Lattice gas models

- Lattice gas : diffusion
- Diffusion Limited Aggregates
- Surface growth models
- Percolation

...

M. Peressi - UniTS - Laurea Magistrale in Physics
Laboratory of Computational Physics - Unit X

Random Walks

Dependence of $\langle R^2(t) \rangle$ on t :

- **normal** behavior: $\langle R^2(t) \rangle \sim t$
  for the brownian motion
- **superdiffusive** behavior: $\langle R^2(t) \rangle \sim t^{2\nu}$ with $\nu > 1/2$
  in models where self-intersections are unfavored
- **subdiffusive** behavior $\langle R^2(t) \rangle \sim t^{2\nu}$ with $\nu < 1/2$
  in models where self-intersections are favored

t (time) $\longleftrightarrow$ N (number of steps); $t = N\,\Delta t$

RW and diffusion

- consider the normal behaviour: $\langle R^2(t) \rangle \sim t$

We define the **autodiffusion coefficient**:

for large t,

$$D(t) = \frac{1}{2dt} \langle \Delta R(t)^2 \rangle$$

where d is the dimensionality of the system;

for large t, D should go asymptotically to a constant value

For d=1: $\langle R_N^2 \rangle = N\ell^2$

$$D = \frac{1}{2dt} \langle \Delta R(t)^2 \rangle = \frac{N\ell^2}{2t} = \frac{1}{2\Delta t} \ell^2$$

(constant, in this case)

RW and diffusion in 1D

The probability that a RW of N steps (N large) ends at position x is given by:

$$P_N(x) = \sqrt{\frac{2}{\pi N}}\ \exp\left(-\frac{x^2}{2N}\right)$$

Considering that $t = N\Delta t$, defining $D = \frac{\ell^2}{2\Delta t}$, and measuring x in units of ℓ, we get:

$$P(x,t) = \sqrt{\frac{1}{\pi D t}}\ \exp\left(-\frac{x^2}{4Dt}\right)$$

which is the fundamental solution of the diffusion equation, a part from a factor of 2 in the normalization due to the spatial discretization. The continuum solution is:

$$P(x,t) = \sqrt{\frac{1}{4\pi D t}}\ \exp\left(-\frac{x^2}{4Dt}\right)$$

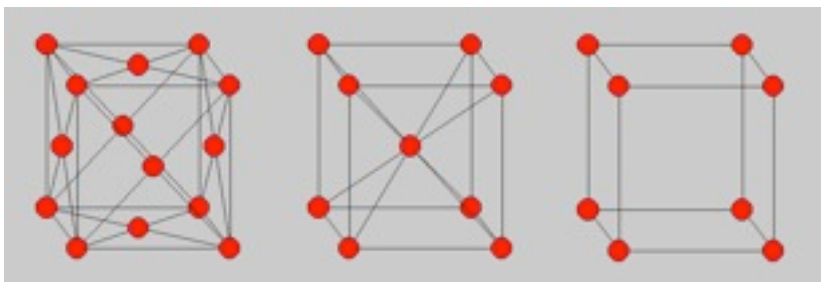
i.e., a Gaussian distribution with $\sigma^2 = 2Dt$ which describes a pulse gradually decreasing in height and broadening in width in such a manner that its area is conserved.

RW and diffusion on a 3D lattice

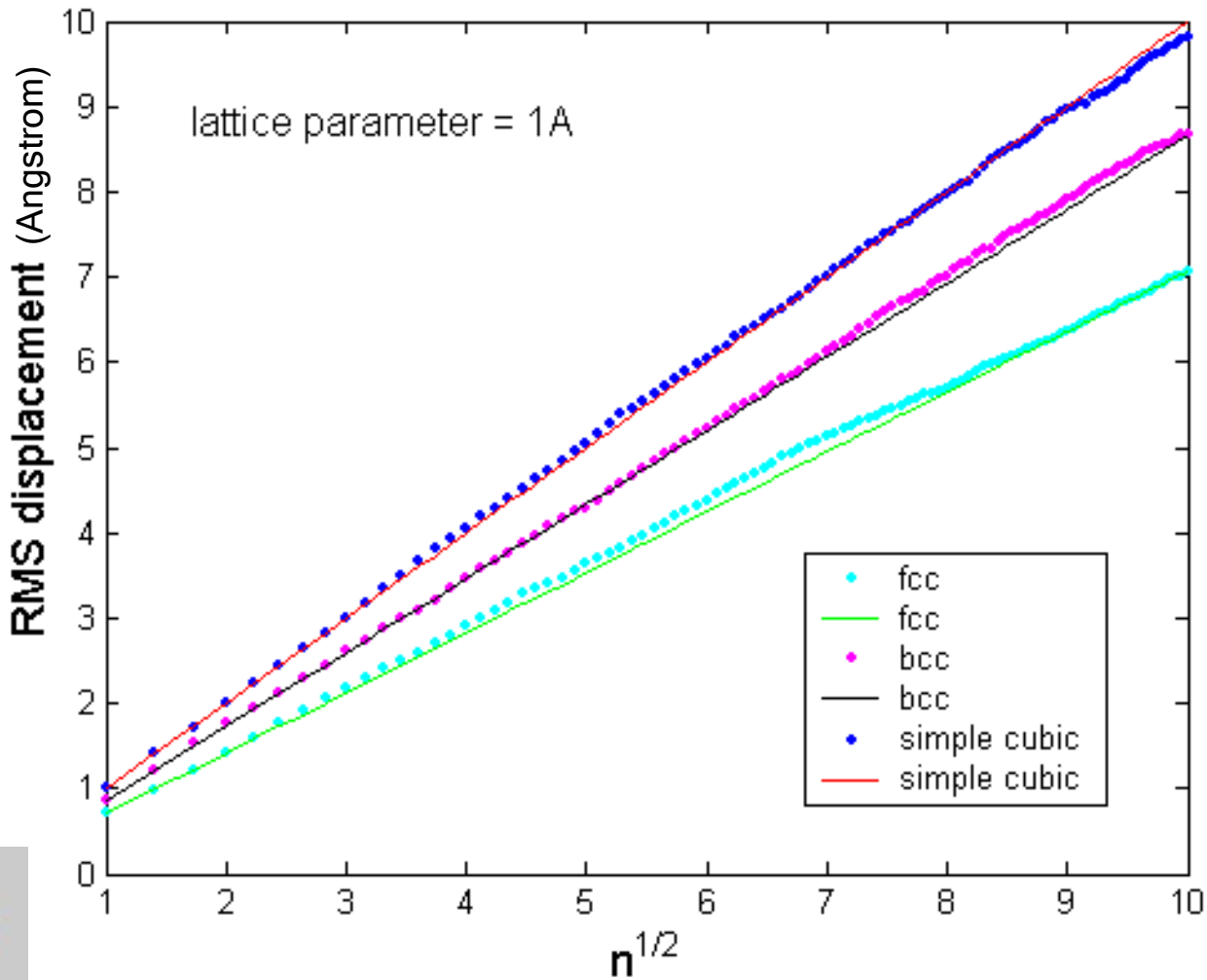
Discretized model:
if 1 step = move by 1 bond lenght,
we expect:

$$RMS = \sqrt{\langle R_N^2 \rangle} = \sqrt{N}\ d_{NN}$$

fcc : $d_{NN} = a_0 \frac{\sqrt{2}}{2} = 0.71 a_0$
bcc : $d_{NN} = a_0 \frac{\sqrt{3}}{2} = 0.87 a_0$
sc : $d_{NN} = a_0$



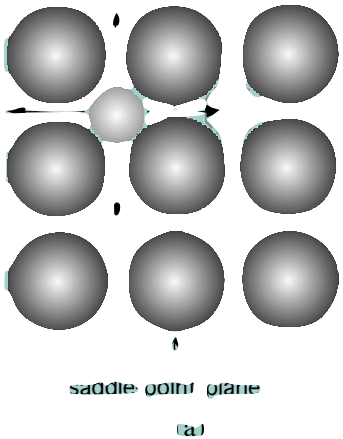
fcc bcc sc



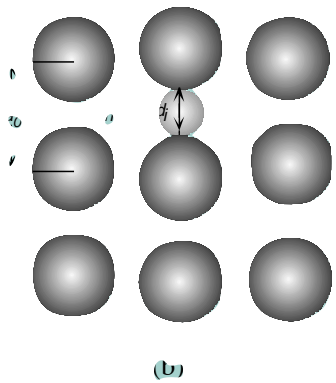
(number of steps)

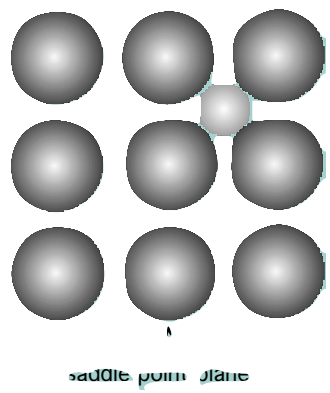
(=> D depends on the structure of the lattice)

Example of diffusion in solids



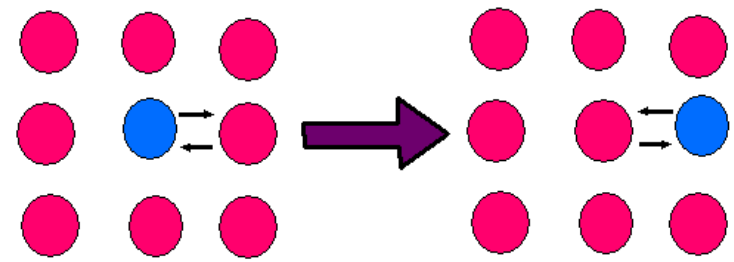
INTERSTITIAL IMPURITIES



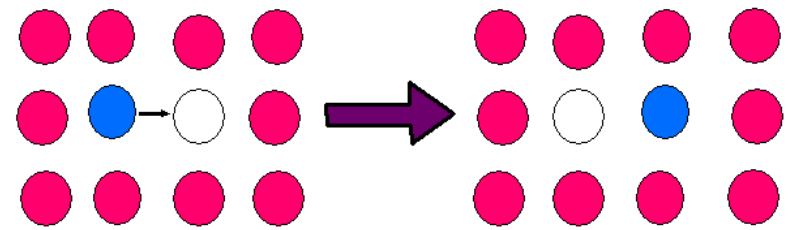


SUBSTITUTIONAL IMPURITIES

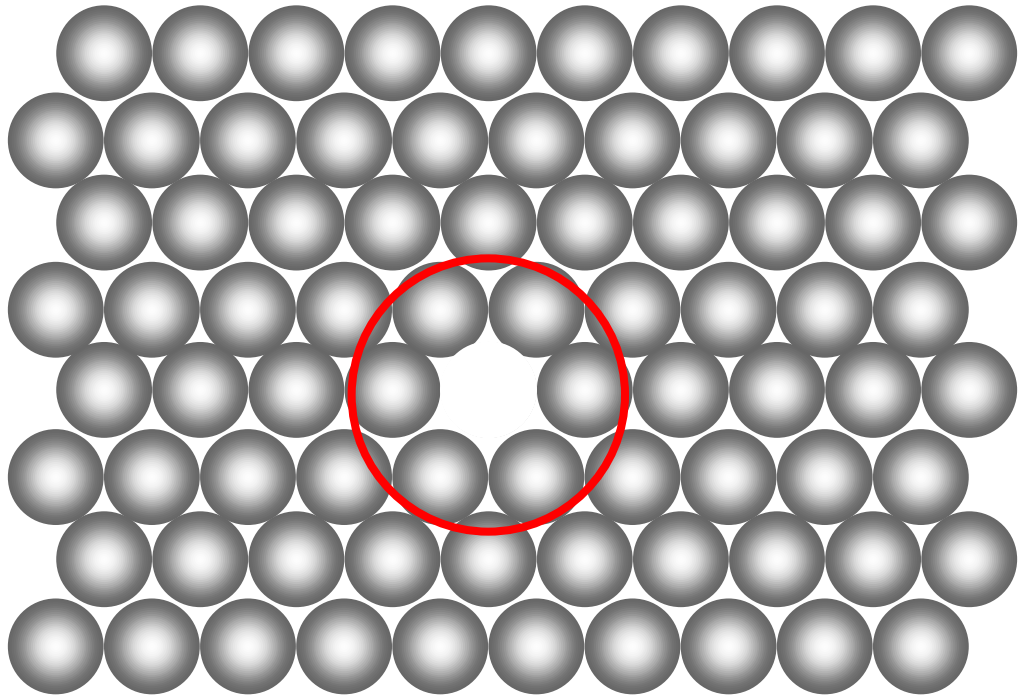
Direct exchange:



Vacancy assisted diffusion:



Vacancies diffusion in solids

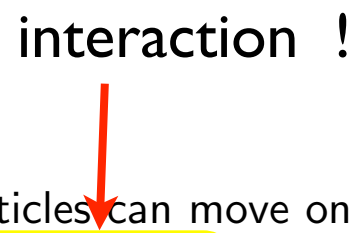


vacancies themselves can diffuse!

... but typically:
more than one single interstitial,
more than one single impurity,
or more than one single vacancy....

A SIMPLE RW MODEL IS NOT ENOUGH!

Lattice Gas model



Consider a finite lattice with some density ρ of N_p particles. The particles can move on the lattice by jumps to the nearest sites, but two particles can not occupy the same site. This is a simple example of a **restricted** random walk (see above). The physical interpretation is e.g. vacancies moving in a lattice.

To simulate this kind of system, we need a bit more of an advanced approach than before. First of all, we need to simulate the motion of all the particles at the same time, not taking the average over many independent single-particle motions as was done before.

2D Lattice Gas model

1° Choose number of particles N_p, number of steps N_{steps}, side length L. Set Δt and lattice size a. $(\text{our old } \ell)$

2° Set all positions in the $L \times L$ grid to be empty

3 a° Generate N_p particle coordinates randomly on the grid, checking that no two particles end up on the same points.

3 b° Mark the points with the particles in the $L \times L$ grid as filled.

4° Loop over MC steps of time Δt

  5° Loop from 1 to N_p

    6° Pick one particle i at random

    7° Find which positions it can jump to. If none, return to step 6° (*)

    8° Let the particle jump to one of the allowed directions j by a displacement $x_i = x_i + \delta x_j, y_i = y_i + \delta y_j$, enforce periodic boundaries on x and y

    9° Set $dx_i = dx_i + \delta x, dy_i = dy_i + \delta x$ (where periodic boundaries **do not** play a role!)

  10° End loop from 1 to N_p

  11° Update time $t = t + \Delta t$

12° End loop over MC steps

13° Output $\langle \Delta R^2 \rangle = \langle dx_i^2 + dy_i^2 \rangle$ and calculate diffusion coefficient $D(t) = \frac{1}{2dt} \langle \Delta R(t)^2 \rangle$

average over the particles

Lattice Gas model

(*) Different dynamics can be implemented, for instance:

- find which nearest neighbor sites are free and jump in one of them randomly chosen (if any) (this is actually mentioned in the previous slide and implemented in the code we are going to discuss) OR
- choose randomly one nearest neighbor site and jump only if it is free

Different dynamics => different behavior with concentration

Lattice Gas model

The crucial difference here to the previous random walk algorithms is that the outer loop goes over MC steps, the inner one over particles. When the walkers are independent of each other ("non-interacting") we can deal with one walker at a time, saving memory since storage of all particles is not needed.

But here the walkers (the particles) are "interacting"

Programs:

on
$/home/peressi/comp-phys/X-latticegas-fract/
[do: $cp /home/peressi/.../X-latticegas-fract/* .]
or on moodle2

latticegas.f90
dla2d.f90
eden.f90

Implementation of the model (latticegas.f90)

```
 ...
 logical,allocatable::lattice(:,:) ! (occ./non occ.=.true./.false.)
 integer,allocatable::x(:),y(:)    ! istantaneous positions
 double precision, allocatable :: dx(:),dy(:) !displ. from beginning
 integer :: free(4),nfree          ! occupation of nearest neighbors
 integer :: dxtrial(4),dytrial(4)  ! trial move on the square latt.
 integer :: xnew(4),ynew(4)        ! 4 new possible pos. in SQ latt.

  .....
 allocate(lattice(0:L-1,0:L-1))
 allocate(x(Np),y(Np))
 allocate(dx(Np),dy(Np))

  ...
 lattice = .false.          ! Mark all positions as empty

  ...
! Enumerate directions: 1=right; 2=left; 3=up; 4=down
 dxtrial(1)=+1; dytrial(1)= 0;
 dxtrial(2)=-1; dytrial(2)= 0;
 dxtrial(3)= 0; dytrial(3)=+1;
 dxtrial(4)= 0; dytrial(4)=-1;
```

```
! INIZIALIZE THE LATTICE : Generate Np particles on LxL lattice
  do i=1,Np

    do ! Loop until empty position found,  UNBOUNDED LOOP!
        call random_number(rnd)    !which has dimension(2)
        x(i)=int(rnd(1)*L)
        y(i)=int(rnd(2)*L)
        if (lattice(x(i),y(i))) then
           ! Position already filled, loop to find new trial
           cycle   !REMEMBER: JUMP AT THE END OF THIS LOOP (NOT EXIT)
        else
           lattice(x(i),y(i))=.true.
           !  Successful, place next particle
           exit
        endif
     enddo
     dx(i)=0.0d0; dy(i)=0.0d0;

  enddo
```

```
! MONTE CARLO LOOP

 do istep=0,Nsteps-1 ! Loop over MC steps
     do isubstep=1,Np ! Move all particles on ave. once every MC step
        ! Pick one particle at random
        call random_number(rnd1)
        i=int(rnd1*Np)+1  ! 1 =< i =<  Np;

     ! Find possible directions (j=1,...,4) for moving, store them
  in free() ... (NOTE: different possible recipes !!!)
     ! If no free positions, get a new particle ; otherwise choose
     ! one possible direction (j) and update (x,y) with (xnew,ynew):
         ......
        !Empty  old  position  and  fill  new
        lattice(x(i),y(i))=.false.
        lattice(xnew(j),ynew(j))=.true.

     enddo
     t=t+deltat
 enddo
```

Another fundamental part (look at it!): calculation of distance from initial pos. for each particle (do not use PBC for that, remember!), accumulation of data...

```
! Get total displacement from dx,dy
dxsum=0.0d0; dysum=0.0d0;
dxsqsum=0.0d0; dysqsum=0.0d0;
do i=1,Np
dxsum=dxsum+dx(i);   dysum=dysum+dy(i);
dxsqsum=dxsqsum+dx(i)*dx(i);
dysqsum=dysqsum+dy(i)*dy(i);
enddo
print *,'dxsum',dxsum,' dysum',dysum
print *,'dxsqsum',dxsqsum,' dysqsum',dysqsum
```

Concentration dependent diffusion coefficient

And here is a series of results:

Np	L	concentration Np/L^2	D (cm^2/s)
---	----	------	------------
10	100	0.001	9.769973881166823E-008
10	100	0.001	1.127346430730184E-007
100	100	0.01	1.028685543050629E-007
100	100	0.01	9.469519884885580E-008
10000	1000	0.01	9.899003879678247E-008
1000	100	0.1	9.111043889255736E-008
1000	100	0.1	9.427090885414200E-008
100000	1000	0.1	9.403952985695557E-008
3000	100	0.3	8.284148565973272E-008
3000	100	0.3	7.915751903784448E-008
6000	100	0.6	5.895798261670045E-008
6000	100	0.6	5.913229928124830E-008
9000	100	0.9	1.771291645136659E-008
9000	100	0.9	1.786338311620434E-008
900000	1000	0.9	1.824779088931029E-008
9900	100	0.99	1.831247452488705E-009
9900	100	0.99	1.860272704661156E-009

Here: 2d example

$1\ \text{MC step} = 1\ \text{ns}$

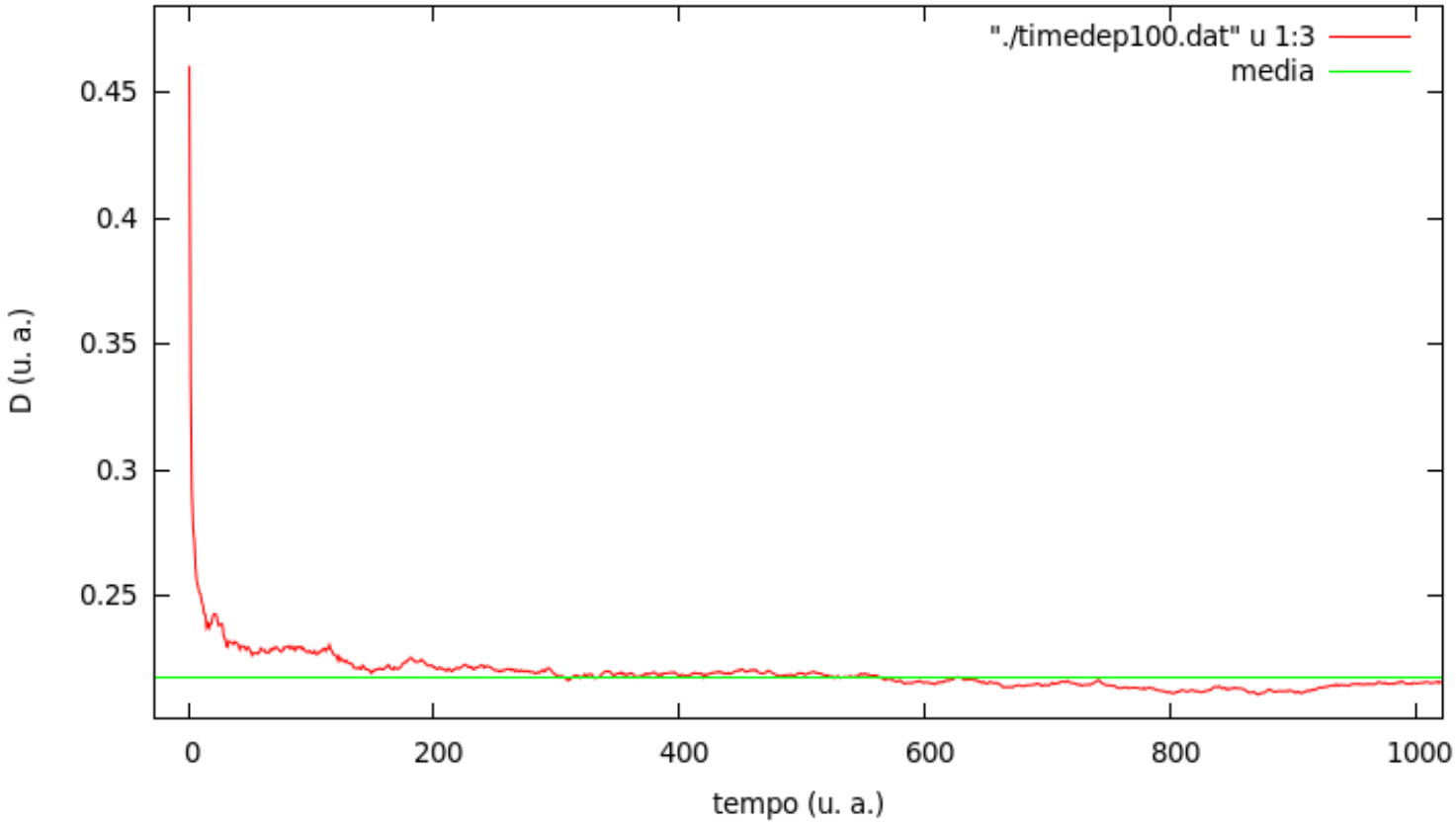
$\text{unit step length} = 2\ \text{Å}$

What does this mean? At small concentrations, the system behaves essentially as an unconstrained random walk. For that one, we know that $\langle \Delta R^2 \rangle$ should be equal to $a^2 N$, where N is the number of steps, and a is the jump distance, and the result for the diffusion coefficient should be

$$D = \frac{\langle \Delta R^2 \rangle}{4t} = \frac{(2\ \text{Å})^2 N}{4N\Delta t} = \frac{(2\ \text{Å})^2}{4 \times 1\ \text{ns}} = 10^{-7} \frac{\text{cm}^2}{\text{s}}$$

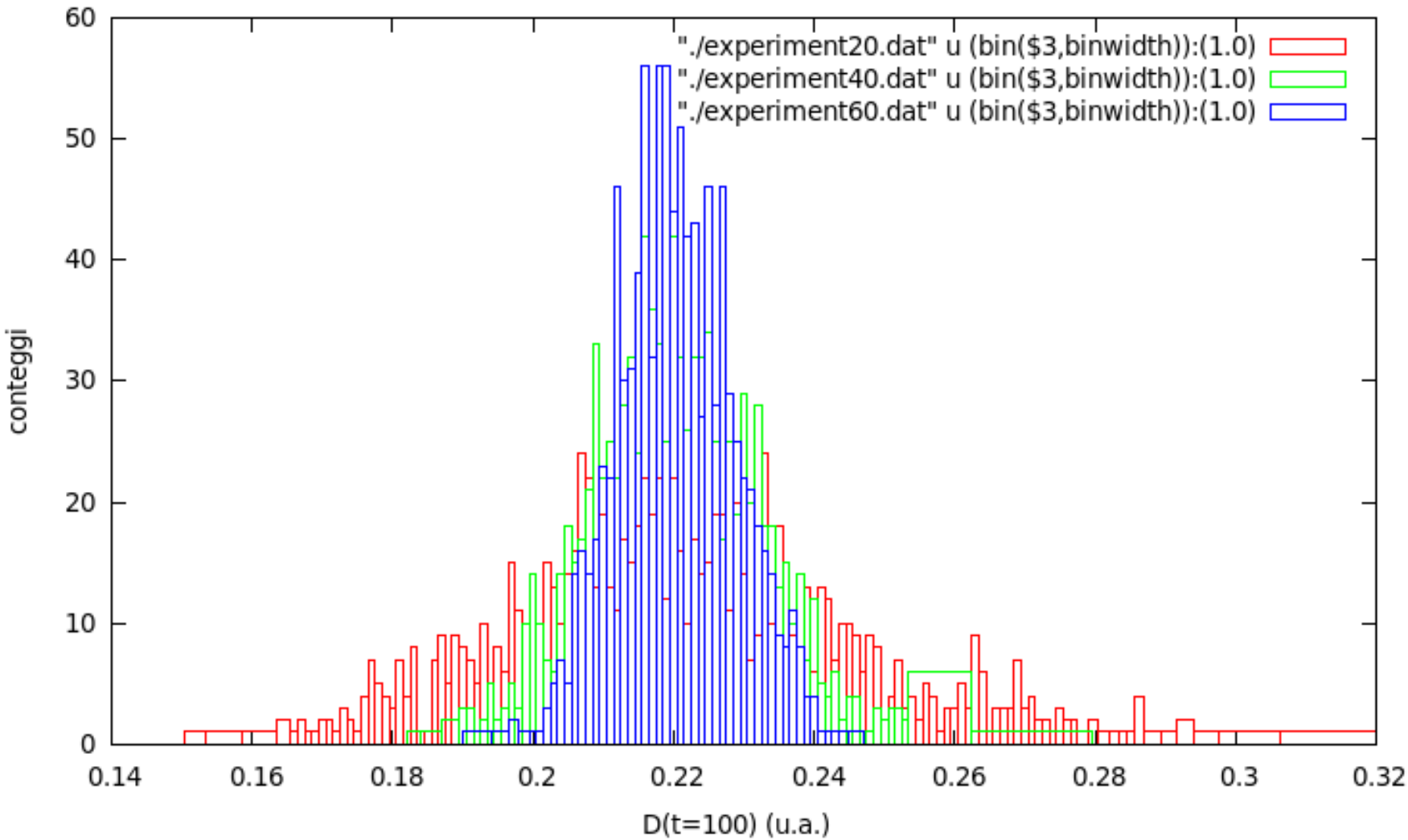
Discussing Ex. 1

(1.a) Study D(t) for a fixed value of ρ, for instance 0.2. Although D is defined as the limit $t \to \infty$, it is instructive to follow D(t) as a function of time: for this model, it fluctuates after a short equilibration time and no appreciable improvements in the statistics are achieved by increasing t.



this is D(t) (averaged over particles);
calculate it for $t \longrightarrow \infty$

A usually, we can estimate the statistical error
associated to the estimate of D
(here: histogram done collecting data in the time evolution of D(t))

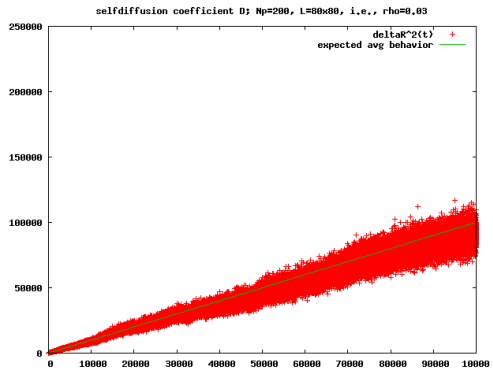


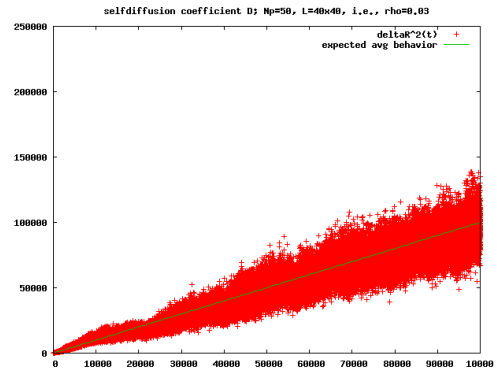
size effect in the determination of D (concentration ρ fixed)!
(more later)

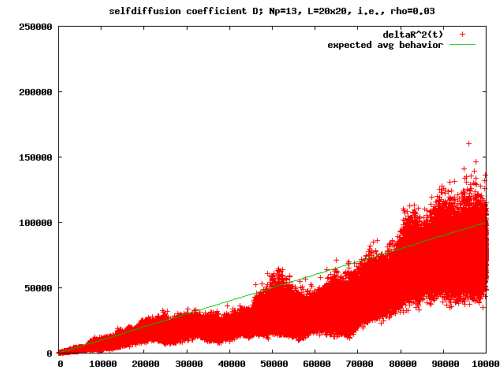
(I.1) ... Better statistics for D can be obtained by averaging D over as many particles as possible (i.e., for a given ρ)... Here **ρ=0.03**

Np=200, 80x80 | Np=50, 40x40 | Np=13, 20x20

$\Delta R^2(t)$ and expected behavior

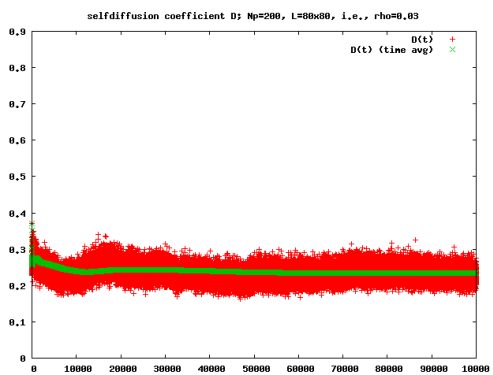


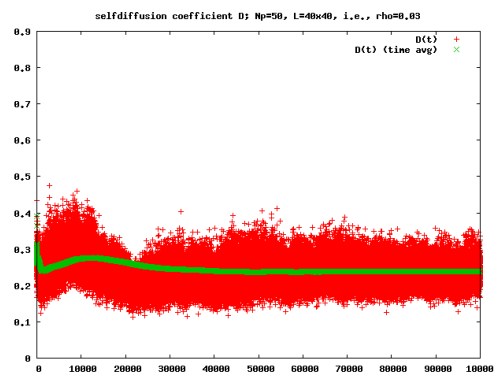


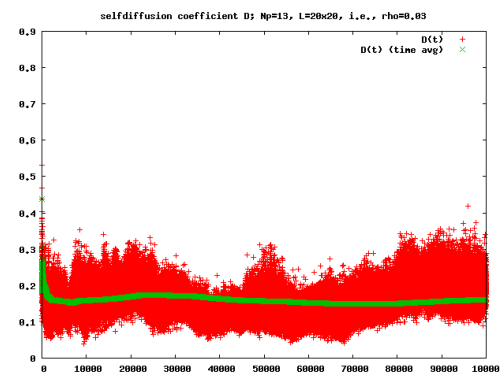


D(t) and <D(t)>

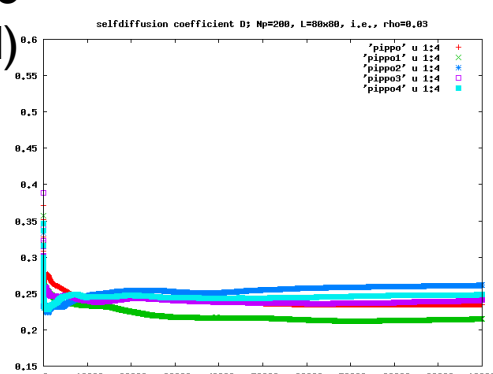
(<D(t)> is also time averaged)

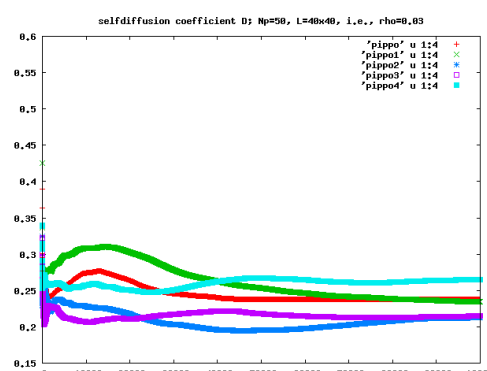


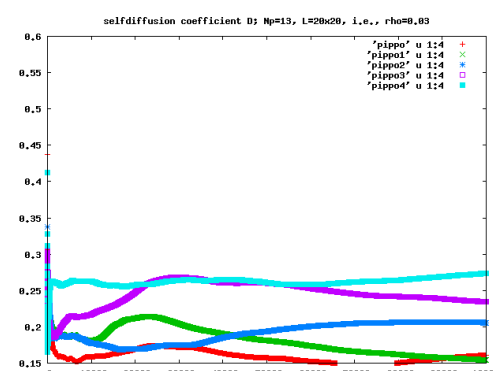




5 runs

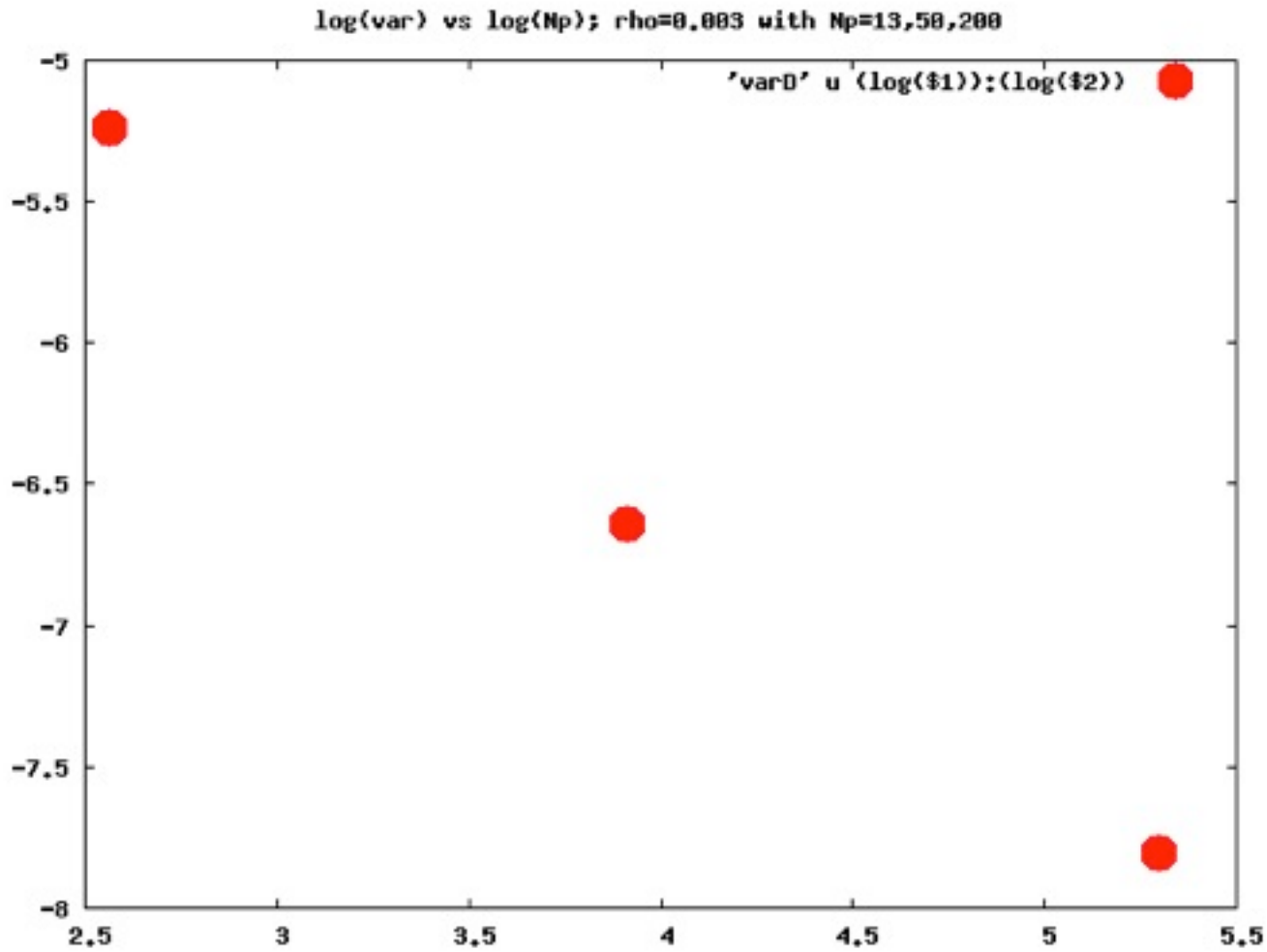






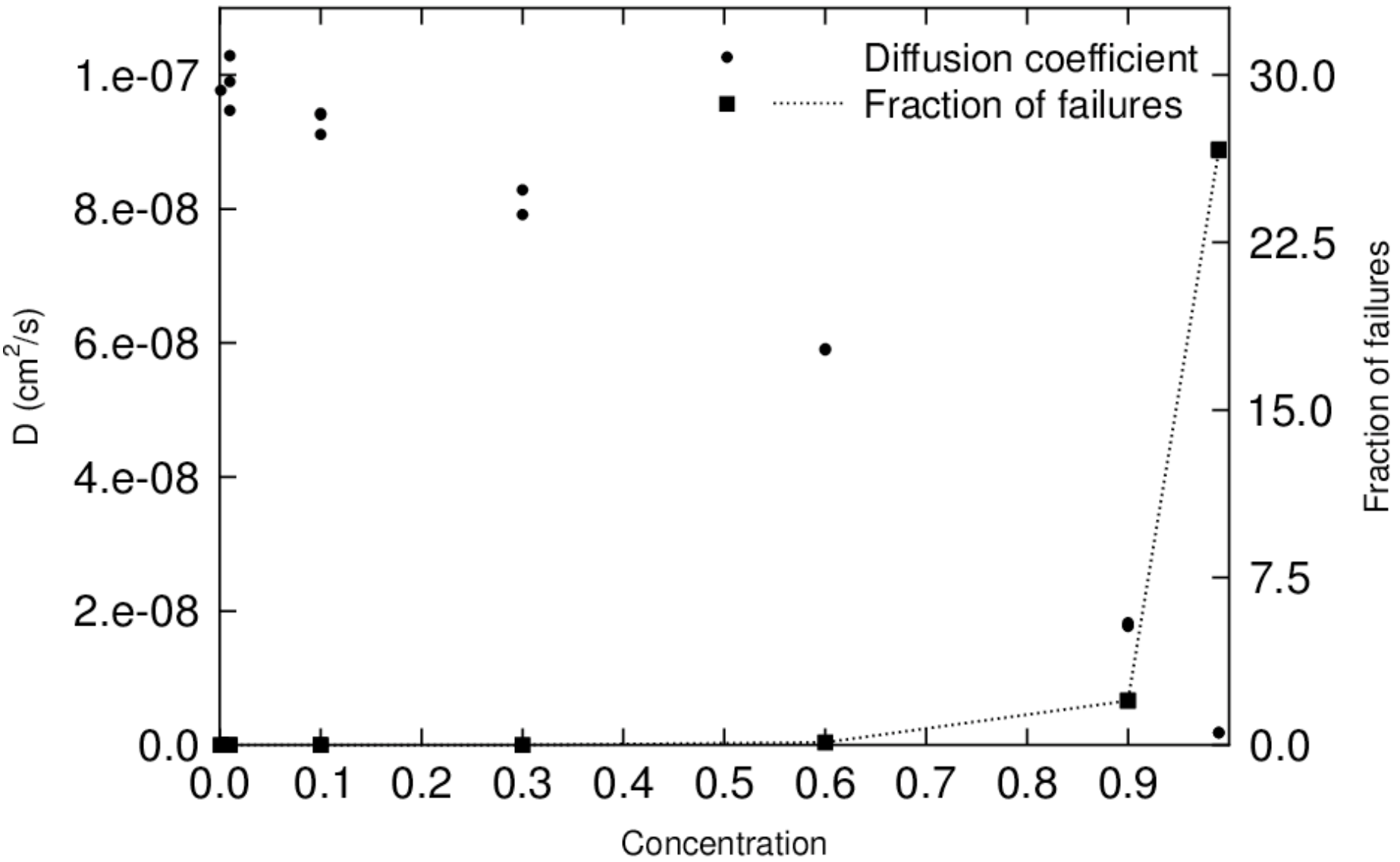
(we expect the limit of the simple 2D RW on a square lattice, with D=0.25)

Ex. 1 (...) Verify that deviations of D(t) from its mean value are proportional to the inverse square root of the total number of particles.



σ^2_D proportional to 1/Np

Concentration dependent diffusion coefficient



Addition of further interactions

- Attractive (J>0) nearest-neighbor (NN) interaction only: total energy of the system:

$$E = -\frac{J}{2}\sum_{\langle ij \rangle} n_i n_j$$

=> Trend to aggregation (diffusive behavior is limited to a transient)

- Add a repulsive (J<0) next-nearest-neighbor (NNN) interaction: total energy of the system:

$$E = -\frac{1}{2}\sum_{\langle ij \rangle} J_{ij} n_i n_j$$

=> The behavior depends on the ratio $R = J_{NNN}/J_{NN}$

- With finite NN and/or NNN interactions, temperature plays a role

Other models related to random walks

- diffusion limited aggregated (DLA)
- percolation

Diffusion Limited Aggregation

Several examples of formation of natural patterns showing common features:

Electrodeposition:
cluster grown from a copper sulfate solution in an electrodeposition cell

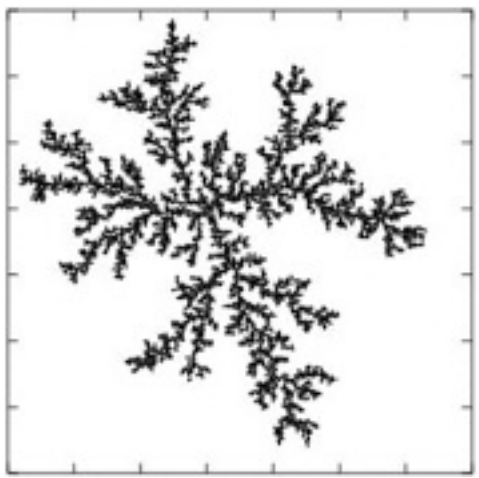
Dielectric breakdown:
High voltage dielectric breakdown within a block of plexiglas

These common features that can be captured by very simple models:

Diffusion Limited Aggregation

- simple model of FRACTALS GROWTH, initially proposed for irreversible colloidal aggregation, although it was quickly realized that the model is very widely applicable.
- by T.A. Witten and L.M. Sander, 1981

REAL IMAGE (Atomic Field Microscopy) of a gold colloid of about 15 nm over a gel substrate



SIMULATION

DLA: algorithm

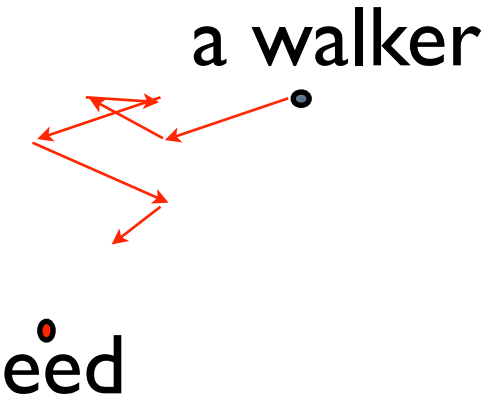
* Start with an immobile seed on the plane

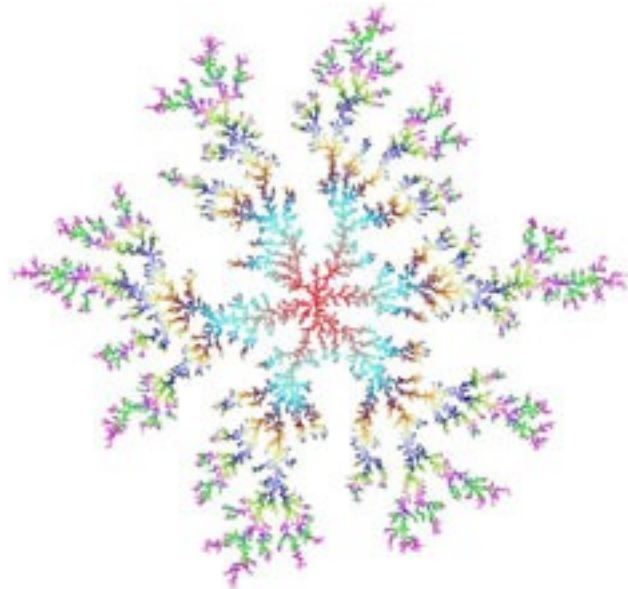
* A walker is then launched from a random position far away and is allowed to diffuse

* If it touches the seed, it is immobilized instantly and becomes part of the aggregate

* We then launch similar walkers one-by-one and each of them stops upon hitting the cluster

* After launching a few hundred particles, a cluster with intricate branch structures results

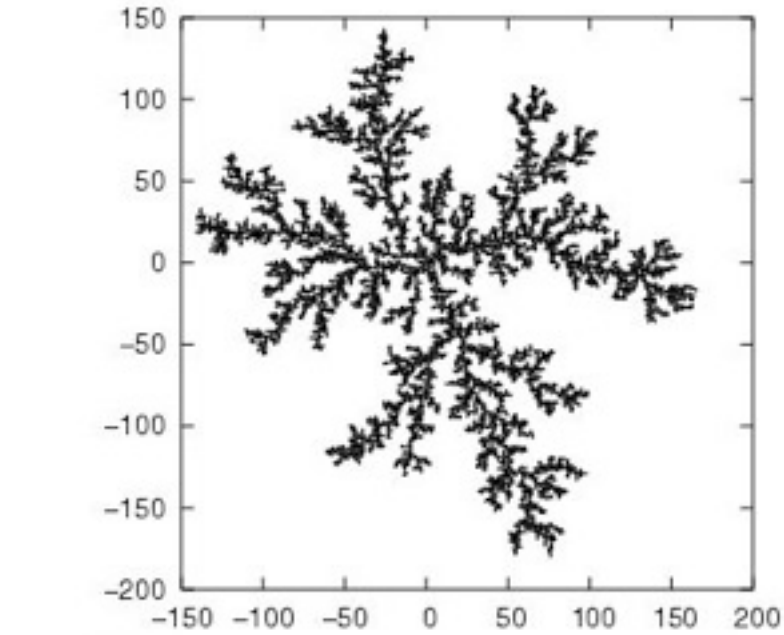


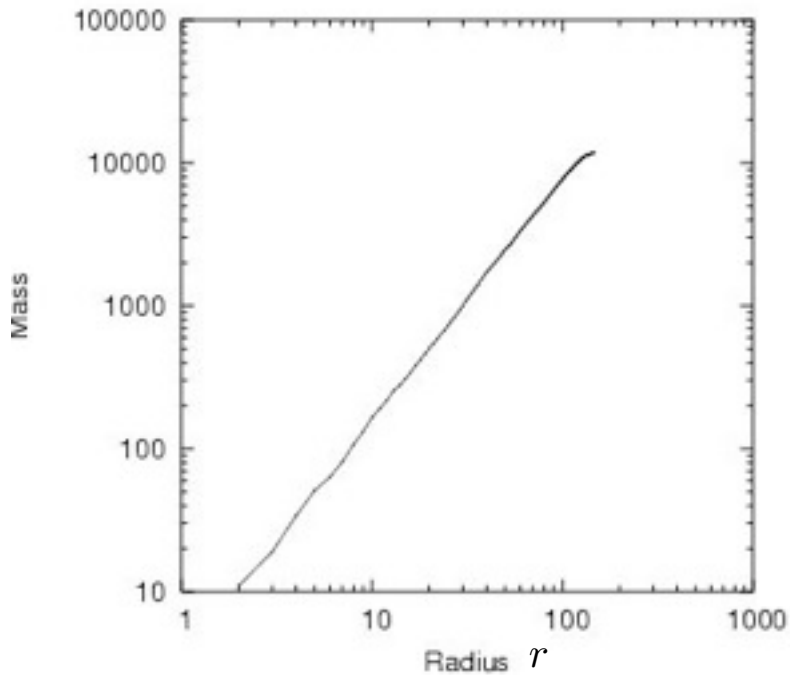


DLA: algorithm - details

- We launch walkers from a "launching circle" which inscribes the cluster
- They are discarded if they wander too far and go beyond a "killing circle"
- The diffusion is simulated by successive displacements in independent random directions
- After every step, all particles on the cluster are checked to detect any overlapping with the walker which would aggregate

DLA: results





(mass M of the cluster = number of particles N)

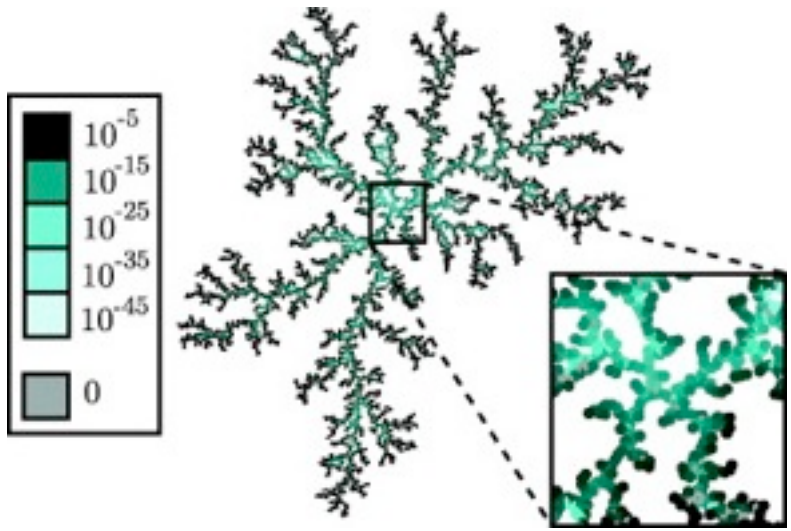
$$\ln N \propto \ln r$$

$$\Downarrow$$

$$N \propto r^k$$

DLA: interesting quantities

- in a "normal" 2D object: $N \propto r^2$
- FRACTAL DIMENSION: the number of particles N with respect to the maximum distance r of a particle of the cluster from its center of mass is $N \propto r^{D_f}$, with $1 < D_f < 2$

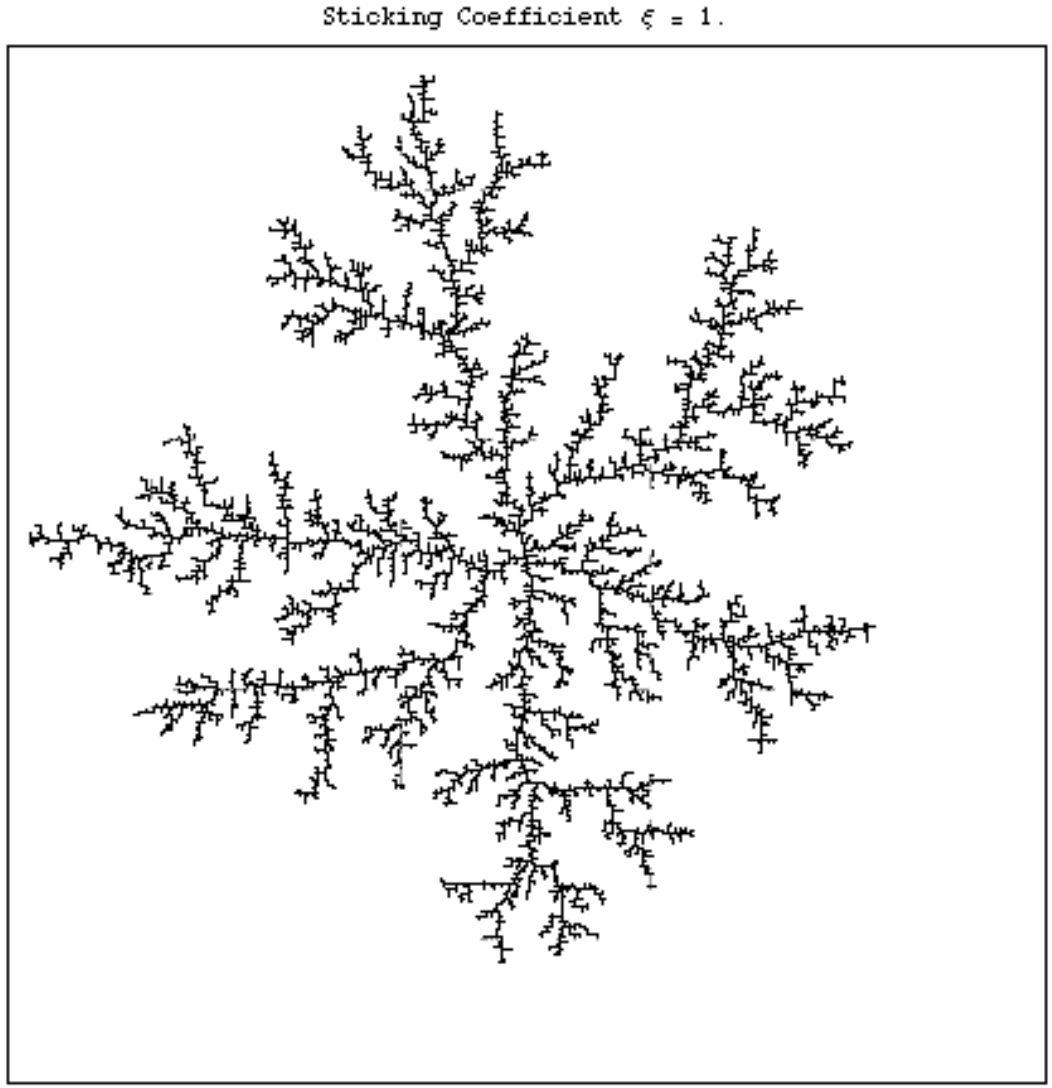


DLA: algorithm - details II

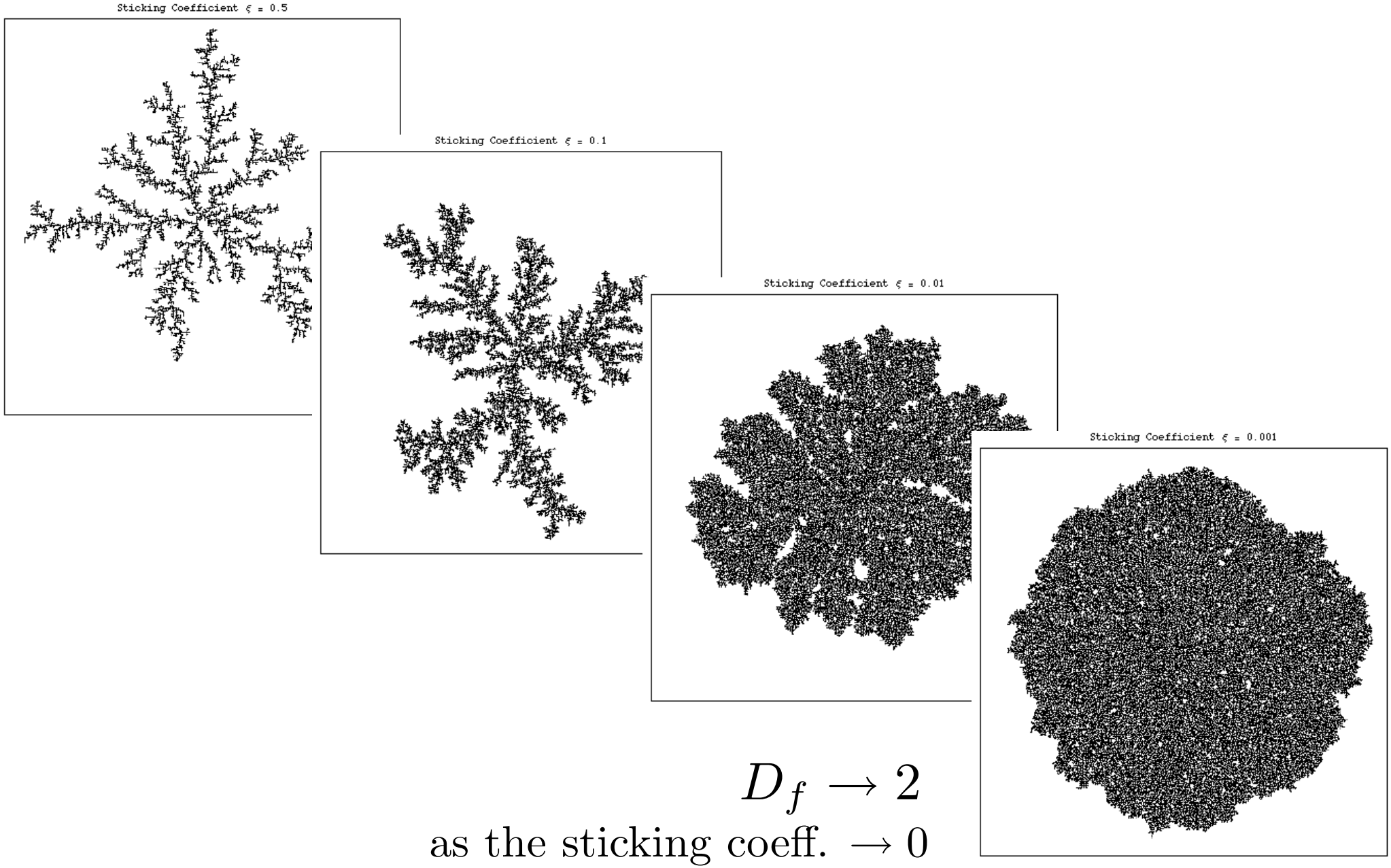
- the simplest DLA models: diffusion on a lattice. On a **square lattice**, 4 adjacent sites are available for the diffusing particle to stick
- It will stick with certain probability (the "sticking coefficient") - to simulate somehow the surface tension
- (a bit more complicate models: with a sort of Brownian diffusion in a continuous way)

DLA: results

$$1 < D_f = 1.6 < 2$$



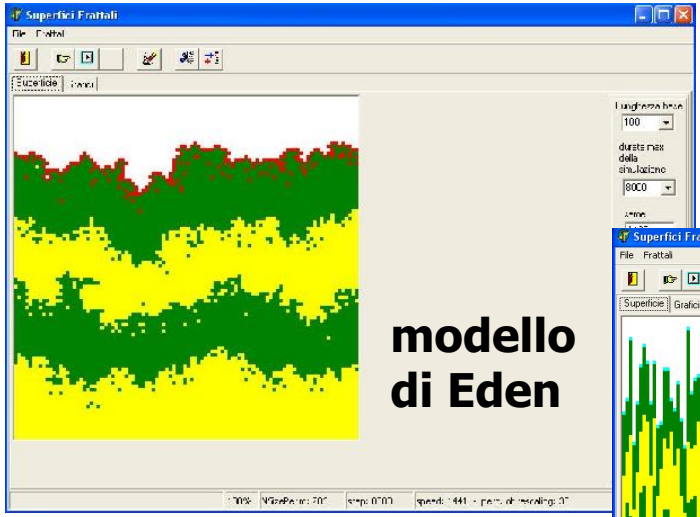
DLA: results



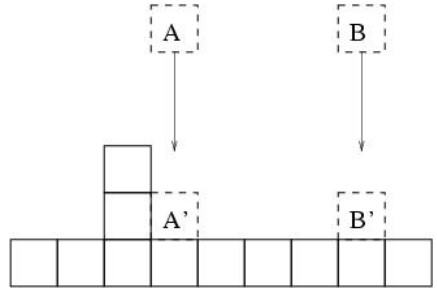
$$D_f \to 2$$

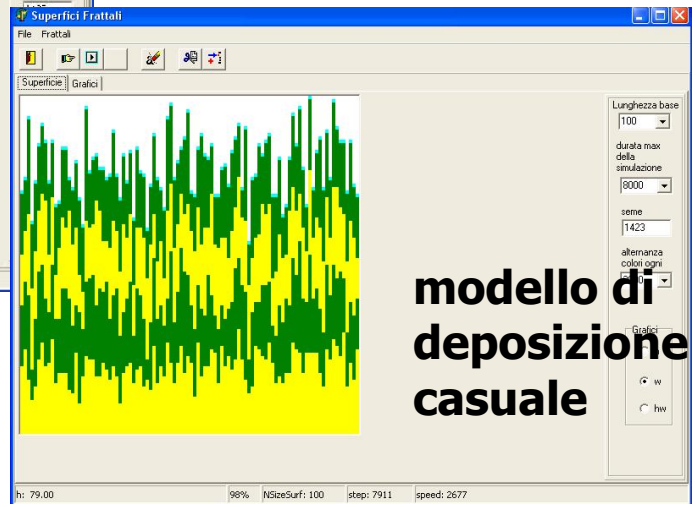
as the sticking coeff. $\to 0$

Models of surface growth

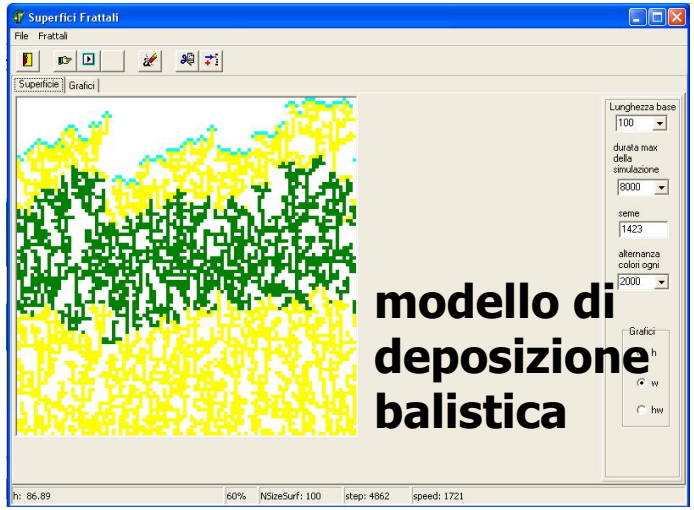


modello di Eden

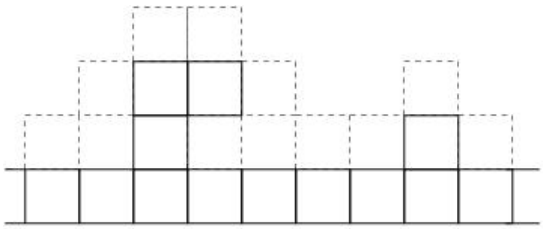


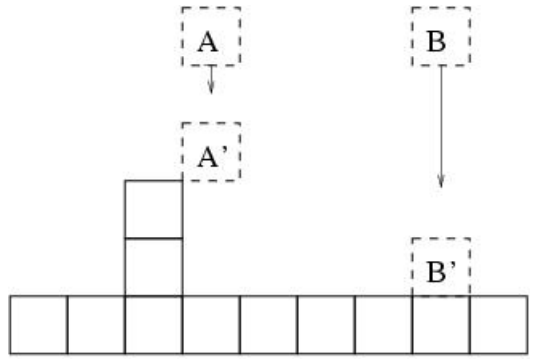


modello di deposizione casuale



modello di deposizione balistica

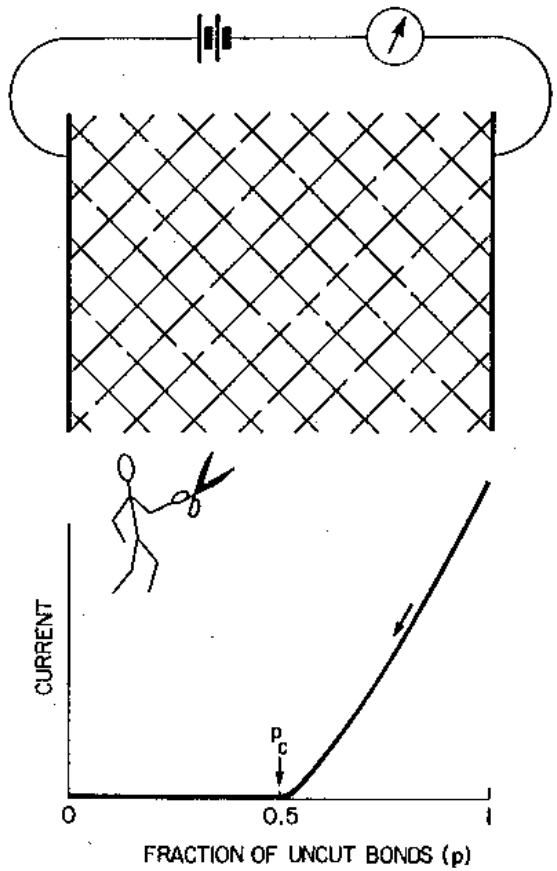


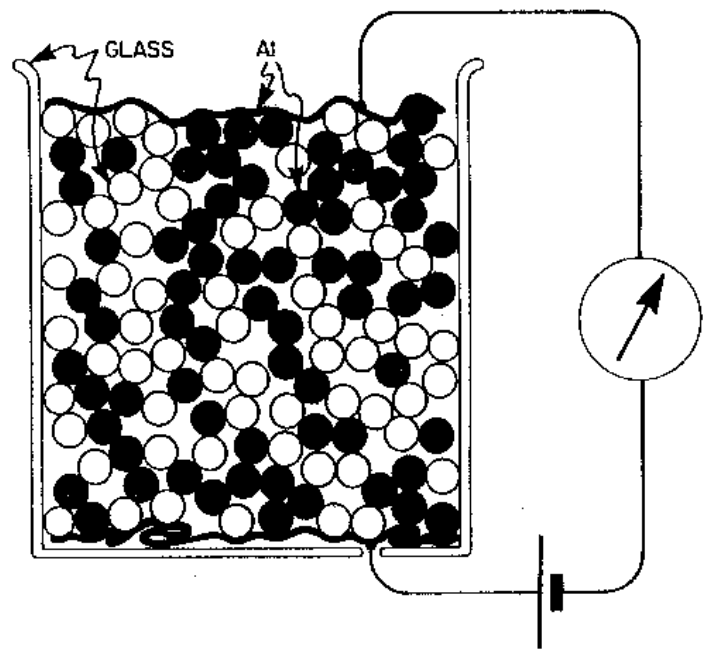


see e.g. Barabasi & Stanley, ***Fractal concepts in surface growth***, Cambridge University Press

Percolation

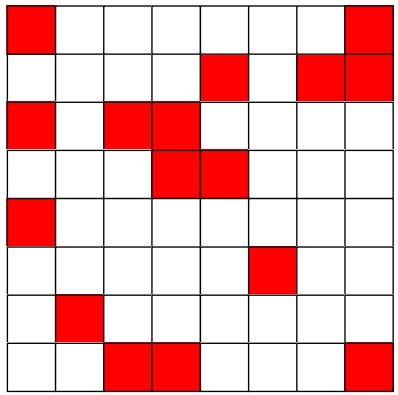
geometric connectivity in a stochastic system; modeling threshold and transition phenomena



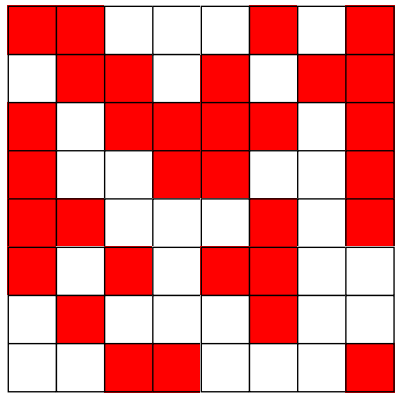


existence of a critical occupation fraction P above which spanning clusters occur (in nature: mixtures of conducting/insulating spheres...; resistor networks..)

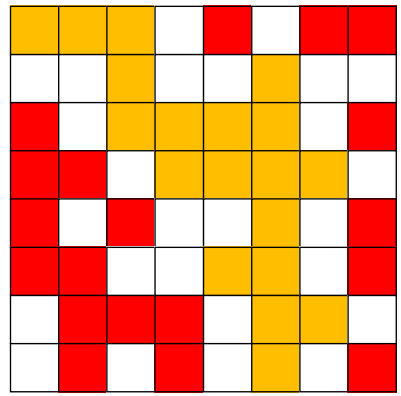
Percolation



L = 8 p = 0.25



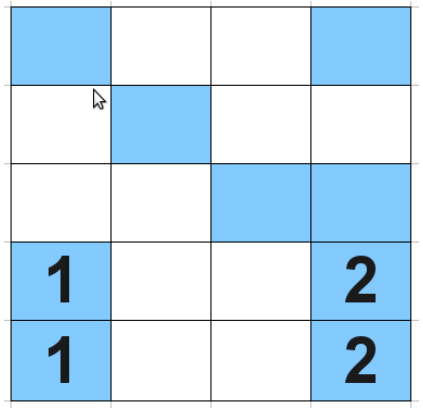
L = 8 p = 0.50



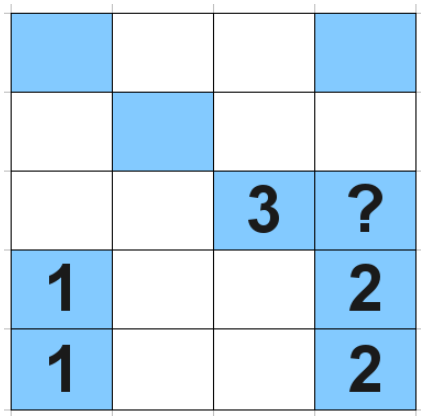
L = 8 p = 0.60

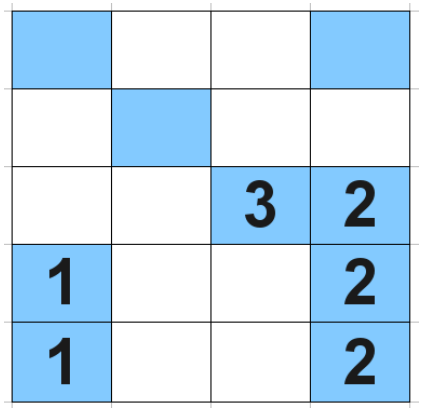
The (non trivial) part of the model:
choose a smart algorithm to identify and label the clusters made of adjacent occupied sites

Percolation

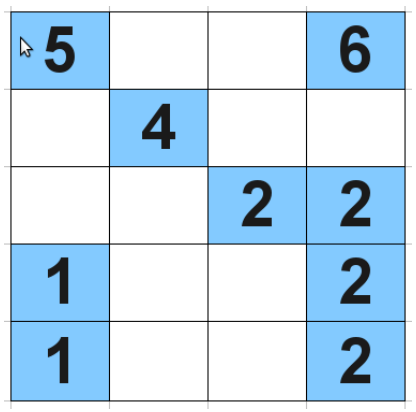


(1): span all the cells
(here: left => right
and bottom => up)
and start labeling





(2): attribute the minimum cluster label
to cells neighboring to different clusters



(3): refine labeling

Hoshen- Kopelman algorithm for clusters labeling