **1D case**



Experimental realization:
the Galton board



Analytical study

Quantities of interest: $\sqrt{\langle r^2 \rangle_N} \sim N^\alpha \quad \alpha = ? \quad P(N, k) = ?$

$P(N, k) = \frac{W_N^k}{W_N}$ W_N^k : number of possible paths with N steps that end up at coordinate k
 W_N : total number of possible paths with N steps

$$W_N = 2^N; \quad W_N^k = C_N^{(N+k)/2} = \frac{N!}{[(N+k)/2]![(N-k)/2]!}$$

$$\ln[P(N, k)] = \ln(N!) - \ln[(N+k)/2!] - \ln[(N-k)/2!] - N \ln(2)$$

We study the $N \gg 1$ and $k \ll N$ limit and use the following approximations:

$$\left. \begin{array}{l} \ln(n!) \approx n \ln(n) - n + \frac{1}{2} \ln(2\pi n) \quad \text{Stirling's formula} \\ \ln(1 + \frac{k}{N}) \approx \frac{k}{N} - \frac{k^2}{2N^2} + \dots \\ \ln(1 - \frac{k}{N}) \approx -\frac{k}{N} - \frac{k^2}{2N^2} - \dots \end{array} \right\} \begin{array}{l} \text{Taylor expansion} \\ \\ \text{In the continuum limit:} \end{array} \left\{ \begin{array}{l} P(N, k) = \sqrt{\frac{2}{\pi N}} \exp\left(-\frac{k^2}{2N}\right) \\ \\ P(N, x) = \frac{1}{\sqrt{2\pi N}} \exp\left(-\frac{x^2}{2N}\right) \end{array} \right.$$

Scaling properties

$$\langle k^2 \rangle_N = \sum_{\{k\}} k^2 P(N, k) = \int_{-\infty}^{\infty} x^2 P(N, x) dx = N \rightarrow \alpha = \frac{1}{2}$$

the 2D and 3D cases: $\langle r^2 \rangle_N^{2D} = \langle x^2 + y^2 \rangle_N^{2D} = \langle x^2 \rangle_{N/2}^{1D} + \langle y^2 \rangle_{N/2}^{1D} = \frac{N}{2} + \frac{N}{2} = N$

$$\langle r^2 \rangle_N^{3D} = \langle x^2 + y^2 + z^2 \rangle_N^{3D} = \langle x^2 \rangle_{N/3}^{1D} + \langle y^2 \rangle_{N/3}^{1D} + \langle z^2 \rangle_{N/3}^{1D} = \frac{N}{3} + \frac{N}{3} + \frac{N}{3} = N$$

The scaling exponent α is independent thus of the dimension!

Interesting problems:

-Random walks with restriction or memory

special case: **self-avoiding random walk** $\rightarrow$ what is α ?

Solving the problem by MC-type simulations

-**the idea:** reproducing the random walk by using “random numbers”, and realizing the experiment with N random steps many time $\rightarrow$ calculating numerically thus $\langle k^2 \rangle_N$

```
# include <stdio.h>
# include <stdlib.h>
```

```
#define N_kezdeti 50
#define m 5000
#define j 50
```

```
FILE *fp;
```

```
float k_N_medium[j+1];
```

```
main()
```

```
{
```

```
    int k, r, i, ii, ij, N;
```

```
    for(i=1; i<j+1; i++)
```

```
    {
```

```
        N=i*N_kezdeti;
```

```
        for(ii=1; ii<m+1
```

```
            ; ii++)
```

```
        { k=0;
```

```
          for(ij=1; ij<N+1; ij++)
```

```
          {
```

```
              k=k+((int)((float)(rand())/(RAND_MAX+1.0)*2)*2-1);
```

```
          }
```

```
          k_N_medium[i]=(float)(k_N_medium[i]*(ii-1)+k*k)/(float)(ii);
```

```
        }
```

```
    }
```

```
    fp=fopen("result.dat","a");
```

```
    for(ii=1; ii<j+1; ii++)
```

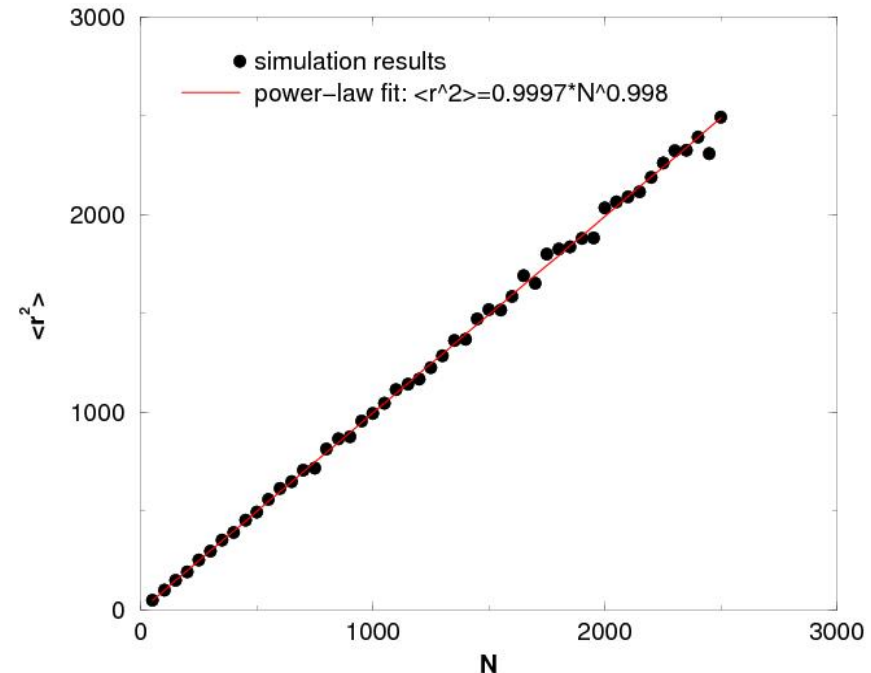
```
        fprintf(fp,"%d  %f\n", ii*N_kezdeti, k_N_medium[ii]);
```

```
    fclose(fp);
```

```
}
```

Result.dat

```
50  49.055225
100 101.964813
150 151.559158
200 192.649689
250 253.033554
300 298.542023
350 354.043884
400 393.019684
450 453.976074
500 496.323608
550 558.835999
600 616.433716
650 649.309265
700 706.541199
750 718.191101
800 813.707275
850 865.757202
900 876.237854
950 956.739868
1000
995.524902
```



simulation results

Phase transition in a sociological system

In a room with sizes $L \times L$, there are N rats. Each rat can be in two states: either calm (state 0) or nervous (state 1). The system of rats obey the following dynamical rules:

1. The rats randomly run through the whole room. From time to time they stop and look around. Each rat can detect only those rats that are within a distance smaller than r .
2. If a nervous rat see no other rat around him, it becomes calm. Otherwise remains nervous.
3. If a calm rat sees a nervous rat around him, it becomes nervous. Otherwise remains calm.
4. With a very small $p \rightarrow 0$ probability a calm rat can become nervous accidentally.

Problem: prove, that in the thermodynamic limit ($L \rightarrow \infty$ and $N \rightarrow \infty$) the rat system exhibits a phase transition as a function of the rats density. I.e. there is a critical rat density (ρ_c) in the system, so that for $\rho < \rho_c$ the stable dynamic equilibrium is that the nervous rats concentration $n_1 = 0$, and for $\rho > \rho_c$ $n_1 > 0$.

Analytical solution

notations: $\rho = \frac{N}{L^2}$, $n_1 = \frac{N_1}{N}$ density of nervous rats, $n_0 = \frac{N_0}{N}$ density of calm rats $(n_1 + n_0 = 1)$

$P_{1 \rightarrow 0}(N, N_1)$ probability/event (time) that a nervous rat becomes calm, $P_{0 \rightarrow 1}(N, N_1)$ probability/event (time) that a calm rat becomes nervous

the master equation of the dynamics:

$$\frac{dN_1}{dt} = P_{0 \rightarrow 1}(N, N_1)N_0 - P_{1 \rightarrow 0}(N, N_1)N_1$$

$$P_{1 \rightarrow 0}(N, N_1) = \left(1 - \frac{\pi r^2}{L^2}\right)^{N-1} ; \quad P_{0 \rightarrow 1}(N, N_1) = 1 - \left(1 - \frac{\pi r^2}{L^2}\right)^{N_1} \quad \text{notation: } g = \frac{\pi r^2}{L^2}$$

in equilibrium: $\frac{dN_1}{dt} = \frac{dN_0}{dt} = 0 \rightarrow [1 - (1 - g)^{n_1 N}](1 - n_1) - n_1(1 - g)^{N-1} = 0 \dots (eq.1)$

Managing the analytical solution for equilibrium

we consider the $g \rightarrow 0$ limit

$$\lim_{g \rightarrow 0} (1-g)^{1/g} = e^{-1} \longrightarrow (1 - e^{-gNn_1})(1 - n_1) - n_1 e^{-(N-1)g} = 0 \quad \boxed{n_1 = ?} \quad \rho_c = \frac{N_c}{N} = ?$$

n_1 is the order-parameter in the system

In the vicinity of the critical point $n_1 \ll 1$

$$e^{-gNn_1} \approx 1 - gNn_1 \longrightarrow \boxed{n_1[(1 - n_1)gN - e^{-(N-1)g}] = 0}$$

$$n_1 = 0 \quad n_1 = 1 - \frac{e^{-(N-1)g}}{gN} \dots (eq.2)$$

Always acceptable acceptable only if $0 \leq n_1 \leq 1$

$$1 - \frac{e^{-(N_c-1)g}}{gN_c} = 0 \longleftarrow N_c$$

$$N_c \approx \frac{1}{2g} + \frac{1}{2} \approx \frac{1}{2g}$$

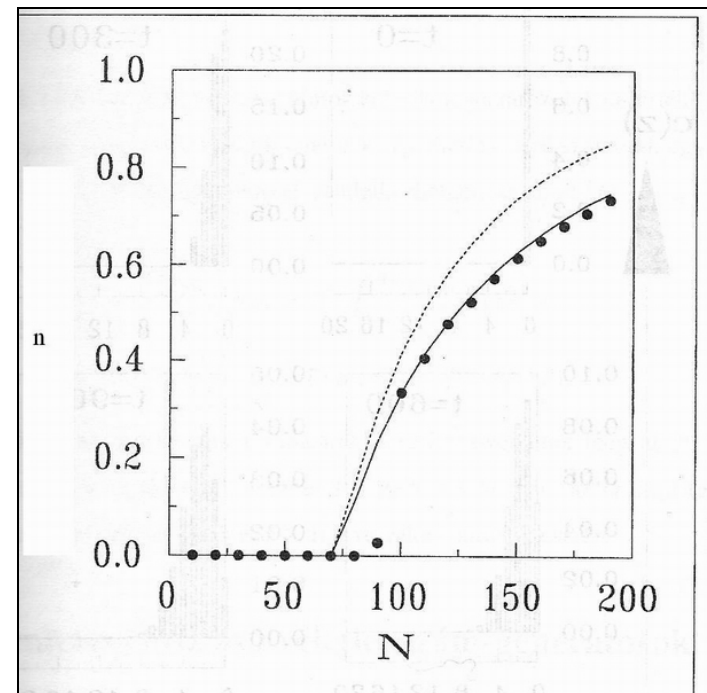
if $N < N_c$ the stable solution is $n_1 = 0$

if $N > N_c$ the stable solution is $n_1 > 0$

numerical solution of eq.1
(continuous line)

eq.2 dashed line

Points: MC simulation



MC simulation of the problem

The C program can be found on the course home-page

the “rats” are placed in new random positions at each simulation steps $\leftrightarrow$ a fast uncorrelated random motion

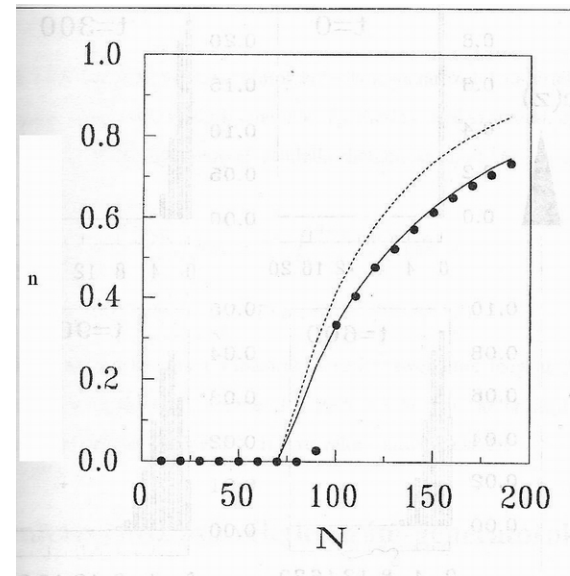
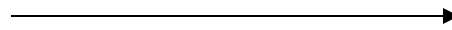
The algorithm:

1. We fix the simulation parameters (r , L , Number of transient steps, $p \rightarrow$ probability of get nervous accidentally , number of steps on which averaging for n_1 is done)
2. We consider simulation with different rats number, outmost cycle..... For each case we initialize the states (calm or nervous) for each rat
3. Using the dynamical rules 1.-4. we give new random positions for the rats, and update their states. We do this many times, first as many times as many transient steps are, and than as many steps as needed for the average
4. We study the average value of n_1 as a function of N

numerical solution of eq.1 (continuous line)

eq.2 dashed line

Points: MC simulation



Random numbers

...the key to MC simulations

- In order to get random numbers we need a real stochastic (random) process like: **throwing a dice** or **tossing a coin**
- In reality there is no stochastic process in our calculator, so in simulations we use **pseudo-random numbers, generated deterministically** by our computer. These numbers will approach a desired random behavior if their statistics satisfy some properties.
- In principle one can design interfaces which will be able to generate real random numbers (using for example tunnel diodes, etc.). The speed of these generators are however very low.

Random numbers

-
- ```
graph LR; A[Random numbers] --> B[- Uniformly distributed on a given interval (real numbers or integers)]; A --> C[- Distributed according to a given distribution];
```
- Uniformly distributed on a given interval (real numbers or integers)
  - Distributed according to a given distribution

# Uniformly distributed pseudo-random numbers

The core of most of them are the **modulo generators**

Primary task is to arrange integers from 0 to  $M-1$  in “random order”:

this is done by:

$$x_n = (ax_{n-1} + c) \bmod(M)$$

(important the proper choice of “**a**” and “**c**”, to get a sequence with a periodicity of  **$M-1$** )

Recepie:

$$a = 2^q + 1 \quad q \rightarrow \text{integer}$$

$M \rightarrow$  large number which is a power of 2

$c \rightarrow$  an odd number below 10

$M \rightarrow$  defined by RAND\_MAX in C  $\rightarrow$  the largest integer

# Uniformly distributed pseudo-random numbers in C

Generating a random “float” in the  $[0,1)$  interval:

$$x = (\text{float})(\text{rand}()) / (\text{RAND\_MAX} + 1.0)$$

Generating a random “float” in the  $[R_{\min}, R_{\max})$  interval:

$$x = R_{\min} + (\text{float})(\text{rand}()) / (\text{RAND\_MAX} + 1.0) * (R_{\max} - R_{\min})$$

Generating a random integer in the  $[R_{\min}, R_{\max}-1]$  interval:

$$k = R_{\min} + (\text{int})((\text{float})(\text{rand}()) / (\text{RAND\_MAX} + 1.0) * (R_{\max} - R_{\min}))$$

rand():

- arranges in random order integers between 0 and RAND\_MAX. (RAND\_MAX is usually the maximal integer-1)
- it's period of repetition is RAND\_MAX
- it starts from the same  $x_0$  seed at each run.
- to start the series from another “random” seed use the randomize() function or srand(i)

# Testing the uniform random number generator

1. Determining the **repetition period** (after how many calls the series will repeat). This must be as big as possible....

2. Testing **the uniformity of the distribution** --> **the histogram test**.

(both for integer and float generators)

idea: construct a histogram for the numbers generated in some fixed constant intervals. Denote by  $y_i$  the frequency of generating a number in the “i” -th bin :  $y_i = N_i / N$  ( $i=1,2,\dots,n$ ). The values of  $y_i$  must converge to  $1/n$  for a uniform distribution as  $N \rightarrow \infty$ . or::

$$\chi = \frac{1}{n} \sum_{i=1}^n \sqrt{(y_i - 1/n)^2}$$
$$\chi \rightarrow 0$$

3. **The return map** test --> testing both the uniformity and the absence of correlation in the  $[0,1)$  interval. A visual test by plotting on an x-y coordinate system  $x_{n+1}$  as a function of  $x_n$ .  
If the generator is a proper one, the points must cover uniformly the  $[0,1) \times [0,1)$  square

4. The absence of short-range correlations --> **the correlation test**

## Pearson correlation

for the total absence of “k” order correlation  
we must have  $C(k) \rightarrow 0$

$$C(k) = \frac{\langle x_i x_{i+k} \rangle - \langle x_i \rangle \langle x_{i+k} \rangle}{\langle x_i^2 \rangle - \langle x_i \rangle^2}$$

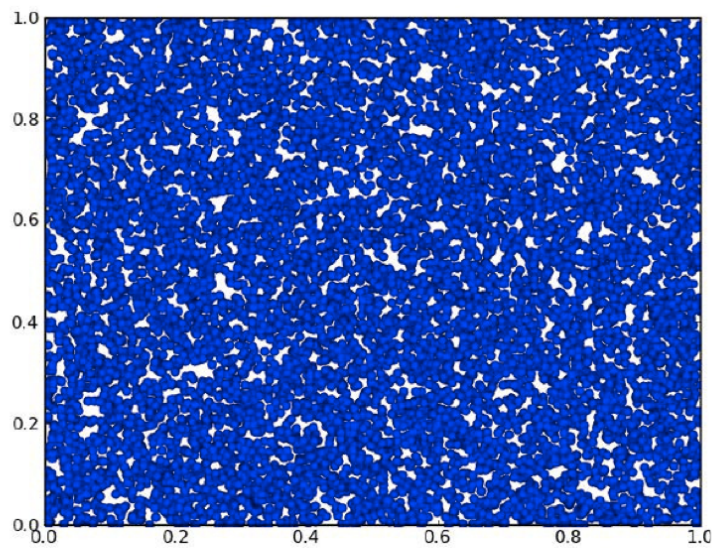


Figure 6.1: *The return map test of a good uniform random number generator*

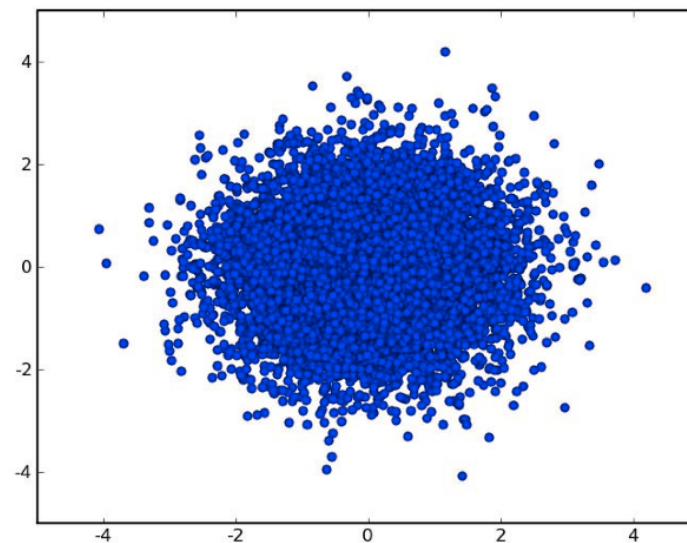


Figure 6.2: *The return map test of a good gaussian random number generator*

# Pseudo-random numbers distributed according to a desired distribution

We are looking for a **Gen2** random number generator, that gives random numbers according to the **g(x)** distribution function, on the  $[R_{\min}, R_{\max})$  interval.

$$\int_{R_{\min}}^{R_{\max}} g(z) dz = 1$$

Let us suppose that **Gen1** gives random numbers distributed uniformly on the  $[0,1)$  interval.

We assume **x** distributed uniformly on  $[0,1)$  and **z** distributed according to **g(z)** on  $[R_{\min}, R_{\max}]$

$$z = F(x) = ?$$

$$f(x)dx = g(z)dz \rightarrow \int_0^y f(x)dx = \int_{R_{\min}}^{F(y)} g(z)dz$$

$$\int_{R_{\min}}^{F(y)} g(z)dz = y$$

$$z = G^{-1}[x + G(R_{\min})];$$
$$G(x) = \int g(x)dx$$

$$Gen1 = 0 \rightarrow Gen2 = R_{\min}$$

$$Gen1 = x \rightarrow Gen2 = y$$

$$Gen1 = 1 \rightarrow Gen2 = R_{\max}$$

$$Gen2 = G^{-1}[Gen1 + G(R_{\min})]$$

# Properties of a good “random number” generator

- the basic generator should have a long period
- no detectable correlation between the terms
- distribution close to the the desired one already for relatively short series
- should be very fast! (should not contain mathematical functions like  $\exp()$ ,  $\sin()$  ....)
- should use small amount of memory
- should be tested before the use
- should be repeatable for optimal debugging purposes

## Gaussian (normally distributed) random number generator (Box-Muller algorithm)

$$\begin{aligned} y_1 &= \sqrt{-2 \cdot \ln(x_1)} \cdot \cos(2\pi x_2) \\ y_2 &= \sqrt{-2 \cdot \ln(x_1)} \cdot \sin(2\pi x_2) \end{aligned} \quad f(y, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y - \mu)^2}{2\sigma^2}\right)$$

- where  $x_1$  and  $x_2$  are random numbers with a uniform distribution on the  $[0,1)$  interval
- $y_1$  and  $y_2$  are normally distributed, mean value 0, standard deviation 1
- If we need random numbers with normal distribution having mean  $\mu$  and variance  $\sigma$  we apply the transformations:

$$z_1 = \mu + \sigma y_1$$

$$z_2 = \mu + \sigma y_2$$

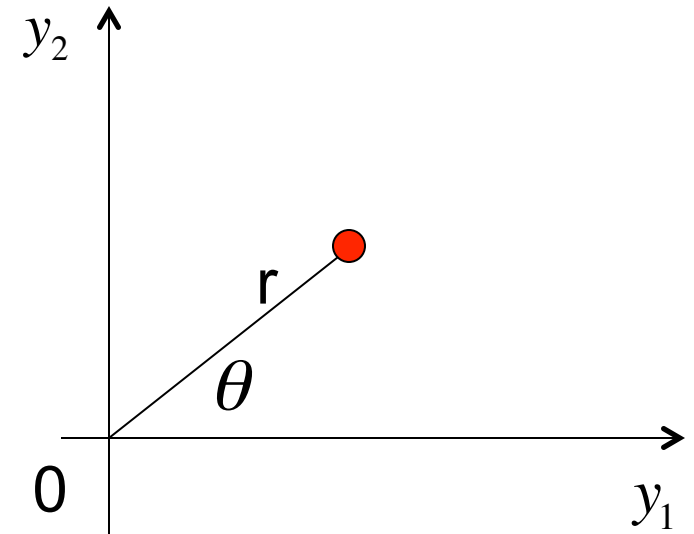
# Proving the Box-Muller algorithm

We need  $y$  distributed according to

$$f(y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right)$$

We consider the distribution in 2D:

$$f(y_1, y_2) = f(y_1)f(y_2) = \frac{1}{2\pi} \exp\left(-\frac{(y_1^2 + y_2^2)}{2}\right)$$



Due to the isotropy of this distribution (not dependent on angle) we can generate it using uniformly generated polar angles in the plane

$$y_1 = R \cos(\theta)$$

$$y_2 = R \sin(\theta)$$

$$\theta = 2\pi x_1 \rightarrow x_1 \text{ is a random variable with uniform distribution on the } [0,1] \text{ interval}$$

$$\rho(R) = \frac{P(R, R + dR)}{dR} = ?$$

$$\left. \begin{array}{l} y_1, y_2 \rightarrow r, \theta \\ f(y_1, y_2) dy_1 dy_2 = f(r) r \cdot dr d\theta \end{array} \right\} \longrightarrow f(r) = \exp\left(-\frac{r^2}{2}\right)$$

$$F(R) = P(r < R) = \int_{r=0}^R \int_{\theta=0}^{2\pi} \frac{1}{2\pi} \exp\left(-\frac{r^2}{2}\right) r dr d\theta = 1 - \exp\left(-\frac{R^2}{2}\right)$$

$$\rho(R) = R \cdot \exp\left(-\frac{R^2}{2}\right) \rightarrow G(R) = -\exp\left(-\frac{R^2}{2}\right) \longrightarrow \text{We need random numbers generated according to this distribution}$$

We apply the general recipe with  $R_{\min}=0$ :

$$\boxed{Gen2 = G^{-1}[Gen1 + G(R_{\min})]}$$

$$G^{-1}(x) = \sqrt{-2 \ln(-x)} \longrightarrow R = \sqrt{-2 \ln[-(x-1)]}$$

$x$  is a random variable with uniform distribution on the  $[0,1]$  interval

$$-(x-1) \rightarrow x_2 \longrightarrow \text{again a random variable with uniform distribution on the } [0,1] \text{ interval}$$

We obtain thus:

$$y_1 = \sqrt{-2 \cdot \ln(x_1)} \cdot \cos(2\pi x_2)$$

$$y_2 = \sqrt{-2 \cdot \ln(x_1)} \cdot \sin(2\pi x_2)$$

# Exercises

1. Write a simple dice-throwing program
2. Using the `rand()` function write a simple BINGO-game program (give the numbers from 1-49 in random order)
3. Using the `rand()` function write a “float” random number generator on the  $[0,1)$  interval and test it!
  - make the return-map test
  - make the histogram test
  - calculate the  $C(2)$ ,  $C(4)$  and  $C(6)$  correlation values
4. Write a random number generator that generates float random numbers on the  $[0,4)$  interval according to the  $g(x)=x^2$  distribution function.
5. Write a random number generator that generates float random numbers on the  $[0,100)$  interval according to a Gaussian distribution. Use the central limit theorem for generating this Gaussian (normal) distribution.

Central limit theorem: the sum of sufficiently large number of identically distributed independent random variables each with finite mean and variance will be approximately normally distributed.

(more on Wikipedia....)

## Exercises

1. Invent a Monte Carlo method for calculating the value of  $\pi$  !
2. Prove numerically that by the formulas of the from

$$y_1 = \sqrt{-2 \cdot \ln(x_1)} \cdot \cos(2\pi x_2)$$

$$y_2 = \sqrt{-2 \cdot \ln(x_1)} \cdot \sin(2\pi x_2)$$

where  $x_1$  and  $x_2$  are random numbers with a uniform distribution on the  $[0,1)$  interval  $\rightarrow$  one generates Gaussian random numbers

3. Invent a Monte Carlo method to calculate the value of  $e$  !

# Elements of statistical physics

- statistical physics deals with systems of large number of particles or stochastic processes
- the (3D) **coordinate** and (3D) **phase-space** of one particle
- the (6D) **state-space** of one particle
- the (6ND) state-space of N particles (the state of the system is characterized by a characteristic point in this 6ND space)
- the **allowed region of the state-space** (region of the state-space where the characteristic point can move; points allowed by the externally imposed conditions)
- the externally imposed conditions --> the ensemble in which the systems is
- **the ergodic principle**: in a very short time (much shorter than the time needed for a physical measurement) the characteristic point of the system visits all the allowed points of the state-space.  
(Not all systems respects the ergodic principle!)
- when we measure one physical quantity, we usually measure it's time-average for the states that are visited during the measurement time by the characteristic point of the system
- systems that respect ergodicity: **time average** --> **ensemble average** (ensemble average is an average over the allowed points of the state-space)
- for non ergodic systems: time average cannot be replaced by ensemble average!

# The microcanonical ensemble

All allowed points of the state-space are equally probably realized.

Important microscopic quantity:

**W** : the number of allowed microstates

$$S = k \ln(W) \quad \text{Boltzmann's equation}$$

Generalization of the Boltzmann equation: **Renyi entropy valid for all ensembles:**

$$S = -k \sum_{\{i\}} p_i \ln(p_i)$$

Useful equation for handling analytically or numerically  $W \rightarrow$

**U, V, N** fixed

(for magnetic systems: **U, B, N** fixed)

relevant thermodynamic potential:  
**S** (entropy)

$$\frac{P}{T} = \left( \frac{\partial S}{\partial V} \right)_{U,N}$$

$$\frac{1}{T} = \left( \frac{\partial S}{\partial U} \right)_{V,N}$$

$$\frac{\mu}{T} = - \left( \frac{\partial S}{\partial N} \right)_{U,V}$$

$$\frac{\vec{M}}{T} = \left( \frac{\partial S}{\partial \vec{B}} \right)_{U,N}$$

$$\frac{1}{T} = \left( \frac{\partial S}{\partial U} \right)_{V,N}$$

$$\frac{\mu}{T} = - \left( \frac{\partial S}{\partial N} \right)_{U,V}$$

$$\ln(N!) \cong N \ln(N) - N + \frac{1}{2} \ln(2\pi N)$$

**Stirling's formula**

# Canonical ensemble

**Different points of the state-space are not visited with the same probabilities**

The probability  $p_i$  of visiting a micro-state with energy  $E_i$  is:

$$p_i = \frac{1}{Z} e^{-\frac{E_i}{kT}}$$

Important microscopic quantity;  $Z$  (the canonic partition function)

$$Z = \sum_{\{i\}} e^{-\frac{E_i}{kT}}$$
$$F = -kT \ln(Z)$$

Average value of a physical quantity:

$$\langle A \rangle = \frac{1}{Z} \sum_{\{i\}} A_i e^{-\frac{E_i}{kT}}$$

**T, V, N fixed**

relevant thermodynamic potential: **F** (free-energy)

$$S = -\left(\frac{\partial F}{\partial T}\right)_{V,N}$$

$$P = -\left(\frac{\partial F}{\partial V}\right)_{T,N}$$

$$\mu = \left(\frac{\partial F}{\partial N}\right)_{T,V}$$

$$S = -\left(\frac{\partial F}{\partial T}\right)_{V,N}$$

$$\vec{M} = -\left(\frac{\partial F}{\partial \vec{B}}\right)_{T,N}$$

$$\mu = \left(\frac{\partial F}{\partial N}\right)_{T,V}$$

$$C_V = \frac{1}{kT^2 N} \left( \langle E^2 \rangle - \langle E \rangle^2 \right)$$

For magnetic systems:

$$p_i = \frac{1}{Z} e^{-\frac{E_i}{kT}}$$

$$Z = \sum_{\{i\}} e^{-\frac{E_i}{kT}}$$
$$F = -kT \ln(Z)$$

for magnetic systems: **T, B, N**  
fixed

relevant thermodynamic  
potential: **F** (free-energy)

$$\vec{M} = - \left( \frac{\partial F}{\partial \vec{B}} \right)_{T, N}$$

$$\chi = \left( \frac{\partial \vec{M}}{\partial \vec{H}} \right)_{\vec{B} \rightarrow 0} = \frac{1}{NkT} (< M^2 > - < M >^2)$$

# For quantum-mechanical systems

Calculating the partition function is rather complicated.

- First have to solve the stationary Schrodinger equation (diagonalizing  $\hat{H}$  )
- Then, have to calculate Z on the obtained energy eigenvalues....

A more viable method: 
$$Z = Tr \left[ \exp \left( -\frac{\hat{H}}{kT} \right) \right]$$

$$\exp \left( -\frac{\hat{H}}{kT} \right) = \hat{I} + \frac{1}{1!} \hat{H} + \frac{1}{2!} \hat{H}^2 + \dots$$

Calculation of Z is possible by some wise approximations

# Elements of critical phenomenon

**Critical phenomenon:** behavior near a phase-transition point

- phase-transitions are usually possible only in systems with interactions among it's constituents.
- real phase-transitions are only in infinite systems!

## Phase transitions

```
graph LR; A[Phase transitions] --> B[First order transitions]; A --> C[Second order phase transitions]; A -.-> D[many other type...];
```

**First order transitions** (the first order derivative of the relevant thermodynamic potential has a discontinuity)

examples: boiling, freezing, hadronization ...

**Second order phase transitions** (the second order derivative of the relevant thermodynamic potential has a singularity or discontinuity)

examples: para-ferromagnetic transition, percolation ....

**many other type...**

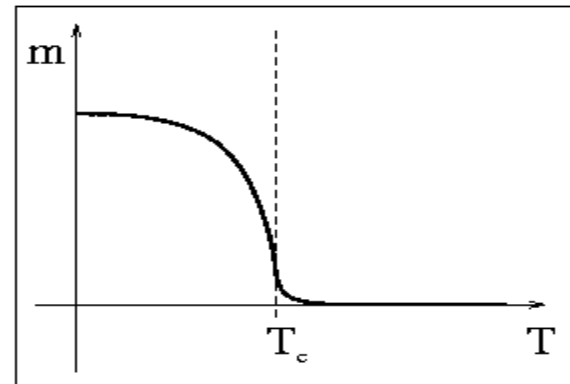
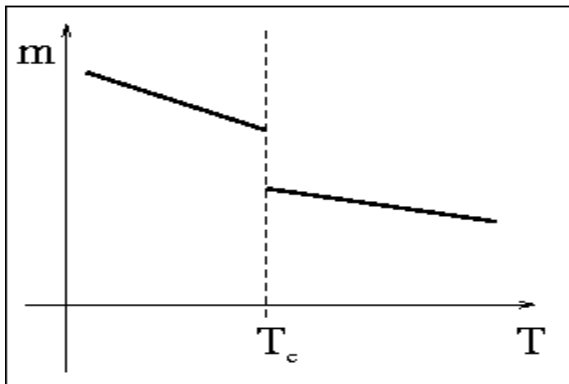
# The order parameter ( $m$ )

- Characterizes the degree of the order in the system and the phases
- It is usually a first derivative of the relevant thermodynamic potential
- It is usually non-dimensional
- It is usually in the  $[0,1]$  interval
- Discontinuity at the transition point for first-order phase-transitions
- It's derivative has a discontinuity for second-order phase-transitions
- Examples of order parameters:

**$\langle S \rangle$** : for the Ising model

**relative size of the largest cluster**: for percolation

**$\rho/\rho_0$** : for freezing or boiling



# Critical exponents

- defined for second-order phase-transitions
- all relevant physical quantities have a power-law behavior in the vicinity of the transition point. The quantities either converge to 0, or diverges.
- important is the **universality of the critical exponents** (they do not depend on the microscopic details of the model, just on the symmetry properties of the order parameter and the dimensionality of the system)
- some important critical exponents:
  - $\beta$ : critical exponent of the order parameter
  - $\alpha$ : critical exponent of  $C_V$
  - $\gamma$ : critical exponent of the susceptibility
  - $\nu$ : critical exponent of the correlation length

**Critical exponents are crucial quantities in order to understand and model phase-transitions!**

$$A \sim t^\alpha$$
$$t = \frac{|T - T_c|}{T_c}$$

-  $\alpha$  is the critical exponent of quantity A

if  $\alpha > 0$   $A \rightarrow 0$  when  $T \rightarrow T_c$

if  $\alpha < 0$   $A \rightarrow \pm\infty$  when  $T \rightarrow T_c$

we can define an  $\alpha_+$  ( $T > T_c$ )  $\alpha_-$  ( $T < T_c$ )

Critical exponents are not independent from each other!  $\rightarrow$  **scaling laws**: universal equations between the critical exponents

Ex. for 2D systems ( $d=2$ )

$$\alpha + 2\beta + \gamma = 2$$

**Hyperscaling equation** (valid in all dimensions)

$$d\nu = 2 - \alpha$$

# Elements of Stochastic Processes

## Markov processes/ Markov chains

**Markov processes** (chain) are characterized by a lack of memory (i.e. the statistical properties of the immediate future are uniquely determined from the present, regardless of the past)

**Example:** random walk --> Markov process; self-avoiding walk is NOT a Markov process

Let  $\underline{x}_i$  be the state of the stochastic system at step “i”, a stochastic variable

The time- evolution of the system is described by a sequence of states:  $\underline{x}_0, \underline{x}_1, \dots, \underline{x}_n, \dots$

The conditional probability that  $\underline{x}_n$  is realized if previously we had:  $\underline{x}_0, \underline{x}_1, \dots, \underline{x}_{n-1}$ :

$$P(\underline{x}_n \mid \underline{x}_{n-1}, \dots, \underline{x}_0)$$

**Definition:** For a Markov process we have:

$$P(\underline{x}_n \mid \underline{x}_{n-1}, \underline{x}_{n-2}, \dots, \underline{x}_0) = P(\underline{x}_n \mid \underline{x}_{n-1})$$

$$P(\underline{x}_0, \dots, \underline{x}_n) = P(\underline{x}_n \mid \underline{x}_{n-1}) \cdot P(\underline{x}_{n-1} \mid \underline{x}_{n-2}) \dots P(\underline{x}_1, \underline{x}_0) \cdot a_0$$

$$P(\underline{x}_k, \underline{x}_j) = P(\underline{x}_k \rightarrow \underline{x}_j) = P_{k,j}$$

one-step transition probabilities,  
elements of the **stochastic matrix**

**Definition:** A probability distribution ( $w_k$ ) over the possible states (k) is called **invariant** or stationary for a given Markov chain if satisfy:

$$w_k > 0;$$

$$\sum_k w_k = 1;$$

$$w_s = \sum_k w_k P_{ks}$$

# Ergodicity of a Markov chain

- A Markov chain is **irreducible** if and only if every state can be reached from every state! (the stochastic matrix is irreducible)

-A Markov chain is **aperiodic**, if all states are **aperiodic**. A state  $x_i$  has a period  $T > 1$  if  $P_{ii}^{(n)} = 0$  unless  $n = zT$  ( $z$ : integer), and  $T$  is the smallest integer with this property. A state is aperiodic if no such  $T > 1$  exist.

(Here we denoted by  $P_{ik}^{(n)}$  the probability to get from state  $i$  to state  $k$  through  $n$  steps)

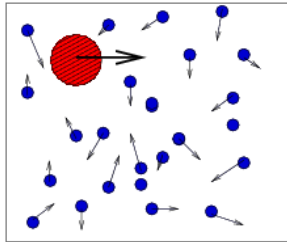
**Definition:** An irreducible and aperiodic Markov chain is called **ergodic**

The **basic theorem for Markov processes:**

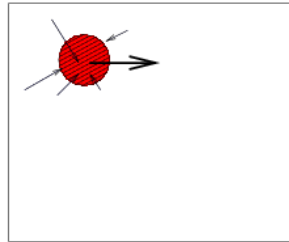
An ergodic Markov chain possesses an invariant distribution  $w_k$  over the possible states

# Brownian Dynamics

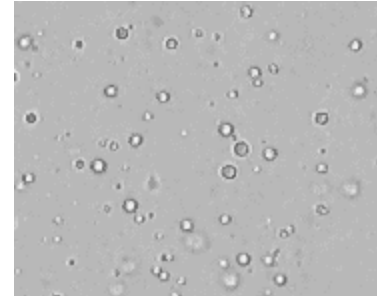
- It is a **hybrid method**, involving both deterministic and stochastic dynamics
- in molecular dynamics methods all degrees of freedom were explicitly taken into account --> classical equation of motion of particles
- in Brownian dynamics some degrees of freedom are represented only through their stochastic influence



Molecular dynamics



Brownian dynamics



- we study Brownian dynamics in canonical ensemble. Basic idea: **The effect of the constant temperature heat-bath --> by a stochastic force-field acting on the particles.**

- in Brownian dynamics simulation methods the system is described by stochastic differential equations. For example the equation of motion of a particle making a Brownian motion:

$$m \frac{dv}{dt} = R(t) - \mu v$$

**Langevin equation of motion**, stochastic equation of motion. The coupling to the heat bath is realized through the  $R(t)$  stochastic force.

**Question:** What properties should  $R(t)$  have, in order to be equivalent with a heat-bath at temperature  $T$ ?

- we are looking for  $R(t)$ , that will lead for  $v$  the classical invariant Maxwell-Boltzmann distribution, expected in canonical ensemble.
- let us work in 1D!
- by using the theory of stochastic processes and Markov chains, it can be shown, that this is achieved when:

$$P(R) = (2\pi \langle R^2 \rangle)^{-1/2} \exp(-R^2 / 2 \langle R^2 \rangle)$$

$$\langle R^2 \rangle = 2\mu k_B T / h$$

$$\langle R(t) \rangle = 0$$

$$\langle R(t)R(0) \rangle = 2\mu k_B T \delta(t)$$

$h$ : is the time-step for numerically integrating the equation of motion.

This choice of  $R(t)$  leads to:

$$P(v) = \sqrt{\frac{m}{2\pi k_B T}} e^{-\frac{mv^2}{2k_B T}}$$

### Algorithm for simulating the Brownian dynamics:

1. Assign initial position and velocity for the particle
2. Draw a random number from a Gaussian distribution with mean zero and variance as described above. This will give us  $R(t)$ .
3. Integrate the equation of motion with the obtained value of  $R$ , and get the new positions and velocities.
4. Proceed with step 2.

## Another way of doing Brownian dynamics:

- by taking into account the coupling of the system to the heat-bath by “statistical” collisions with virtual particles. In this approach no friction is necessary.
- each stochastic collision is assumed to be an instantaneous event
- the colliding virtual particles have a Maxwell-Boltzmann momentum distribution
- The time intervals at which a particle suffers a collision is distributed according to

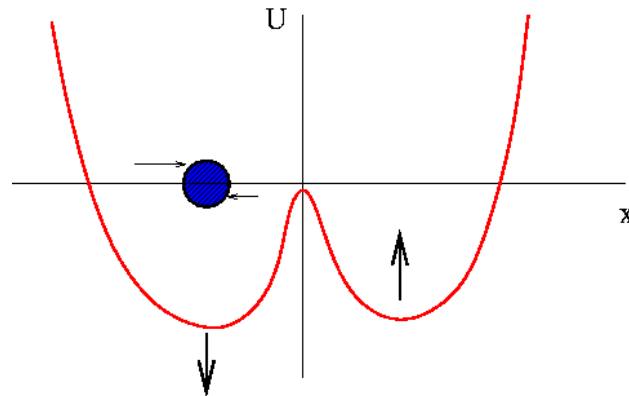
$$P(t) = \lambda e^{-\lambda t} \quad (\lambda \text{ is the mean collision time})$$

### Algorithm II. for making Brownian Dynamics:

1. Get initial position and velocity for the particle
2. Choose time intervals according to the above distribution
3. Integrate the equations of motion until the time of a stochastic collision.
4. Choose a momentum at random from the Maxwell-Boltzmann distribution at temperature T.
5. Proceed with step 3

# Exercises

1. Prove by computer simulations that the given recipe for  $R(t)$  leads to a Maxwell-Boltzmann distribution of the particles velocities (in 1D)
2. Study the motion of a particle in a harmonic potential and subject to a heat-bath at temperature  $T$ . (in 1D)
3. Study the motion of a particle in a  $W$  potential valley, in contact with a heat-bath at temperature  $T$ . Both parts of the  $W$  potential valley are harmonic.(in 1D)
4. Study problem nr. 3 when the two minimum of the  $W$  potential valley is modulated in anti-phase by a time-like harmonic component. Calculate the correlation function between the particle' s position and the external modulating field. (1D case) (the phenomenon of stochastic resonance)



# The Monte Carlo method

**Definition:** Monte Carlo methods use random sequence of numbers to calculate statistical estimates on a sample population for a desired parameter

**known examples:** calculating PI, calculating percolation thresholds ..

**other examples:** calculating average magnetization and energy for the Ising model

**in general:** applications are enormous and fascinating ....

## The outline of MC methods:

1. Description of the system in terms of a Hamiltonian
2. Selecting an appropriate ensemble for the problem
3. Observables are computed using an associated distribution function. Ultimately the goal is to compute quantities appearing as results of high-dimensional integrations on the state-space. The idea is to sample the main contributions to get an estimate for the observable.

**Starting point:** one dimensional Monte Carlo integration

# One dimensional Monte Carlo integration

**Problem:** given a function  $f(x)$ , compute the integral:

$$I = \int_a^b f(x) dx$$

The integral  $I$  can be computed by choosing  $n$  points  $(x_i)$  randomly on the  $[a,b]$  interval, and with a uniform distribution:

Straightforward sampling

$$I = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$$

The **strong law of large numbers** guarantees us that for a sufficiently large sample one can come arbitrary close to the desired integral!

Let  $x_1, x_2, \dots, x_n$  be random numbers selected according to a normalized probability density  $\mu(x)$ , then :

(!) *the above affirmation is also true if the random numbers are correlated, or the interval is finite*

$$I = \int_{-\infty}^{\infty} f(x) \mu(x) dx;$$
$$P\left(\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i) = I\right) = 1$$

**How rapidly the method converge? --> for  $\mu(x) = \text{const.}$  very badly!!!**

**Central limit theorem:** if:

$$\sigma^2 = \int_{-\infty}^{\infty} f^2(x) \mu(x) dx - I^2$$

then:

$$P\left(\left|\frac{1}{n} \sum_{i=1}^n f(x_i) - I\right| \leq \frac{\omega \sigma}{\sqrt{n}}\right) = \sqrt{2\pi} \int_{-\omega}^{\omega} \exp\left(-\frac{x^2}{2}\right) dx + O\left(\frac{1}{\sqrt{n}}\right)$$

- For straightforward sampling ( $\mu(x) = \text{const.}$ ) the error  $\sim 1/n^{1/2}$  !!
- The error is dependent on the choice of  $f(x)$  and  $\mu(x)$ ! --> influencing  $\sigma$

A better method for calculating

$$I = \int_a^b f(x) dx$$

$$I = \int_a^b \frac{f(x)}{p(x)} p(x) dx$$

$$\int_a^b p(x) dx = 1$$

We generate random  $x_1, x_2, \dots, x_n$  points according to the  $p(x)$  distribution

$$I = \frac{1}{n} \sum_{i=1}^n \frac{f(x_i)}{p(x_i)}$$

**The basic idea:** If we choose  $p(x)$  as close as possible to: we get  $\sigma \rightarrow 0$  and the method converges rapidly for small values of  $n$ !!!

$$p(x) = \frac{f(x)}{\int_a^b f(x) dx}$$

**Problem:** The method needs advance knowledge of  $I$ !

One way to overcome the problem is by guessing some  $p(x)$  functions, that mimics well the behavior of  $f(x)$ !. The error is also considerably reduced!

importance sampling

Sampling in the neighborhood where  $f(x)$  is large!

# Monte Carlo for statistical physics problems

We want to compute integrals like:

$f(\underline{x}) \rightarrow$  an appropriate ensemble distribution

$\underline{x} \rightarrow$  elements of the state-space

$\Omega \rightarrow$  the entire state-space

$H(\underline{x}) \rightarrow$  the Hamiltonian of the system

$$\langle A \rangle = \frac{1}{Z} \int_{\Omega} A(\underline{x}) f[H(\underline{x})] d\underline{x}$$
$$Z = \int_{\Omega} f[H(\underline{x})] d\underline{x}$$

Very high dimensional integral which is exactly computable only for a limited number of problems!!!

**Basic idea:** to use the **importance sampling** for calculating these integrals

If in the MC integration we choose the states with probability  $P(\underline{x}) \rightarrow$

$$\langle A \rangle = \frac{\sum_{i=1}^n A(\underline{x}_i) P^{-1}(\underline{x}_i) f[H(\underline{x}_i)]}{\sum_{i=1}^n P^{-1}(\underline{x}_i) f[H(\underline{x}_i)]}$$

By choosing

$$P(\underline{x}) = \frac{f[H(\underline{x})]}{Z} \quad \sigma \rightarrow 0 \quad \text{and thus the error} \rightarrow 0$$

$$\langle A \rangle = \frac{1}{n} \sum_{i=1}^n A(\underline{x}_i)$$

**Problem:** we still don't know Z!

## The Metropolis et al. idea...

An algorithm has to be derived that generates states according to the desired  $P(\underline{x})$ !

**Basic idea:** using a Markov chain, such that starting from an initial state  $\underline{x}_0$  further states are generated which are ultimately distributed according to  $P(\underline{x})$

For this Markov chain need to specify the  $W(\underline{x}, \underline{x}')$  transition probabilities from state  $\underline{x}$  to state  $\underline{x}'$ . In order that the limiting distribution be  $P(\underline{x})$  we need:

- 1. For all complementary pairs  $(S, S')$  of sets of phase points there exist  $\underline{x} \in S$  and  $\underline{x}' \in S'$  such that  $W(\underline{x}, \underline{x}') > 0$  (ergodicity)
- 2. For all  $\underline{x}$ :  $\sum_{\underline{x}'} W(\underline{x}, \underline{x}') = 1$
- 3. For all  $\underline{x}$ :  $\sum_{\underline{x}'} W(\underline{x}, \underline{x}') P(\underline{x}') = P(\underline{x})$  (existence of the limiting distribution)

Instead of 4. A stronger but simpler condition can be used, the so called **detailed balance**:

$$W(\underline{x}, \underline{x}') P(\underline{x}) = W(\underline{x}', \underline{x}) P(\underline{x}')$$

**Result:** We can construct Markov chains leading to the desired  $P(\underline{x})$  distribution, without the prior knowledge of  $Z$  !!!

**Example:** the canonical ensemble:

$$f[H(\underline{x})] \propto \exp\left[-\frac{H(\underline{x})}{k_B T}\right]$$

**Metropolis dynamics:**

$$W(\underline{x}, \underline{x}') = \exp[-\Delta E(\underline{x}, \underline{x}')/k_B T] \quad \text{if } \Delta E(\underline{x}, \underline{x}') > 0;$$

$$W(\underline{x}, \underline{x}') = 1 \quad \text{if } \Delta E(\underline{x}, \underline{x}') < 0 \quad \{\Delta E(\underline{x}, \underline{x}') = H(\underline{x}') - H(\underline{x})\}$$

**Glauber dynamics:**

$$W(\underline{x}, \underline{x}') = \exp[-\Delta E(\underline{x}, \underline{x}')/k_B T] / \{1 + \exp[-\Delta E(\underline{x}, \underline{x}')/k_B T]\}$$

$$\{\Delta E(\underline{x}, \underline{x}') = H(\underline{x}') - H(\underline{x})\}$$

Detailed balance  
satisfied

**Algorithm for Monte Carlo simulations:**

1. Specify an initial point  $\underline{x}_0$  in the phase space
2. Generate a new state  $\underline{x}$
3. Compute the  $W(\underline{x}, \underline{x}')$  transition probability
4. Generate a uniform random number  $r$  between  $[0, 1]$ .
5. If  $r < W$  --> jump to the new state, and return to 2.  
    If  $r \geq W$  --> count the old state as new and return 2.
6. Average the desired  $A$  quantity on all states after the initial  $N \gg 1$  "transient" states

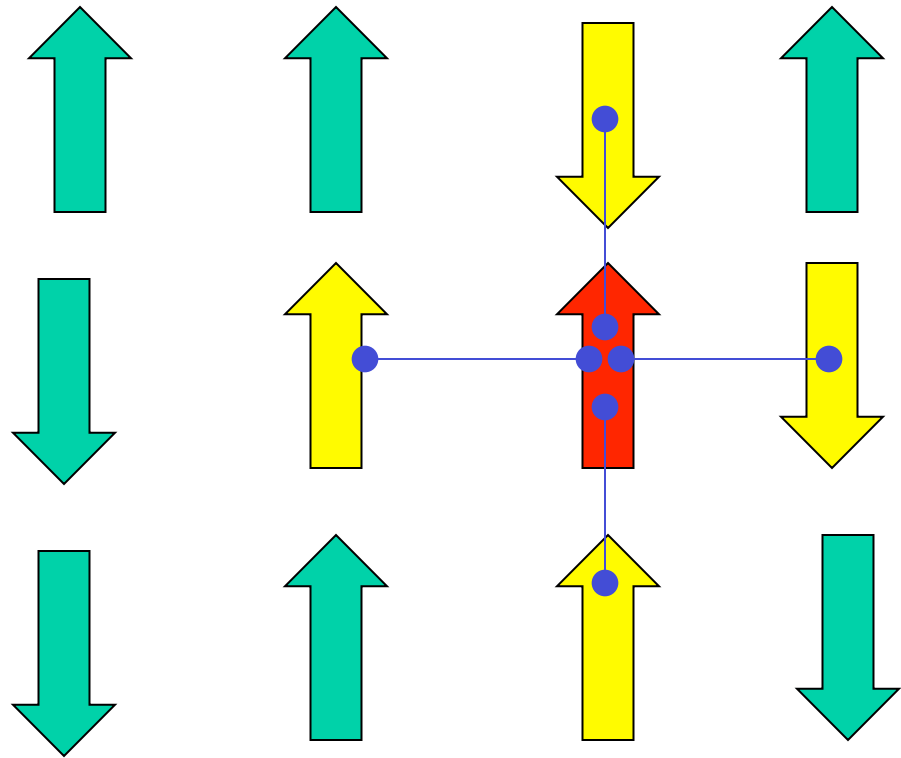
# The Ising model

$$H = -J \sum_{\langle i,j \rangle} S_i S_j - \mu B \sum_{\{i\}} S_i$$

$$S_i = \pm 1$$

Interaction with nearest neighbors only!

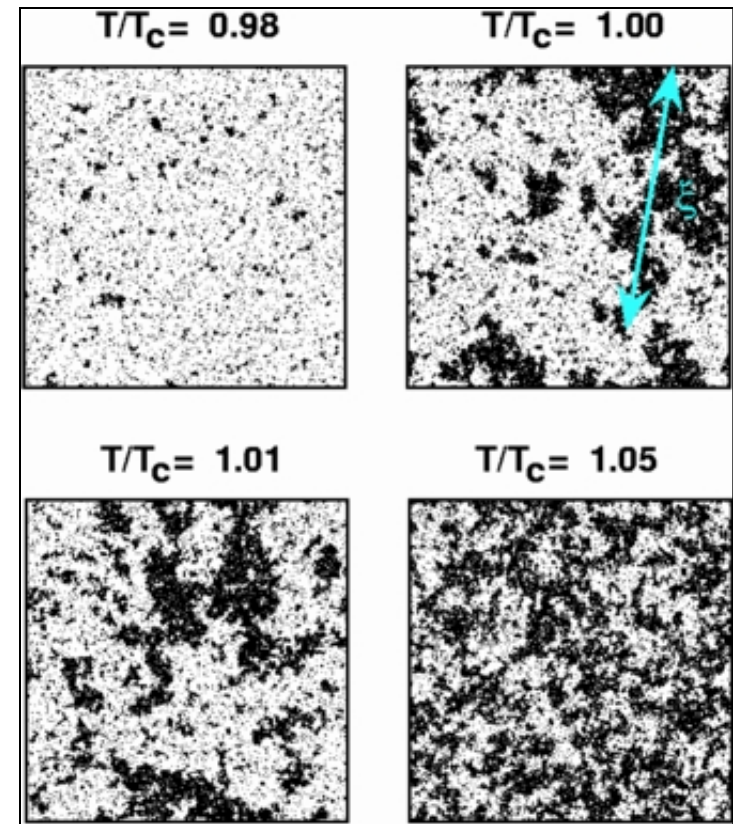
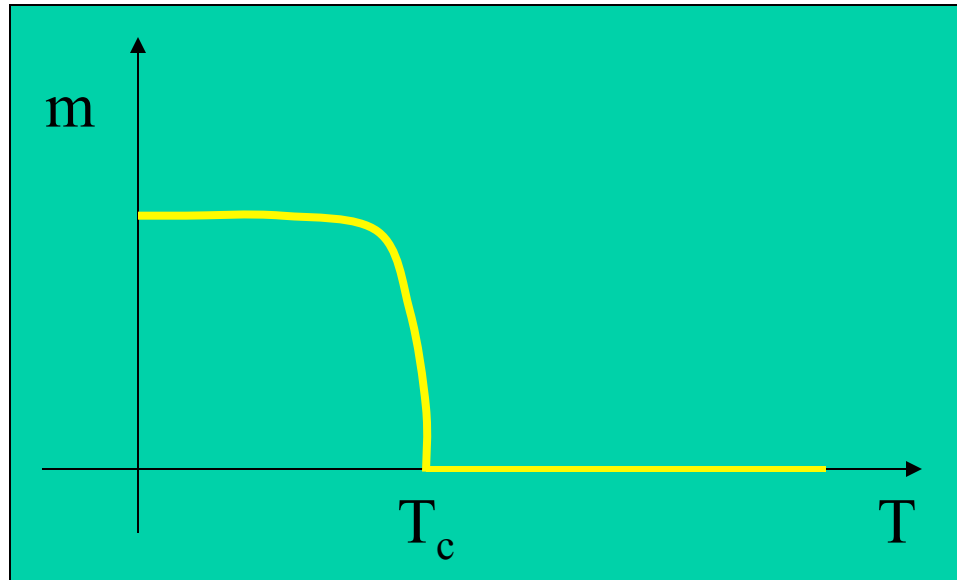
- spontaneous magnetization is possible ( $M \neq 0$  for  $B=0$ )
- first model for understanding ferro- and anti-ferromagnetism for localized spins
- for  $J>0$  --> **ferromagnetic** order
- for  $J<0$  --> **anti-ferromagnetic** order
- no phase transition in 1D
- ferro-paramagnetic phase transition for  $D>1$
- second order phase transition (order-disorder)



- In 1D and 2D exactly solvable!
- Due to the local interactions calculating  $Z$  is difficult.
- exact solution very difficult in 2D
- No exact solution in 3D
- **Approximation methods:** mean-field theory, renormalization, high and low temperature expansion

Order parameter:

$$m = |\langle S \rangle| = \frac{|M|}{N}$$



### Important quantities

- $m(T)$  curve
- $T_c$
- the critical exponent of susceptibility( $\gamma$ ), order parameter( $\beta$ ), specific heat( $\alpha$ ) and correlation length( $\nu$ )

### Exact results:

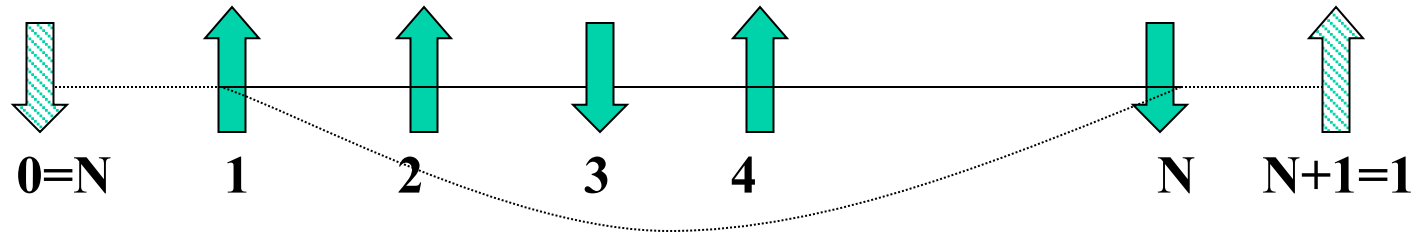
(1D)  $T_c=0$ ;

(2D)  $T_c=2.26918 J/k_B$  (square lattice);  $\beta=1/8$ ;  $\alpha=0$  (logarithmic divergence!);  $\gamma=7/4$ ;  $\nu=1$

(3D) no exact results (believed that:

$T_c= 4.44 J/k_B$  (square lattice);  $\alpha=1/8$ ;  $\beta=5/16$ ;  $\gamma=5/4$ ;  $\nu=5/8$ )

# The transfer matrix solution to the 1D Ising chain



$$H = -J \sum_{i=1}^N \sigma(i) \sigma(i+1) - h \sum_{i=1}^N \sigma(i) = - \sum_{i=1}^N \left[ J \sigma(i) \sigma(i+1) + \frac{h}{2} (\sigma(i) + \sigma(i+1)) \right]$$

$$Z = \sum_{\sigma(1)=\pm 1} \dots \sum_{\sigma(N)=\pm 1} \exp \left\{ \beta \sum_{i=1}^N \left[ J \sigma(i) \sigma(i+1) + \frac{h}{2} (\sigma(i) + \sigma(i+1)) \right] \right\} = \sum_{\{\sigma(i)\}} \prod_{i=1}^N \exp \left\{ \beta \left[ J \sigma(i) \sigma(i+1) + \frac{h}{2} (\sigma(i) + \sigma(i+1)) \right] \right\}$$

$$Z = \sum_{\{\sigma(i)\}} \langle \sigma(1) | \hat{P} | \sigma(2) \rangle \cdot \langle \sigma(2) | \hat{P} | \sigma(3) \rangle \dots \langle \sigma(N) | \hat{P} | \sigma(N+1) \rangle = \sum_{\{\sigma(1)\}} \langle \sigma(1) | \hat{P}^N | \sigma(1) \rangle = \text{Tr}(\hat{P}^N)$$

$$\begin{array}{|l} |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ | -1 \rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{array}$$

$$\hat{P} = \begin{bmatrix} \exp[\beta(J+h)] & \exp(-\beta J) \\ \exp(-\beta J) & \exp[\beta(J-h)] \end{bmatrix}$$

$$Z = \text{Tr}(\hat{P}^N) = \lambda_1^N + \lambda_2^N$$

$\lambda_{1,2}$  the eigenvalues of  $\hat{P}$ !

$$F = -k_B T \ln(Z) = -k_B T \ln(\lambda_1^N + \lambda_2^N) \approx -k_B T N \ln(\lambda_1) \quad \text{when } N \rightarrow \infty \text{ and } \lambda_1 > \lambda_2$$

$$F = -N k_B T \ln \left\{ e^{\beta J} \cosh(\beta h) + \sqrt{e^{2\beta J} \sinh^2(\beta h) + e^{-2\beta J}} \right\}$$

Exact results in 1D:

$$m = \langle \sigma(i) \rangle = -\frac{1}{N} \frac{\partial F}{\partial h} = \frac{\sinh(\beta h)}{\sqrt{\sinh^2(\beta h) + e^{-4\beta J}}}$$

For  $h=0$ :

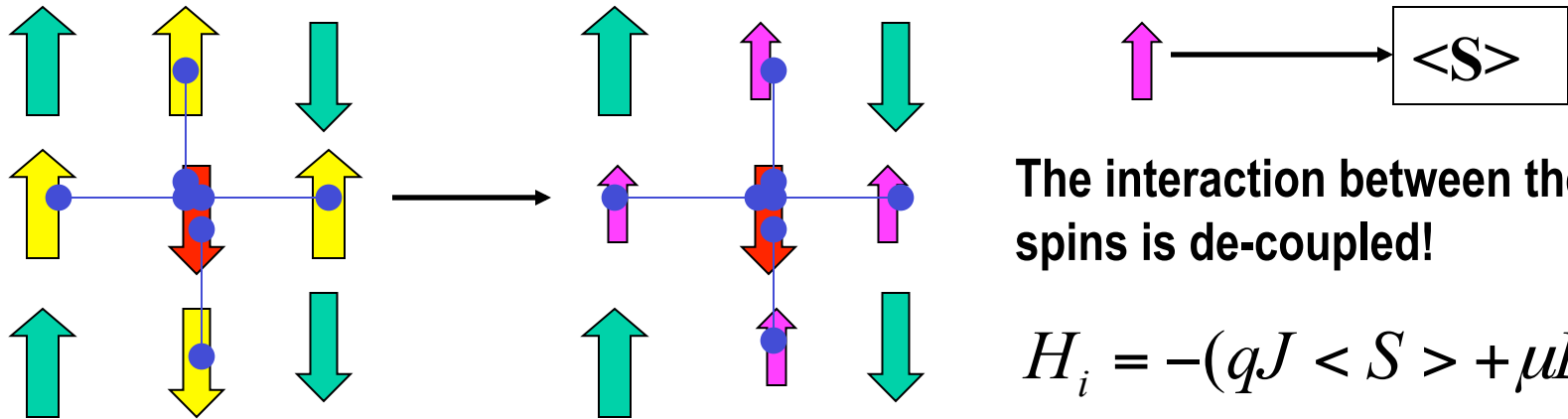
$$\langle \sigma(i)\sigma(i+j) \rangle = [\tanh(\beta J)]^j = e^{-\frac{j}{\xi}}$$

$\xi$  the correlation length  
 $\xi = -[\ln(\tanh(\beta J))]^{-1}$

no phase transition at  $T>0$  !!! ( $T_c=0$ )

For  $T \rightarrow 0$  we have that  $\xi \rightarrow \infty$

# The mean-field approximation of the Ising model



$$Z_i = \sum_{\{S_i\}} \exp[\beta(Jq \langle S \rangle + \mu B) S_i] = e^x + e^{-x} = 2 \cosh(x)$$

$$x = \beta(Jq \langle S \rangle + \mu B)$$

$$\langle S_i \rangle = 1P_+ + (-1)P_- = \frac{1}{Z_i} (e^x - e^{-x}) = \tanh(x)$$

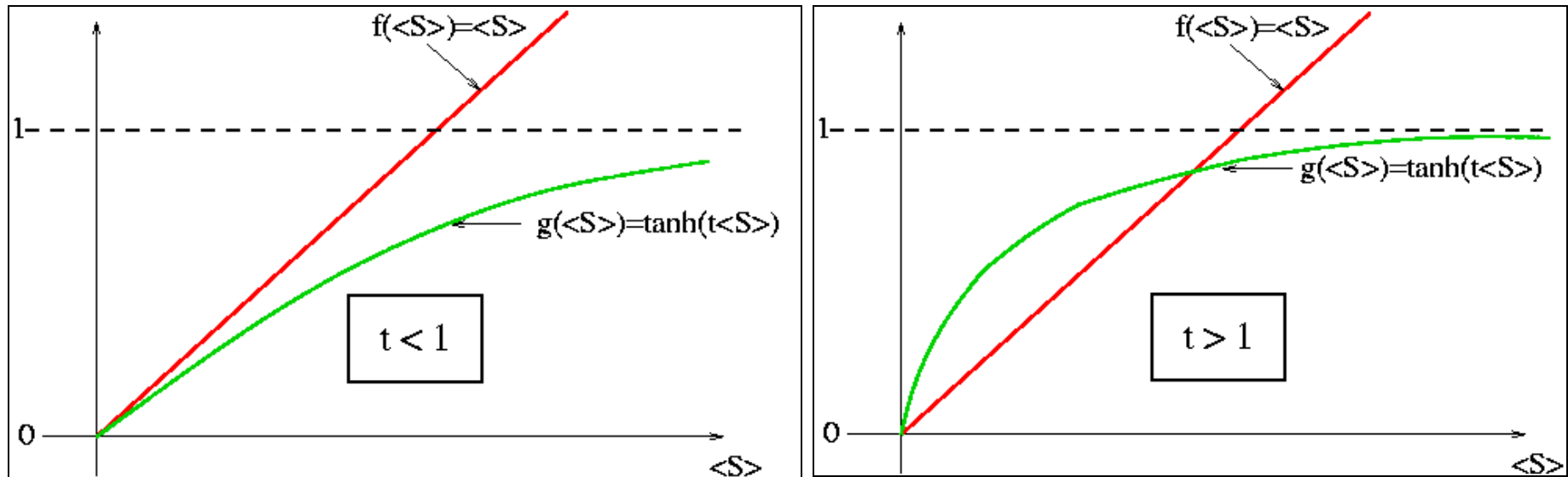
$$\langle S \rangle = \tanh[\beta(Jq \langle S \rangle + \mu B)] \quad \leftarrow \text{An implicit equation for } \langle S \rangle$$

for B=0

$$\langle S \rangle = \tanh(t \langle S \rangle)$$

$$t = \beta J q$$

a graphical solution:



- for  $t \leq 1$  the only possible solution:  $\langle S \rangle = 0$  --> paramagnetic behavior
- for  $t > 1$  two solutions;  $\langle S \rangle = 0$  (unstable solution) and  $\langle S \rangle > 0$  (stable solution) --> ferromagnetic behavior

- $t = 1$  the critical point -->

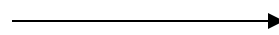
$$T_c = \frac{Jq}{k_B}$$

on the square lattice  $q=4$ ;

on the cubic lattice  $q=6$ ;

.....

in the neighborhood of  $T_c$  ( $T < T_c$ ):



$$\langle S \rangle = \sqrt{3} \left( \frac{T}{T_c} \right)^{\frac{3}{2}} \left( \frac{T_c}{T} - 1 \right)^{\frac{1}{2}}$$

$T_c = 4 J/k_B$  (square lattice) , exact:  $2.2692 J/k_B$

$= 6 J/k_B$  (cubic lattice), believed  $4.44 J/k_B$

$= 2 J/k_B$  (Ising chain), exact: 0!

$\beta = 1/2$ ; exact 2D:  $1/8$ ; believed 3D:  $0.31$

$\alpha = 0$ ; exact 2D: 0; believed 3D:  $0.12$

$\gamma = 1$ ; exact 2D:  $7/4$ ; believed 3D:  $1.25$

$\nu = 1/2$ ; exact 2D: 1; believed 3D:  $0.64$

-as the dimensionality of the problem increase, mean-field approaches become better and better!

**- mean-field is totally wrong in 1D !!**

# Implementing the Metropolis and Glauber Monte Carlo for the 2D Ising model

**Problem:** Study  $\langle m(T) \rangle$ ,  $\langle E(T) \rangle$ ,  $\langle C_v(T) \rangle$ ,  $\langle \chi(T) \rangle$  and  $T_c$  for 2D Ising models by using the Metropolis or Glauber algorithm.

We consider  $B=0$ , and fix  $J=1$ . The units are considered that  $k_B=1$ .

$$\boxed{\begin{aligned} H &= - \sum_{\langle i,j \rangle} S_i S_j \\ S_k &= \pm 1 \end{aligned}}$$

-Let us assume a lattice  $N \times N$  with **free** boundary conditions

-We consider a canonical ensemble and fix thus **N** and **T**

**We would like to calculate:**

$$\langle m(T) \rangle = \left\langle \frac{M}{N^2} \right\rangle = \left\langle \frac{\sum_i S_i}{N^2} \right\rangle = \frac{\left\langle \sum_i S_i \right\rangle}{N^2}$$

$$\langle C_v(T) \rangle = \frac{1}{N^2 k_B T^2} \left[ \langle E^2(T) \rangle - \langle E(T) \rangle^2 \right]$$

$$\langle \chi(T) \rangle = \frac{1}{N^2 k T} \left[ \langle M^2(T) \rangle - \langle M(T) \rangle^2 \right]$$

$$\langle E(T) \rangle = \langle H(\{S_i\}) \rangle = - \left\langle \sum_{\langle i,j \rangle} S_i S_j \right\rangle$$

$T_c$  will be determined from the maxima of  $\langle C_v(T) \rangle$  and  $\langle \chi(T) \rangle$

In order to get the desired quantities we have to calculate the following **NxN** dimensional sums (integrals):

$$\langle M(T) \rangle = \langle \sum_i S_i \rangle = \sum_{\{S_i\}} \left[ \left( \sum_i S_i \right) \frac{1}{Z} \exp \left( -\frac{H(\{S_i\})}{k_B T} \right) \right]$$

$$\langle M^2(T) \rangle = \langle \left( \sum_i S_i \right)^2 \rangle = \sum_{\{S_i\}} \left[ \left( \sum_i S_i \right)^2 \frac{1}{Z} \exp \left( -\frac{H(\{S_i\})}{k_B T} \right) \right]$$

$$\langle E(T) \rangle = - \langle \sum_{\langle i,j \rangle} S_i S_j \rangle = - \sum_{\{S_i\}} \left[ \left( \sum_{\langle i,j \rangle} S_i S_j \right) \frac{1}{Z} \exp \left( -\frac{H(\{S_i\})}{k_B T} \right) \right]$$

$$\langle E^2(T) \rangle = \langle \left( \sum_{\langle i,j \rangle} S_i S_j \right)^2 \rangle = \sum_{\{S_i\}} \left[ \left( \sum_{\langle i,j \rangle} S_i S_j \right)^2 \frac{1}{Z} \exp \left( -\frac{H(\{S_i\})}{k_B T} \right) \right]$$

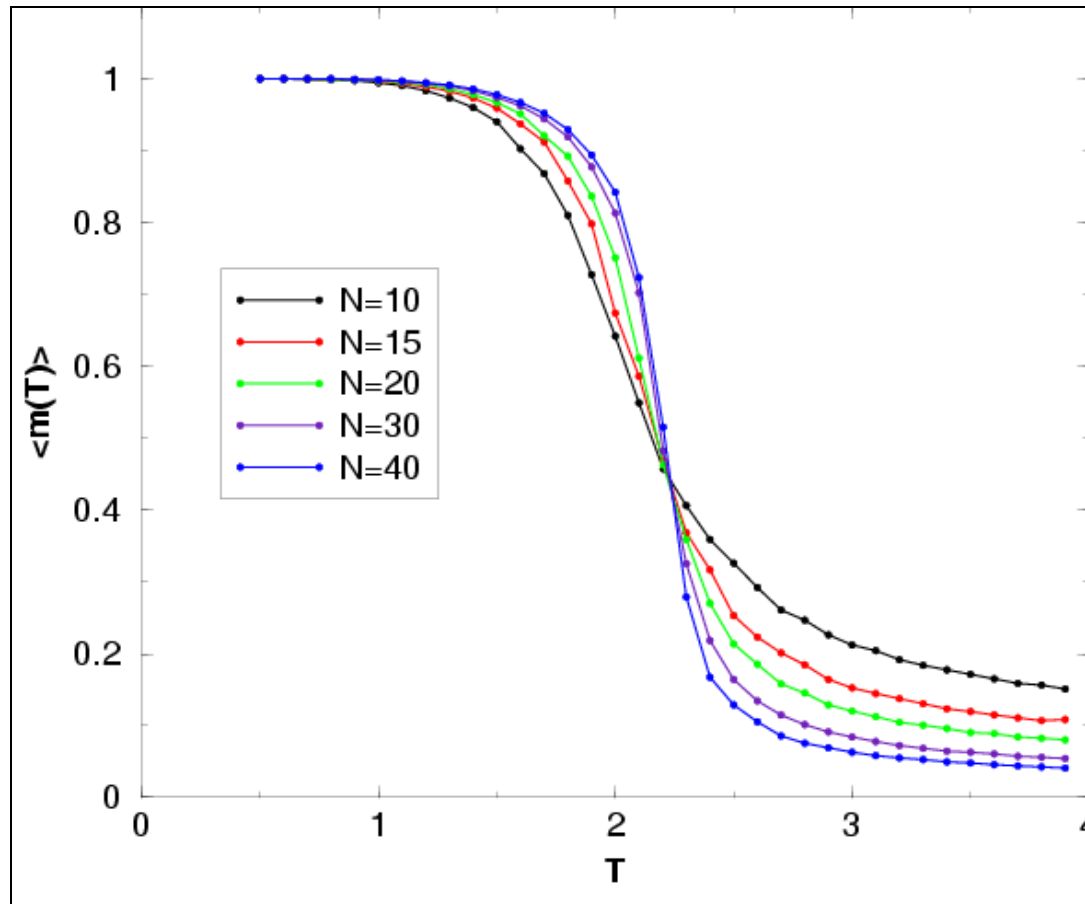
We will use the Metropolis MC method to calculate these sums (integrals)!

## The algorithm (Metropolis and Glauber MC for the Ising model):

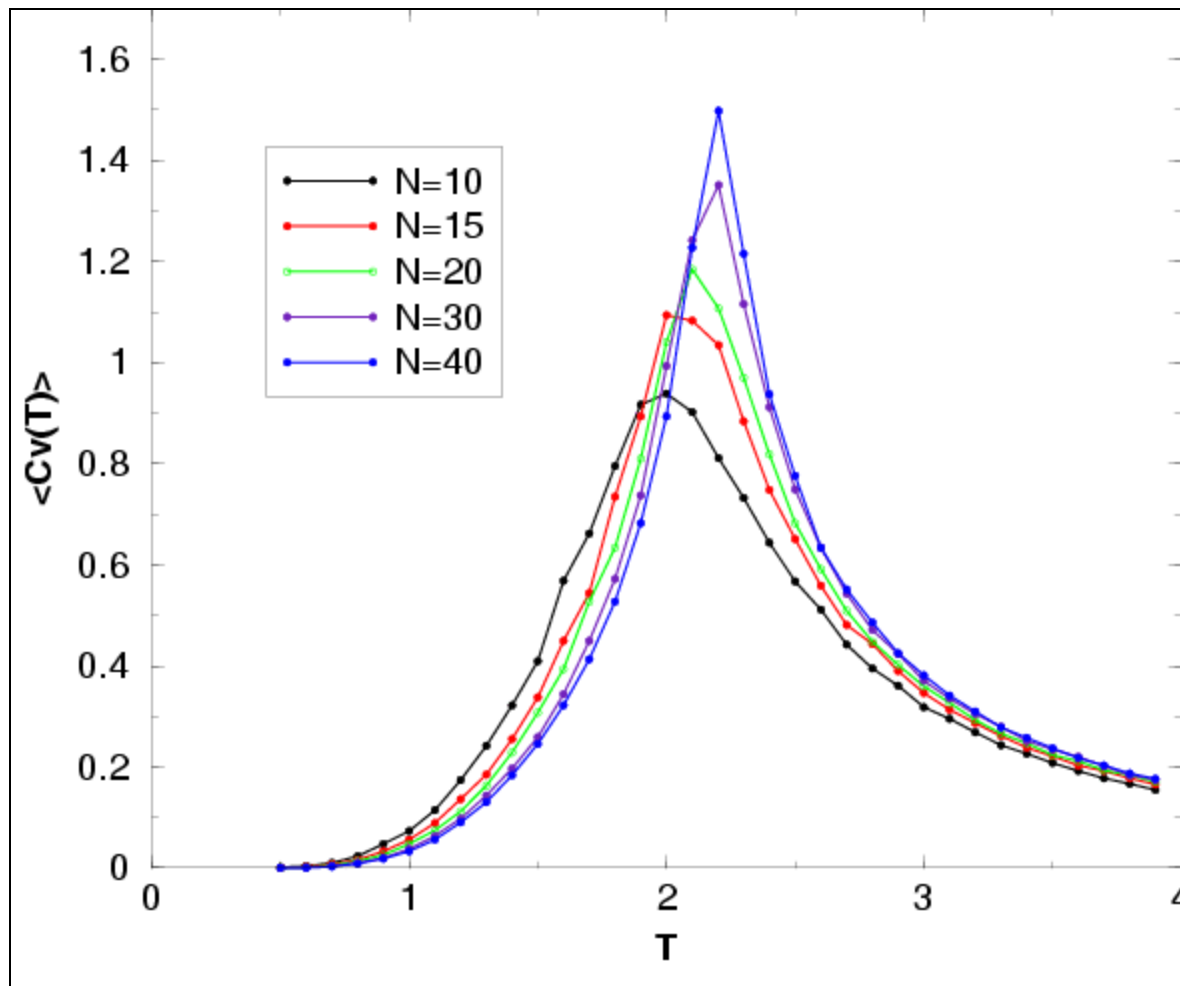
1. Fix a given temperature
2. Fix an initial spin configuration ( $\underline{x}$ )
3. Calculate the initial value of  $E$  and  $M$
4. Consider a new spin configuration by virtually “flipping” one randomly selected spin ( $\underline{x}'$ )
5. Calculate the energy  $E'$  of the new configuration, and the energy change due to this spin-flip
6. Calculate the Metropolis (Glauber)  $W(\underline{x} \rightarrow \underline{x}')$  probabilities for this move
7. Generate a random number “ $r$ ” between 0 and 1
8. If  $r \leq W(\underline{x} \rightarrow \underline{x}')$  accept the flip and update the value of the energy to  $E'$  and magnetization to  $M'$
9. Repeat the steps 4 - 8 many times (drive the system to the desired canonical distribution of the states)
10. Repeat the steps 4 -8 by collecting the values of  $E$ ,  $E^2$ ,  $M$ ,  $M^2$ , and calculate their average
11. Compute this average for a large number of steps
12. Calculate the value of  $\langle m(T) \rangle$ ,  $\langle E(T) \rangle$ ,  $\langle C_v(T) \rangle$  and  $\langle \chi(T) \rangle$  by the given formulas
13. Change the temperature and repeat the algorithm for the new temperatures as well.
14. Construct the desired  $\langle m(T) \rangle$ ,  $\langle E(T) \rangle$ ,  $\langle C_v(T) \rangle$ ,  $\langle \chi(T) \rangle$  curves

writing the code--> see the computer code

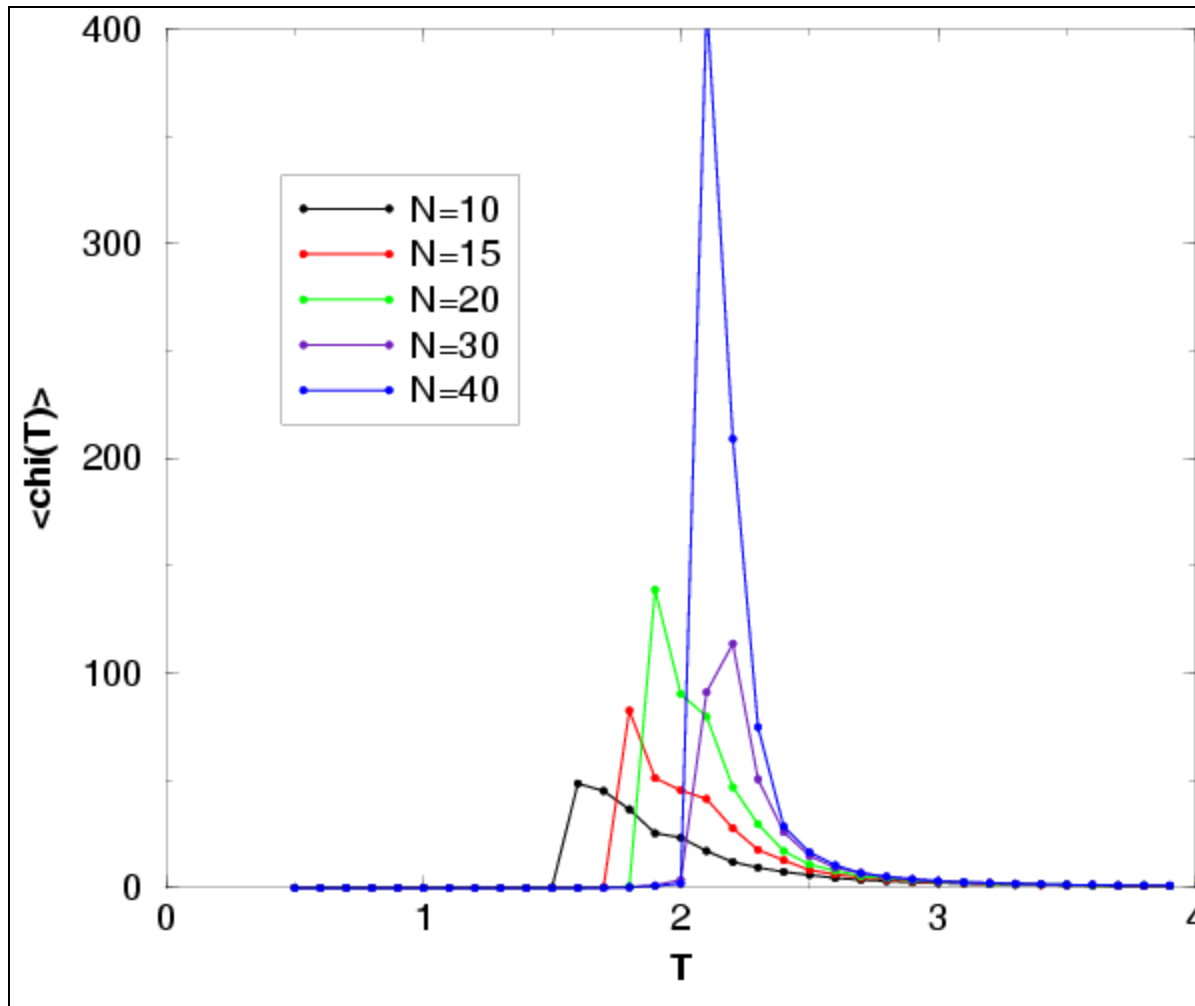
## A simple 2D Glauber dynamics code



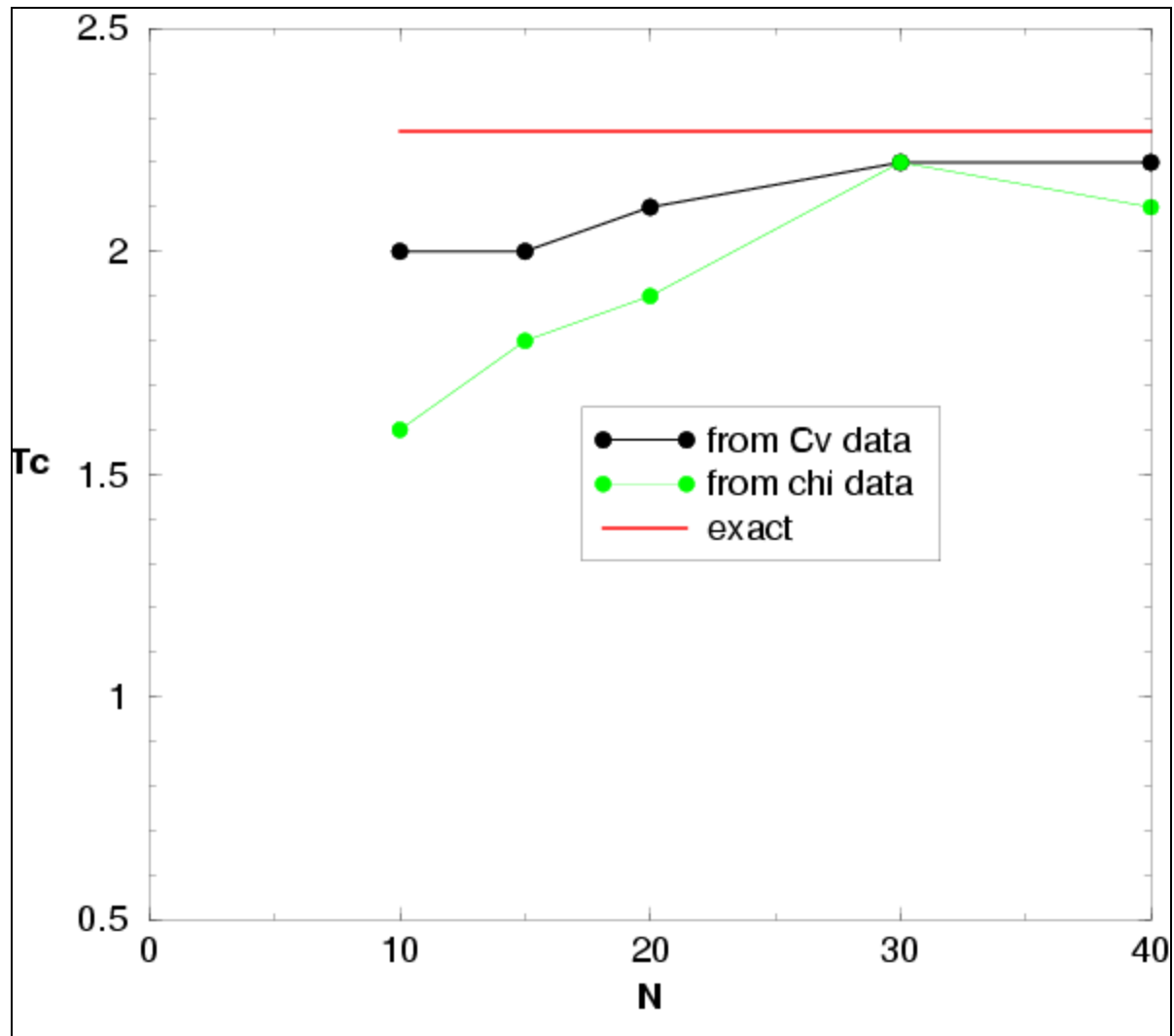
Variation of  $\langle m(T) \rangle$  as a function of  $T$  for different system sizes



Variation of  $\langle C_v(T) \rangle$  as a function of  $T$  for various system sizes



Variation of  $\langle \chi(T) \rangle$  as a function of  $T$  for various system sizes



Estimates for  $T_c$  for various system sizes (step in  $T$  is 0.1)

- the considered  $W(\underline{x} \rightarrow \underline{x}')$  transitions leads to an **ergodic Markov process**
- **one MC step** is defined as  $N \times N$  spin flip trials !
- By applying the above algorithm for  $T < T_c$  one can also follow how the order arises in the system. This dynamics might not necessarily be the “real one”. The Metropolis MC method is intended to yield equilibrium properties and not dynamical simulation of the system!
- It is believed that the Glauber transition probabilities gives a realistic picture for the dynamics as well!
- Influence of finite lattice size is strong --> finite size effects (finite lattice size cuts the correlation length!, no real phase transitions, and no real divergences!) --> important **problem of scaling**: get the desired quantities for the  $N \rightarrow \infty$  limit.
- One way of making the system quasi-infinite is to impose **periodic boundary conditions**. (see the exercise in the computer codes!)

# The BKL Monte Carlo (or: kinetic MC, or resident time MC) method

## Used for:

- computing quickly equilibrium properties at low temperatures in the Metropolis or Glauber algorithm
- simulating jump-like stochastic processes with  $\exp(-c \Delta E)$  type activation probability.  
*examples:* - diffusion of atoms on crystal substrates
  - grain growth process
  - dynamics of defects in crystals
  - dynamics of spins at low temperature

Reference: Bortz, Kalos, Lebowitz; *J. Computational Physics*, vol. 17, pp. 10 (1975)

## Difficulties that are solved by the BKL method:

1. The Metropolis and Glauber dynamics is very ineffective at low temperatures where  $\exp(-\beta \Delta E) \ll 1$  !! (too many rejected steps, where nothing is changed in the system)

(at low temperatures usually  $\Delta E$  and  $\beta$  are both big!)

2. For jump-like stochastic processes the Glauber and Metropolis dynamics is also ineffective due to the largely different time-scales that are present in real problems

(more concrete examples in this sense later ....)

# The BKL method for low temperature Ising systems

- When we want to compute the specific heat or the susceptibility at low temperatures we need to follow the fluctuation of the energy or magnetization for many MC steps
- At low temperature the Metropolis or Glauber algorithm is very ineffective since the system most of the time waits...and nothing happens.
- A quick solution: make a transition at each flip-attempt and update the “resident time” of each state (needed in the average) according to the probability of each transition.
- We give now the algorithm and make the theoretical basis later.

1. Set the time  $t=0$ .
2. Form a list of the Metropolis or Glauber rates  $r_i$  (probabilities) for all the possible  $W(\underline{x} \rightarrow \underline{x}')$  transitions (spin flips) in the system.
3. Calculate the cumulative function  $R_i = \sum_{j=1}^i r_j$  for  $i=1,2,\dots,N$ , where  $N$  is the total number of possible transitions in the system (let  $R=R_N$ )
4. Get a uniform random number  $u \in [0,1)$ .
5. Find the event  $i$  to carry out by finding the value of  $i$  for which  $R_{i-1} < uR \leq R_i$
6. Carry out event  $i$ .
7. Recalculate all  $r_i$  values that might have been changed due to the transition
8. Get a new uniform random number  $u \in [0,1)$ .
9. Update the time (simulation step, or weight in the average) by  $t=t+\Delta t$  where  $\Delta t = -\log(u)/R$
10. return to step 2.

# Motivation for the basic algorithm

- Let us presume we have a system with 3 possible transitions:

$w(1) \rightarrow$  with rate (probability)  $r_1=0.1$ ;  $w(2) \rightarrow$  with rate (probability)  $r_2=0.4$

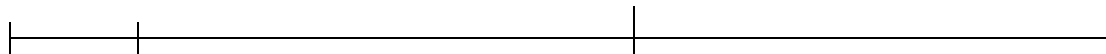
$w(3) \rightarrow$  with rate (probability)  $r_3=0.5$

- In the basic Metropolis (or Glauber) algorithm we make the transitions by attempts with the given probabilities. The probability to choose a given transition from the possible three transition is proportional to the values of  $r_i$ .

- With  $r_1$ ,  $r_2$ , and  $r_3$  we get the cumulative functions  $R_1$ ,  $R_2$  and  $R_3$ :

$R_1=r_1=0.1$ ;  $R_2=r_1+r_2=0.5$ ;  $R_3=r_1+r_2+r_3=1$

- We can plot now the values of  $r_i$  as regions, and  $R_i$  as points on a line as follows



- If we generate a random number  $u \in [0, 1)$  and multiply it by  $R_3=1$ , this will correspond to one point on the line. The probability to get transition  $w(2)$  will be proportional with the distance between  $R_1$  and  $R_2$ , which is  $r_2/R \rightarrow$  this is what we wanted to do!

- At each simulation step we make than a transition, and the rates at which different transitions occur is the desired one!

**Problem:** how to update the “time” in order to get the good values of the averages

# Motivation for the equation of time

-It is easy to determine the  $f_i(t)$  probability, that the transition  $w_i$  with rate  $r_i$  didn't occurred up to time  $t$ :

$$df_i(t) = -r_i f_i(t) dt \longrightarrow f_i(t) = C \exp(-r_i t)$$

The  $C=1$  constant results from the  $f(0)=1$  condition  $\longrightarrow f_i(t) = \exp(-r_i t)$

- the individual transitions form a Poisson process, i.e. the probability to have a transition in a time  $dt$  is linearly proportional with the length of the interval  $dt$ .
- a useful feature of Poisson processes is that a large number  $N$  of Poisson processes, with rates  $r_i$ , will behave as another large Poisson process with properties similar to the single processes. --> denoting by  $F(t)$  the probability that no  $w_i$  transition occurs up to time  $t$ , we get:

$$F(t) = \exp(-Rt) \quad \text{with} \quad R = \sum_i r_i$$

The probability to have the first transition between time  $t \in [0, \infty)$  and  $t+dt$  is

In order to have time steps satisfying this probability distribution, we need to generate thus  $q$  random numbers according to the  $g(t) = R \exp(-Rt)$

distribution! This can be done with the known

method, after generating uniformly distributed  $u$

random numbers (see random number generators):

$$q = -\frac{\ln(1-u)}{R} = -\frac{\ln(u')}{R}$$

# Simulating dynamic processes

## 1. Simulating grain-growth by using a T=0 temperature Potts- model (a trivial application)

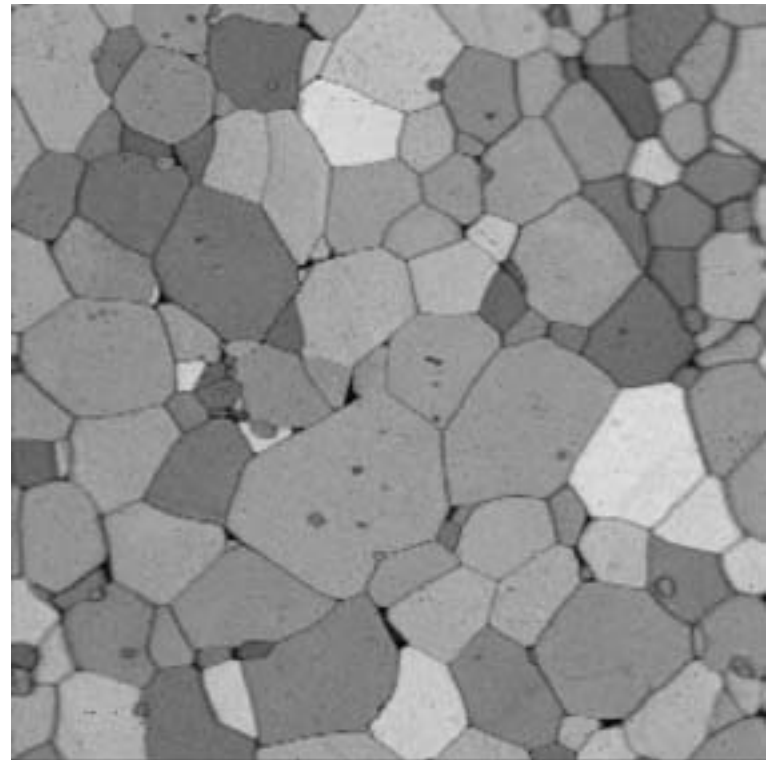
Grain-growth in metals: growth of the size of the mono-crystalline domains. Small domains are “eated up” by larger ones, the grain-boundary moves. Experimentally observed that the  $\langle d \rangle$  mean-grain size increases as:

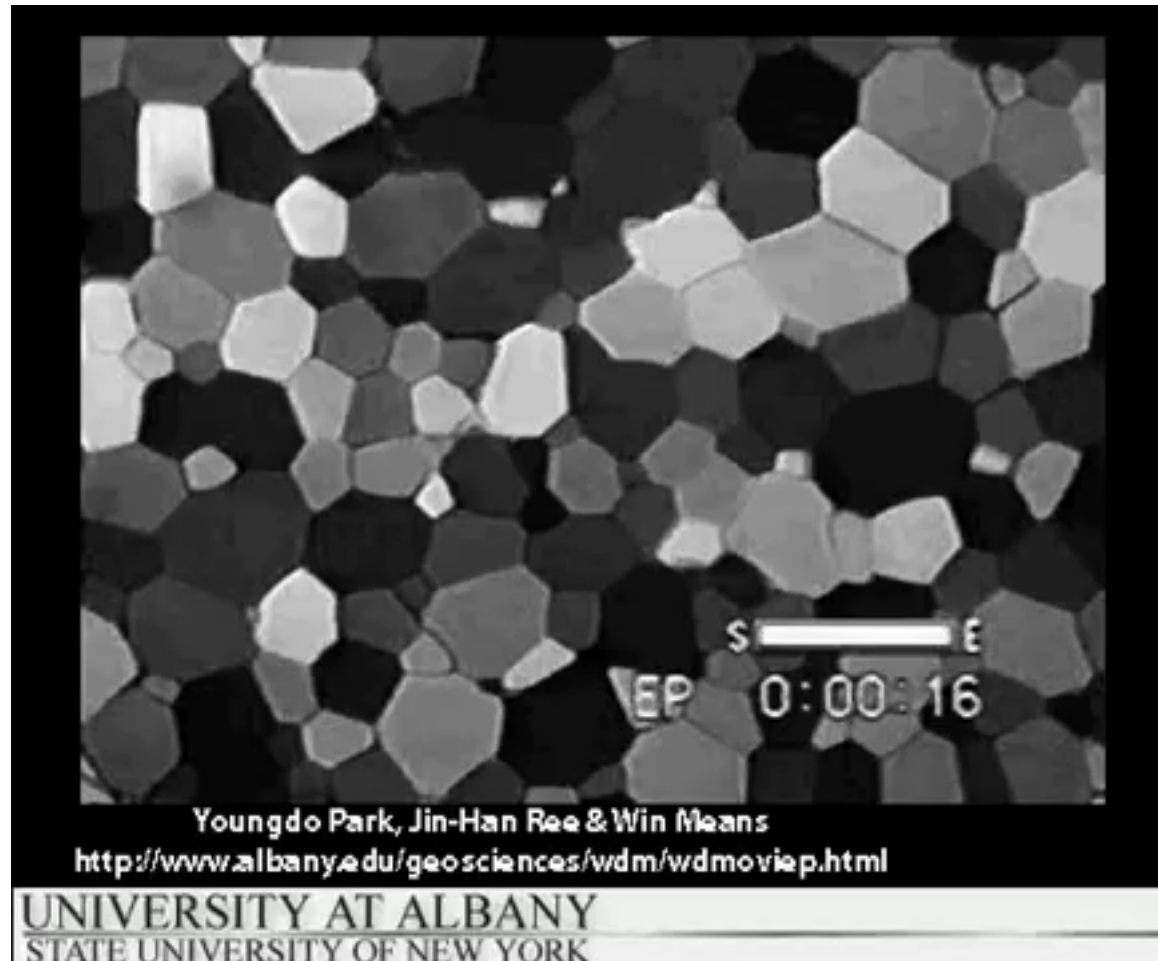
$$\langle d(t) \rangle \sim (\langle d(0) \rangle^2 + at)^{1/2}$$

The Potts model:

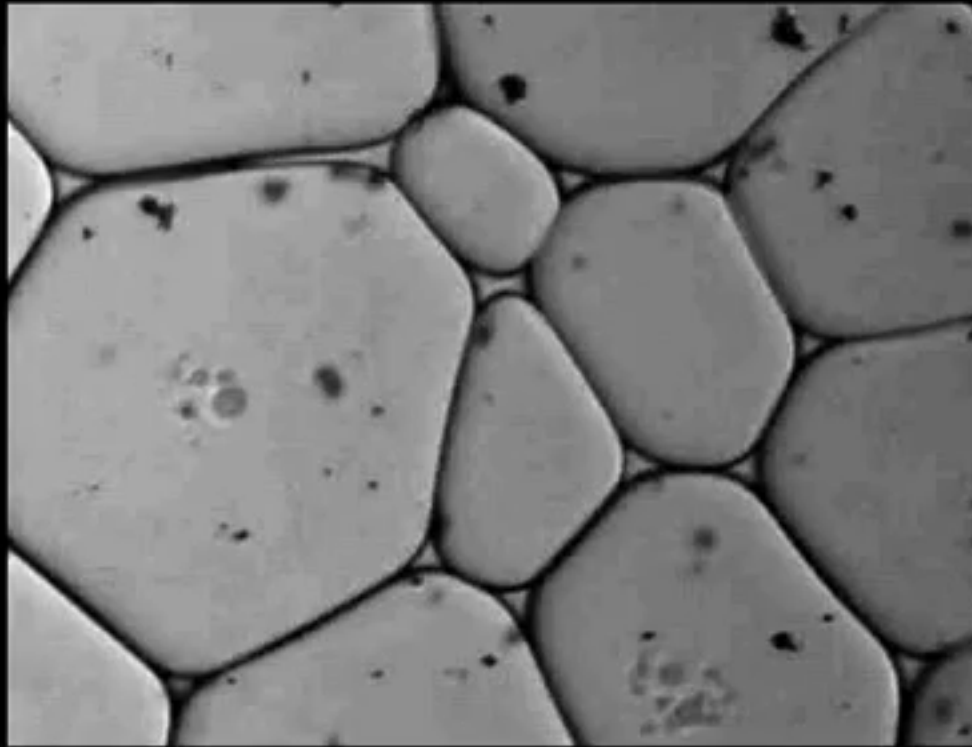
$$H = -q \sum_{\langle i,j \rangle} \delta_{\sigma(i), \sigma(j)}$$
$$\sigma(i) = 1, 2, \dots, q$$

- we use a square or triangular lattice
- we start with a large q number of initial states
- each lattice site will have a randomly chosen Potts variable
- Make a Metropolis or Glauber dynamics at T=0 (accept only steps that do NOT increase the energy of the system)





Grain growth in octachloropropane



Youngdo Park, Jin-Han Ree & Win Means

<http://www.albany.edu/geosciences/wdm/wdmoviep.html>

UNIVERSITY AT ALBANY  
STATE UNIVERSITY OF NEW YORK

Grain growth in octachloropropane  
in a partially melt case

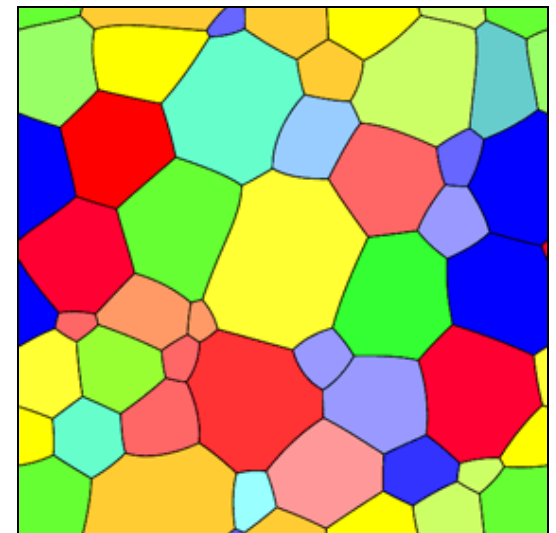
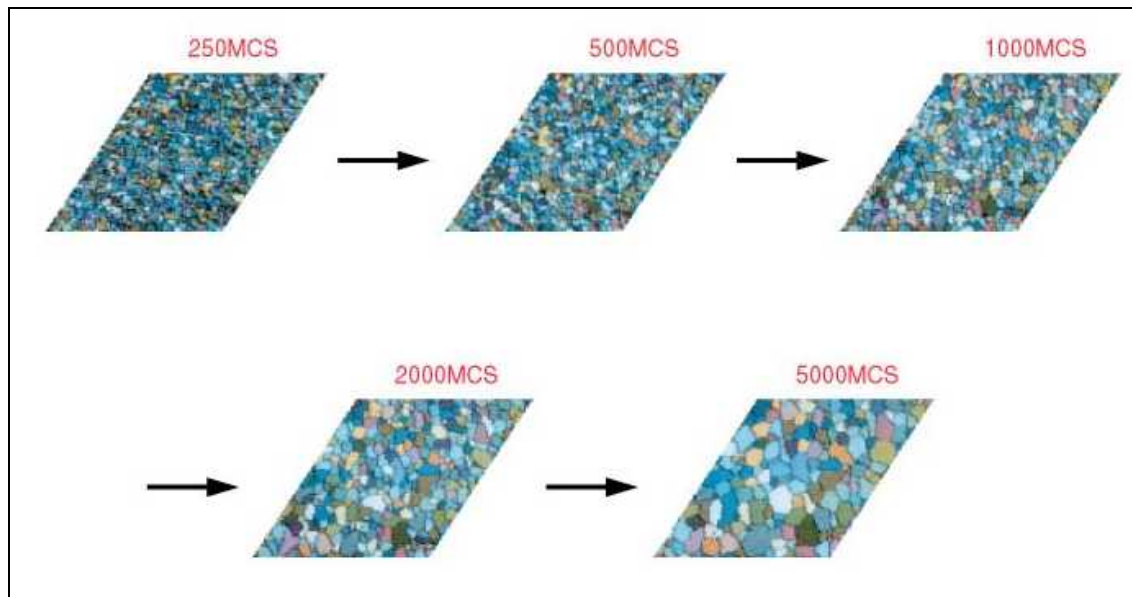
- In the classical Metropolis or Glauber algorithm, one would randomly select between all the lattice sites, and accept moves according to the Metropolis or Glauber transition rates--> assuming the used Metropolis or Glauber dynamics real, the elapsed time here would be proportional with the number of attempts...one MC step (trial equal with the number of spins) would be considered as a  $\tau$  unit time.

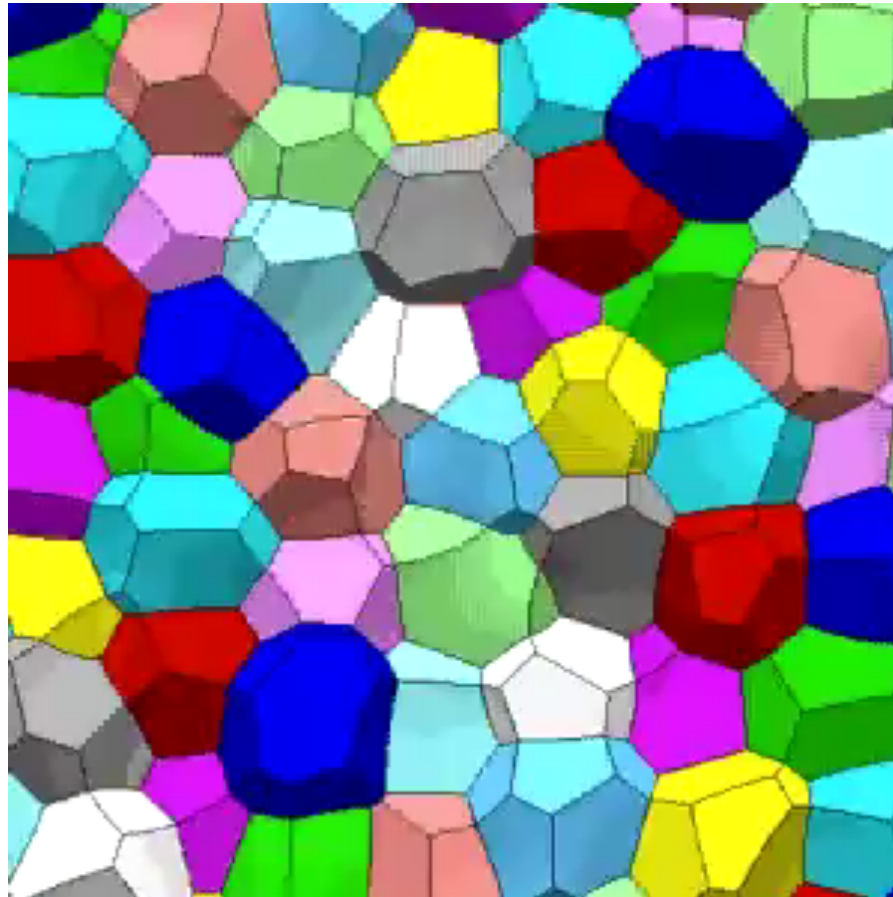
-In the BKL algorithm at  $T \rightarrow 0$  temperature, we visit ONLY the GRAIN BOUNDARIES and calculate for these sites the  $r$  transition rates to the Potts variable of the neighboring states. The transition rates in the bulk and other Potts states are assumed to be zero. We select one site according to the BKL scheme and flip it to a neighboring state. The time, however has to be updated according to the BKL scheme!

- The update in time after each GB transition is:

- The increase of  $\langle d(t) \rangle$  follows the expected form!

$$dt = -\frac{\ln(u)}{R} \tau$$

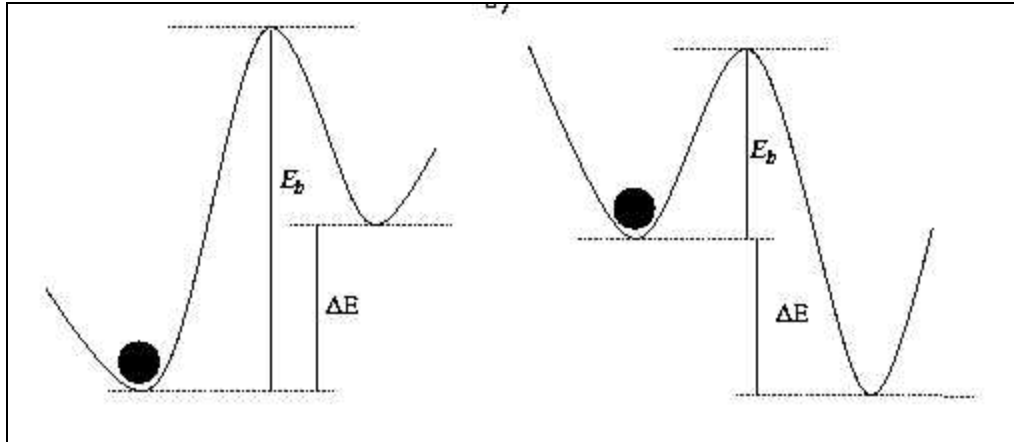




3D kMC simulation of grain-growth

## 2. Simulating dynamics of atoms on crystal surfaces, or/and the deposition of atoms on substrates

- atoms deposited on surfaces can jump from one site to a nearby site, after escaping from a potential barrier (bounding energy to nearby atoms)



### The escape probability

$$P \sim \exp\left(-\frac{E_b}{kT}\right)$$

### The transition rate

$$r = w \exp\left(-\frac{E_b}{kT}\right)$$

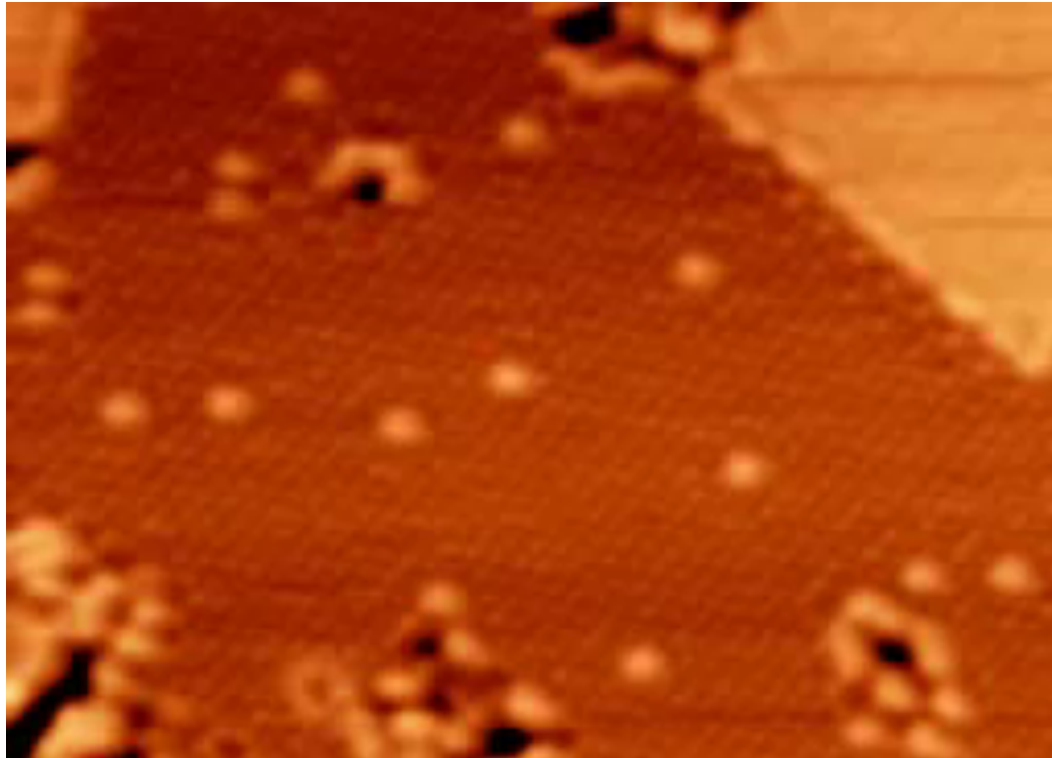
$w$ : is the attempt frequency per unit time, or the frequency of the vibration (of order  $10^{12}$ - $10^{13}$ Hz)

- the bounding energy is proportional with the number of it' s nearest neighbors.
- usually there is also a deposition process coexisting with the diffusion one, with a rate  $r_{\text{dep}}$  (number of atoms deposited per unit time per unit site)
- many times there is a co-deposition or co-diffusion of more types of atoms, let' s say A and B.
- **problem**: to simulate the dynamics of the deposited atoms, and the formed structures.

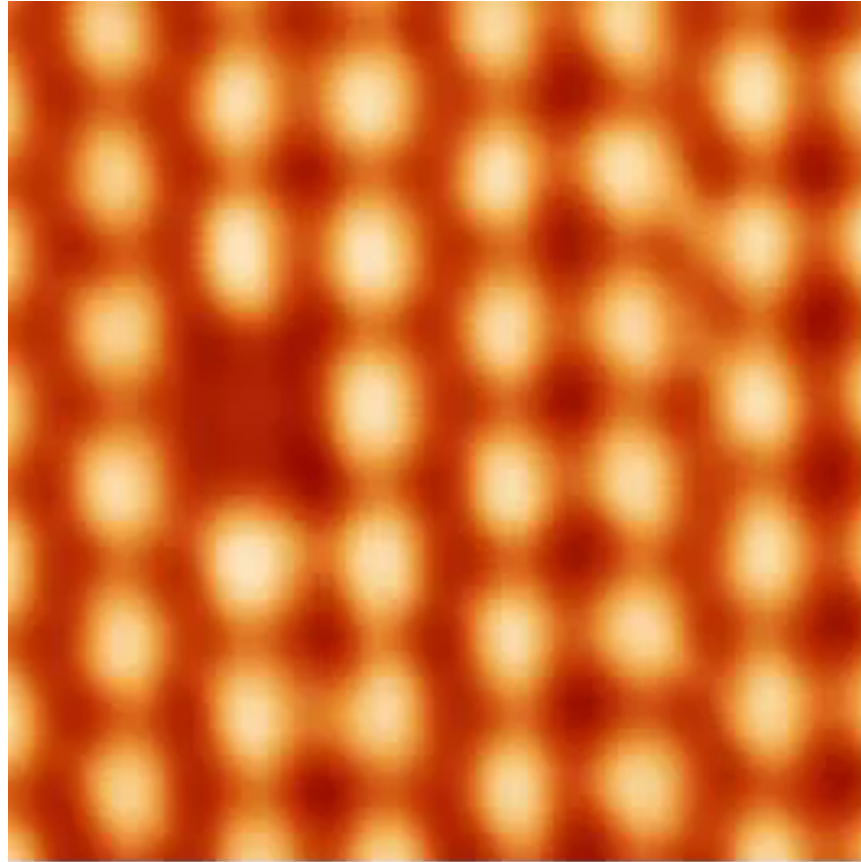
**Main difficulty**: the largely different time-scales (transition rates) for the possible stochastic processes --> makes the simulation very lengthy in time



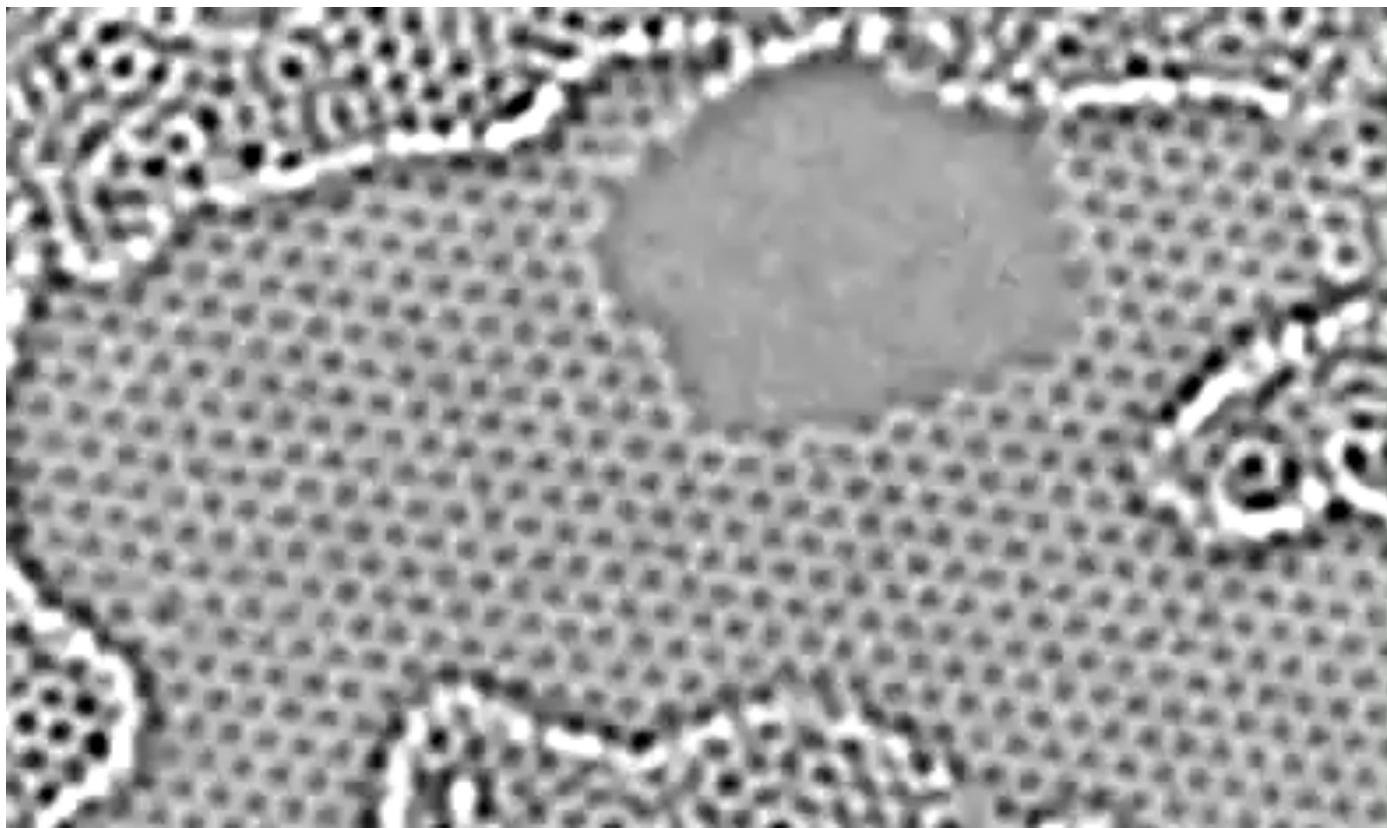
In Situ Real time observation of thin film deposition: Growth of a Polycrystalline film



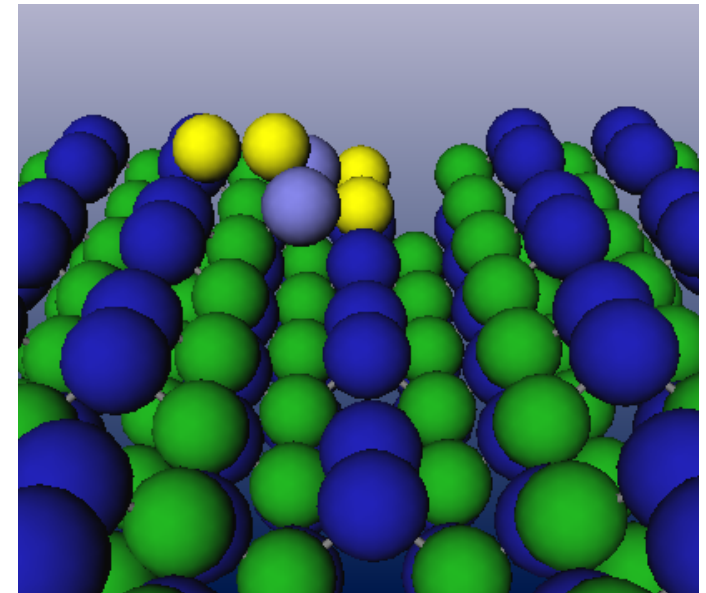
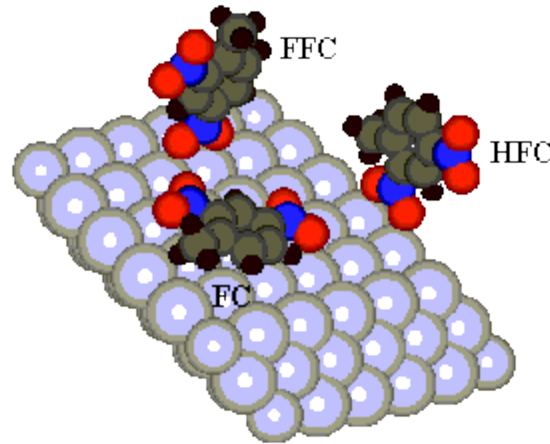
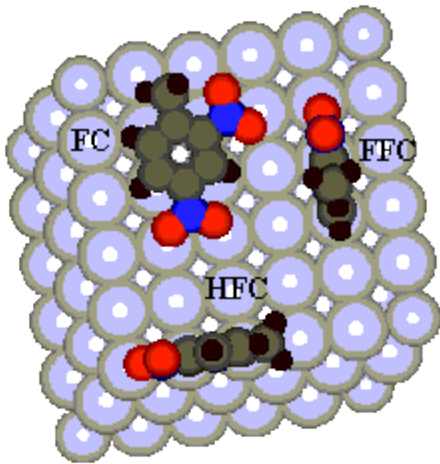
Pb atoms moving on a Si(111) surface at low Temperatures (STM movie)



Atoms moving in a germanium surface (STM movie)



Carbon atoms moving at the edge of a hole in graphene



### A classical algorithm:

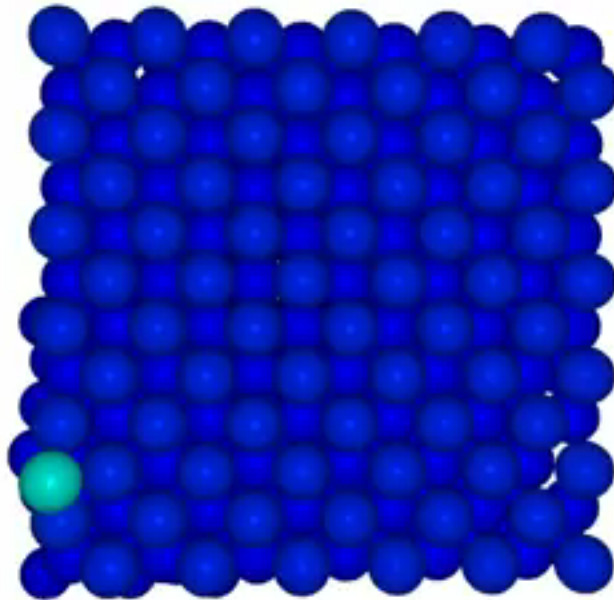
- consider a lattice for the substrate
- fix the interaction constants (fix the bounding energies to nearby atoms), fix the temperature and deposition rates
- calculate the transition rates for different situations (different species and different number of nearest neighbors)
- update the time in units  $\tau$ , smaller than the smallest period for the transitions or deposition
- at each time-step randomly choose as many sites as many atoms are on the substrate, and attempt their transition to a nearby site with the probability:  $P_i = \tau * r_i$
- at each time-step deposit atom on a randomly chosen site with probability  $P_{\text{dep}} = \tau * r_{\text{dep}}$
- continue the simulation until the desired time

**Problem and solution:** due to the largely different transition rates (effect of the exponential in the transition rate formula) the simulation is very slow, and ineffective. Most of the time nothing happens --> can be much improved by the BKL method!

Step:0

Barrier:0.680465119998

Time:6.61601237816e-05



Kinetic Monte Carlo simulation of AG  
growth on AG surfaces

Case study:

Diffusion and formation of Pt atom clusters on  
Pt(111) surfaces.

# Cluster Algorithms (Swendsen and Wang MC method and the Wolff algorithm)

- it is not a single spin flip, but a cluster flip algorithm
  - in one simulation step we flip a cluster of spins
  - these clusters becomes very large as  $T$  decreases and approaches  $T_c$
  - it is very effective in the neighborhood of  $T_c$
- eliminates the **critical slowing down** in the neighborhood of  $T_c$
- very appropriate to study the critical exponents

**1. R.H. Swendsen, J-S. Wang**, “Nonuniversal critical dynamics in Monte Carlo simulations”, Phys. Rev. Lett. 58, 86–88 (1987)

Abstract: *A new approach to Monte Carlo simulations is presented, giving a highly efficient method of simulation for large systems near criticality. The algorithm violates dynamic universality at second-order phase transitions, producing unusually small values of the dynamical critical exponent.*

**2. U. Wolff**, Collective Monte Carlo Updating for Spin System, Phys. Rev. Lett. 62, 361–364 (1989)

Abstract: *A Monte Carlo algorithm is presented that updates large clusters of spins simultaneously in systems at and near criticality. We demonstrate its efficiency in the two-dimensional  $O(n)$  sigma models for  $n=1$  (Ising) and  $n=2$  (x-y) at their critical temperatures, and for  $n=3$  (Heisenberg) with correlation lengths around 10 and 20. On lattices up to  $128^2$  no sign of critical slowing down is visible with autocorrelation times of 1-2 steps per spin for estimators of long-range quantities.*

# Spin-spin correlation and the correlation length

The correlation between spin  $S_0$  and spin  $S_n$  is measured by:

$$\langle S_0 S_n \rangle$$

(the average is an ensemble average). If two spins are uncorrelated  $\langle S_0 S_n \rangle = 0$ .

- At  $T=0$  the spins are lined up in one direction  $\rightarrow \langle S_0 S_n \rangle = 1$  (this is a trivial correlation, because flipping  $S_0$  will hardly affect  $S_n$ , unless if it is not a neighbor of  $S_0$ )
- Near  $T_c$  the situation is very different, the spins are constantly changing, but not independently  $\rightarrow$  there are large domains of parallel spins which persist for long periods of time. Thus, spins far apart from one another are strongly correlated
- At high temperatures, the spins fluctuate rapidly but almost independently of each other.
- To describe the real spin correlation (and eliminate the trivial correlation) we define the **pair correlation function**

$$g_{0n} = \langle S_0 S_n \rangle - \langle S_0 \rangle \langle S_n \rangle$$

- In general  $g_{0n}$  depends only on the distance between the two spins:  $\mathbf{g} \rightarrow \mathbf{g}(\mathbf{r})$  (for large systems,  $r$  can be considered a continuous variable. For  $r \gg a$  (lattice spacing) we have:

$$g(r) \sim \frac{\exp(-r/\xi)}{r^{d-2+\eta}} \quad (\xi(T) \text{ is the correlation length}) \quad \xi(T) \text{ diverges at } T_c \text{ as } \rightarrow \quad \xi(T) \sim \frac{1}{(T - T_c)^\nu}$$

At  $T = T_c$  we obtain thus

$$g(r) \sim \frac{1}{r^{d-2+\eta}}$$

$\eta$  is another critical exponent ( $\eta = 1/4$ ) for the 2D Ising model

# The critical slowing down (a first picture)

- The Ising model does not have real dynamics built into it (there is no kinetic energy term associated with the spins, so there is no real time evolution)
- In a real dynamical system, the dynamical variables are functions of time. The relaxation time characterizes the time scale over which the system approaches equilibrium. If  $A(t)$  is a quantity that relaxes towards its equilibrium value  $A_0$ , the relaxation time can be characterized as:
$$\tau = \frac{\int_0^\infty t |A(t) - A_0| dt}{\int_0^\infty |A(t) - A_0| dt}$$
- The Monte Carlo method generates successive spin configurations
- Although there is no real time evolution the MC simulation process might be viewed as kind of dynamics...
- We can follow how a configuration of spins relaxes to another configuration as the simulation goes by ---> this defines a relaxation time
- The relaxation time is linked to the correlation length by the **dynamical critical exponent**,  $z$ :
$$\tau \sim \xi^z$$
 as  $T \rightarrow T_c$  we have  $\xi \rightarrow \infty$  and get that  $\tau \rightarrow \infty$
- The big problem: for the Metropolis or the Glauber dynamics  $z=2$  !!! ---> the relaxation time diverges more quickly than the correlation length ---> **critical slowing down**
- There is necessary many MC steps to generate independent (uncorrelated configurations) --> the sampling is restricted only to a small portion of the state-space
- The system has a long memory....

# The relaxation time in Metropolis MC simulations --> the autocorrelation time

- The relaxation time in the Metropolis MC characterizes how many MC steps to skip in order to generate statistically independent configurations.
- If the relaxation time is of the order of a single MC step, every configuration can be used in measuring averages
- If the relaxation time is longer, than approximately  $\tau$  MC steps should be discarded between every point.
- to define a relaxation time we first define the autocorrelation function for a quantity A:

$$c_{AA}(k) = \langle A_n A_{n+k} \rangle - \langle A_n \rangle^2$$

A<sub>n</sub> it' s value at the MC step: n

- if this autocorrelation function decays exponentially

$$c_{AA}(t) \sim \exp(-t / \tau_{\text{exp}})$$

which defines the **exponential correlation (relaxation) time** ( $\tau_{\text{exp}}$ )

- starting from the equalities from below one can suggest another correlation time:

$$\int_0^\infty c_{AA}(t) dt = \int_0^\infty c_{AA}(0) e^{-t/\tau} dt = \tau c_{AA}(0)$$

$$\longrightarrow \tau_{\text{int}} = \frac{1}{2} + \sum_{k=1}^\infty \frac{c_{AA}(k)}{c_{AA}(0)}$$

$\tau_{\text{int}}$ : **integrated correlation time**;

$\longrightarrow$

This is what we determine usually in computer simulations

## Critical slowing down (a second look)

- for finite lattices when  $T \rightarrow T_c$  we get  $\xi \rightarrow L$  (size of the lattice)  $\rightarrow$  no real divergence

- we get thus:  $\tau \sim L^z$

- for the Metropolis and Glauber dynamics  $z=2$ , and we get  $\tau \sim L^2$   
(Metropolis, Glauber and BKL are *local* algorithms....)

and the simulation is very inefficient on large lattices (on small lattices on the other hand there are important finite-size effects!)

- good news: the value of the dynamical critical exponent is NOT universal, it depends on the MC algorithm (paper of Swendsen and Wang!)

- the problem: elaborate a MC method for which the value of the dynamical critical exponent is smaller!  $\rightarrow$  these must be *non-local* algorithms!

### Cluster algorithms:



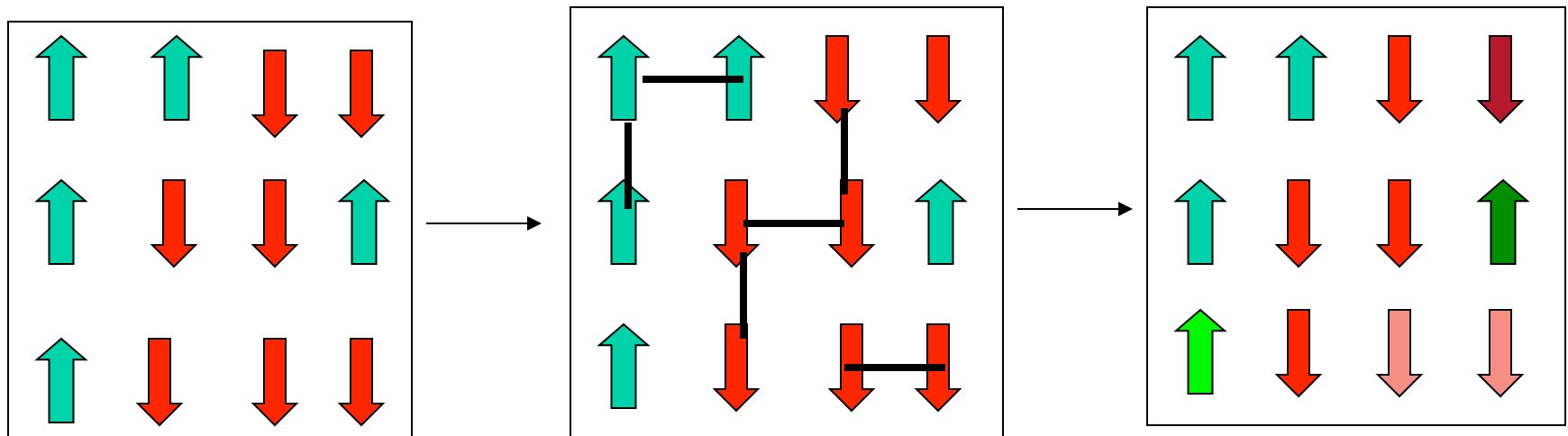
- we flip together all correlated spins.
- in each flip we generate statistically independent configurations.
- the value of the dynamic exponent becomes as low as  $z=0.15$  !

# The Swendsen and Wang cluster algorithm for $J>0$ Ising model

- **the basic idea**: is to identify the clusters of like and correlated spins and treat the clusters as a giant spin, flipping it according to a random criterion.
- it is necessary that the algorithm should lead to an ergodic Markov process and the detailed balance condition is satisfied!
- the algorithm can be generalized for arbitrary ( $J>0$  or  $J<0$ ) Potts models

## Construction of the clusters of correlated spins:

- the simple clusters of like nearest neighbor spins are NOT the clusters of correlated spins, these are too large... (At  $T=\infty$  there are still spins with like orientation, although the correlation between them in this case should vanish)
- the way of constructing the clusters of correlated spins, is to put a link between nearest neighbors and like spins, with a probability  $p=1-\exp(-2J/k_B T)$ .



## Flipping the clusters:

- all the clusters are flipped with probability  $1/2$ ! (we assign a new common value,  $\pm 1$  to all spins in the cluster)
- the spins in the whole lattice are in this manner updated!
- the algorithm satisfies detailed balance --> appropriate for important sampling

## The Swendsen and Wang algorithm for the 2D Ising model:

1. consider a lattice of spins with size  $N \times N$
2. fix the parameters ( $T, J=1, k_B=1$ )
3. consider an initial configuration of the spins
4. put “virtual bonds” with probability  $1 - \exp(-2J/k_B T)$  between nearest neighbor and like spins
5. construct the clusters of correlated spins
6. “flip” the clusters with probability  $1/2$  (this is one MC step)
7. get the new configuration of the spin system, and count it in the calculation of the desired averages
8. Identify like nearest neighbor spins and repeat the algorithm starting from 4.

**Main difficulty:** --> the construction of the clusters of correlated spins  
need of clever and fast cluster identification algorithms

**An example code** --> program nr. 12 (identification of clusters with **recursion**)

# The Wolff single cluster algorithm

- even more efficient than the S-W algorithm
- difference: constructing and flipping only one cluster at a time!
- the way of constructing the correlated spins cluster is the same as in the S-W algorithm

## The basic of the Wolff algorithm:

- choose a spin randomly in the lattice
- construct the cluster of correlated spins starting from this spin as a “seed”, by connecting nearest neighbor and like spins with a probability  $p=1-\exp(-2J/k_B T)$ . Do this process recursively until the cluster cannot grow more.
- flip this cluster of correlated spins (this will be one MC step)
- update the time proportionally with the number of flipped spins
- count the new configuration in the average of the desired quantities

- The Wolff algorithm is more effective, because we construct only one cluster and always flip it, + the probability to choose a cluster is proportional with the size of the cluster --> we will usually flip bigger clusters --> we generate statistically independent configurations!

- An example program is given as code nr. ?????

- A simple visual program to compare the effectiveness of the Metropolis, BKL and Wolff algorithm,:

[LMC.exe](#)

## Explaining the Wolf algorithm

In order to generate these configurations with the appropriate weight (and optimal speed), the Monte Carlo algorithm moves (in one iteration) from configuration  $a$  to configuration  $b$  with probability  $P(a \rightarrow b)$ . This transition probability is chosen to satisfy the fundamental condition of detailed balance

$$\pi(a)P(a \rightarrow b) = \pi(b)P(b \rightarrow a) \quad (1)$$

which is implemented using the Metropolis algorithm

$$P(a \rightarrow b) = \min \left( 1, \frac{\pi(b)}{\pi(a)} \right) \quad (2)$$

or one of its variants.

For the prototypical Ising model, the stationary probability distribution (the statistical weight) of a configuration is the Boltzmann distribution with an energy given by

$$E = -J \sum_{\langle i,j \rangle} S_i S_j \quad L > 0 \quad (3)$$

An essential concept is the one of an a priori probability: it accounts for the fact that the probability  $P(a \rightarrow b)$  is a composite object, constructed from the probability of *considering* the move from  $a$  to  $b$ , and the probability of *accepting* it.

$$P(a \rightarrow b) = \underbrace{\mathcal{A}(a \rightarrow b)}_{\text{consider } a \rightarrow b} \times \underbrace{\tilde{P}(a \rightarrow b)}_{\text{accept } a \rightarrow b}$$

In usual Monte Carlo terminology, if  $a \rightarrow b$  is rejected (after having been considered), then the ‘move’  $a \rightarrow a$  is chosen instead and the system remains where it is.

With these definitions, the detailed balance condition eq. (1) can be written as

$$\frac{\tilde{P}(a \rightarrow b)}{\tilde{P}(b \rightarrow a)} = \frac{\pi(b)}{\mathcal{A}(a \rightarrow b)} \frac{\mathcal{A}(b \rightarrow a)}{\pi(a)}$$

and implemented by a Metropolis algorithm generalized from eq. (2):

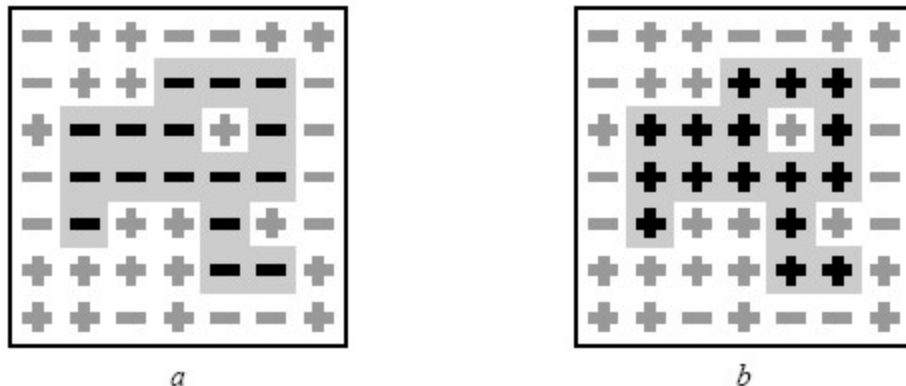
$$\tilde{P}(a \rightarrow b) = \min \left\{ 1, \frac{\pi(b)}{\mathcal{A}(a \rightarrow b)} \frac{\mathcal{A}(b \rightarrow a)}{\pi(a)} \right\} \quad (4)$$

It is very important to realize that the expression “a priori probability  $\mathcal{A}(a \rightarrow b)$ ” is synonymous to “Monte Carlo algorithm”. A Monte Carlo algorithm  $\mathcal{A}(a \rightarrow b)$  of our own conception must satisfy three conditions:

1. It must lead the state of the system from a configuration  $a$  to a configuration  $b$ , in such a way that, eventually, all configurations in phase space can be reached (ergodicity).
2. It must allow to compute the ratio  $\pi(a)/\pi(b)$ . This is trivially satisfied, at least for classical systems, as the statistical weight is simply a function of the energy.
3. It must allow, for any possible transition  $a \rightarrow b$ , to compute both the probabilities  $\mathcal{A}(a \rightarrow b)$  and  $\mathcal{A}(b \rightarrow a)$ . Again, it is the ratio of probabilities which is important.

The local spin-flip Monte Carlo algorithm not being satisfactory, it would be much better to move large parts of the system, so called clusters. This cannot be done by a blind flip of one or many spins (with  $\mathcal{A}(a \rightarrow b) = \mathcal{A}(b \rightarrow a)$ ), which allows unit acceptance rate both for the move  $a \rightarrow b$  and its reverse  $b \rightarrow a$  only if the energies of both configurations are the same. One needs an algorithm whose a priori probabilities  $\mathcal{A}(a \rightarrow b)$  and  $\mathcal{A}(b \rightarrow a)$  soak up any differences in statistical weight  $\pi(a)$  and  $\pi(b)$ .

This can be done by starting the construction of a cluster with a randomly sampled spin and by iteratively adding neighboring sites of the same magnetization with a probability  $p$ . To be precise, one should speak about ‘links’: if site  $i$  is in the cluster and a neighboring site  $j$  is not, and if, furthermore,  $S_i = S_j$ , then one should add link  $\langle i, j \rangle$  with probability  $p$ . A site is added to the cluster if it is connected by at least one link. In configuration  $a$  of Fig. 3 the cluster construction has stopped in the presence of 9 links “--” across the boundary. Each of these links could have been accepted with probability  $p$ , but has been rejected. This gives a term  $(1 - p)^9$  in the a priori probability. Flipping the cluster brings us to configuration  $b$ . The construction of the cluster for configuration  $b$  would stop in the presence of 19 links “++” across the boundary (a priori probability  $\propto (1 - p)^{19}$ ).



This allows us to compute the a priori probabilities

$$\begin{aligned}
\mathcal{A}(a \rightarrow b) &= \mathcal{A}_{\text{interior}} \times (1 - p)^9 \\
\mathcal{A}(b \rightarrow a) &= \mathcal{A}_{\text{interior}} \times (1 - p)^{19} \\
E_a &= E_{\text{interior}} + E_{\text{exterior}} - 9 \times J + 19 \times J \quad (\pi_a = \exp[-\beta E_a]) \\
E_b &= E_{\text{interior}} + E_{\text{exterior}} - 19 \times J + 9 \times J \quad (\pi_b = \exp[-\beta E_b])
\end{aligned}$$

In these equations, the ‘interior’ refers to the part of the cluster which does not touch the boundary. By construction, the ‘interior’ and ‘exterior’ energies and a priori probabilities are the same for any pair of configurations  $a$  and  $b$  which are connected through a single cluster flip.

We thus dispose of all the information needed to evaluate the acceptance probability  $\tilde{P}$  in eq. (4), which we write more generally in terms of the number of “same” and of “different” links in the configuration  $a$ . These notions are interchanged for configuration  $b$  (in Fig. 3, we have  $n_{\text{same}} = 9$ ,  $n_{\text{diff}} = 19$ ). With the energy scale  $J$  set to 1, we find

$$\begin{aligned}
\tilde{P}(a \rightarrow b) &= \min \left\{ 1, \frac{e^{\beta n_{\text{diff}}} e^{-\beta n_{\text{same}}}}{(1 - p)^{n_{\text{same}}}} \frac{(1 - p)^{n_{\text{diff}}}}{e^{-\beta n_{\text{diff}}} e^{\beta n_{\text{same}}}} \right\} \\
&= \min \left\{ 1, \left[ \frac{e^{-2\beta}}{1 - p} \right]^{n_{\text{same}}} \left[ \frac{1 - p}{e^{-2\beta}} \right]^{n_{\text{diff}}} \right\} \tag{5}
\end{aligned}$$

Once the cluster construction stops, we know the configuration  $b$ , may count  $n_{\text{same}}$  and  $n_{\text{diff}}$ , and evaluate  $\tilde{P}(a \rightarrow b)$ . Of course, a lucky coincidence<sup>1</sup> occurs for  $p = 1 - \exp[-2J\beta]$ . This special choice yields a rejection-free algorithm whose acceptance probability is unity for all possible moves and is implemented in the celebrated Wolff cluster algorithm [1], the fastest currently known simulation

# The Histogram Monte Carlo technique

*A.M. Ferrenberg, R.H. Swendsen; Phys. Rev. Lett., vol. 61, 2635 (1988)*

Idea: During normal Metropolis or SW Monte Carlo procedures a big quantity of information is lost!!! ( we calculate only the averages  $\langle E \rangle$ ,  $\langle E^2 \rangle$ ,  $\langle M \rangle$  and  $\langle M^2 \rangle$ )

- the distribution functions  $f(E)$  and  $g(M)$  can be however also obtained, and they carry a lot of important statistical information

$$f(E)dE = P(E, E + dE)$$

$$g(M)dM = P(M, M + dM)$$

- with the knowledge of  $f(E)$  and  $g(M)$  at a temperature  $T$  we can then calculate averages at other temperatures WITHOUT making ANY additional simulations at the desired temperature.

## Basics of the method

Let  $P(E_i, T_0)$  be the probability to get in the equilibrium at temperature  $T_0$ , a configuration with energy  $E_i$

$$P(E_i, T_0) = \frac{1}{Z(T_0)} D(E_i) \exp\left(-\frac{E_i}{k_B T}\right)$$

(corresponding to the used canonical ensemble)

$D(E_i)$  is the number of possible microstates for  $E_i$  (independent of temperature)

During the MC simulation at temperature  $T_0$  we can construct a histogram characterizing  $P(E_i, T_0)$ .

$$P(E_i, T_0) = \frac{h(E_i, T_0)}{\sum_j h(E_j, T_0)}$$

Where  $h(E_i, T_0)$  is the number of spin configurations achieved with energy  $E_i \rightarrow$  histogram of  $E$

By using the previous equations we get:

$$D(E_i) = C(T_0)h(E_i, T_0)\exp\left(\frac{E_i}{k_B T_0}\right)$$

Now, we can calculate  $P(E_i, T)$  at any desired temperature

$$P(E_i, T) = \frac{1}{Z(T)} D(E_i) \exp\left(-\frac{E_i}{k_B T}\right)$$

$$P(E_i, T) = \frac{C(T_0)}{Z(T)} h(E_i, T_0) \exp\left[-\frac{E_i}{k_B} \left(\frac{1}{T} - \frac{1}{T_0}\right)\right]$$

The  $C(T_0)/Z(T)$  constant can be obtained from the normalization condition  $\sum_i P(E_i, T) = 1$

- we get thus for any T value:

$$P(E_i, T) = \frac{h(E_i, T_0) \exp\left[-\frac{E_i}{k_B} \left(\frac{1}{T} - \frac{1}{T_0}\right)\right]}{\sum_j h(E_j, T_0) \exp\left[-\frac{E_j}{k_B} \left(\frac{1}{T} - \frac{1}{T_0}\right)\right]}$$

Any average of the energy can be than immediately calculated:

$$\begin{aligned} \langle E(T) \rangle &= \sum_i E_i \cdot P(E_i, T) \\ \langle E^2(T) \rangle &= \sum_i E_i^2 P(E_i, T) \end{aligned}$$

# Calculating other than $E$ averages $\rightarrow$ multiple (double) histogram method

$X \rightarrow$  another quantity of interest, for example  $M$  (magnetization)

Following the previous ideas, the joint probability to get in a state where the energy is  $E_i$  and the  $X$  is  $X_j$

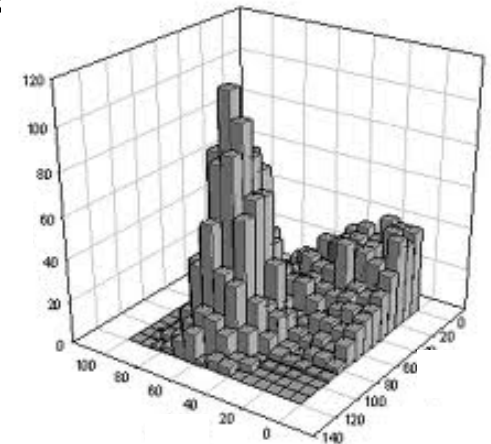
$$P(E_i, X_j, T_0) = \frac{1}{Z(T_0)} D(E_i, X_j) \exp\left(-\frac{E_i}{k_B T}\right)$$

$D(E_i, X_j) \rightarrow$  the number of microstates where the energy is  $E_i$  and  $X$  is  $X_j$

Using the same method as for the single histogram case, we

$$P(E_i, X_j, T) = \frac{h(E_i, X_j, T_0) \exp\left[-\frac{E_i}{k_B} \left(\frac{1}{T} - \frac{1}{T_0}\right)\right]}{\sum_k \sum_l h(E_k, X_l, T_0) \exp\left[-\frac{E_k}{k_B} \left(\frac{1}{T} - \frac{1}{T_0}\right)\right]}$$

$h(E_i, X_j, T_0) \rightarrow$  is a multi-histogram in the  $E$ - $X$  space



Averages can now be computed at any  $T$  temperature

$$\langle X \rangle_T = \sum_i \sum_j X_j \cdot P(E_i, X_j, T)$$

$$\langle X^2 \rangle_T = \sum_i \sum_j X_j^2 \cdot P(E_i, X_j, T)$$

## A basic algorithm for the histogram MC technique

1. consider a lattice with appropriate boundary conditions
2. fix the relevant parameters (temperature, lattice size, exchange constant, etc...)
3. fix a simulation temperature  $T_0$
4. make a Metropolis (Glauber, SW, Wolf..etc) MC simulation at this temperature.
5. after the initial transient steps, collect the values of E and X
6. if time-step is constant (Metropolis, Glauber, SW) algorithm, than simply collect the E and X values in a vector  $W[E][X]$
7. If time step is variable, than collect the E, X values with weight and  $\Delta t$  values in a vector  $W[E][X]$
8. Construct the histograms  $h(E,X,T_0)$  from the data in W.
9. Calculate the average  $\langle E(T_0) \rangle$  and  $\langle E^2(T_0) \rangle$ ,  $\langle X(T_0) \rangle$  ... .
10. By using the proved formula, from  $h(E,X, T_0)$  compute the averages  $\langle E(T) \rangle$  ,  $\langle E^2(T) \rangle$  ,  $\langle X(T) \rangle$  ...for many values of T in the neighborhood of  $T_0$
11. consider another  $T_1$  temperature (far from  $T_0$ ) and repeat the algorithm from 4.

# The Microcanonical Monte Carlo method

*M. Creutz, "Microcanonical Monte Carlo Simulation" Phys. Rev. Lett., Vol. 50, 1411-1415 (1983)*

- Studies the thermodynamic system not in a canonical but in microcanonical ensemble  $\rightarrow$   $E, V, N$  are fixed!

$$S(E, V, N) = k \ln(W);$$

$W$ : the number of microstates corresponding to a macro-state with fixed  $E, V, N$

- All microstates have the same probabilities  $\rightarrow$  needing no extra random number generation to compute averages for the given ensemble

$$\langle A \rangle = \sum_{\{i\}} A_i / W \quad \text{where the summation is on all possible microstates}$$

$A_i$  is the value of the physical quantity  $A$  in microstate  $i$

- **The main problem is how to sample states with fixed  $E$  values!**
- An immediate possibility would be to randomly try to change the state of the system (using an ergodic Markov chain), and accepting only the changes that keeps the total energy of the system constant.  $\rightarrow$  the big problem is that many trials are in vain, and are disregarded  $\rightarrow$  losing a lot of computation time!
- The idea for improving this is the “**demon algorithm**”

# The demon algorithm

- We add an extra degree of freedom to the original macroscopic system of interest → this is the demon
- The total energy of the demon + thermodynamic system (TS) is conserved!
- There is an energy transfer between the demon and the thermodynamic system, the demon can receive energy from the TS and can give energy to the TS. The only constraint is that the demon energy cannot become negative!
- We realize the following ergodic dynamics to map the micro-canonical ensemble of the TS (specific case of the Ising model):

## The Demon algorithm

- choose an initial spin configuration with the desired energy
- choose another spin configuration by flipping one spin
- compute  $\Delta E$ , the change in the energy of the system due to the considered change
- if  $\Delta E < 0$  the system gives the amount  $|\Delta E|$  to the demon and the trial configuration is accepted.
- If  $\Delta E > 0$  and the demon has sufficient energy for this change then the demon gives the necessary energy for the system and new configuration is again accepted. If the demon has not enough energy the new trial configuration is rejected.
- the above steps are continued until a representative sample is obtained, and averages on this are calculated

Since the demon is only one degree of freedom in comparison to the many degrees of freedom of the system, the energy fluctuations of the system will be of order  $1/N$ , which is very small for  $N \gg 1$ , so the calculated average will approximate well the conditions for the micro-canonical ensemble at the fixed energy

# The demon energy distribution, the demon as a thermometer

- The demon in many respects acts as an ideal thermometer since it is only one degree of freedom in comparison to the many degrees of freedom system (TS) with which it changes energy
- From classical statistical physics we know, that in the considered setup, the probability that the energy of the demon at a given step is  $E_d$ :

$$P(E_d) = \frac{1}{Z} \exp\left(-\frac{E_d}{kT}\right) \quad \text{where} \quad Z = \sum_{\{s\}} \exp\left(-\frac{E_s}{kT}\right) \quad \text{is the partition function for the demon}$$

- One way to characterize thus the “temperature” that the demon (thermometer) experiences is by plotting  $\ln[P(E_d)]$  as a function of  $E_d \rightarrow$  the slope will give indication for  $1/kT$
- A simpler way to determine the temperature experienced by the demon is by calculating  $\langle E_d \rangle$ . For the case of the Ising model it can be shown, that:

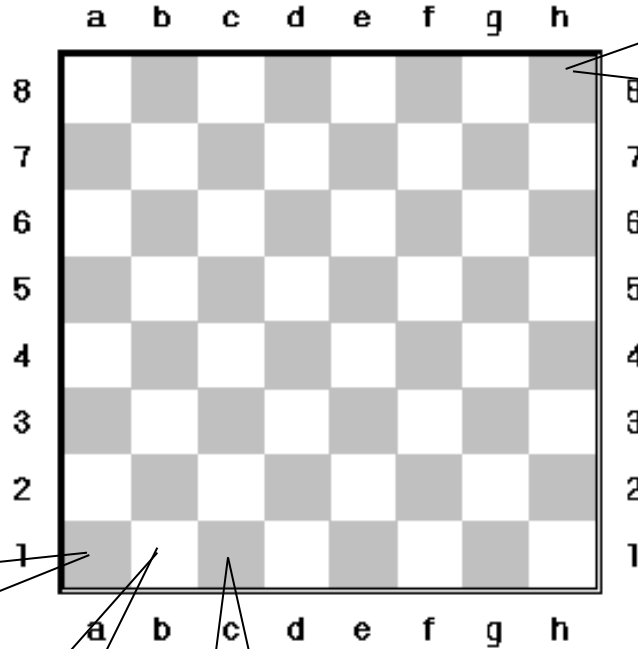
$$kT / J = \frac{4}{\ln(1 + 4J / \langle E_d \rangle)} \quad \text{for } J/\langle E_d \rangle \ll 1 \rightarrow kT = \langle E_d \rangle$$

## Problems to solve by microcanonical Monte Carlo

1. For the 1d Ising model compute  $T$  for  $N=100$  and  $E=-20, -40, -60$  and  $-80$  from the inverse slope of  $P(E_d)$  and  $\langle E_d \rangle$ . Compare Your results to the exact results of the infinite one-dimensional lattice  $E/N = -\tanh(J/kT)$ . Consider 500 MC steps to heat up the system, and than 1000 Monte Carlo steps to study the statistics for  $E_d$ .
2. Simulate the two-dimensional Ising model on a square lattice by using the demon algorithm. Compute  $\langle E_d \rangle$ ,  $\langle M^2 \rangle$  and  $\langle |M| \rangle$  as a function of  $E$ . From here study the susceptibility as a function of the temperature of the system. Consider a lattice with sizes  $100 \times 100$ , 500 MCS to heat up the system and 1000 additional MC steps to study the necessary averages.

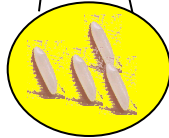
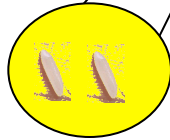
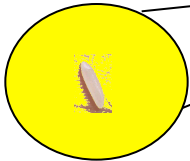
# Monte Carlo optimizations for NP hard problems

What are NP hard problems? - the danger of exponential increase...



$2^{64}$

If the mass of one grain is 0.1 g this is roughly.  $10^{15}$  kg....



The exponential function increases quicker than any power-law:

$$\lim_{x \rightarrow \infty} \frac{a^x}{x^n} = \infty$$

$$n \in \mathbb{N}; n \geq 1$$

$$a \in \mathbb{R}^+; a > 1$$

# Typical problems on computers

**Surfing on the web** : time needed for the download usually scales linearly with the size of the document.

**Text editing**: Time and memory needed increases linearly with the size of the documents.

**Simple arithmetic operations with numbers** (adding, product): Time and memory needed for the operation scales linearly with the size of the numbers

**Searching in a database**: time and memory needed growth linearly with the size of the database

**Complex mathematical functions** (logarithm, sinus, stb...) time needed for completing the operation growth in a complicated manner with the size of the problem, BUT not quicker than a power-law...

**Numerical solution of equations**: The needed time depends on the needed precision. In general the needed time grows as a power-law with the inverse of the precision of the solution.

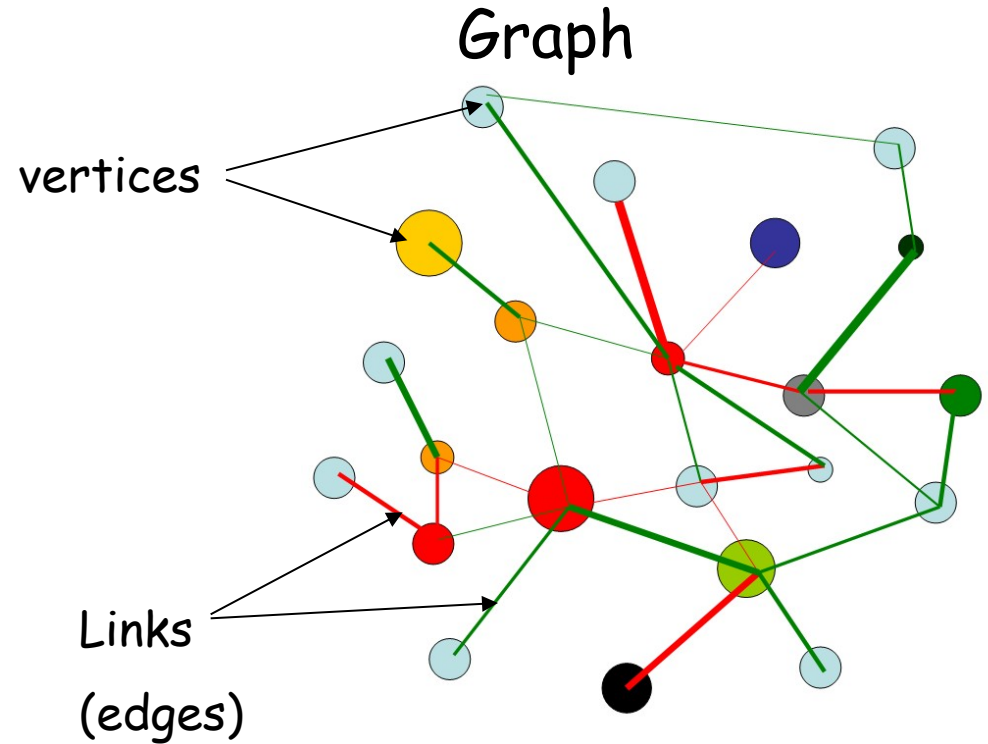
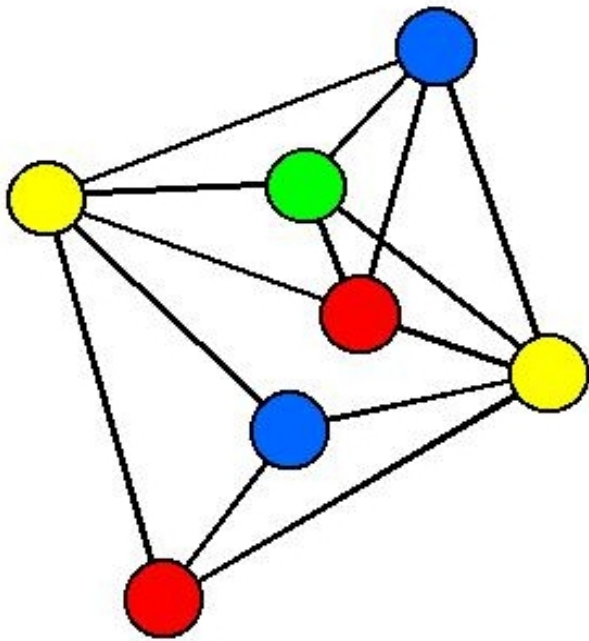
**Calculating determinants, inverting matrices**: time needed grows as a power-law with the size of the matrix..

.....

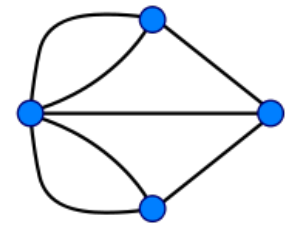
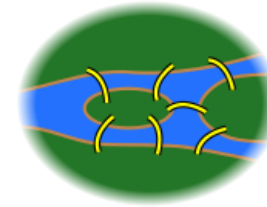
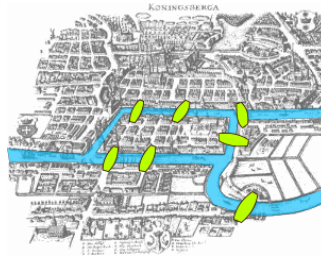
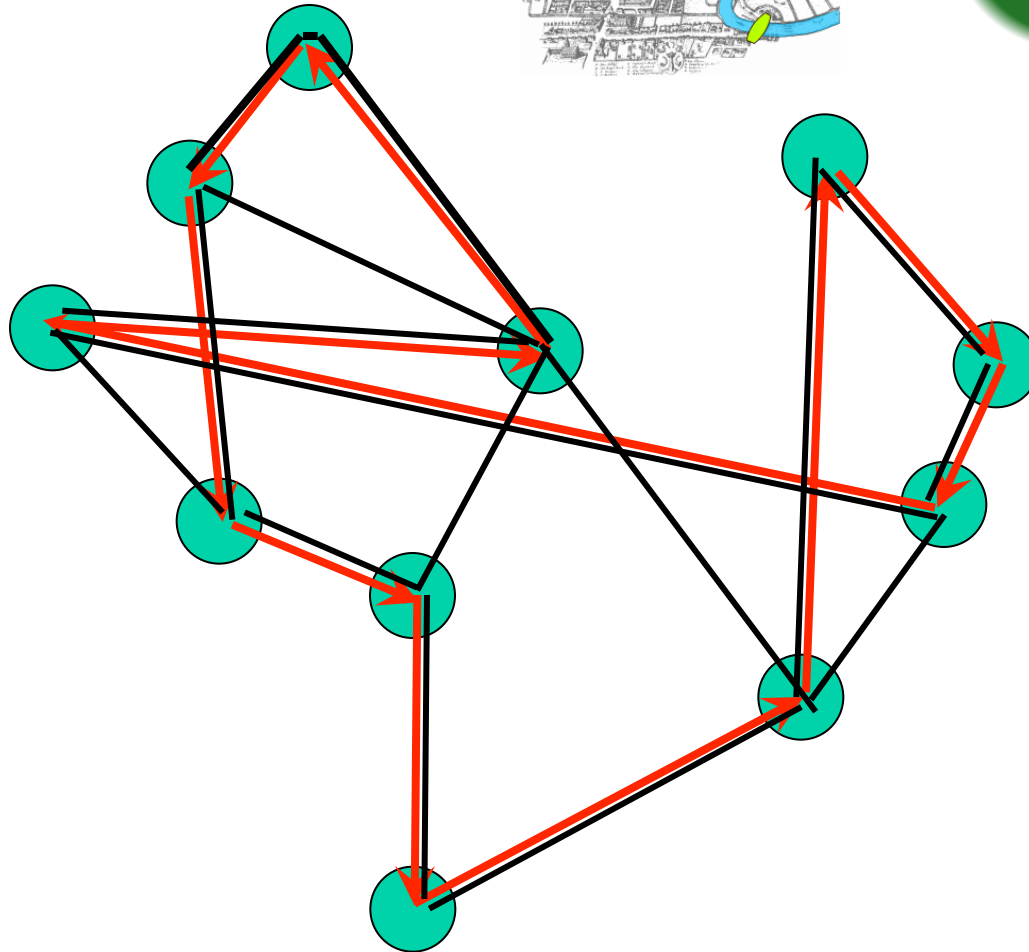
**Image recognition, many parameter optimization or fitting, searching for minima in multidimensional spaces**: the needed time grows exponentially with the size of the system....we can get in problem even for small systems....problems unsolvable even on supercomputers...

Some more complex problems where computational time does not scale in a polynomial manner with system size ....

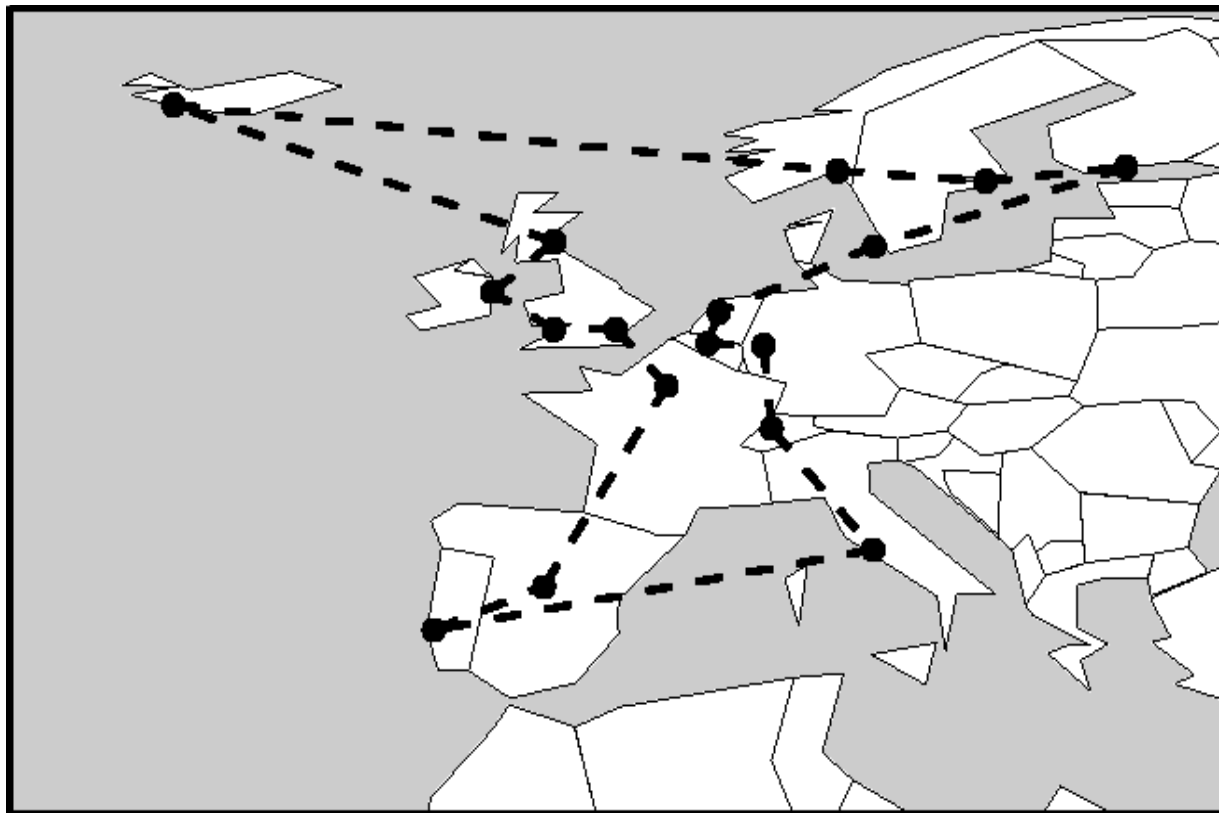
## Graph coloring



# Hamilton cycle

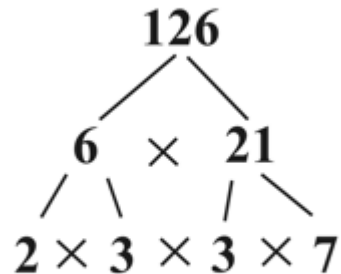


## Euler problem



The traveling salesmen problem

# Factorization (decomposing in primes)



$$18=2*3*3$$

$$364650=2*3*5*5*11*13*17$$

$$205839357486 = 2*3*7*7*19*19*23*37*43*53$$

$$560987965432109896754976009786=?*?*.....$$

**coding, passwords .....**

# SAT (Satisfiability problem)

$$(x_{11} \oplus x_{12} \oplus x_{13}) \otimes (x_{21} \oplus x_{22} \oplus x_{23}) \otimes (x_{31} \oplus x_{32} \oplus x_{33}) \otimes \dots$$

$$x_{ij} \in \{X_1, X_2, X_3 \dots X_s, \bar{X}_1, \bar{X}_2, \dots \bar{X}_s\}$$

$$\oplus = OR$$

$$\otimes = AND$$

|                      | $X$ | 0 | 1 |
|----------------------|-----|---|---|
| $\oplus \rightarrow$ | 0   | 0 | 1 |
|                      | 1   | 1 | 1 |

|                       | $X$ | 0 | 1 |
|-----------------------|-----|---|---|
| $\otimes \rightarrow$ | 0   | 0 | 0 |
|                       | 1   | 0 | 1 |

# Protein folding (ternary structure)

**Protein:** a macromolecular chain constructed by a sequence of amino-acids joined by peptide bonds. The basis of all living cells with important biological functions.

**Primary structure:** the sequence of the amino-acids.

**Secondary structure:** the relative rotation angle of the amino-acides in the chain.

**Tertial structure:** the folding of the chain in water.

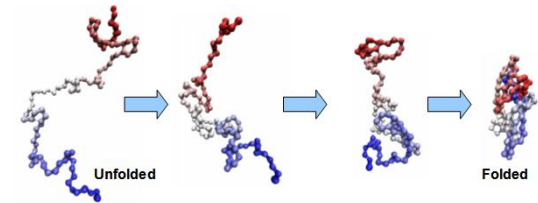
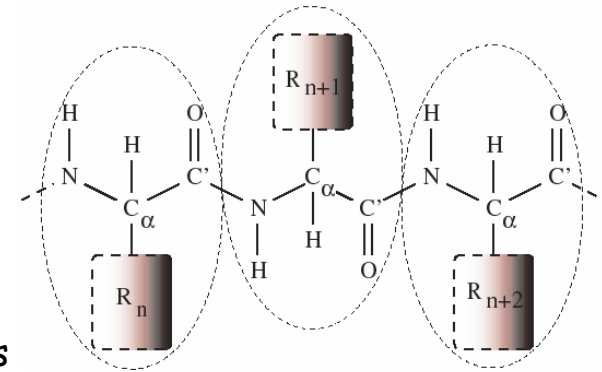
-Proteins have biological functions only in a folded configuration

-Understanding protein folding is a basic problem in chemistry and pharmaceutics

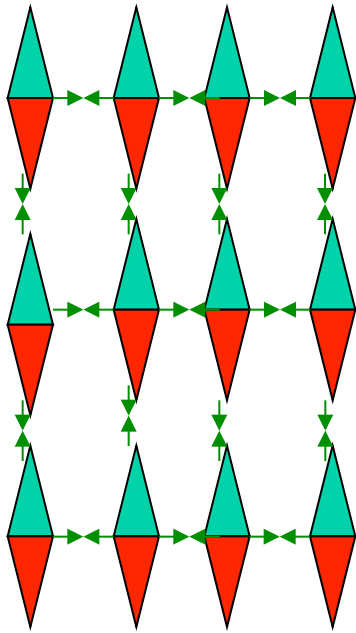
-The folding is determined by Van der Waals forces between the molecules. This is a frustrated system with a lot of local energy minima.

-The biologically relevant proteins have a unique way of folding. Folding drives the system in a global energy minima.

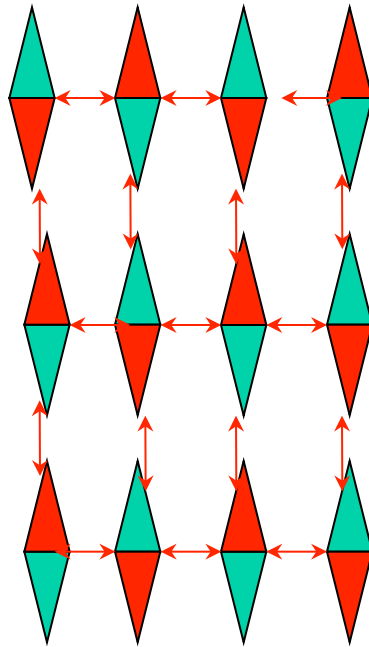
- **Levinthal paradox:** If the protein chain would map all possible folded states and would look for the energy minima in such manner, it would be necessary an astronomical time to fold. Folding in reality is very fast....BUT how?



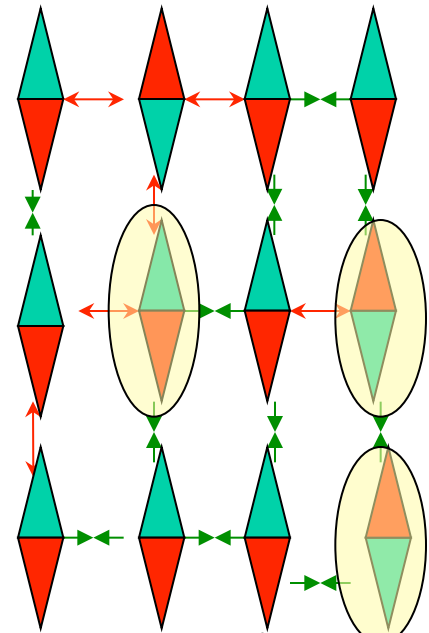
# Spin-glasses



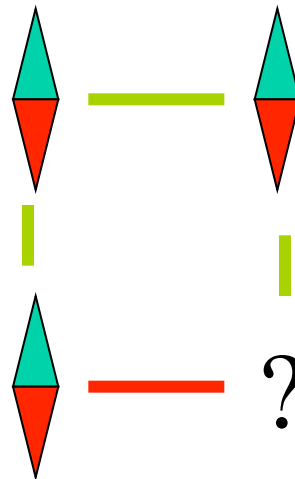
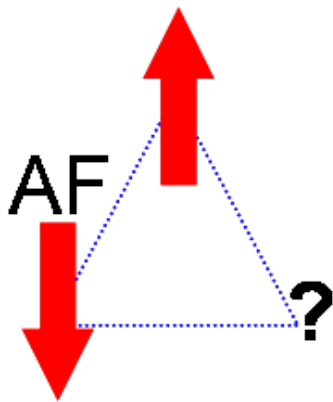
ferromagnets



antiferromagnets



Spin-glasses

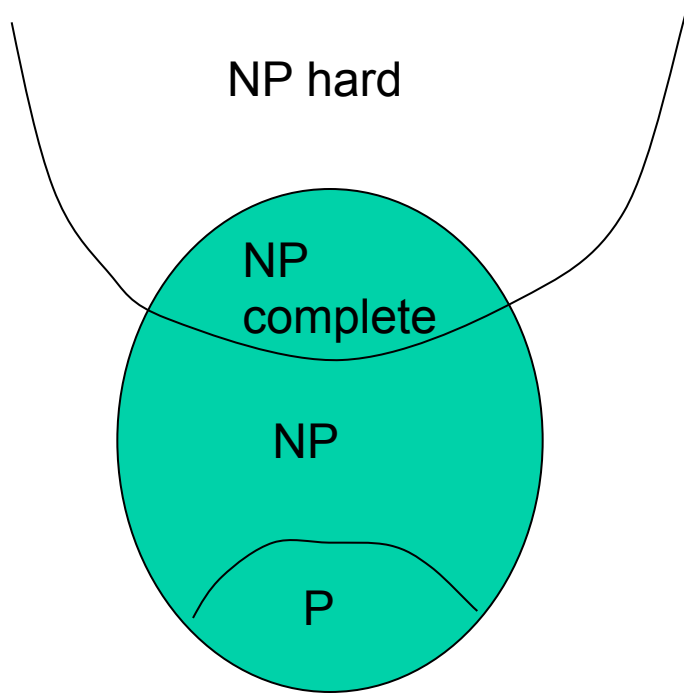


$$H = - \sum_{\langle i,j \rangle} J_{ij} S_i S_j$$

$$S_k = \{-1, +1\}$$

$$J_{ij} = \{-1, +1\} \rightarrow \text{fixed}$$

# Computational complexity - NP hard problems



The main question: How the time needed for solution grows with the “size” of the problem?

## Complexity classes

**P:** time needed for the solution grows polynomially (as a power-law) with the size of the problem (multiplication, adding, matrix determinant, matrix inverting, graph connectivity...etc...)

**NP:** computational amount needed to check the solution grows polynomially with the size of the problems (all P problems... factorization, graph coloring with a given number of colors....etc...)

**NP hard:** at least as hard as any NP complete problem ....in fact harder than any NP complete...

-time for checking the solution grows quicker than a polynomial with the size of the system...

(traveling sales-man problem, protein-folding, spin-glasses...)

On classical computers we can deal easy only with P type problems....!

**NP Complete:** the hardest NP problems. If we can solve these with a P algorithm, all P problems are solvable with P algorithms and then  $NP=P$  (this is a one million USD problems)... these problems are equivalent can be mapped one in the other.. (SAT, graph coloring, Hamilton cycle....)

# A general MC method for NP hard optimization → **simulated annealing**

- A method for searching the GLOBAL minimum in systems with many deep local minima
- First proposed by Kirkpatrick, Gelett and Vecchi (1983) and Cerny (1985)
  - S. Kirkpatrick, C.D. Gelett and M.P. Vecchi, Science vol. 220, p.621-630 (1983)
  - V. Cerny, J. Optim. Theory Appl. , vol 45, pp. 41-51 (1985)
- Idea comes from annealing in metals to minimize defects
  1. Heat the solid state metal to a high temperature
  2. Cool it down very slowly according to a specific schedule
  3. If the heating temperature is sufficiently high to ensure a random state and cooling process is slow enough to ensure thermal equilibrium, then the atoms will place themselves in a pattern that corresponds to the global energy minimum of a perfect crystal.
- The method is appropriate for driving the system to it's GLOBAL energy minimum

# Local search algorithms

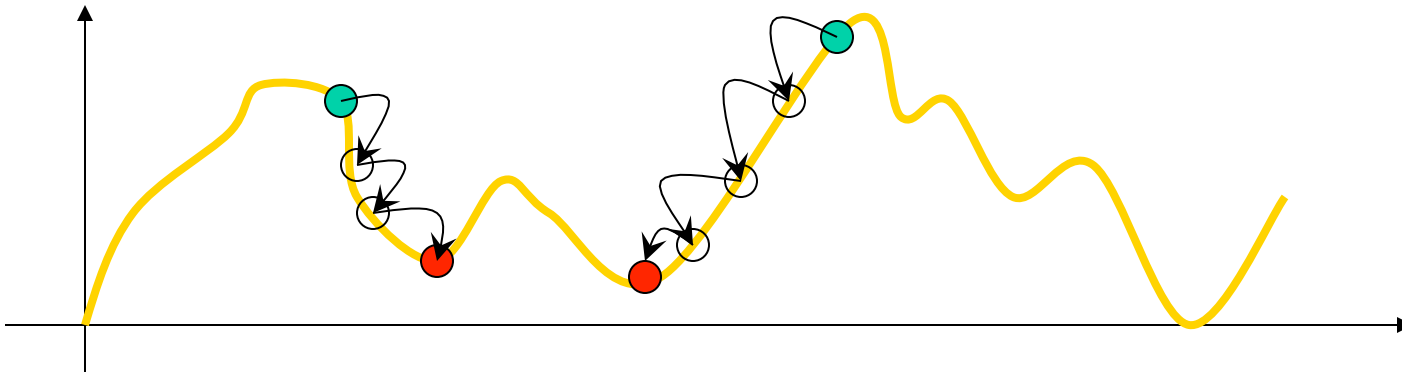
- traditional local search always moves in a direction of improvement  
these are “Greedy” optimization methods

## Standard strategy

- Generate trial point based on current estimates
- Evaluate function at proposed location
- Accept new value if it improves solution

problem: One Can get trapped at local minima

Outcome might depend on starting point



- simulated annealing allows non-improving moves to avoid getting stuck at a local optimum (sometimes we select new points that do not improve solution)
- SA is not a single algorithm but rather a **heuristic Strategy**. There are several decisions to make and several parameters to set to implement SA for a particular problem

# Simulated annealing

- Neighborhood is sampled at random
  - SA allows probabilistic acceptance of non-improving moves
- Acceptance probability is dependent on the control parameter, (temperature)  $t$ , and magnitude of cost increase,  $d$
- When  $d$  is small, acceptance probability is high
- When  $t$  is high, acceptance probability is high
- As  $t$  decreases, acceptance probability decreases

It can be formulated in a thermodynamic language

Incorporate a temperature parameter ( $t$ ) into the minimization procedure

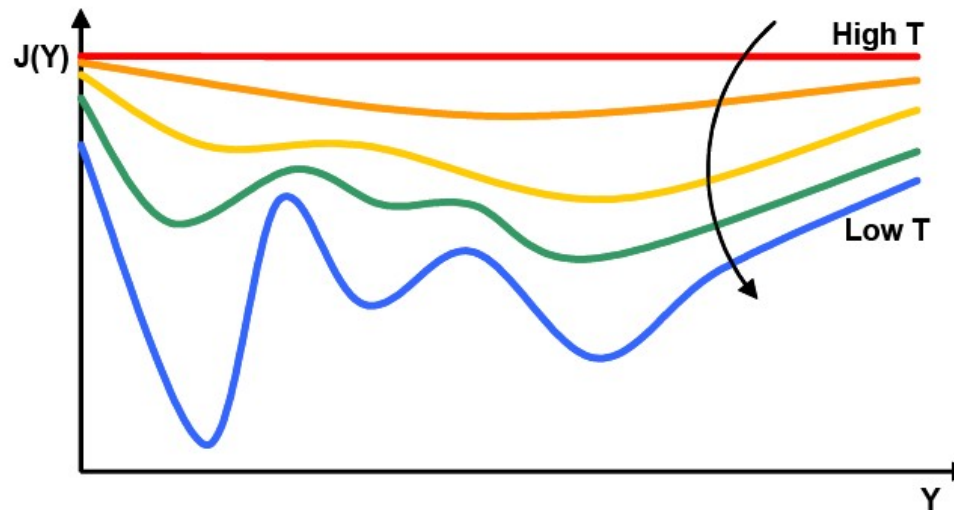
At high temperatures, explore parameter space

At lower temperatures, restrict exploration

| Thermodynamic simulation | Combinatorial optimization |
|--------------------------|----------------------------|
| System states            | Feasible solutions         |
| Energy                   | Cost                       |
| Change of state          | Neighbouring solution      |
| Temperature              | Control parameter          |
| Frozen state             | Heuristic solution         |

# Simulated annealing strategy

- Consider decreasing series of temperatures
- For each temperature, iterate these steps:
  - Propose an update and evaluate function
  - Accept updates that improve solution
  - Accept some updates that don't improve solution
  - Acceptance probability depends on "*temperature*" parameter
- If cooling is sufficiently slow, the global minimum will be reached



At high temperature the algorithm is only looking at the gross features of the optimization surface, while at low temperatures, the finer details of the surface start to appear

## Details:

-The choice of new states: irreducible Markov chain....

-Accepting an update: Metropolis criterion

-Change from  $E$  to  $E'$  :  $\longrightarrow$

$$P = \min \left\{ 1, e^{-\frac{E' - E}{kT}} \right\}$$

- Main problems:

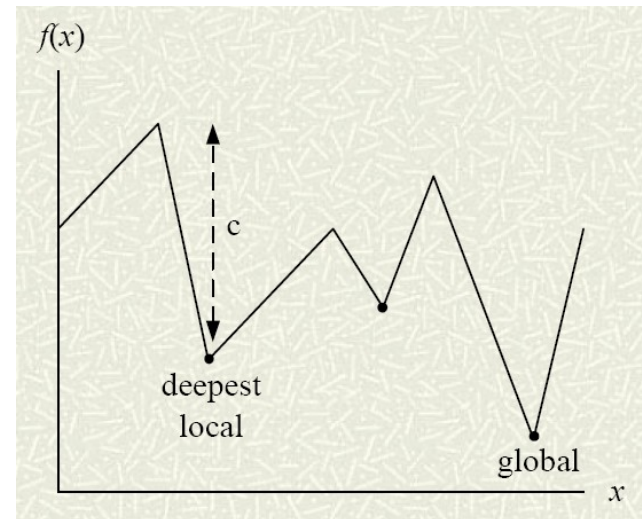
- The maximum temperature?
- Scheme for decreasing temperature?
- Stopping condition ?
- Strategy for proposing updates?

## Initial temperature

Choose  $t$  large enough so that the algorithm can escape the deepest local optimum and has a chance to reach the global

It depends on the investigated problem...

-It must be hot enough to allow free exchange of neighboring solutions and to make the final solution independent of the starting solution



# Temperature decrease scheme

- The system should get close to its stationary distribution at the current temperature before temperature is lowered
- The temperature should go from the initial temperature to approximately 0 at the end...

Two strategies:

(1) Make a large number of iterations at few temperatures

a common cooling schedule is geometric,

$t' = at$  where  $0 < a < 1$ ;  $a = 0.8-0.99$  implying a slow cooling

(2) Make a small number of iterations at many temperatures

a common cooling schedule is  $t' = t / (1 + bt)$  where  $b$  is small, implying very slow cooling; only one iteration is made at each temperature;

- Other cooling schedules are proposed but not as widely used as the above two
- Rate of cooling is more important than manner of cooling (geometric or otherwise)

## Stopping condition

-As  $t$  approaches zero, probability of accepting non-improving moves becomes virtually zero. Following this probability one can decide a stopping limit.

## Strategy for proposing updates

- Problem specific
- Every solution should be **reachable** from every other directly or indirectly (for convergence and validity)
- Neighborhood should be small enough to be searched adequately in relatively fewer iterations, but large enough to provide drastic cost improvements in a single move (tradeoff between computation time and solution quality)
- Random generation of a neighboring solution should be fast for effective use of computation time (since this must be done at every iteration)

# CORRELATION CLUSTERING

## community detection in frustrated networks

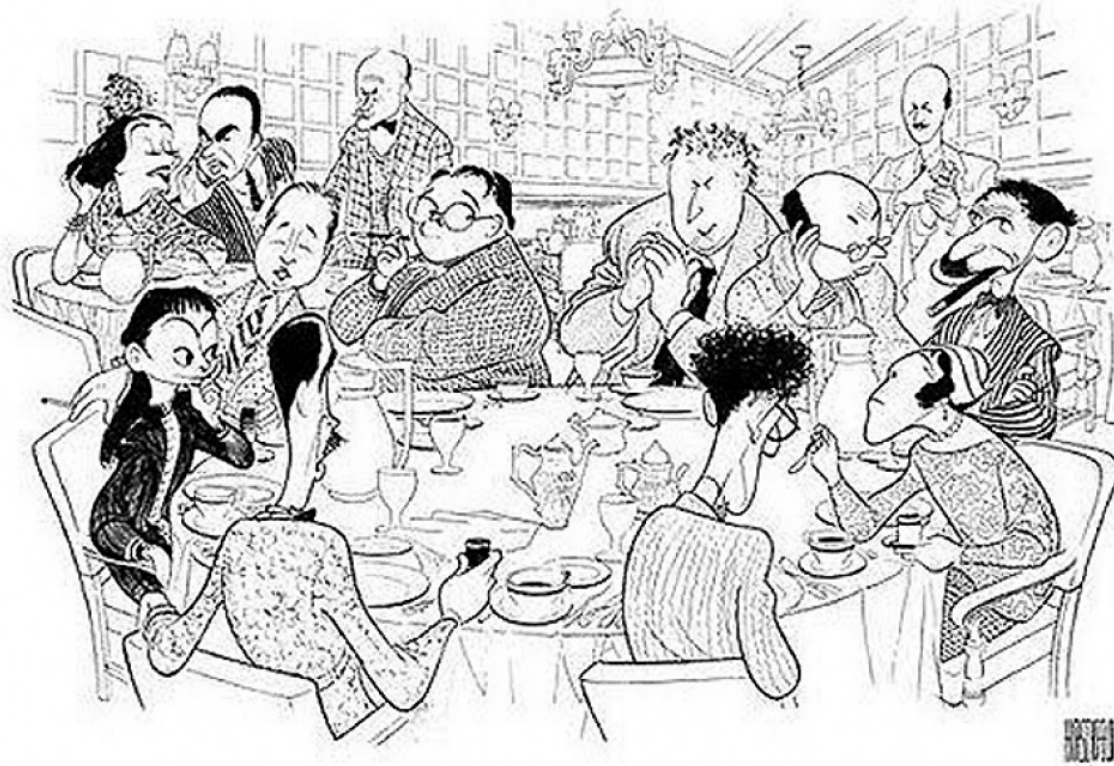
### The problem in laymen terms

given a group of elements with a complex interrelationship between them (positive as well negative links) the goal is to produce a clustering that:

- maximizes the number of positive edges **within** the clusters
- maximizes the number of negative edges **between** the clusters

In general it is an **NP hard** problem!

- N. Bansal, A. Blum and S. Chawla, *Machine Learning, Special Issue on clustering*, vol. 56 (2004) 89.
- M. Charikar, V. Guruswami and A. Wirth, *Journal of Computer and System Sciences*, vol. 71 (3) (2005) 360.
- I. Giotis and V. Guruswami, *Theory of Computing*, vol. 2 (2006) 249.



positive links



negative links



frustrated links

We look for a “statistical” solution similar with the one given for graph coloring and partitioning in Kanter & Sompolinsky (*Graph optimisation problems and the Potts glass*) , J. Phys. A, Vol. 20, L673-679 (1987)

# The "better" defined problem

In a complete network, where the vertex points have  $S_i > 0$  weights (importance) and we have +/- type bonds with  $Z_{ij}$  strength, we want to group the vertices in clusters, so that the bonds are optimally satisfied.

$(i, j = 1, 2, \dots, N) \rightarrow$  the vertex points

$(Z_{ij} = Z_{ji}) \rightarrow$  bonds

$\sigma(i) \rightarrow$  the cluster to which the element  $i$  belongs

Rigorous definition of the problem:

**find  $\sigma(i)$  that minimizes the K cost-function**

$$K = \sum_{i < j} \left[ \frac{1}{2} (Z_{ij} + |Z_{ij}|) S_i S_j (1 - \delta_{\sigma(i)\sigma(j)}) - \frac{1}{2} (Z_{ij} - |Z_{ij}|) S_i S_j \delta_{\sigma(i)\sigma(j)} \right]$$

$$K = - \sum_{i < j} \delta_{\sigma(i)\sigma(j)} Z_{ij} S_i S_j + \frac{1}{2} \sum_{i < j} (Z_{ij} + |Z_{ij}|) S_i S_j \Rightarrow - \sum_{i < j} \delta_{\sigma(i)\sigma(j)} Z_{ij} S_i S_j$$

Two cases:

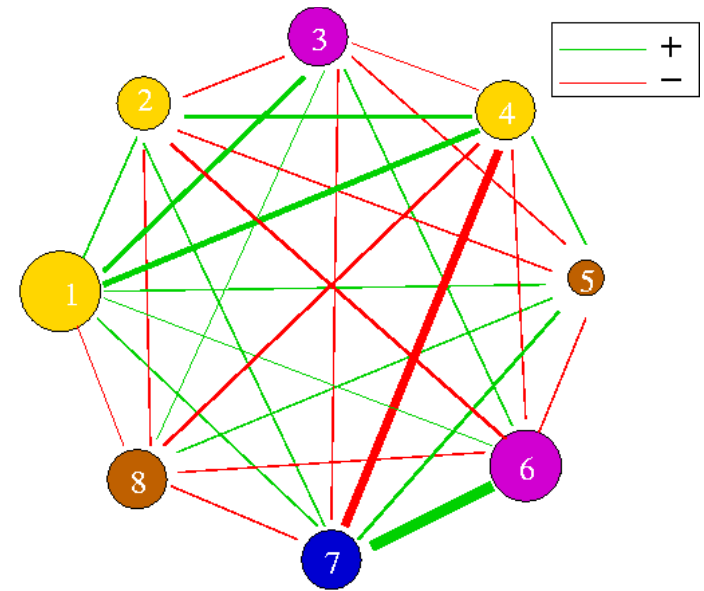
(a)  $W = N$

(b)  $W < N$  and fixed

an NP hard problem!

$\sigma(i) \in \{0, 1, 2, \dots, W - 1\}$

$i = 1, 2, \dots, N;$



# The "simpler" problem on a one-component graph

$$Z_{ij} S_i S_j = J_{ij} = \begin{cases} \pm C \\ 0 \end{cases} \longrightarrow \begin{cases} K = - \sum_{i < j} J_{ij} \delta_{\sigma(i)\sigma(j)} \\ \sigma(i) \in \{0, 1, 2, \dots, N-1\} \\ i = 1, 2, \dots, N \end{cases}$$

$$J_{ij} = \begin{cases} 0 & \text{with probability } (1-p), \\ +C & \text{with probability } pq \\ -C & \text{with probability } p(1-q) \end{cases}$$

In general an NP hard problem!  $B_N \propto \frac{N^N}{(\ln N)^N \cdot e^N}$  total number of configurations for  $W=N$

## Attempt for a statistical solution

order parameter:

$$r_1(q, p) = \left\langle \max_{(i)} \left\{ \frac{C_x(i, q, p)}{N} \right\} \right\rangle_x$$

the relative size of the largest cluster in case of optimal clusterization (for smallest value of  $K$ )

$C_x(i, q, p)$ : the number of elements in the  $i$  cluster for an  $x$  realization of the  $J_{ij}$  bonds (where  $q$  is the probability of the + bonds, and  $p$  the probability that there is a non-zero bond)

**overline:** an averaging on "degeneration" (if any)

$\langle \cdot \rangle_x$ : ensemble average for different realization of  $J_{ij}$

The interesting quantity: order-parameter as a function of  $q \rightarrow r(q, p)$ : for  $N \rightarrow \infty$

# Connection with the Potts-glass

$$H = - \sum_{i < j} J_{ij} \delta_{\sigma(i)\sigma(j)}$$

$$\sigma(i) \in \{0, 1, 2, \dots, N\}$$

$$i = 1, 2, \dots, N$$

## The infinite-range p-state Potts glass

• D. Elderfield and D. Sherrington; J. Phys. C vol. 16, L497 (1983)

• E. De Santes, G. Parisi and F. Ritort; J. Phys. A, vol. 28, 3025 (1995)

The disorder is irrelevant (scales out) if:  $\lim_{N \rightarrow \infty} \frac{\Delta J}{\langle J \rangle} = 0$

In such cases the  $J_{ij} = \frac{J_0}{N}$  mean-field solution is exact  $\longrightarrow$

trivial solutions for one-component graphs at  $T=0$

$$r_{1(2)}(q, p) = \begin{cases} 1/N \dots q < 0.5 \\ 1 \dots q > 0.5 \end{cases}$$

complex spin-glass behavior is obtained when

$$\langle J_{ij} \rangle = \frac{J_0}{N}$$

$$\Delta(J_{ij}) = \langle J_{ij}^2 \rangle - \langle J_{ij} \rangle^2 \sim \frac{1}{N}$$

$$\lim_{N \rightarrow \infty} \langle \langle H \rangle_T \rangle_x \propto N$$

$$\langle \rangle_T \longrightarrow$$

denotes a thermal (ensemble average)

Is our simplified problem equivalent with a  $T=0$ ,  $p=N$  Potts-glass?

in order to use results from statistical physics for one component graphs in the limit  $T \rightarrow 0$

$$\lim_{N \rightarrow \infty} \left\langle \left\langle K(\{\sigma_i\}) \right\rangle_{T=0} \right\rangle_x \propto N$$

$$K = - \sum_{i < j} J_{ij} \delta_{\sigma(i)\sigma(j)}$$

$$\sigma(i) \in \{0, 1, 2, \dots, N-1\}$$

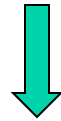
$$i = 1, 2, \dots, N$$

$$J_{ij} = \begin{cases} 0 & \text{with probability } (1-p), \\ +C & \text{with probability } pq \\ -C & \text{with probability } p(1-q) \end{cases}$$

this leads to:  $qpCN^2 \propto N \Rightarrow C \propto \frac{1}{qpN}$

Z. Nédá, R. Florian, M. Ravasz, A. Libál and G. Györgyi:  
**Physica A**, vol. **362**, 357-369 (2006)

Z. Neda, R. Sumi, M. Ercsey-Ravasz, ..., J. Phys. A, vol.  
42, 345003 (2009)



$$\langle J_{ij} \rangle \propto \frac{(2q-1)}{q} \frac{1}{N} \quad \Delta J_{ij} \propto \frac{(1 - p(2q-1)^2)}{pqN^2} = \frac{1}{pqN^2} - \frac{(2q-1)^2}{qN^2}$$

for all DENSE graphs ( $p > 0$ ) the disorder is irrelevant!

the statistical solution is:

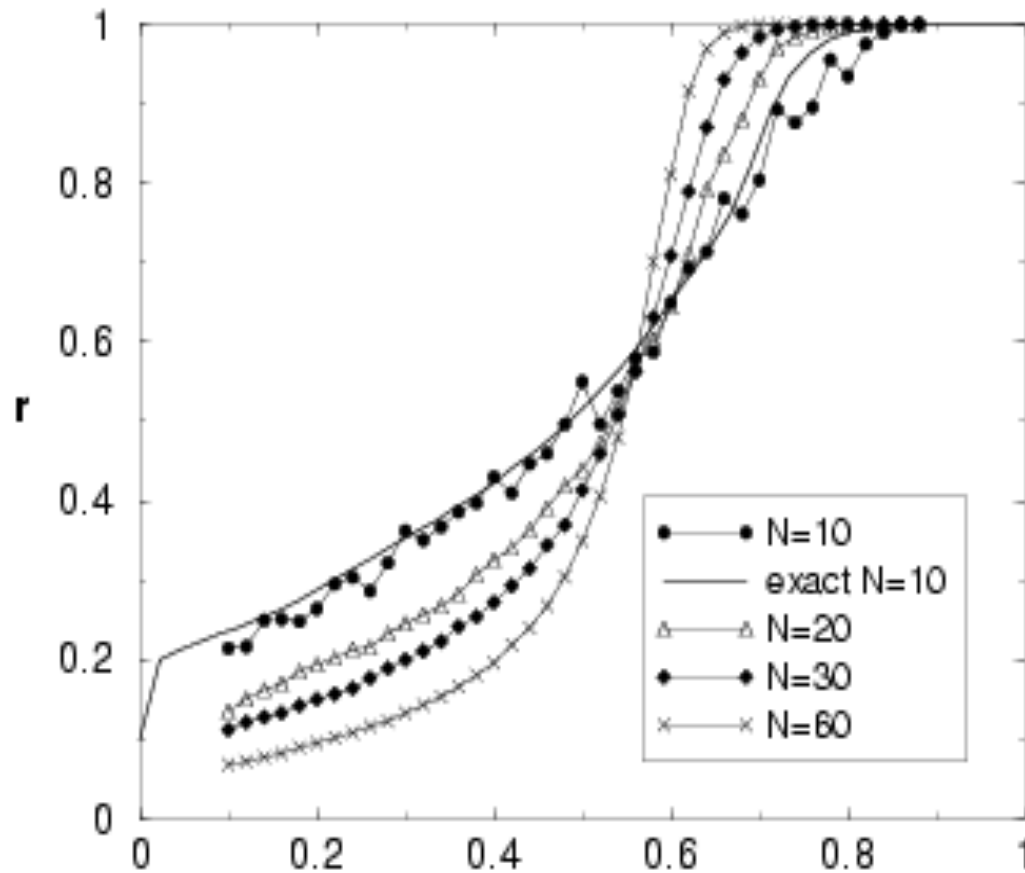
$$r_{1(2)}(q, p) = \begin{cases} 1/N \dots q < 0.5 \\ 1 \dots q > 0.5 \end{cases}$$

for sparse enough graphs ( $p \propto N^{-\alpha}$ ,  $\alpha \geq 1$ )  $\longrightarrow$  complex, spin-glass type behavior

# Simulated Annealing for correlation clustering

- Consider as many clusters as elements in the system
- Consider an initial temperature  $T_k = N / [-4 * \log(0.8)]$ . This temperature allows a big change in energy: ( $\Delta E = N/4$ ) with 80% probability.
- Consider an initial cluster structure for the system, put the elements randomly in these clusters
- Calculate the  $E = K$  cost-function (energy) in this condition
- Choose one element randomly and put it in another cluster
- Calculate the  $E' = K'$  cost-function (energy) in this case
- Accept the new configuration with probability  $\longrightarrow P = \min \left\{ 1, e^{-\frac{E' - E}{T}} \right\}$
- The temperature is cooled by a rate:  $T' = T * 0.98$  until  $T_f = 0.1$
- For  $N = 10 - 40$  we make 200 MC steps/temperature
- For  $N = 100$  we need approx 1000 MC/steps, the number of needed steps increases strongly with system size
- repeat the whole SA strategy many times....
- One can study systems up to  $N = 10$  by exact enumeration and systems  $N < 150$  with SA

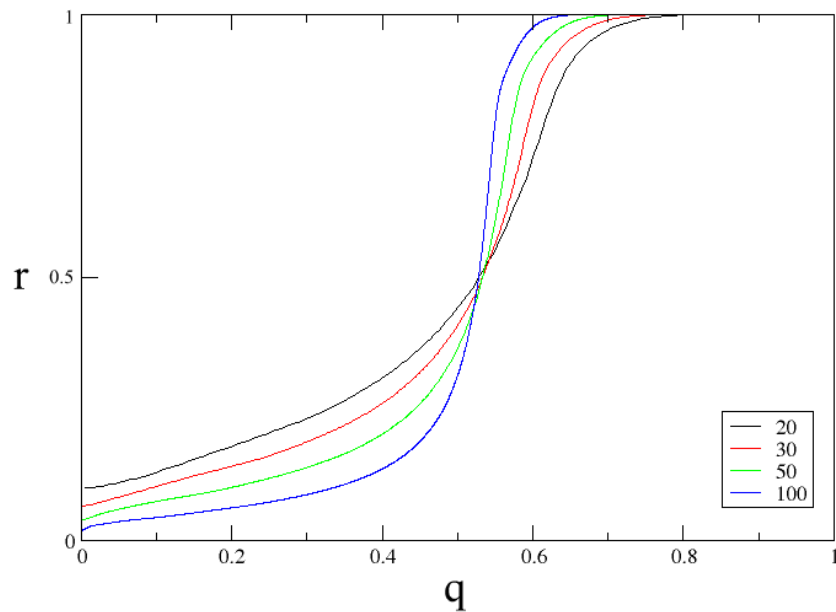
Simulated annealing results for correlation clustering  
on a globally connected (complete) graph



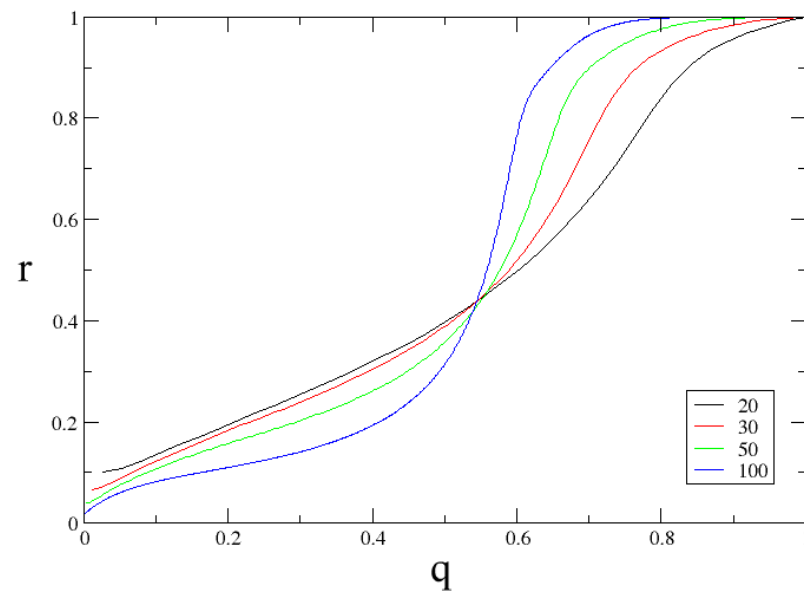
# Numerical results on finite and dense random graphs

**random dilution:** -a portion  $1-p$  of a complete graph links are randomly selected and given  $J_{ij} = 0$  values

$p=95\%$



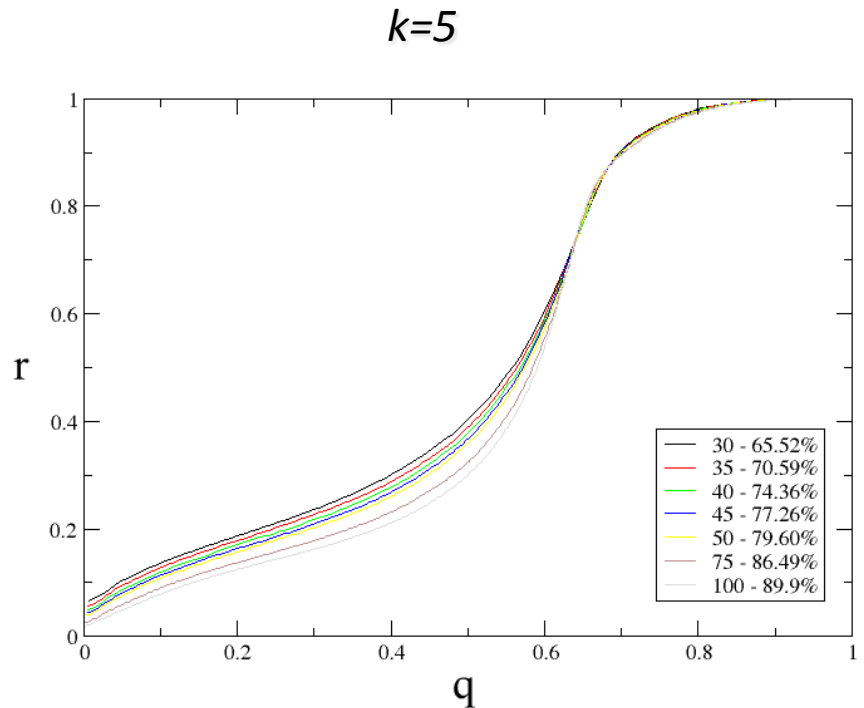
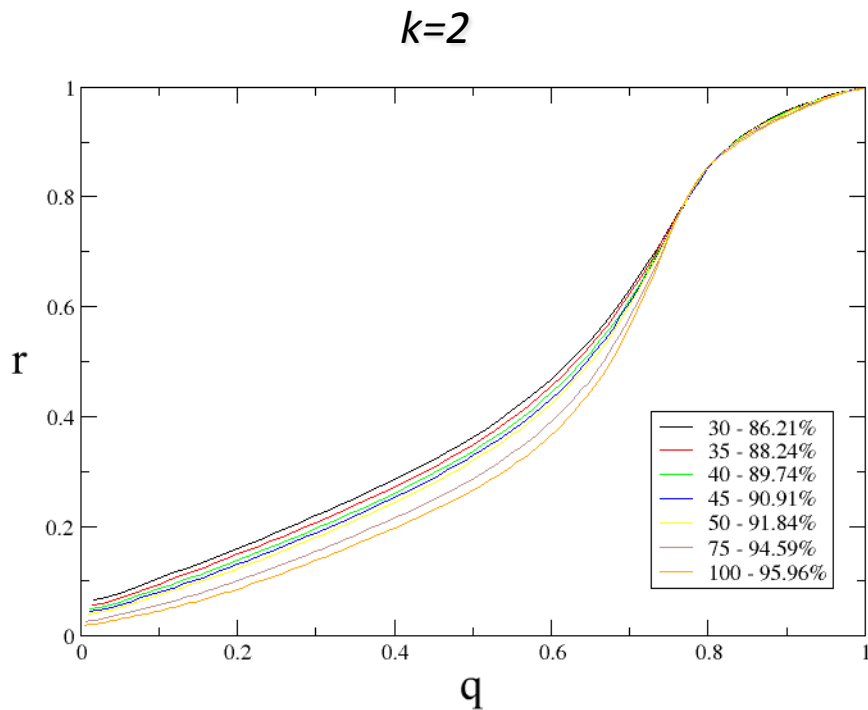
$p=20\%$

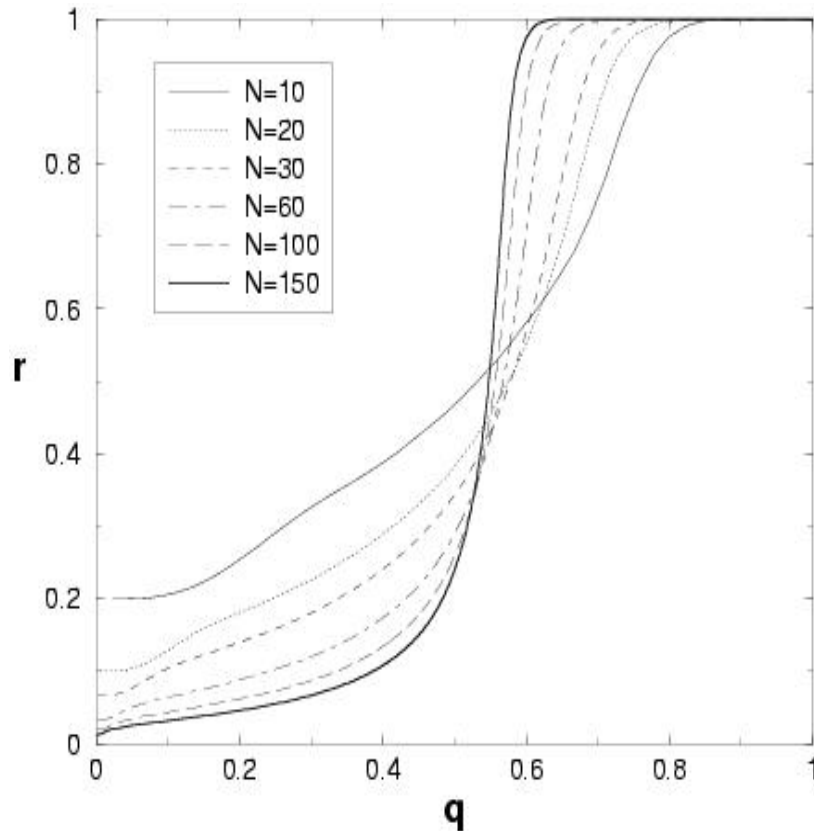


# Numerical results on sparse random networks with fixed number of degrees/node

k: number of connection/node

Disorder is relevant, it does not scale out  $\rightarrow$  the system is disordered as a spin-glass!  $\rightarrow$  phase transition is a highly nontrivial one





The order parameter as a function of  $q$

## MC type renormalization

### Method:

1. We start from  $J_{ij}=-1$  (all  $(i,j)$  pairs), and  $\sigma(i)=i-1$  state (all bonds negative and individuals in different clusters)
2. A randomly chosen bond is changed to  $+1$
3. We verify whether  $K$  is decreasing or not if the two vertices united by that bond are placed in one cluster. If yes, we put them in one cluster (or the two clusters are united).
4. All elements that are linked with  $+$  bonds to the chosen vertex are verified and if  $K$  decreases it is putted in the same cluster. Recursively this is done then with these vertices too...
5. Another bond is changed to  $+$ , and points 3.-4. repeated....this is continued until all bonds become  $+$
6. The MC procedure 1.-5. is repeated many times and the  $r(q)$  curves are thus constructed.

# Conclusions

I . Solution of the correlation-clustering problem on DENSE one-component graphs is simple in the  $N \rightarrow \infty$  thermodynamic limit

whenever there are more positive links than negative ones put every node in the same cluster

whenever there are more negative links than positive ones put everybody in separate clusters

II. Solution of the correlation-clustering problem on SPARSE enough one-component graphs remains a complex spin-glass type problem even in the thermodynamic limit

numerical optimizations on finite systems shows that the behavior of the defined order parameters is more complex than for dense graphs

numerical results suggests that one-community solution is possible ONLY in the  $q=1$  limit!

Moretti....J. Stat. mech. (2011) P03032  
Pre vol. 80, 020102 (2009)

# The Quantum Monte Carlo Method

Quantum-statistical systems --> quantum mechanical systems in contact with a heatbath  
 Quantum spin models (taking in consideration the quantum nature of the spins)

$$\hat{H} = -J_{xy} \sum_{i,j} (\hat{S}_j^x \hat{S}_i^x + \hat{S}_j^y \hat{S}_i^y) - J_z \sum_{i,j} \hat{S}_i^z \hat{S}_j^z$$

$$[\hat{S}_i^x, \hat{S}_i^y] = i\hbar \hat{S}_i^z; [\hat{S}_i^y, \hat{S}_i^z] = i\hbar \hat{S}_i^x; [\hat{S}_i^z, \hat{S}_i^x] = i\hbar \hat{S}_i^y$$

$$\hat{S}_i^z |s, m_s\rangle = \hbar m_s |s, m_s\rangle; \hat{S}_i^2 |s, m_s\rangle = \hbar^2 m_s(m_s + 1) |s, m_s\rangle$$

$J_{xy}=J_z$  --> Heisenberg model;  $J_{xy}=0$  Ising model (classical) ;  $J_z=0$  XY model

Introducing the raising and lowering operators

$$\hat{S}^+ = \hat{S}^x + i\hat{S}^y; \hat{S}^+ |S, m_s\rangle = \hbar \sqrt{S(S+1) - m_s(m_s + 1)} |S, m_s + 1\rangle$$

$$\hat{S}^- = \hat{S}^x - i\hat{S}^y; \hat{S}^- |S, m_s\rangle = \hbar \sqrt{S(S+1) - m_s(m_s - 1)} |S, m_s - 1\rangle \quad [\hat{S}^+, \hat{S}^-] = 2\hat{S}_z \neq 0$$

$$\hat{H} = -\frac{J_{x,y}}{2} \sum_{i,j} (\hat{S}_j^+ \hat{S}_i^- + \hat{S}_i^+ \hat{S}_j^-) - J_z \sum_{i,j} \hat{S}_j^z \hat{S}_i^z$$

in 1D  $\longrightarrow$

$$\hat{H} = -\frac{J_{x,y}}{2} \sum_i (\hat{S}_{i+1}^+ \hat{S}_i^- + \hat{S}_i^+ \hat{S}_{i+1}^-) - J_z \sum_i \hat{S}_{i+1}^z \hat{S}_i^z$$

# The Hubbard model

- many body hamiltonian in a second quantized form
- describes interacting electrons in the periodic potential of a lattice
- second quantization in Wannier states (quasi-localized) electron states on a given orbital and with a given spin orientation at a given ion at position  $\mathbf{R}_\alpha$
- Hamiltonian written with the creation ( $\hat{a}^\dagger$ ) and destruction ( $\hat{a}$ ) operators, acting on states described with occupational number
- interaction is strongly screened by the electron gas, thus it is restricted at the same site

$$\hat{H} = - \sum_{\alpha, \alpha', \sigma} t_{\alpha, \alpha'} \hat{a}_{\alpha', \sigma}^\dagger \hat{a}_{\alpha, \sigma} + U \sum_{\alpha, \sigma} \hat{n}_{\alpha, \sigma} \hat{n}_{\alpha, -\sigma}$$

$$t_{\alpha, \alpha'} = \frac{1}{V} \sum_{\vec{k}} \epsilon_{\vec{k}} \exp[i\vec{k}(\vec{R}_\alpha - \vec{R}_{\alpha'})]$$

In 1D, and considering jumps only at nearest neighbor lattice sites, the hamiltonian becomes

$$\hat{H} = -t \sum_{\alpha, \sigma} (\hat{a}_{\alpha, \sigma}^\dagger \hat{a}_{\alpha+1, \sigma} + \hat{a}_{\alpha+1, \sigma}^\dagger \hat{a}_{\alpha, \sigma}) + U \sum_{\alpha, \sigma} \hat{n}_{\alpha, \sigma} \hat{n}_{\alpha, -\sigma}$$

$n_{\alpha, \sigma}$  are the particle number operators in state  $(\alpha, \sigma)$

$$[\hat{a}_{\alpha, \sigma}^\dagger, \hat{a}_{\alpha, \sigma}] \neq 0$$

# A lattice model for itinerant electrons in 1D

For 1 electron:

$$\hat{H}\psi = -\frac{h^2}{8\pi^2 m} \Delta\psi + V(x) = E\psi$$

Discretizing it on a lattice with sites “a” --> L coupled linear equations (L.a=l; l the length of the considered space);

$\psi_i$  is the medium of  $\psi(x)$  in box “i”

$V_i$  is the medium of  $V(x)$  in box “i”

$$\left(-\frac{h^2}{8\pi^2 m a^2}\right)(\psi_{i+1} + \psi_{i-1} - 2\psi_i) + V_i \psi_i = E\psi_i$$

introducing “t” and “ $W_i$ ” we get:

$$t = \frac{h^2}{8\pi^2 m a^2}; W_i = V_i + \frac{h^2}{4\pi^2 m a^2}$$

$$-t(\psi_{i+1} + \psi_{i-1}) + W_i \psi_i = E\psi_i$$

- the above equations can be written in a second quantized form

state-vectors -->  $n_i$  occupation numbers of the cells

$$|n_1, n_2, \dots, n_i, \dots, n_L\rangle$$

$$\hat{c}_i^+ |n_1, n_2, \dots, n_i, \dots, n_L\rangle = \sqrt{n_i + 1} |n_1, n_2, \dots, n_i + 1, \dots, n_L\rangle$$

$$\hat{c}_i |n_1, n_2, \dots, n_i, \dots, n_L\rangle = \sqrt{n_i} |n_1, n_2, \dots, n_i - 1, \dots, n_L\rangle$$

$$\hat{n}_i |n_1, n_2, \dots, n_i, \dots, n_L\rangle = n_i |n_1, n_2, \dots, n_i, \dots, n_L\rangle$$

$$[\hat{c}_i^+, \hat{c}_i] \neq 0$$

$$\psi = \sum_{i=1}^L \psi_i \hat{c}_i^+ |0, 0, \dots, 0\rangle$$

$$\hat{H} = -t \sum_i [\hat{c}_i^+ \hat{c}_{i+1} + \hat{c}_{i+1}^+ \hat{c}_i] + \sum_i W_i \hat{n}_i$$

Considering more than one interacting electrons on the same lattice is possible by considering extra terms.  $H_0$  is an on-site repulsion and  $H_1$  is interaction between particles in neighboring cells.

$$\hat{H}_0 = V_0 \sum_i \hat{n}_i (\hat{n}_i - 1)$$

$$\hat{H}_1 = V_1 \sum_i \hat{n}_i \hat{n}_{i+1}$$

Taking into account the spin of the particles --> by an additional discrete parameter characterizing the projection of the spin

## Basic idea for the QMC method

Transferring the d dimensional quantum-statistical problem in a d+1 dimensional classical statistical physics problem

we consider the system in canonical ensemble

$$Z = \text{Tr}[\exp(-\beta\hat{H})] = \sum_{\{n_i\}} \langle n_1, n_2, \dots, n_L | \exp(-\beta\hat{H}) | n_1, n_2, \dots, n_L \rangle$$

**First possibility** to tackle the problem: diagonalize H and consider after that the Metropolis or other MC schemes for the resulted states

problem: if there are only 2 possible states in each of the L possible cells --> we have to diagonalize a  $2^L \times 2^L$  matrix which is practically impossible for  $L \gg 1$ .

**Second possibility:** make a classical MC algorithm without diagonalization

- consider an  $\{n_i^0\}$  initial configuration
- consider a possible new  $\{n_i^f\}$  configuration
- accept the change  $\{n_i^0\} \rightarrow \{n_i^f\}$  with probability:

$$P = \frac{\langle \{n_i^f\} | \exp(-\beta \hat{H}) | \{n_i^f\} \rangle}{\langle \{n_i^f\} | \exp(-\beta \hat{H}) | \{n_i^f\} \rangle + \langle \{n_i^0\} | \exp(-\beta \hat{H}) | \{n_i^0\} \rangle}$$

- continue the algorithm until the thermodynamic equilibrium is reached
- collect periodically the relevant data

problem: calculation of the matrix elements are VERY TIME CONSUMING and it is not clear from the start which changes will yield reasonable transition probabilities!

$$e^{\hat{A}+\hat{B}} = e^{-\frac{1}{2}[\hat{A},\hat{B}]} e^{\hat{A}} e^{\hat{B}} \longrightarrow \text{Complicated if A and B does not commute}$$

**Third possibility** (wise solution): to rewrite Z in a form in which the calculation of the P probabilities are easy when only a few  $n_i$  numbers are changed.

- this is possible by the use of the Trotter-Suzuki approximation

# The Trotter-Suzuki approximation

We consider the general form of the H Hamiltonian:

$$\hat{H} = -t \sum_i [\hat{c}_i^+ \hat{c}_{i+1} + \hat{c}_{i+1}^+ \hat{c}_i] + \sum_i W_i \hat{n}_i + \sum_i V_0 \hat{n}_i (\hat{n}_i - 1) + \sum_i V_1 \hat{n}_i \hat{n}_{i+1}$$

and denote: 
$$V_i = \sum_i W_i \hat{n}_i + \sum_i V_0 \hat{n}_i (\hat{n}_i - 1) + \sum_i V_1 \hat{n}_i \hat{n}_{i+1}$$

we divide H in two parts: 
$$\hat{H} = \hat{H}_a + \hat{H}_b$$

$$\hat{H}_a = (-t\hat{c}_1^+ \hat{c}_2 - t\hat{c}_2^+ \hat{c}_1 + \frac{V_1}{2} + \frac{V_2}{2}) + (-t\hat{c}_3^+ \hat{c}_4 - t\hat{c}_4^+ \hat{c}_3 + \frac{V_3}{2} + \frac{V_4}{2}) + \dots$$

$$\dots + (-t\hat{c}_{L-1}^+ \hat{c}_L - t\hat{c}_L^+ \hat{c}_{L-1} + \frac{V_{L-1}}{2} + \frac{V_L}{2}) = \hat{H}_1 + \hat{H}_3 + \dots + \hat{H}_{L-1}$$

$$\hat{H}_b = \hat{H}_2 + \hat{H}_4 + \dots + \hat{H}_L; \text{ where } : \hat{H}_i = -t\hat{c}_i^+ \hat{c}_{i+1} - t\hat{c}_{i+1}^+ \hat{c}_i + \frac{V_i}{2} + \frac{V_{i+1}}{2}$$

The terms inside  $H_a$  and  $H_b$  commute with each other, but:

$$[\hat{H}_a, \hat{H}_b] \neq 0$$

we can write thus:

$$\langle \{n_i\} | \exp(-\beta \hat{H}_a) | \{n_j\} \rangle = \langle \{n_i\} | \exp(-\beta \hat{H}_1) \cdot \exp(-\beta \hat{H}_3) \cdot \dots \cdot \exp(-\beta \hat{H}_{L-1}) | \{n_j\} \rangle$$

Ideal would be to do this for the whole  $H=H_a+H_b$ ! This is not possible however, because  $H_a$  and  $H_b$  do not commute and have terms acting on the same  $n_i$  numbers.

- **SOLUTION** the Trotter-Suzuki (TS) approximation: if A and B are sufficiently SMALL operators:

$$e^{\hat{A}} \cdot e^{\hat{B}} = e^{\hat{A}+\hat{B}+\frac{1}{2}[\hat{A},\hat{B}]} \approx e^{\hat{A}+\hat{B}}$$

Applying the TS approximation: if  $M \gg 1$  (integer), so that  $\beta H_a / M$  and  $\beta H_b / M$  is small enough we can write

$$Z = \sum_{\{n_i\}} \langle \{n_i\} | \exp[-\frac{\beta}{M}(\hat{H}_a + \hat{H}_b)]^M | \{n_i\} \rangle \approx \sum_{\{n_i\}} \langle \{n_i\} | [\exp(-\frac{\beta}{M}\hat{H}_a) \cdot \exp(-\frac{\beta}{M}\hat{H}_b)]^M | \{n_i\} \rangle$$

To write Z as product of simple terms we insert  $2M-1$  complete sets of state vectors

$$\sum_{\{n_i^\alpha\}} |n_{1,\alpha}, n_{2,\alpha}, \dots, n_{L,\alpha}\rangle \langle n_{1,\alpha}, n_{2,\alpha}, \dots, n_{L,\alpha}|$$

between the exponential terms

Taking into account that  $H_i$  acts only on the occupation numbers  $n_i$  and  $n_{i+1}$  we can write

- $i$  and  $\alpha$  label occupation numbers
- from a 1D lattice --> 2D lattice
- all  $n_{i,\alpha}$  numbers are independent!
- due to the original trace:  $n_{i,2M+1} = n_{i,1}$

$$Z = \sum_{\{n_{i,\alpha}\}} \prod_{\alpha=1}^{2M} \prod_{i=1}^L P_{i,\alpha}$$

$$P_{i,\alpha} = \left\langle n_{i,\alpha}; n_{i+1,\alpha} \left| e^{-(\Delta\tau \hat{H}_i)} \right| n_{i,\alpha+1}; n_{i+1,\alpha+1} \right\rangle$$

where:  $\Delta\tau = \frac{\beta}{M}$

We transform now the quantum-statistical problem to a classical one!  $\longrightarrow$

$$Z = \sum_{\{n_{i,\alpha}\}} \exp(-\beta \sum_{i,\alpha} e_{i,\alpha})$$

$$-\beta e_{i,\alpha} = \ln P_{i,\alpha}$$

- $e_{i,\alpha}$  (effective energies) can be calculated from the occupation numbers at sites:  $(i,\alpha)$ ,  $(i+1,\alpha)$ ,  $(i,\alpha+1)$ ,  $(i+1,\alpha+1)$ ,
- dimension indexed by  $i$  --> spatial dimension
- dimension indexed by  $\alpha$  --> Trotter or imaginary time dimension
- the method is equivalent with the PATH integral formulation of the Quantum Mechanics

## Performing the MC simulation:

- we have four-site interaction between neighboring sites  $(i,\alpha)$ ,  $(i+1,\alpha)$ ,  $(i,\alpha+1)$ ,  $(i+1,\alpha+1)$  with the values of  $i$  and  $\alpha$  both even or odd.

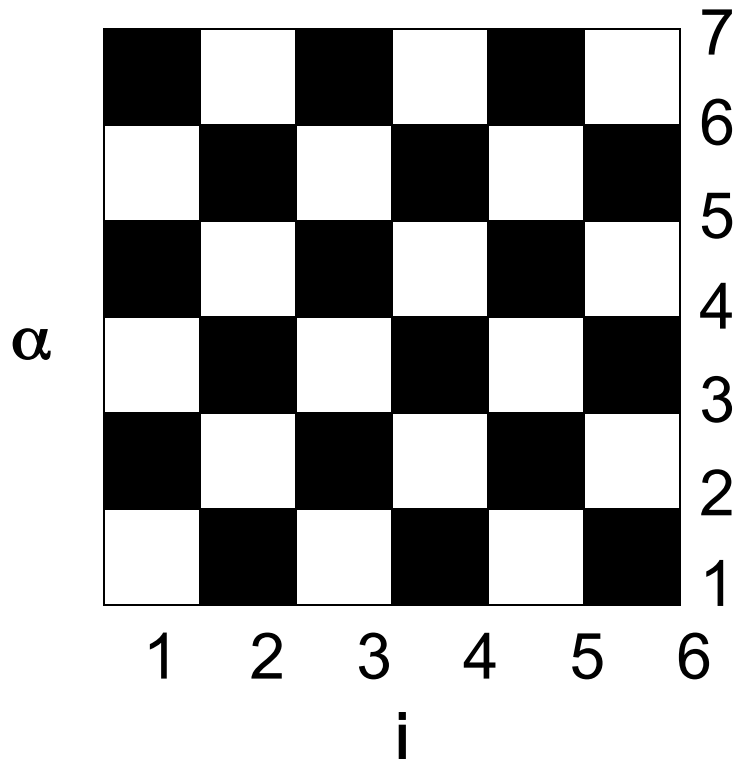
- in the  $i$  and  $\alpha$  space this interaction is represented by a check-board pattern where the interaction is around the positive quarter of the white plaquets

- the condition for  $P_{i,\alpha} \neq 0$  is to satisfy:

$$n_{i,\alpha} + n_{i+1,\alpha} = n_{i,\alpha+1} + n_{i+1,\alpha+1}$$

acceptable changes

- the occupation numbers cannot become negative!



## The QMC algorithm

- we fix  $M$  so that the values of  $\beta H_a/M$  and  $\beta H_b/M$  is much smaller than 1
- we consider an initial acceptable configuration for the  $n_{i,\alpha}$  numbers
- we change the values of the occupation numbers around a white plaquet as described earlier

- we calculate the change in the total energy of the system

$$\Delta E = \Delta \sum_{i,\alpha} e_{i,\alpha}$$

the sum being done on the neighboring dark plaquets

- we accept this change with a probability  
(Glauber dynamics) (Metropolis or other  
dynamics is also OK!)

$$P = \frac{\exp(-\beta\Delta E)}{1 + \exp(-\beta\Delta E)}$$

- we continue the algorithm until thermodynamic equilibrium is reached
- after equilibrium is approached, we collect periodically the relevant data
- 1 MC step is defined as  $2M \cdot L$  trials

## Calculation of the $e_{i,\alpha}$ factors

- in general these “energies” are only numerically calculable, after Taylor expansion of the formula for  $P_{i,\alpha}$ !
- in some simple cases however, it is analytically possible
- example: spinless fermions with only nearest neighbor interactions. ( $n_{i,\alpha}=0,1$ )

$$e(1,1,1,1) = V_1 \frac{\Delta\tau}{\beta}$$

$$e(0,0,0,0) = 0$$

$$e(0,1,0,1) = e(1,0,1,0) = \frac{p}{\beta} - \frac{1}{\beta} \ln[\cosh(p)]$$

$$e(0,1,1,0) = e(1,0,0,1) = \frac{p}{\beta} - \frac{1}{\beta} \ln[\sinh(p)]$$

$$p = t\Delta\tau$$

$$t = \frac{h^2}{8\pi^2 m a^2}$$

# Determination of the relevant physical quantities

$$E = -\frac{\partial \ln Z}{\partial \beta}; E = \frac{1}{Z} \sum_{\{n_{i,\alpha}\}} \frac{\partial}{\partial \beta} [\beta \sum_{i,\alpha} e_{i,\alpha}] \exp(-\beta \sum_{i,\alpha} e_{i,\alpha})$$
$$E = \langle e_{i,\alpha} \rangle + \left\langle \beta \frac{\partial e_{i,\alpha}}{\partial \beta} \right\rangle$$
$$C = -\beta^2 \frac{\partial E}{\partial \beta}$$

The formula for  $M$  and  $\chi$  in the  $z$  direction is the same as in the classical case!

## 2D and 3D cases

- the TS approximation is again applicable
- in the QMC method the dimensionality of the problem increased by one
- for the applicability of the TS transformation  $M$  has to be big --> the problem becomes technically difficult... to computer-time demanding