993SM - Laboratory of Computational Physics lecture 9 - part 1 May 6, 2020

Maria Peressi
Università degli Studi di Trieste - Dipartimento di Fisica
Sede di Miramare (Strada Costiera 11, Trieste)
e-mail: peressi@ts.infn.it
tel.: +39 040 2240242

Modelling other random processes

- Fractals & Diffusion Limited Aggregates
- Percolation

- Simulated annealing
- genetic algorithms

Other models related to random walks

- diffusion limited aggregated (DLA)
- percolation

Diffusion Limited Aggregation

Several examples of formation of natural patterns showing common features:

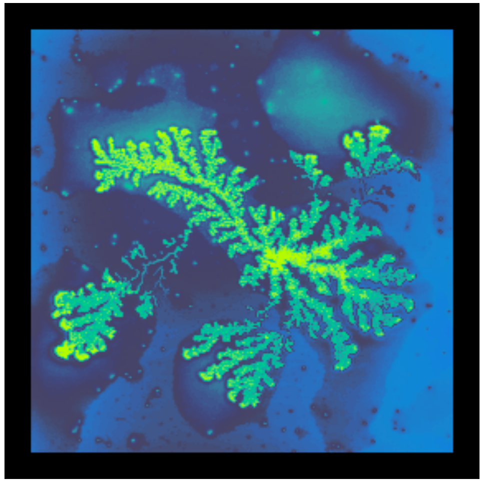
Electrodeposition:
cluster grown from a copper sulfate solution in an electrodeposition cell

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

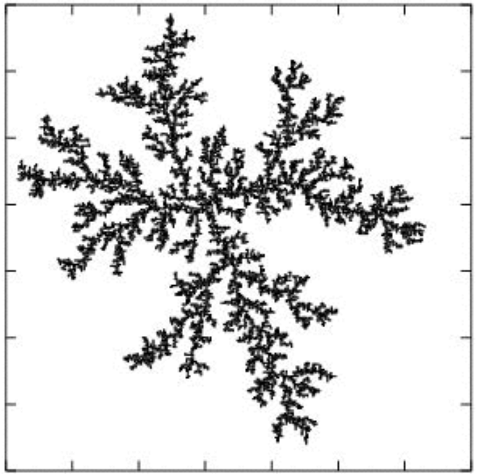
These common features that can be captured by very simple models:

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

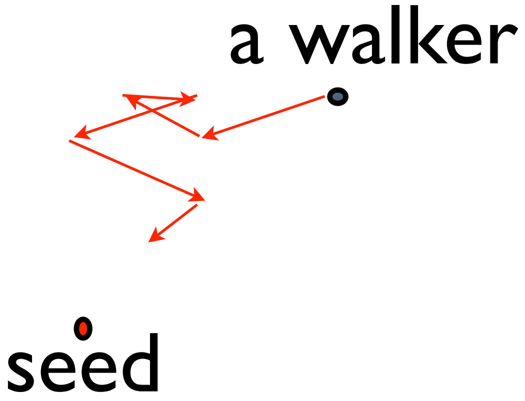
* Start with an immobile seed on the plane

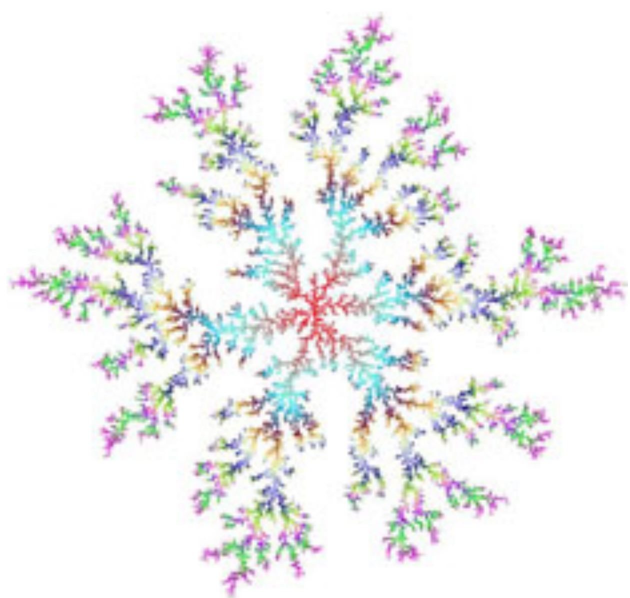
* A walker is then launched from a random position far away and is allowed to diffuse

* 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

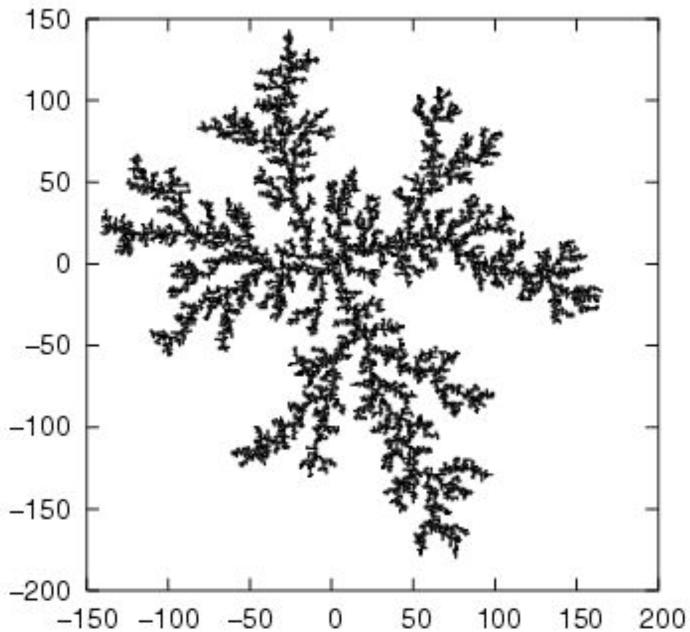


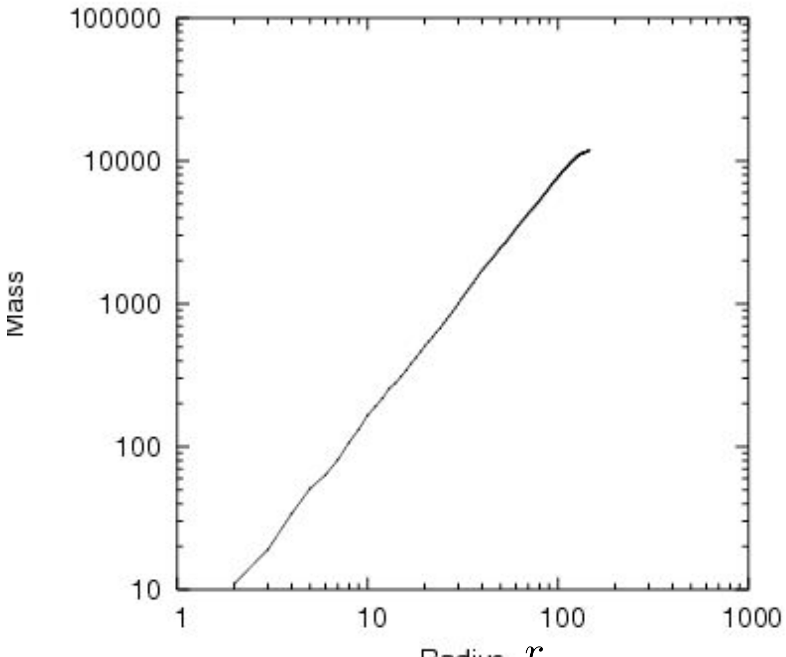


DLA: algorithm - details

- 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





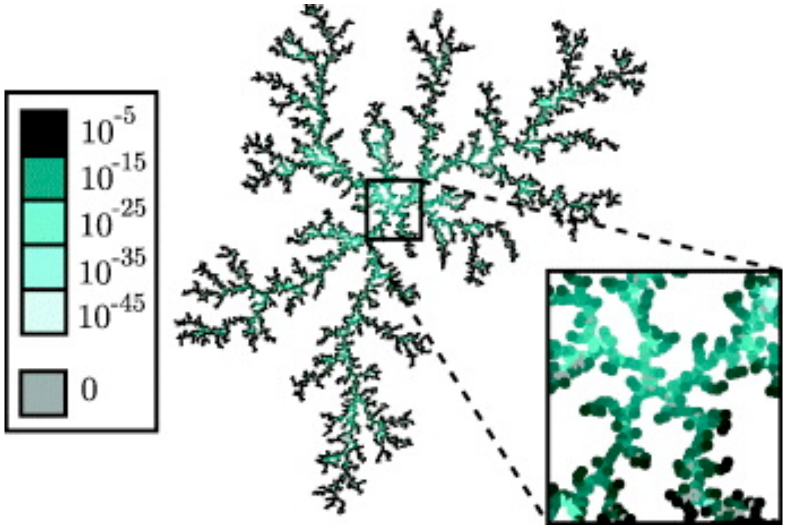
r

(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 $1 < D_f < 2$

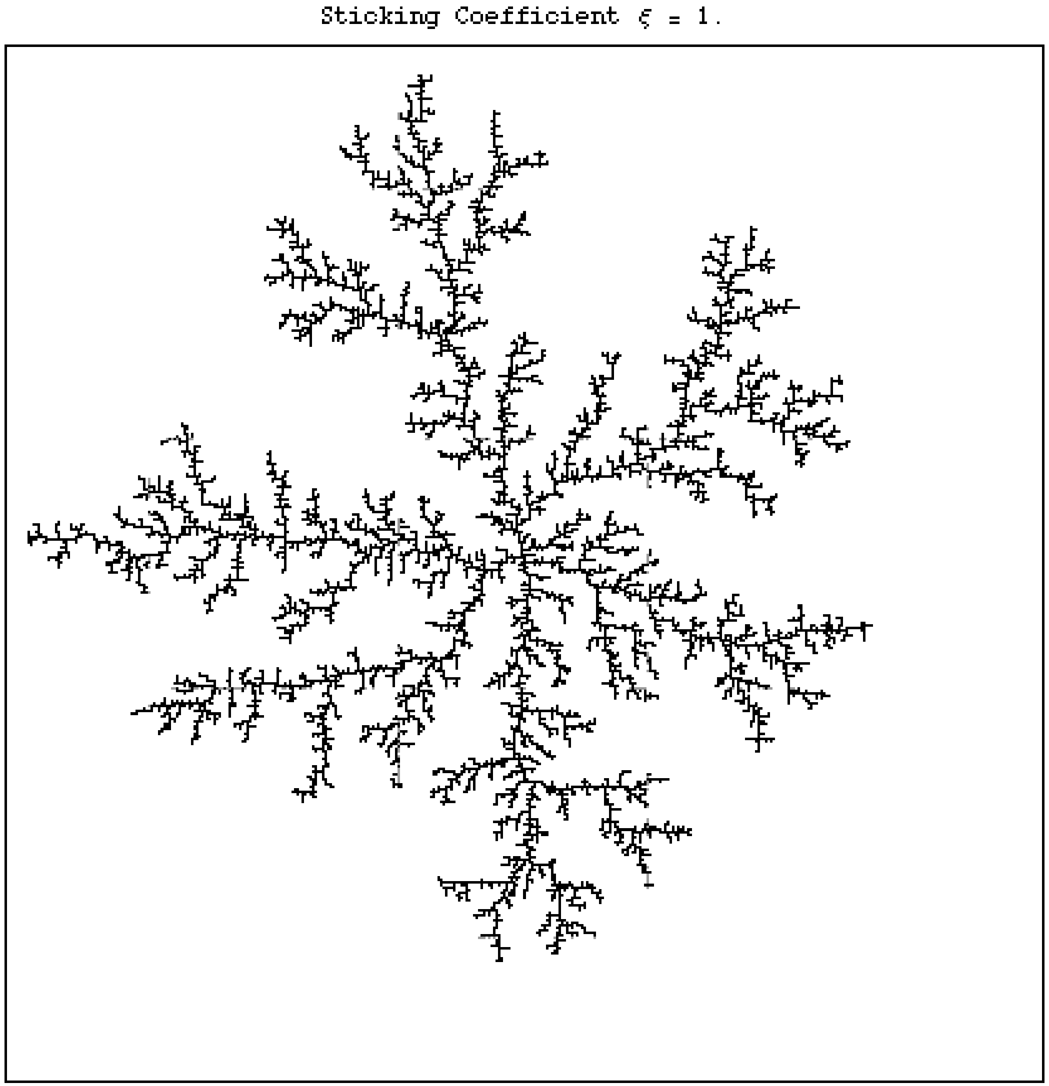


DLA: algorithm - details II

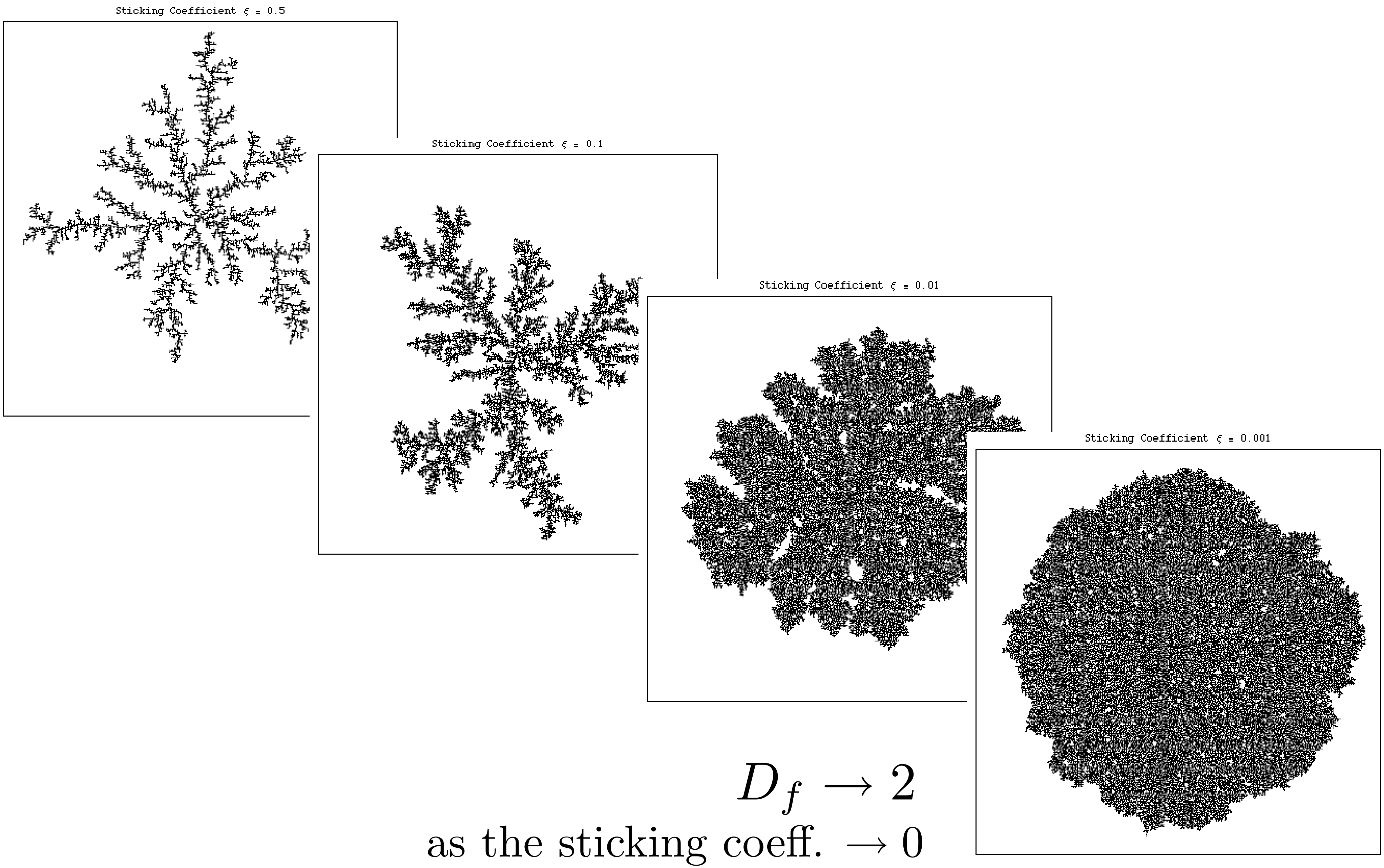
- the simplest DLA models: diffusion on a lattice. On a **square lattice**, 4 adjacent sites are available for the diffusing particle to stick
- It will stick with certain probability (the "sticking coefficient") - to simulate somehow the surface tension
- (a bit more complicate models: with a sort of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

DLA: results

$$1 < D_f = 1.6 < 2$$



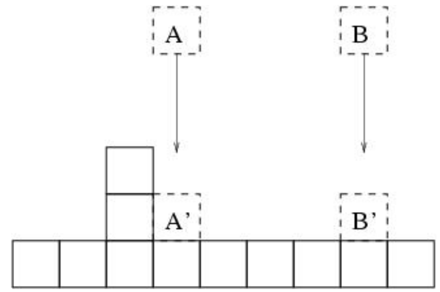
DLA: results

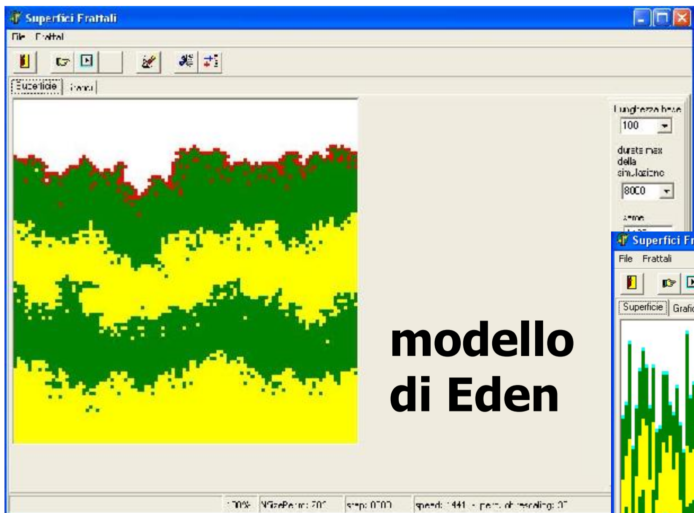


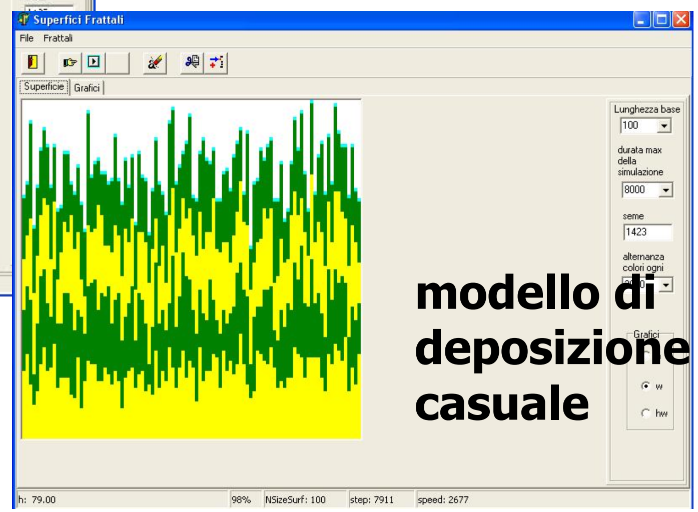
$$D_f \to 2$$

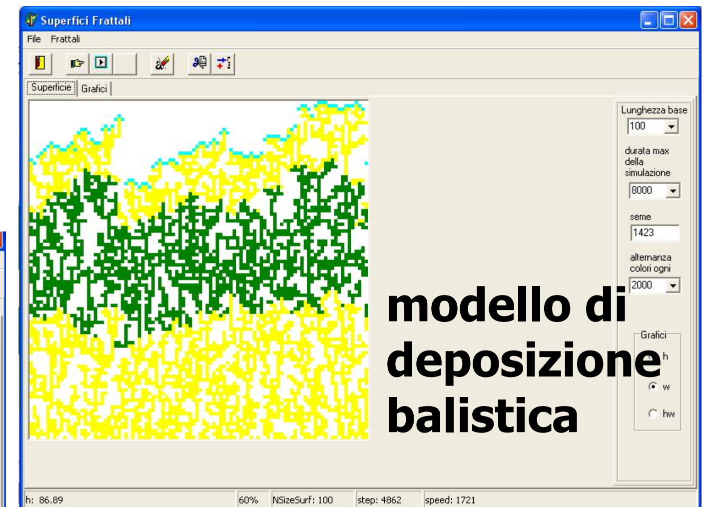
as the sticking coeff. $\to 0$

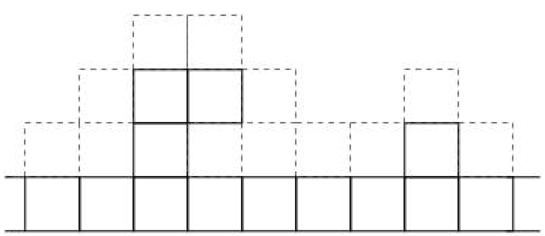
Models of surface growth

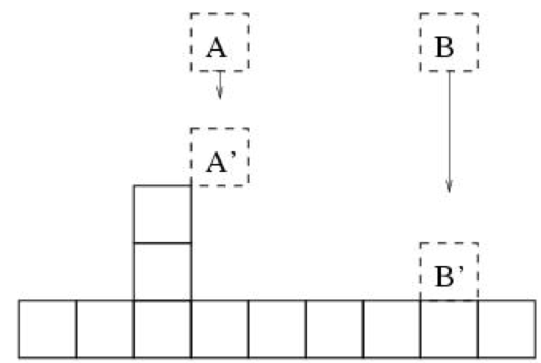








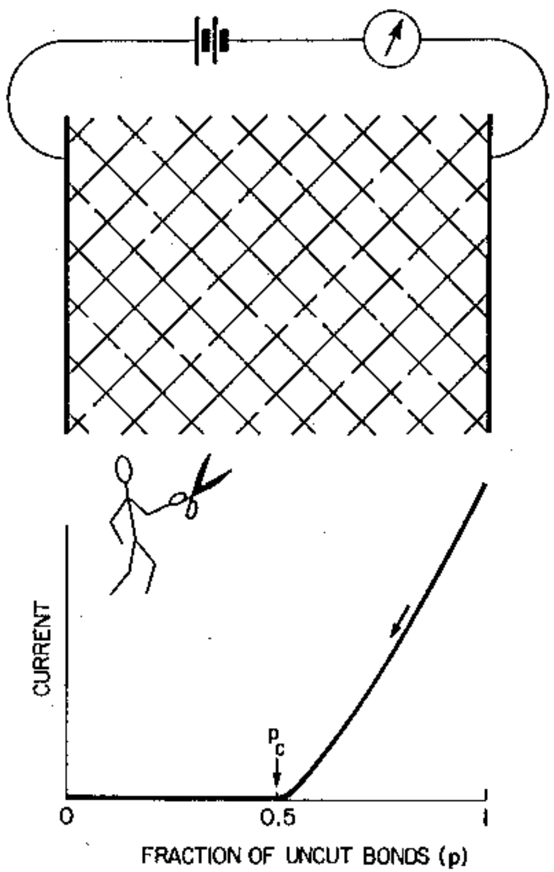


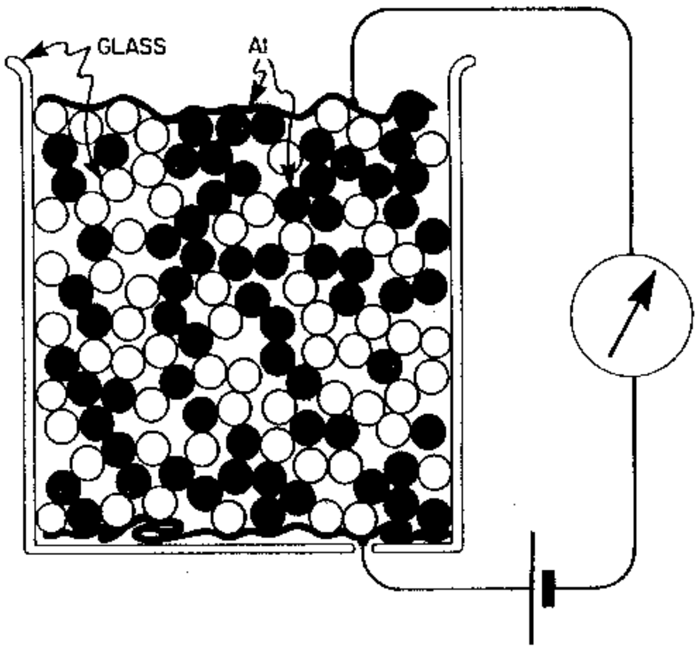


see e.g. Barabasi & Stanley, ***Fractal concepts in surface growth***, Cambridge University Press

Percolation

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





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

Percolation

- metal/insulator threshold behavior in resistor networks (discrete percolation) and in alloys (continuous percolation)

Other examples:
- fluid adsorption in a porous medium
- spreading of a disease in a population
- spreading of a forest fire...
- liquid/glass transition...
- ...

Percolation

Definitions:

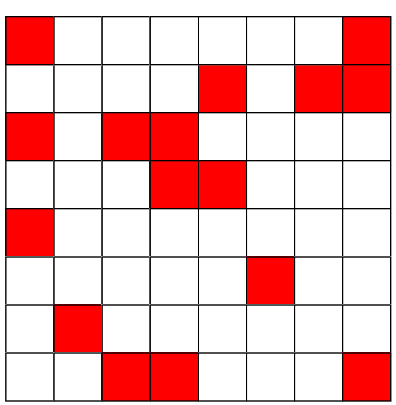
p: occupation probability of each identity (site, bond)

Cluster: group of identities (sites, bonds,...) connected by nearest neighboring bonds

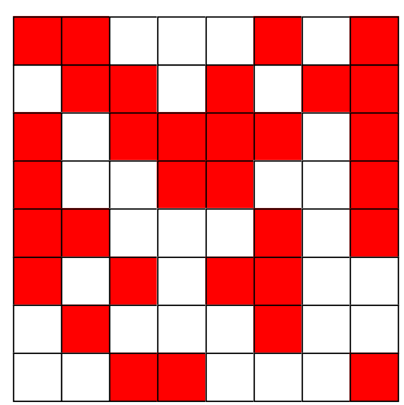
Percolating clusters: connecting two boundaries

which is the critical percolation threshold p_c ?

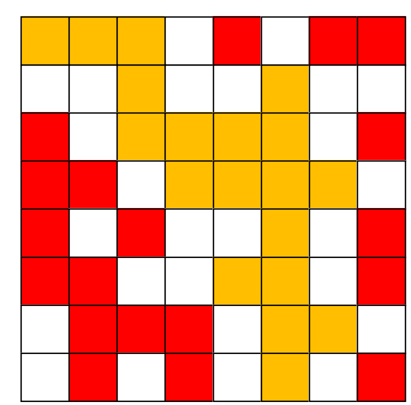
Example of site percolation on a lattice:



L = 8 p = 0.25



L = 8 p = 0.50



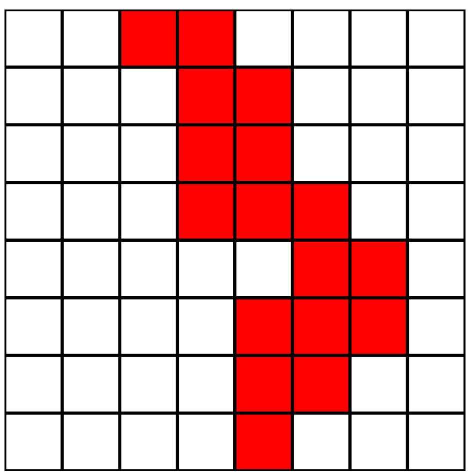
L = 8 p = 0.60

Percolation threshold

p_c depends on the criteria (different possible):

Connection along one fixed direction

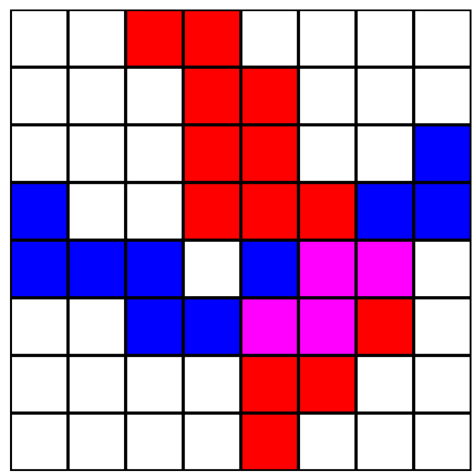
Percolazione verticale



Connection along one (any, horizontal or vertical) direction

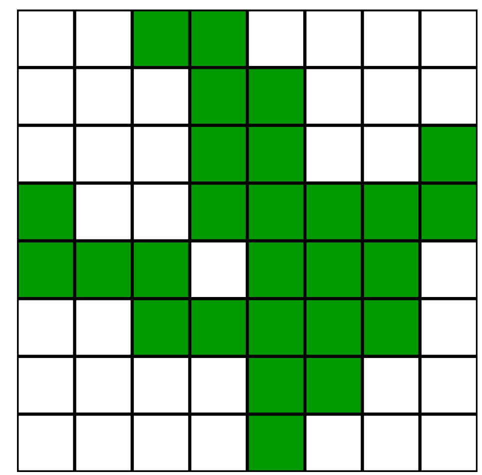
Percolazione verticale

Percolazione orizzontale



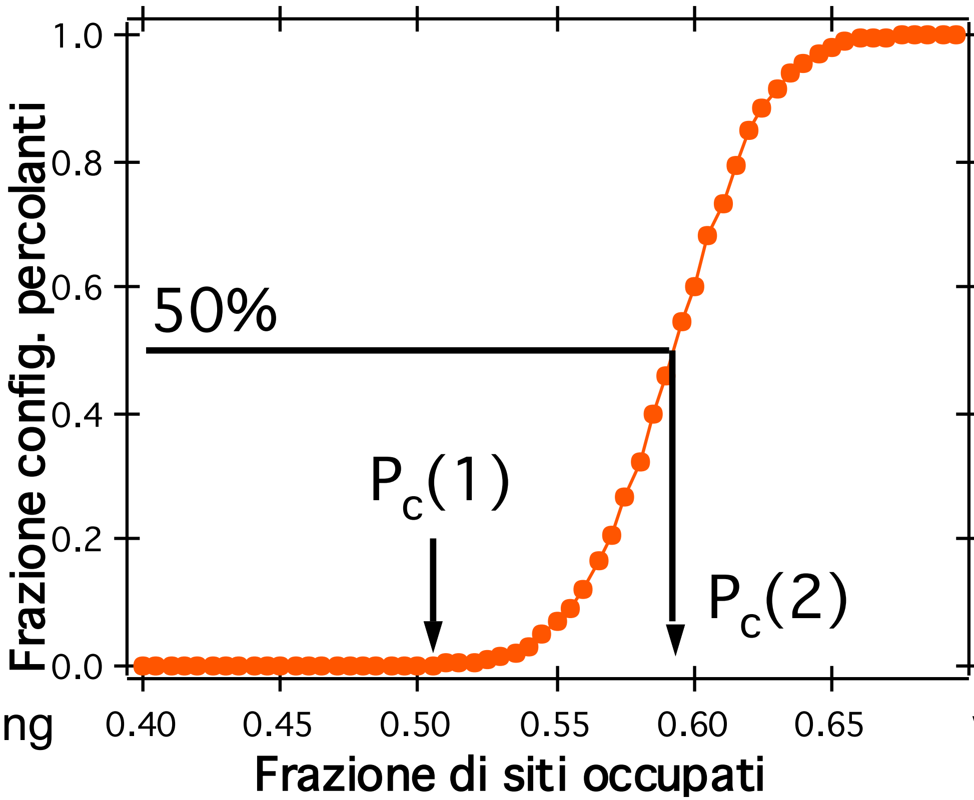
Connection in all directions

Percolazione in entrambe le direzioni



Percolation threshold

p_c depends on the criteria (different possible):



$P_c(1)$: fraction of occupied sites when the first percolating cluster is established

$P_c(2)$: fraction of occupied sites when 50% of the clusters are percolating

$$P_C(1) \equiv P_C(2) \quad \text{for} \quad L \to \infty$$

Monte Carlo approach

fix **L** => Lattice description

fix **p** => Site (or bond) filling accordingly

Identification and characterization of the clusters

```
do i,j=1,L
    r(i,j)=random(seed)
    if r(i,j) < p then index (i,j) = -1
    if r(i,j) > p then index (i,j) = 0
end do
```

use some algorithm of cluster labelling to identify the different clusters

generation of many configurations for each **p**

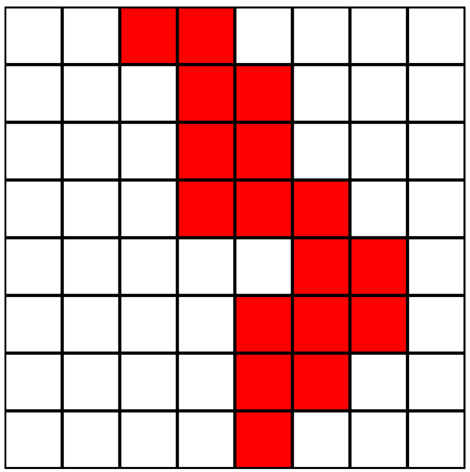
data analysis;
account for size effect (vary **L**)!

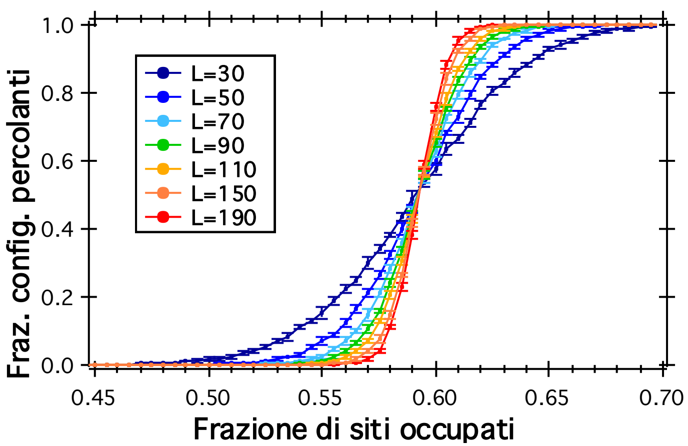
Results

for different percolation criteria and different size

Connection along one fixed direction

Percolazione verticale

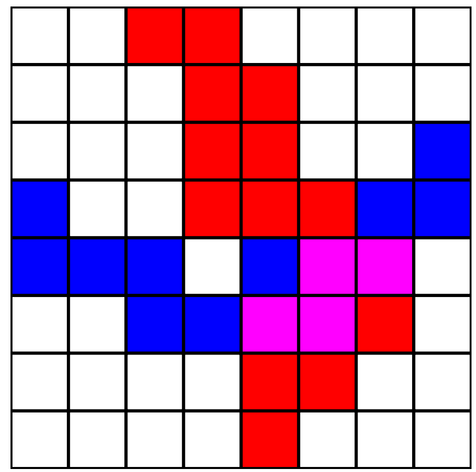


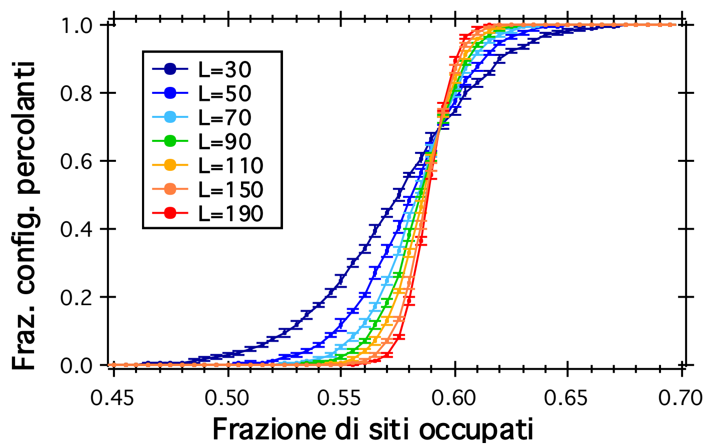


Connection along one (any, horizontal or vertical) direction

Percolazione verticale

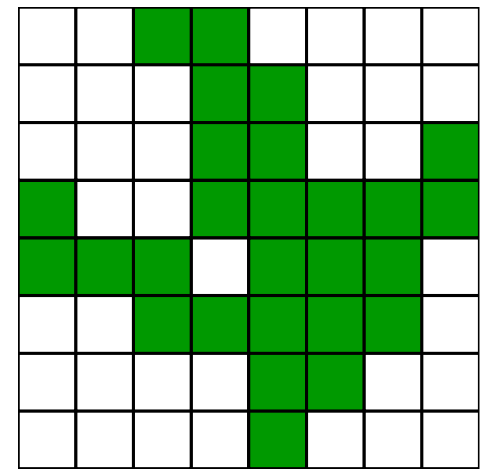
Percolazione orizzontale

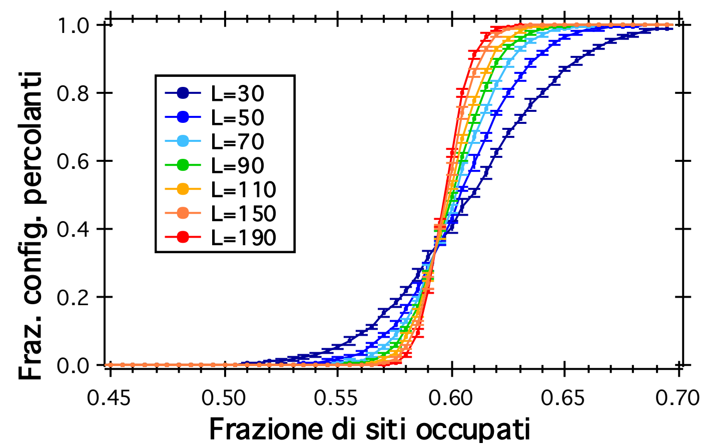




Connection in all directions

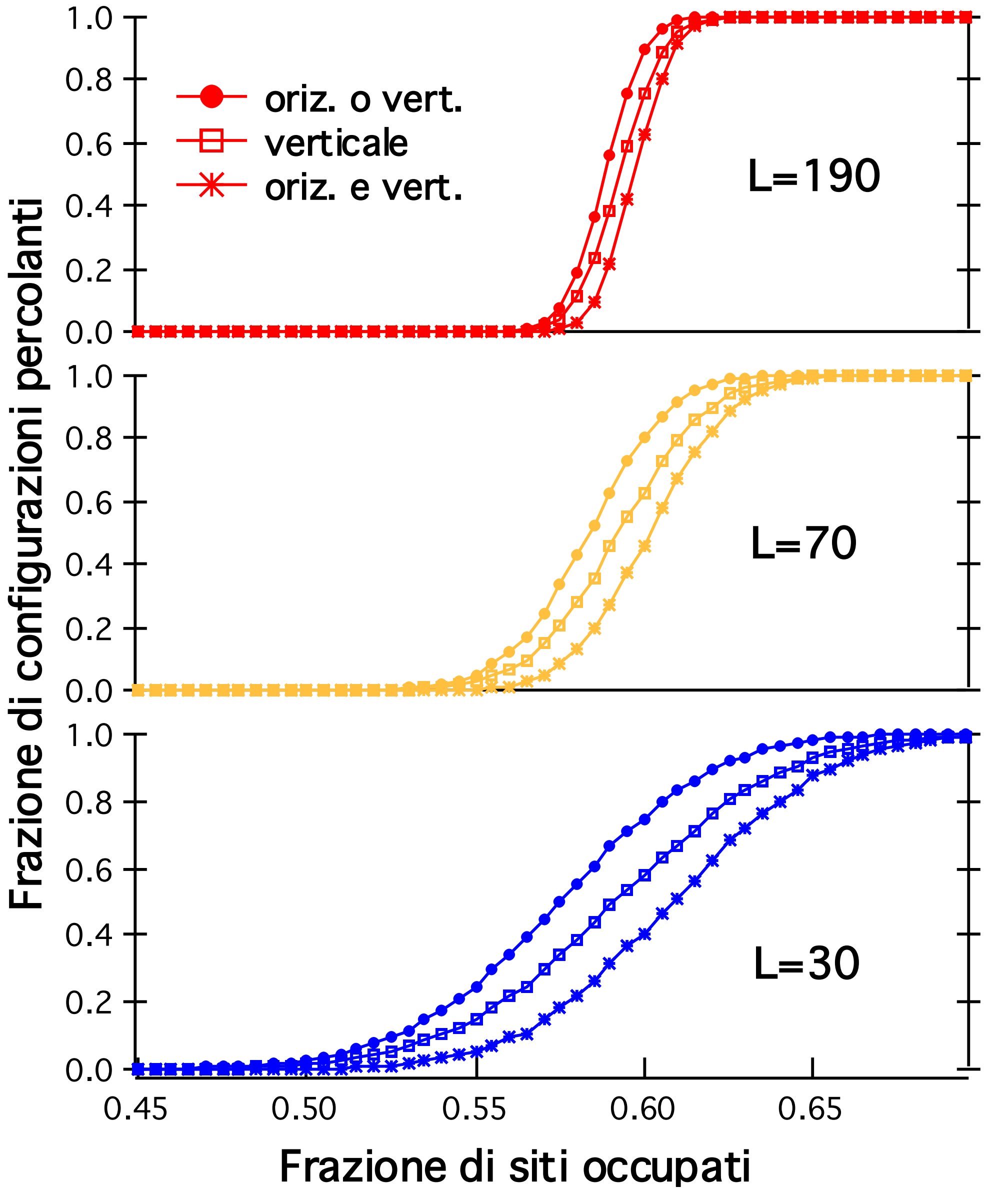
Percolazione in entrambe le direzioni





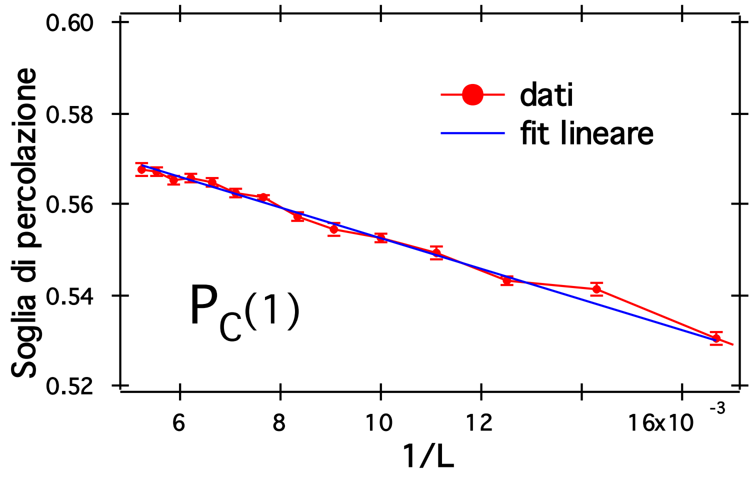
Results

for different percolation criteria and different size



Results

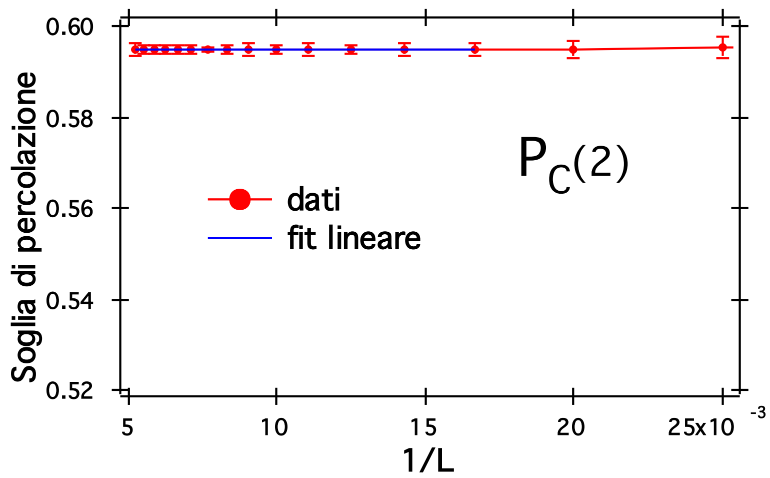
for different percolation criteria and different size



extrapolate the behavior for

$$L \to \infty$$

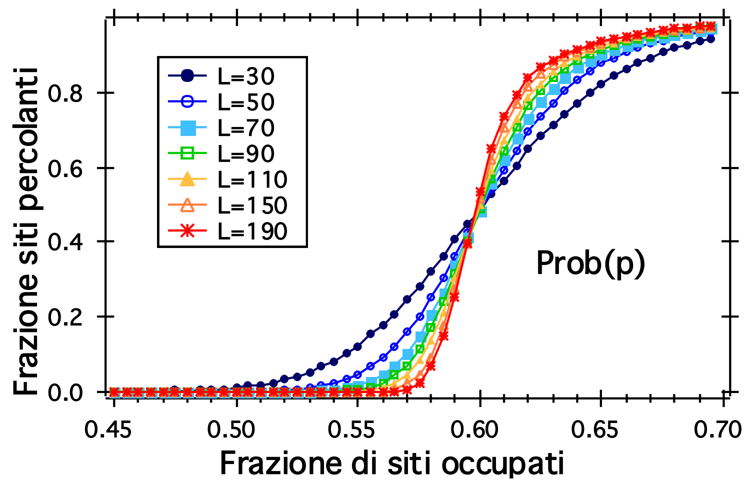
$$1/L \to 0$$



$$P_C^{\infty}(1) = P_C^{\infty}(2) = 0.59 \pm 0.05$$

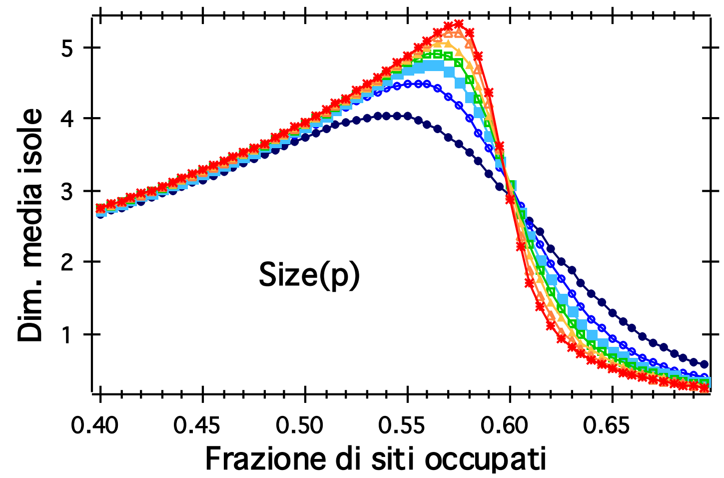
Results

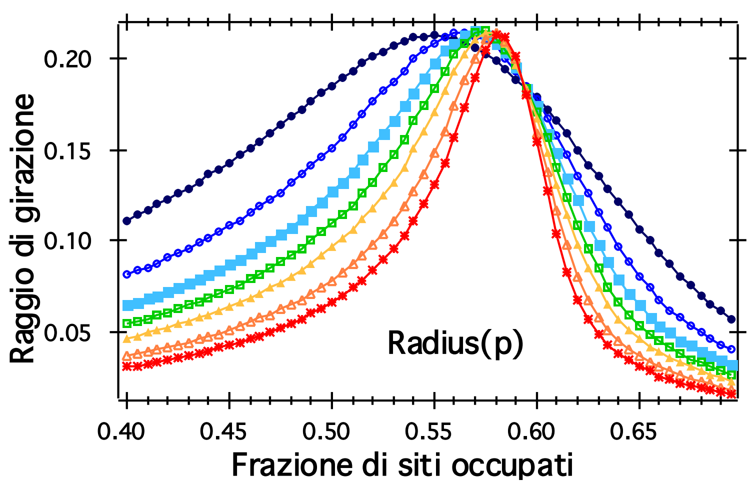
other interesting quantities



probability Prob(p) for a site to be included in a percolating cluster

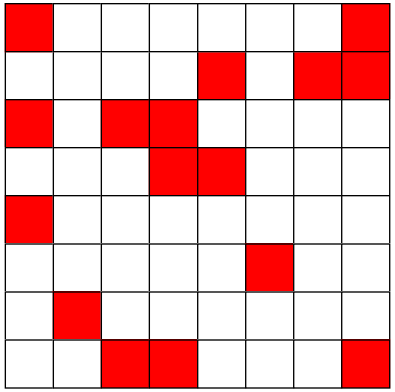
average size Size(p) of a non-percolating cluster



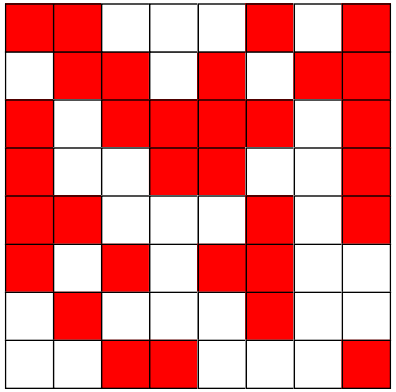


radius of gyration Radius(p) = $\sqrt{\frac{\sum_i^N (\vec{r}_i - \vec{r}_{cm})^2}{N}}$

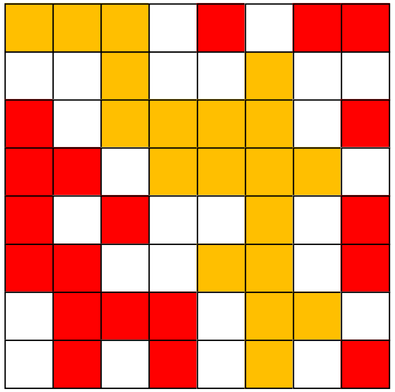
Cluster labeling



L = 8 p = 0.25



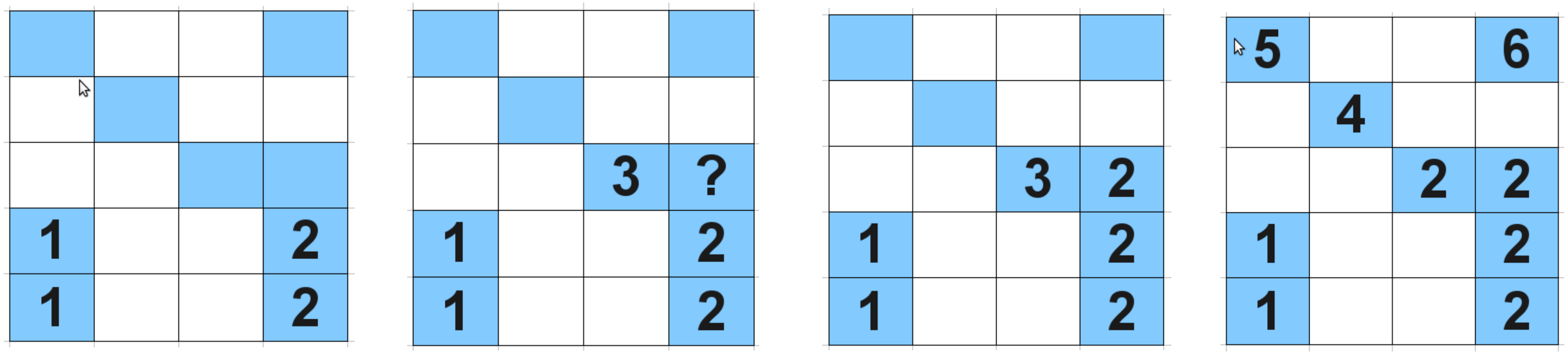
L = 8 p = 0.50



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

Cluster labeling



(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

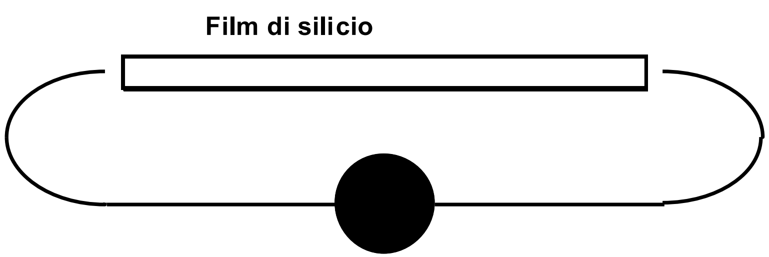
Hoshen- Kopelman algorithm for clusters labeling

Example of application in solid state physics

Dynamical Percolation Model of Conductance Fluctuations in Hydrogenated Amorphous Silicon,

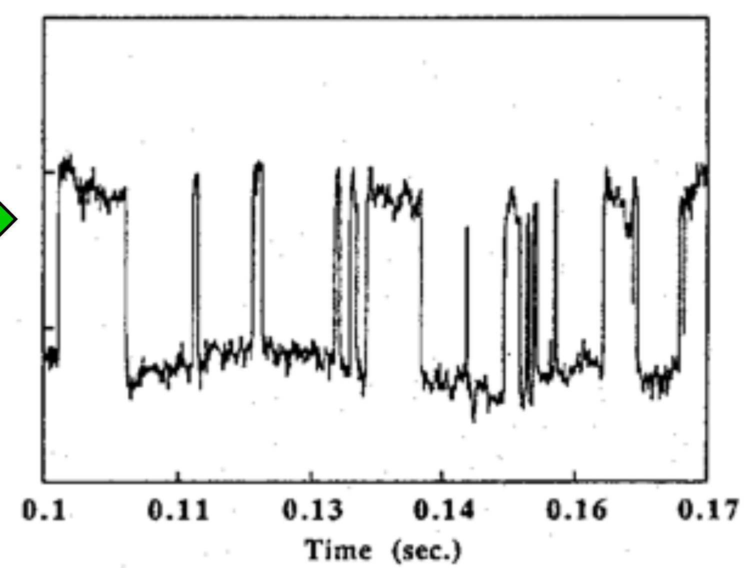
Lust and Kakalios, Phys. Rev. Lett. 75, 2192 (1995)

Fluttuazioni di conduttività nel silicio amorfo idrogenato (*a*-Si:H) sono simulate utilizzando un modello dinamico di diffusione di resistenze in un reticolo in condizioni di soglia di percolazione. Una frazione di siti di reticolo è designata come una trappola tale per cui quando un resistore diffonde in una di esse, rimane localizzato per un periodo finito di tempo.

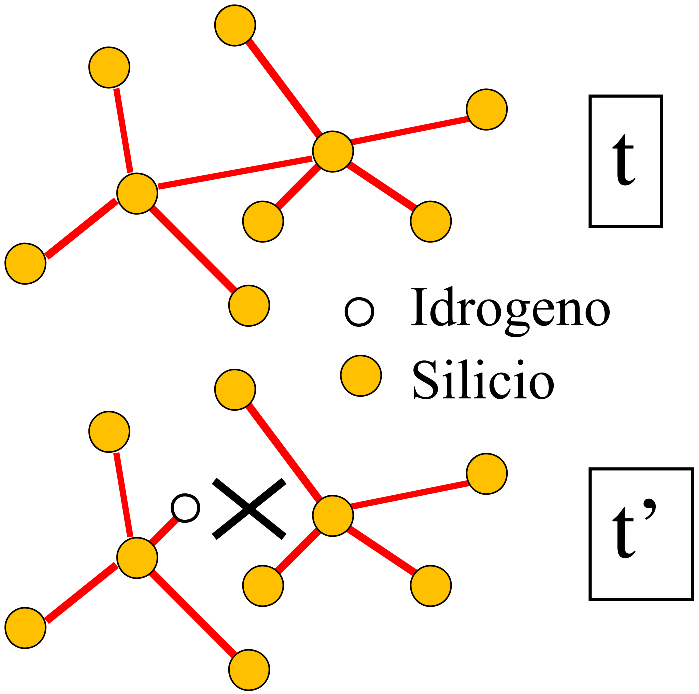


Fluttuazioni di conduttività misurate sperimentalmente

Fluttuazioni di tipo "telegrafico"



Spiegazione qualitativa del fenomeno



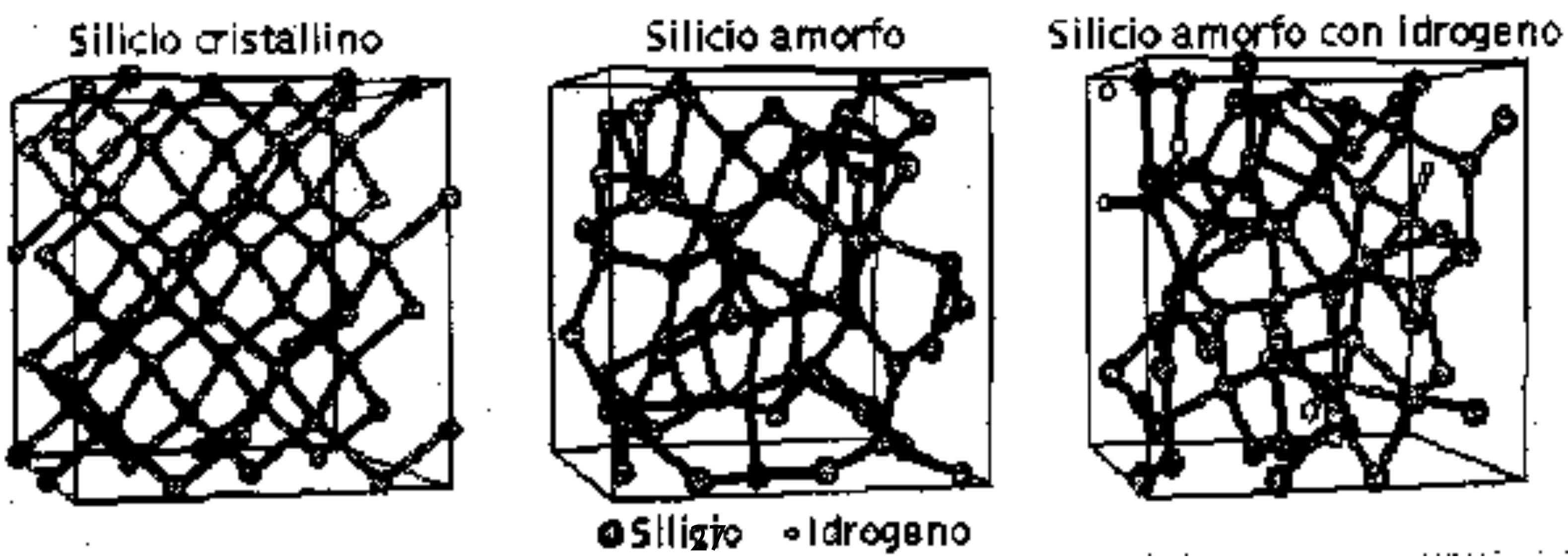
Diffusione idrogeno nel campione per effetto termico

↓

Riarrangiamento dei legami

↓

Modifica rete di resistenze



Spiegazione quantitativa

Quale modello?

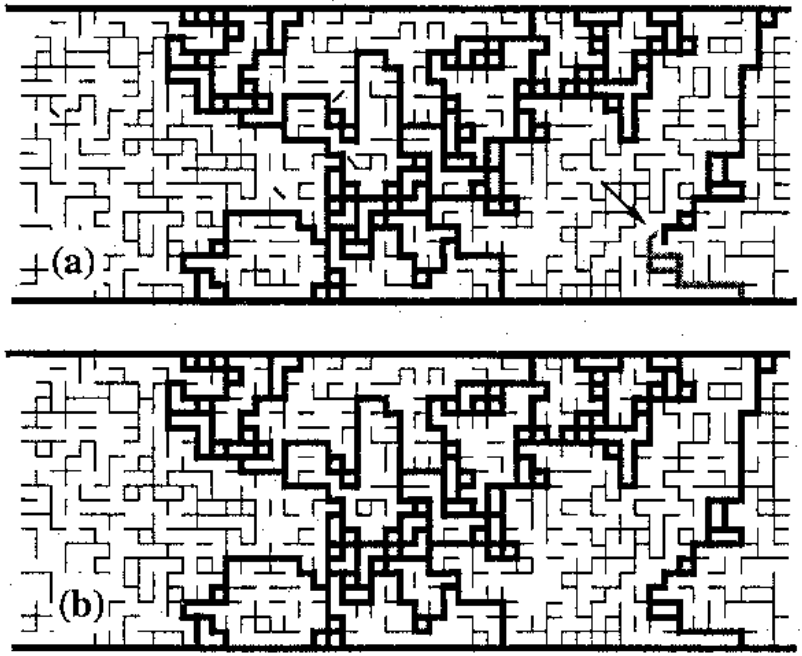
Simulazione in un mezzo disordinato (*a*-Si:H)

Percolazione statica:

modella il trasporto elettronico
(rete casuale di resistenze = legami Si-Si)

Percolazione dinamica:

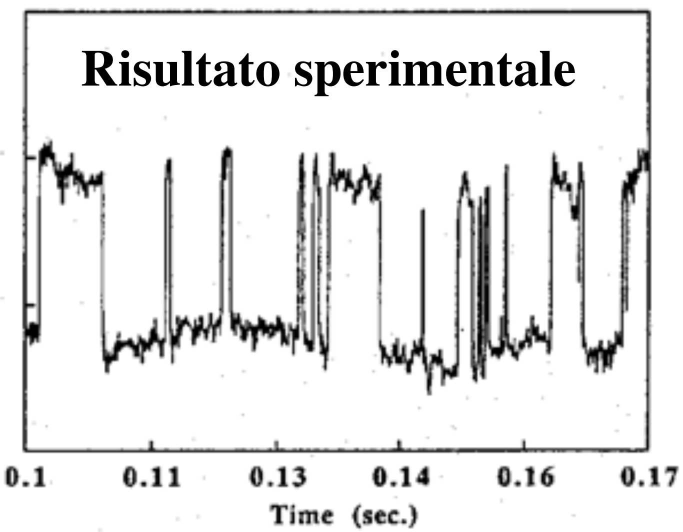
modella le fluttuazioni casuali
nel tempo della struttura locale
(fluttuazioni nella configurazione
della rete di resistenze)

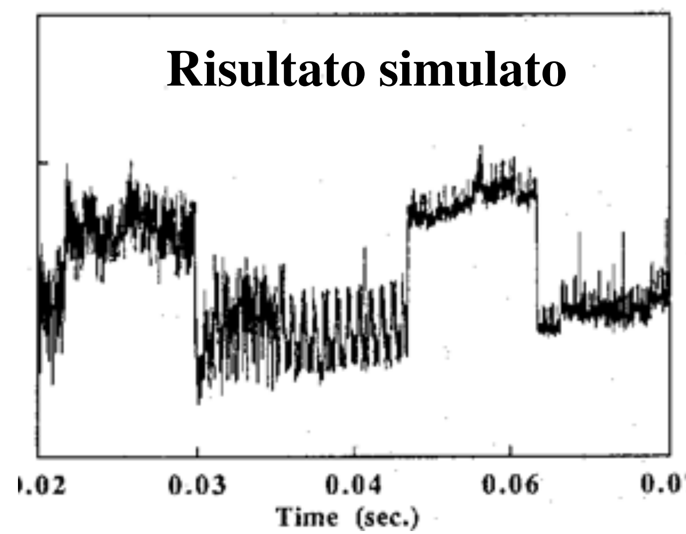


Rete casuale di resistenze con $P \sim P_C$ (fisso)

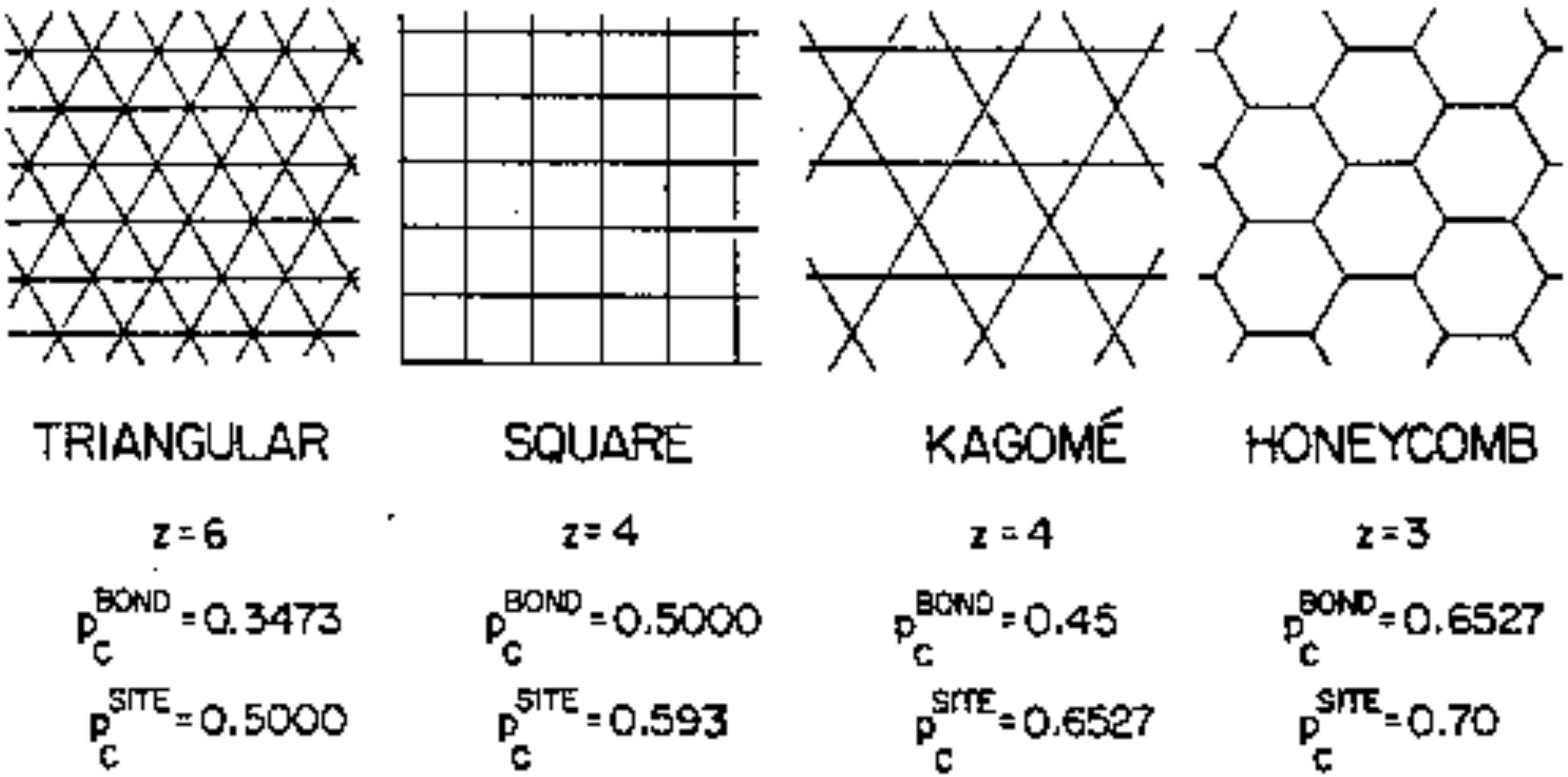
Configurazione dopo un riarrangiamento casuale dei legami

Diffusione H: Creazione/distruzione canali di conduttività





Percolation on different lattices



Metropolis method in the canonical ensemble and the

simulated annealing

a general purpose global optimization algorithm

(Kirkpatrick S, Gelatt CD Jr, Vecchi MP

Science. 1983 May 13; 220(4598):671-80)

Metropolis and simulated annealing - I

- Stochastic search for global minimum. Monte Carlo optimization.

- The concept is based on the manner in which liquids freeze or metals recrystallize. Sufficiently high starting temperature and slow cooling are important to avoid freezing out in metastable states.

mimics the physical process of annealing by treating the cost function as an "energy" E and sampling the value of E according to the Boltzmann distribution at some artificial temperature T using the Metropolis algorithm

Convergence to the global minimum has been proved for a schedule in which the temperature at the kth iteration $T_k \propto 1/\ln(k)$ and moves are drawn from a Gaussian distribution [8, 9], and also for a schedule where $T_k \propto 1/k$ and moves are drawn from a Cauchy distribution [10].

In practice, a much faster cooling schedule without a convergence proof was used in both the original Kirkpatrick paper, and most applications. In this schedule, $T_k \propto e^{-\lambda k}$ where λ is sometimes adjusted adaptively based on sampling statistics [11] => $T_{k+1} = (1 - \lambda)T_k$.
λ is a positive number very close to 0 that controls the cooling speed. The larger λ is, the faster the system cools.

Adaptive schedule are often used

We need:

a cooling schedule, a move generation strategy, and a stopping criterion

8. Geman S, Geman D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions On Pattern Analysis And Machine Intelligence. 1984;6:721–741. [PubMed] [Google Scholar]
9. Hajek B. Cooling schedules for optimal annealing. Mathematics of Operations Research. 1988;13:311–329. [Google Scholar]
10. Szu H, Hartley R. Fast simulated annealing. Physics Letters A. 1987;122:157–162. [Google Scholar]

Metropolis and simulated annealing - II

usual Metropolis procedure in the canonical ensemble

- **Thermodynamic system at temperature T, energy E.**
- *Perturb configuration (generate a new one).*
- *Compute change in energy dE. If dE is negative the new configuration is accepted. If dE is positive it is accepted with a probability given by the Boltzmann factor : exp(-dE/kT).*
- *The process is repeated many times for good sampling of configuration space.*
- **then the temperature is slightly lowered and the entire procedure repeated, and so on, until a frozen state is achieved at T = 0.**

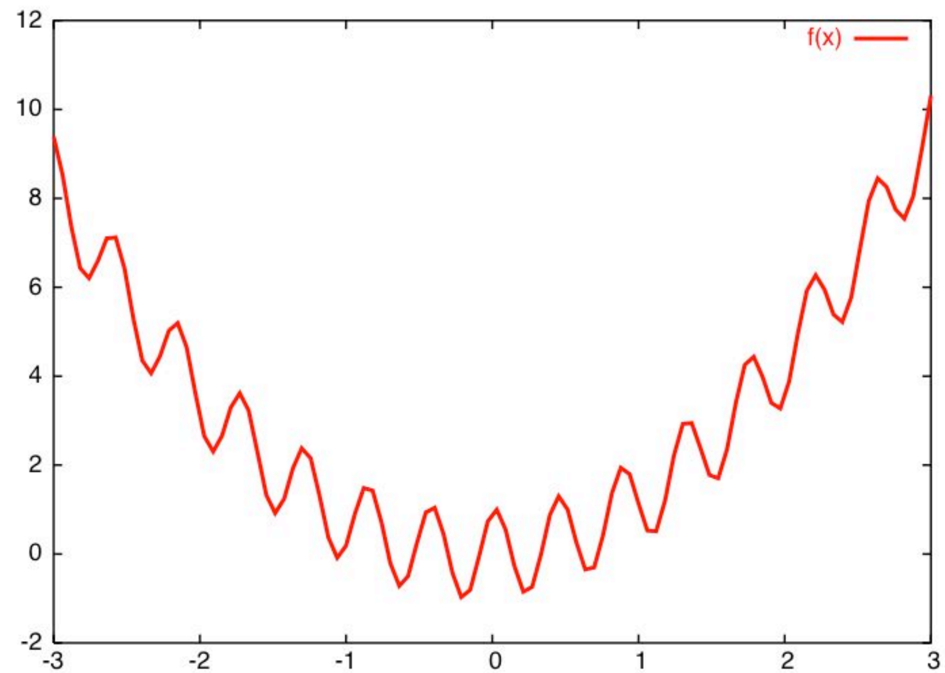
Example

in **`simulated_annealing.f90:`**

minimization of

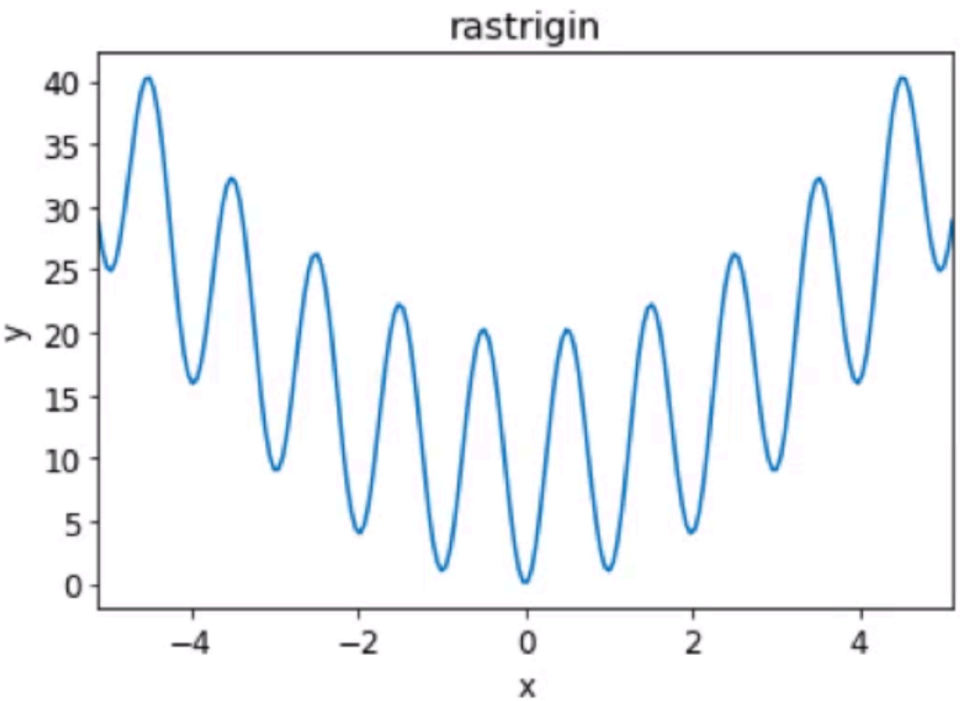
`f(x)=(x+0.2)*x+cos(14.5*x-0.3)`

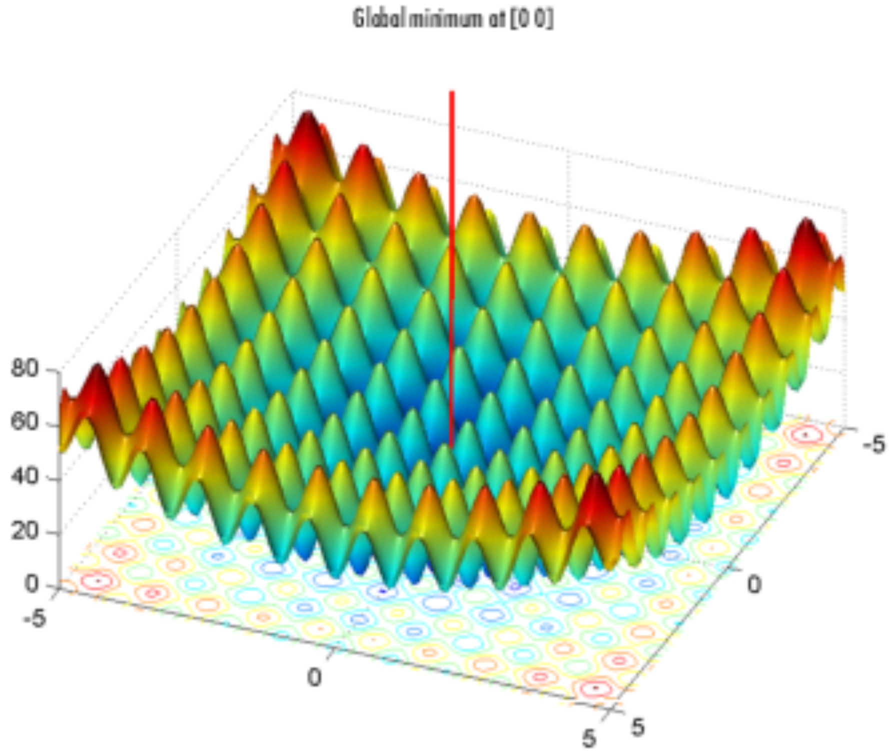
considered as an energy function and using a fictitious temperature



***Rastrigin function*:**

- non-convex *function* used as a performance test problem for optimization algorithm
- typical example of non-linear multimodal *function;*
- first proposed by *Rastrigin* as a 2-dimensional *function;* later generalized by Rudolph.





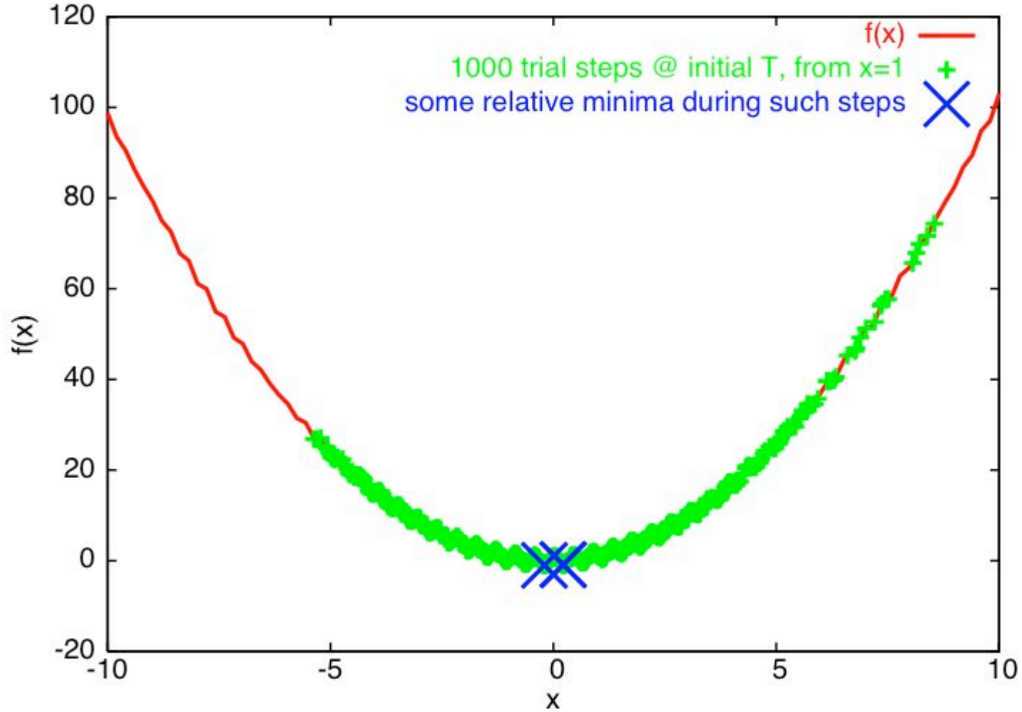
$$f(\boldsymbol{x}) = nA + \sum_{i=1}^{n} [x_i^2 - A\cos(2\pi x_i)]$$

Function to be minimized: **f(x)** ; Starting point: **x, fx=f(x)**

Annealing schedule:	initial (high) temperature:	**temp**
	annealing temperature reduction factor:	**tfactor (<1)**
	number of steps per block:	**nsteps**

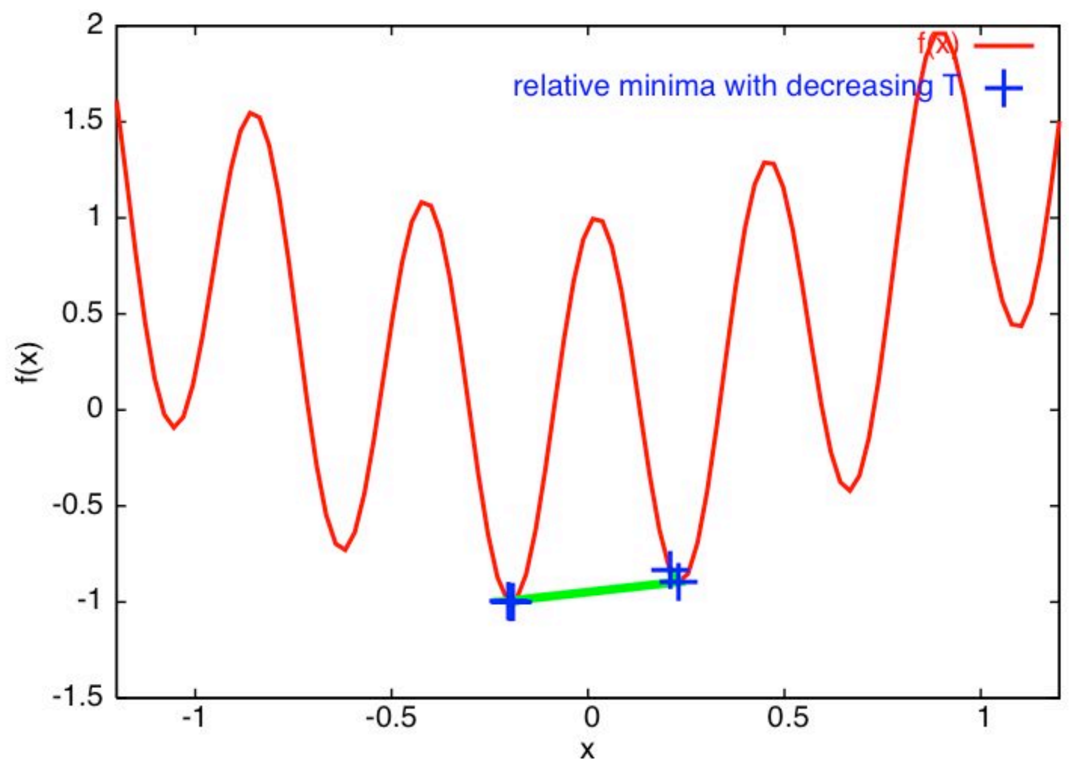
'ad hoc' parameter for trial move: **scale**

```
DO WHILE (temp > 1E-5) ! anneal cycle

  DO istep = 1, nsteps
    CALL RANDOM_NUMBER(rand) ! generate 2 random numbers; dimension(2) :: rand
    x_new = x + scale*SQRT(temp)*(rand(1) - 0.5) ! stochastic move
    fx_new = func(x_new) ! new object function value
    IF (EXP(-(fx_new - fx)/temp) > rand(2)) THEN ! success, save
      fx = fx_new
      x = x_new
    END IF
    IF (fx < fx_min) THEN
      fx_min = fx
      x_min = x
      PRINT '(3ES13.5)', temp, x_min, fx_min
    END IF
  END DO

  temp = temp * tfactor ! decrease temperature
END DO
```




```
initial T:   10 (K_B units)
initial x:        1.000000
initial f(x):     1.137208
```

```
final T:      2.50315E-01
final x:     -1.95067E-01
final f(x):-1.00088E+00
```



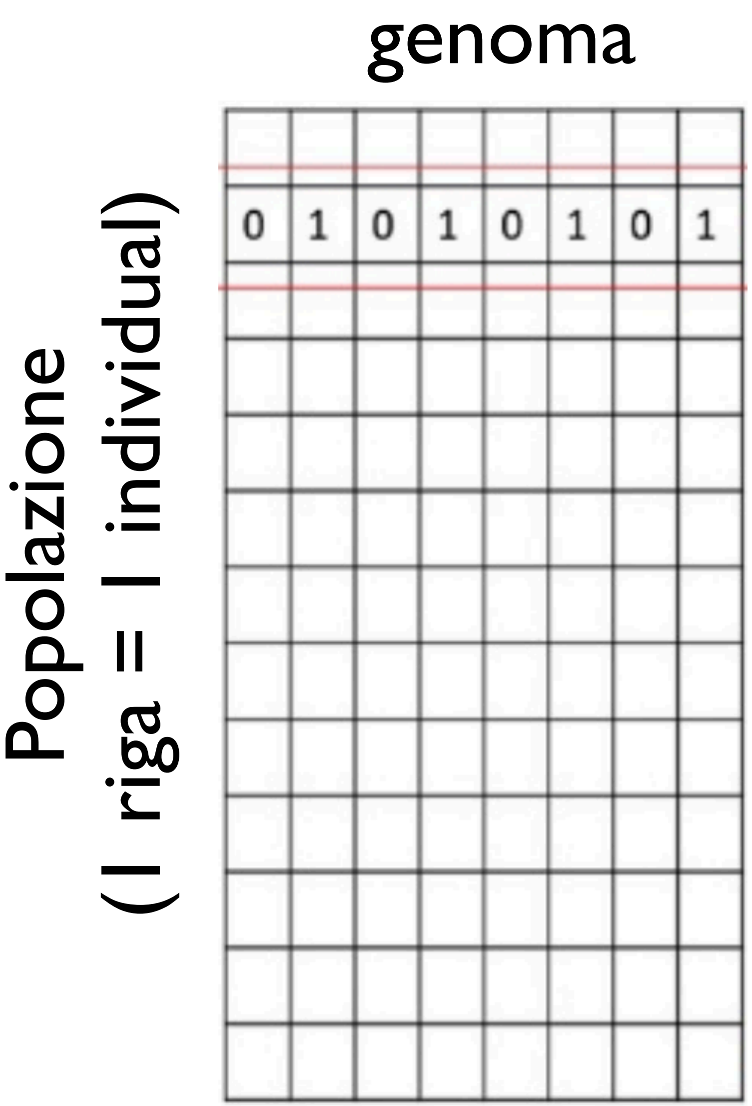
Algoritmi genetici
un problema di ottimizzazione

idea: applicare i principi dell'evoluzionismo naturale a sistemi artificiali,

codificando in modo numerico configurazioni di input, processi evolutivi, soluzioni, etc etc

Un po' di terminologia:

- popolazione
- genoma
- fitness
- processi di:
  crossover, mutazione, ...



1. Generazione casual di una popolazione iniziale costituita da un certo numero di “individui”;

2. ciascun individuo ha una propria **fitness**, che è un indice della qualità della “soluzione” che egli rappresenta; es: 0+1+ 0+1+ 0+1+ 0+1=8

3. Inizio di ciclo evolutivo ha inizio: la **selezione** simula la selezione naturale darwniana <u>proporzionalmente alla loro fitness</u> *(=> roulette wheel)*. Questi sono considerati “genitori” e danno luogo a “nuovi individui”;

4. la ricombinazione (*crossover*) agisce sulla popolazione intermedia dei genitori accoppiandoli due a due e scambiandone porzioni di DNA;

5. Una mutazione può cambiare singoli elementi costitutivi del filamento di DNA e li muta in nuovi;

6. l’algoritmo ricomincia e può procedere anche fino all’infinito: si decide un certo criterio di stop (soddisfatta una certa richiesta, ad es. Una certa fitness media)

PROVARE TEST CON:

- popolazione: da 100 a 10000 (dipende molto dal problema);
- - Pcross: circa 0.8;
  - Pmut: attorno a 0.01.