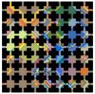


# Computer modeling and simulation of natural phenomena

Bastien Chopard

*Computer Science Department,*
*University of Geneva, Switzerland*

**Lugano, Oct 18, 2016**

# Examples of models and methods

- N-body systems, molecular dynamics
- Mathematical equations : ODE, PDE
- Monte-Carlo methods (equilibrium, dynamic, kinetic)
- **Cellular Automata and Lattice Boltzmann method**
- multi-agent systems
- Discrete event simulations
- Complex network
- L-systems.......

# What is a model ?

Many definitions :

- Simplifying abstraction of reality
- containing only the essential elements with respect to the problem
- A mathematical or rule-based representation of a phenomena
- But a model may also be :
  - A fit of data
  - An animal (medical study)
  - ...

# What is a good model ?

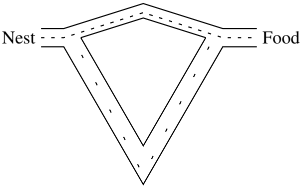
A Einstein :

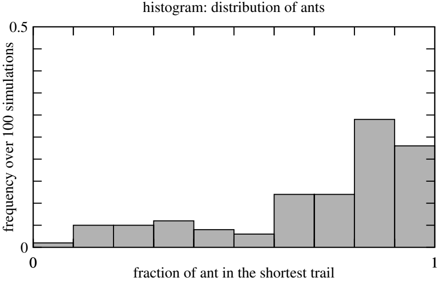
*Everything should be made as simple as possible, but not simnpler*

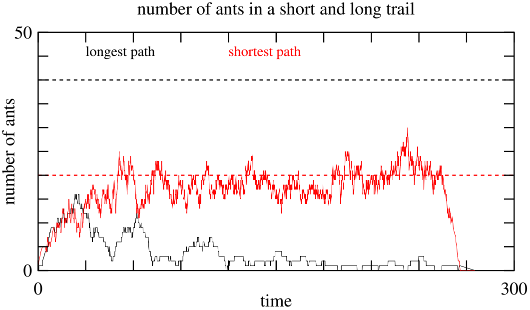
- In silico simulations : understand, predict and control a process
- Allows scientists to formulate new questions that can be addressed experimentally or theoretically
- Adapt the model to the problem

# Discrete Event Simulations

Do ants find the shortest path between nest and food ?



# Magritte's apple

## Magritte's apple


- A model is only a model, not reality

# Same reality, different models, different languages

Hydrodynamics

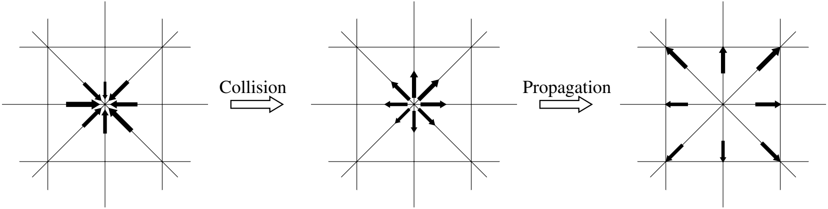
$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u}$$

**phenomenon $\rightarrow$ PDE$\rightarrow$ discretization $\rightarrow$ numerical solution**

# From PDEs to virual universe

One defines a discret universe as an abstraction of the real world

$\boxed{\textbf{phenomenon} \rightarrow \textbf{computer model}}$

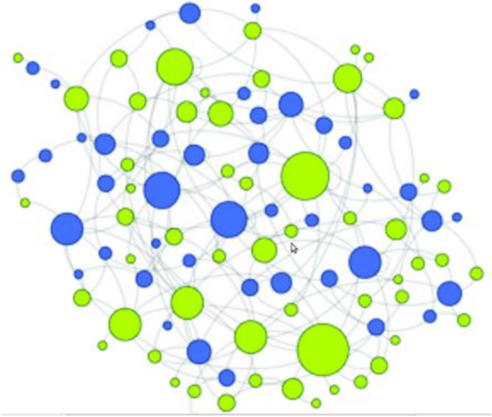


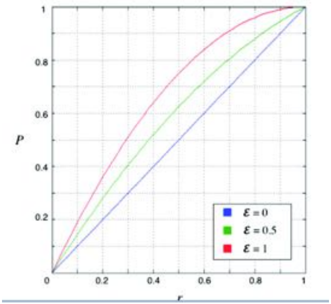
# Multi-Agent Model

- Set of bacteria moving in space with concetration $\rho(x, y)$ of nutrient
- Let $\rho_i(t)$ be the concentration seen by bacteria $b_i$ at time $t$
- if $\rho_i(t) \geq \rho_i(t - \delta t)$, the bacteria move straight with probability 0.9
- if $\rho_i(t) < \rho_i(t - \delta t)$, the bacteria move straight with probability 0.5
- Otherwise it makes a random turn
- Movie

# Beyond the physical space : complex network

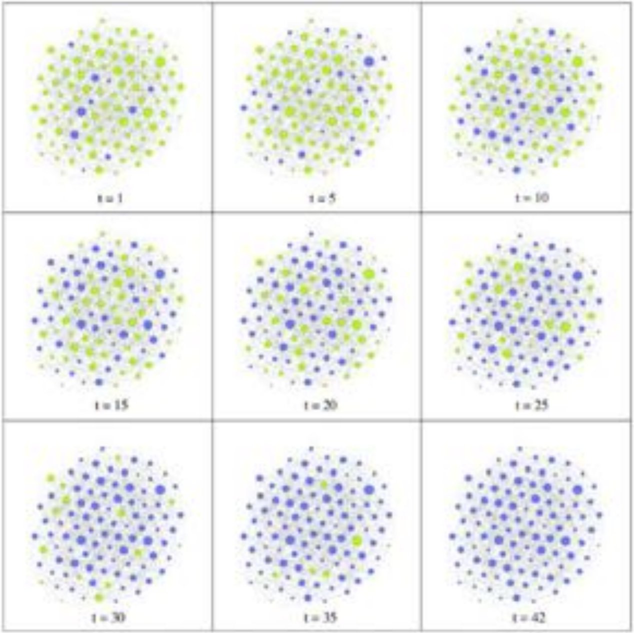
A model of opinion propagation in a social network






(Lino Velasquez, UNIGE)

# Voter model : time evolution

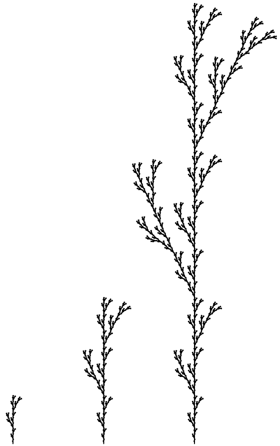



Réseau aléatoire, N = 100, p = 0.05, $\mathcal{E}$ = 0.3

# L-systems $F \rightarrow F[+F]F[-F]F$, $\beta = 25^o$.

```
F [ + F ] F [ - F ] F [ + F [ + F ] F [ -
F ] F ] F [ + F ] F [ - F ] F [ - F [ + F ]
F [ - F ] F ] F [ + F ] F [ - F ] F [ + F [
+ F ] F [ - F ] F [ + F [ + F ] F [ - F ]
F ] F [ + F ] F [ - F ] F [ - F [ + F ] F [
- F ] F ] F [ + F ] F [ - F ] F ] F [ + F ]
F [ - F ] F [ + F [ + F ] F [ - F ] F ] F [
+ F ] F [ - F ] F [ - F [ + F ] F [ - F ]
F ] F [ + F ] F [ - F ] F [ - F [ + F ] F [
- F ] F [ + F [ + F ] F [ - F ] F ] F [ +
F ] F [ - F ] F [ - F [ + F ] F [ - F ] F ]
F [ + F ] F [ - F ] F ] F [ + F ] F [ - F ]
F [ + F [ + F ] F [ - F ] F ] F [ + F ] F
[ - F ] F [ - F [ + F ] F [ - F ] F ] F [ +
F ] F [ - F ] F
```

Code for 3 iterations



Iterations 3,4 and 5

# Cellular Automata

B. Chopard et M. Droz : Cellular Automata Modeling of Physical Systems, Cambridge University Press, 1998.

B. Chopard, *Cellular Automata and lattice Boltzmann modeling of physical systems*, Handbook of Natural Computing, Rozenberg, Grzegorz ; Bäck, Thomas ; Kok, Joost N. (Eds.) Springer, pp. 287–331, 2013

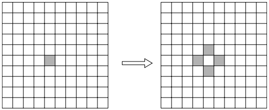
# Definition

What is a Cellular Automata ?

- Mathematical abstraction of the real world, modeling framework
- Fictitious Universe in which everything is discrete
- But, it is also a mathematical object, new paradigm for computation
- Elucidate some links between **complex systems**, **universal computations**, **algorithmic complexity**, **intractability**.

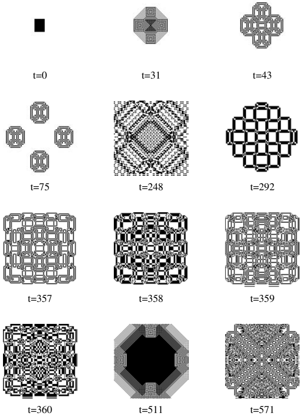
# Example : the Parity Rule

- Square lattice (chessboard)
- Possible states $s_{ij} = 0, 1$
- Rule : each cell sums up the states of its 4 neighbors (north, east, south and west).
- If the sum is even, the new state is $s_{ij} = 0$ ; otherwise $s_{ij} = 1$



Generate "complex" patterns out of a simple initial condition.

# Pattern generated by the Parity Rule

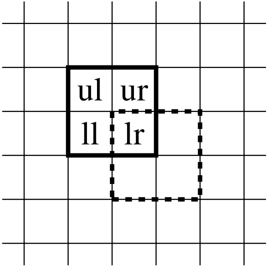


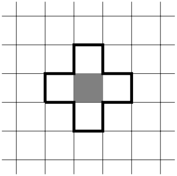
# CA Definition

- Discrete space $A$ : regular lattice of cells/sites in $d$ dimensions.
- Discrete time
- Possible states for the cells : discrete set $S$
- Local, homogeneous **evolution rule** $\Phi$ (defined for a neighborhood $\mathcal{N}$).
- Synchronous (parallel) updating of the cells
- Tuple : $< A, S, \mathcal{N}, \Phi >$

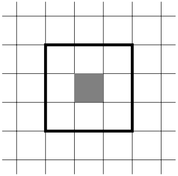
# Neighborhood

- von Newmann
- Moore
- Margolus
- ...



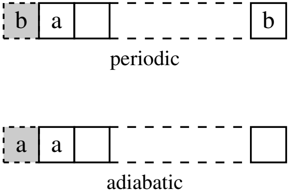
(a)



(b)

# Boundary conditions

- periodic
- fixed
- reflexive
- ....



# Generalization

- Stochastic CA
- Asynchronous update : loss of parallelism, but avoid oscillations
- Non-uniform CA

# Implementation of the evolution rule

$m$ states per cells, $k$ neighbors.

- On-the-fly calculation
- Lookup table
- Finite number of possible universes : $m^{m^k}$ possible rules where $m$ is the number of states per cell and $k$ the number of neighbors.

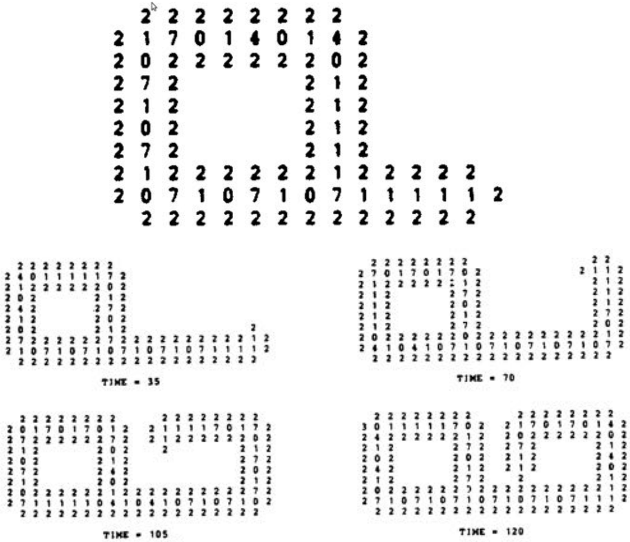
# Historical notes

- Origin of the CA's (1940s) : John von Neumann and S. Ulam
- Design a better computer with self-repair and self-correction mechanisms
- Simpler problem : find the logical mechanisms for self-reproduction :
- Before the discovery of DNA : find an algorithmic way (transcription and translation)
- Formalization in a fully discrete world
- Automaton with 29 states, arrangement of thousands of cells which can self-reproduce
- Universal computer

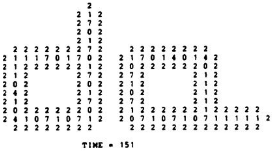
# Langton's CA

- Simplified version (8 states).
- Not a universal computer
- Structures with their own fabrication recipe
- Using a reading and transformation mechanism

# Langton's CA : basic cell replication



# Langton's Automaton : spatial and temporal evolution



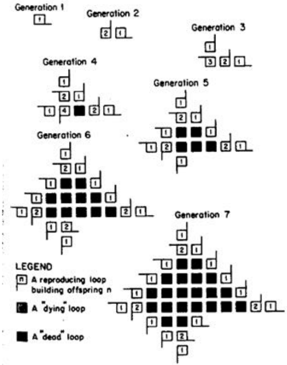




Fig. 10. Growth of loop colony. Seven generations of growth in a colony of loops.

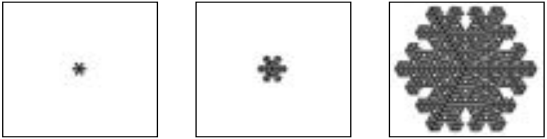
# Langton's CA : some conclusions

- Not a biological model, but an algorithmic abstraction
- Reproduction can be seen from a mechanistic point of view (Energy and matter are needed)
- No need of a hierarchical structure in which the more compicated builds the less complicated
- Evolving Hardware.

# CA as a mathematical abstraction of reality

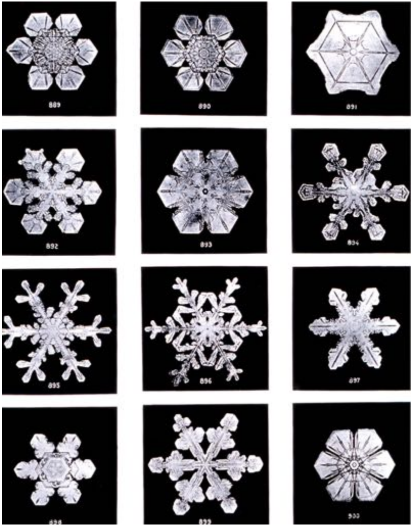
- Several levels of reality : macroscopic, mesoscopic and microscopic.
- The macroscopic behavior depends very little on the details of the microscopic interactions.
- Only "symmetries" or conservation laws survive. The challenge is to find them.
- Consider a fictitious world, particularly easy to simulate on a (parallel) computer with the desired macroscopic behavior.

# A Caricature of reality



**What is this ?**

# The real thing



Wilson Bentley, 1902

# Snowflakes model

- Very rich reality, many different shapes
- Complicated true microscopic description
- Yet a simple growth mechanism can capture some essential features

**• A vapor molecule solidifies ($\rightarrow$ice) if one and only one already solidified molecule is in its vicinity**
**• Growth is constrained by $60^o$ angles**

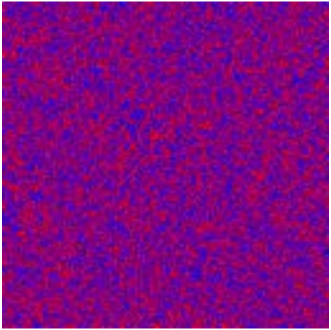
# Examples of CA rules

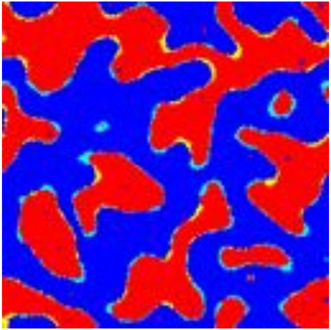
Cooperation models : annealing rule

- Growth model in physics : droplet, interface, etc
- Biased majority rule : (almost copy what the neighbors do)

**Rule :**

| | |
|---|---|
| $\text{sum}_{ij}(t)$ | 0 1 2 3 4 5 6 7 8 9 |
| $s_{ij}(t+1)$ | 0 0 0 0 1 0 1 1 1 1 |





The rule sees the curvature radius of domains

# Cells differentiation in drosophila

In the embryo all the cells are identical. Then during evolution they differentiate

- slightly less than 25% become neural cells (neuroblasts)
- the rest becomes body cells (epidermioblasts).

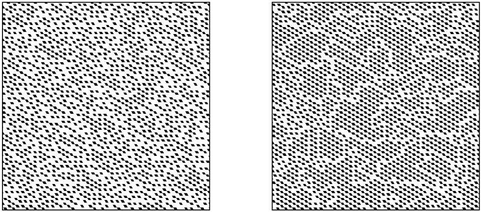
**Biological mechanisms :**

- Cells produce a substance $S$ (protein) which leads to differentiation when a threshold $S_0$ is reached.
- Neighboring cells inhibit the local $S$ production.

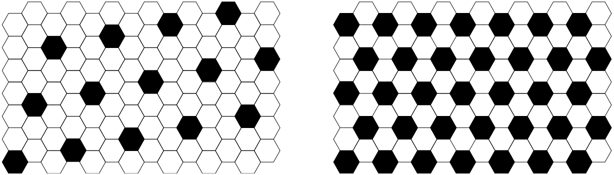
# CA model for a competition/inhibition process

- Hexagonal lattice
- The values of $S$ can be 0 (inhibited) or 1 (active) in each lattice cell.
- A $S = 0$ cell will grow (i.e. turn to $S = 1$) with probability $p_{grow}$ provided that all its neighbors are 0. Otherwise, it stays inhibited.
- A cell in state $S = 1$ will decay (i.e. turn to $S = 0$) with probability $p_{decay}$ if it is surrounded by at least one active cell. If the active cell is isolated (all the neighbors are in state 0) it remains in state 1.

# Differentiation : results



(a) (b)



The two limit solutions with density 1/3 and 1/7, respectively.

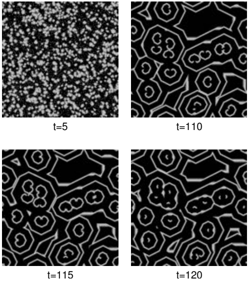
- CA produces situations with about **23%** of active cells, for almost any value of $p_{anihil}$ and $p_{growth}$.
- Model robust to the lack of details, but need for hexagonal cells

# Excitable Media, contagion models

- 3 states : (1) normal (resting), (2) excited (contagious), (3) refractory (immuned)
  1. excited $\rightarrow$ refractory
  2. refractory$\rightarrow$ normal
  3. normal $\rightarrow$ excited, if there exists excited neighbors (otherwise, normal $\rightarrow$ normal).

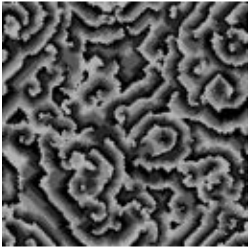
# Greenberg-Hastings Model

- $s \in \{0, 1, 2, ..., n-1\}$
- normal : $s = 0$ ; excited $s = 1, 2, ..., n/2$ ; the remaining states are refractory
- contamination if at least $k$ contaminated neighbors.



# Belousov-Zhabotinski (tube worm)

The state of each site is either 0 or 1 ; a local timer with values 0, 1, 2 or 3 controls the 0 period.



(i) where the timer is zero, the state is excited ;

(ii) the timer is reset to 3 for the excited sites which have two, or more than four, excited sites in their Moore neighborhood.

(iii) the timer is decreased by 1 unless it is 0 ;

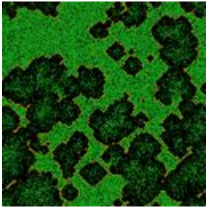
# Forest fire

(1) a burning tree becomes an empty site ;

(2) a green tree becomes a burning tree if at least one of its nearest neighbors is burning ;

(3) at an empty site, a tree grows with probability $p$ ;

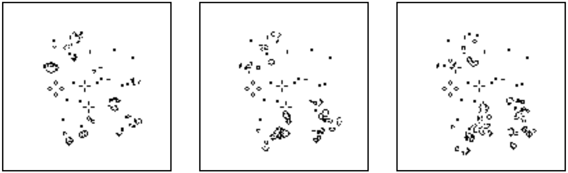
(4) A tree without a burning nearest neighbor becomes a burning tree during one time step with probability $f$ (lightning).



# Complex systems

**Rule of the Game of Life :**

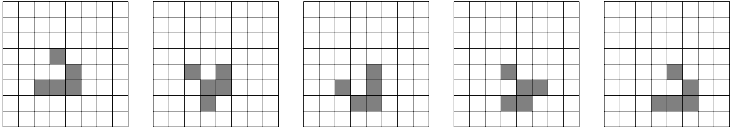
- Square lattice, 8 neighbors
- Cells are dead or alive (0/1)
- Birth if exactly 3 living neighbors
- Death if less than 2 or more than 3 neighbors



t t+10 t+20

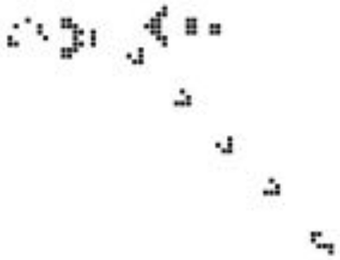
# Complex Behavior in the game of life

Collective behaviors develop (beyond the local rule)
**"Gliders"** (organized structures of cell) can emerge and can **move** collectively.



t=0 t=1 t=2 t=3 t=4

# Complex Behavior in the game of life



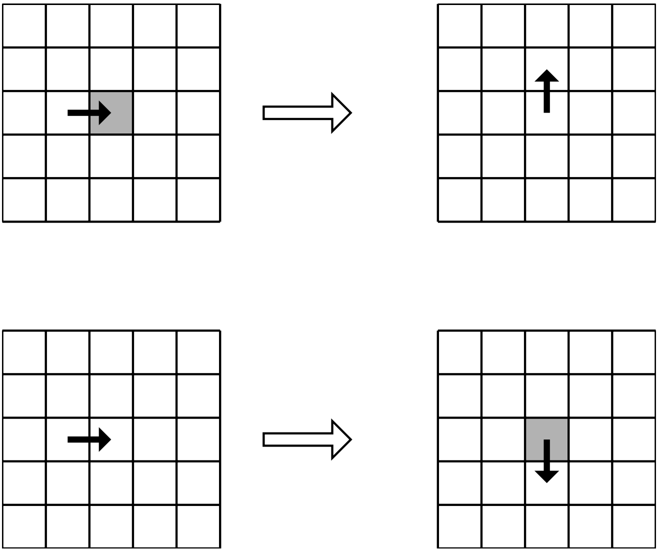
A glider gun (image : Internet)

- There are more complex structures with more complex behavior : a zoology of organisms.
- The game of life is a *Universal computer*

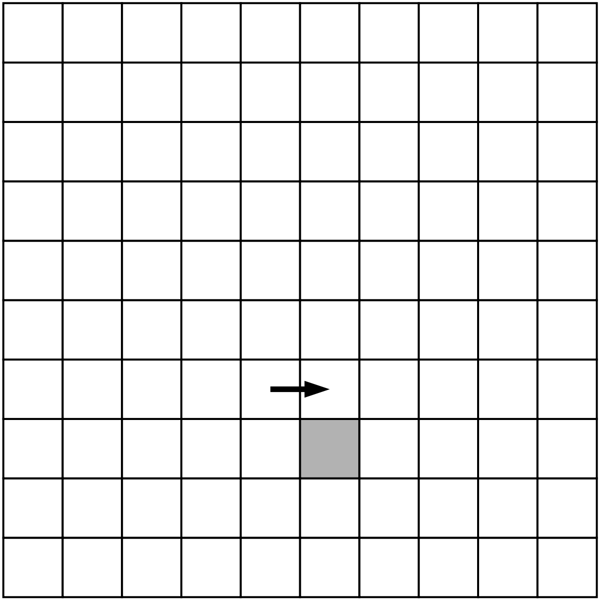
# Langton's ant

This is an hypothetical animal moving on a 2D lattice, acoring to simple rules, which depend on the color of the cell on which the ant sits.

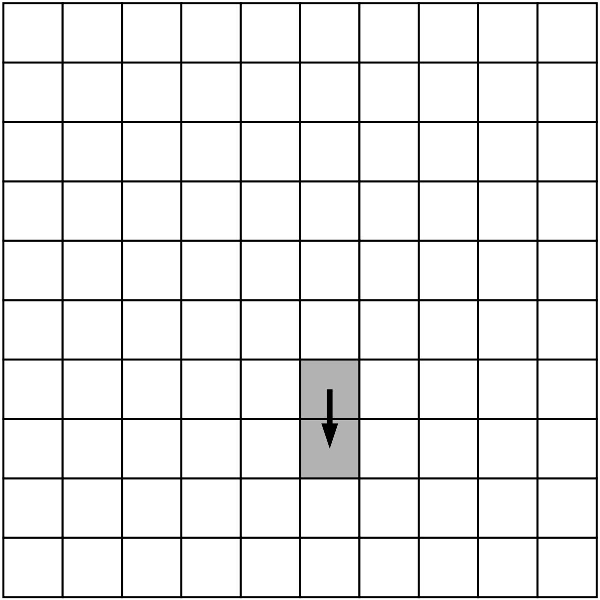
# The rules



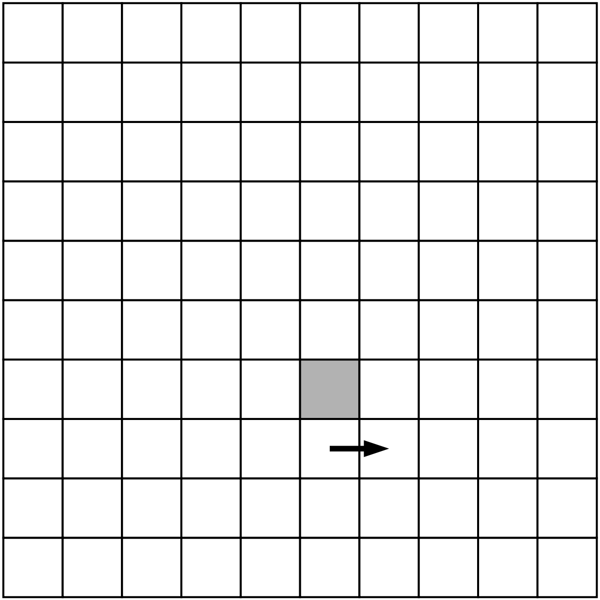
# Some evolution steps



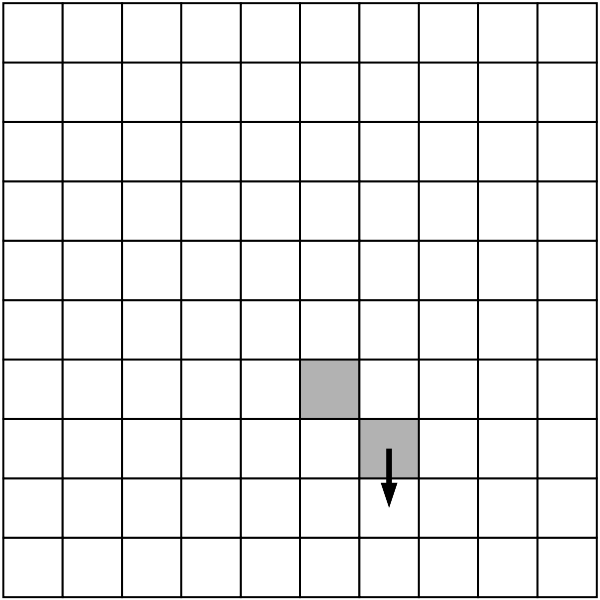
## Some evolution steps



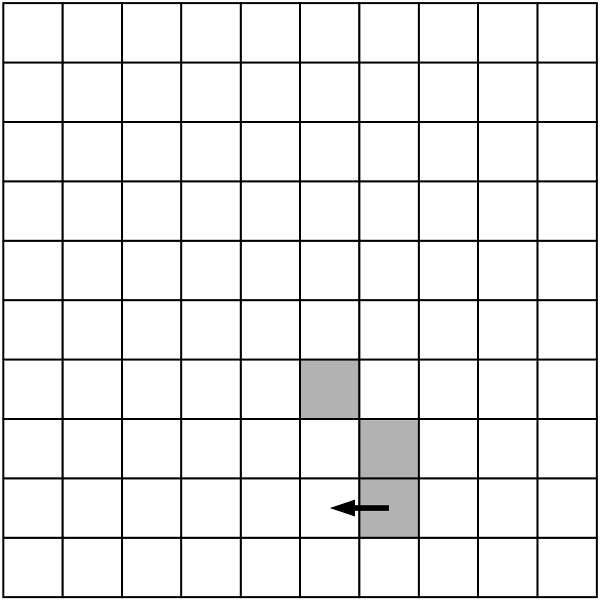
## Some evolution steps



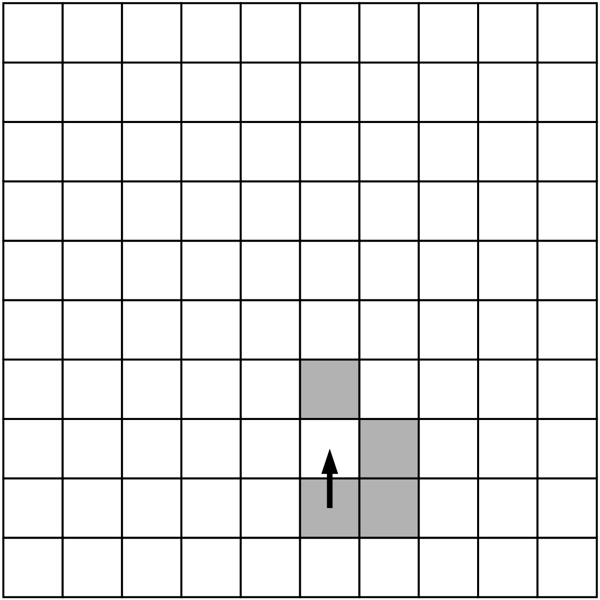
# Some evolution steps



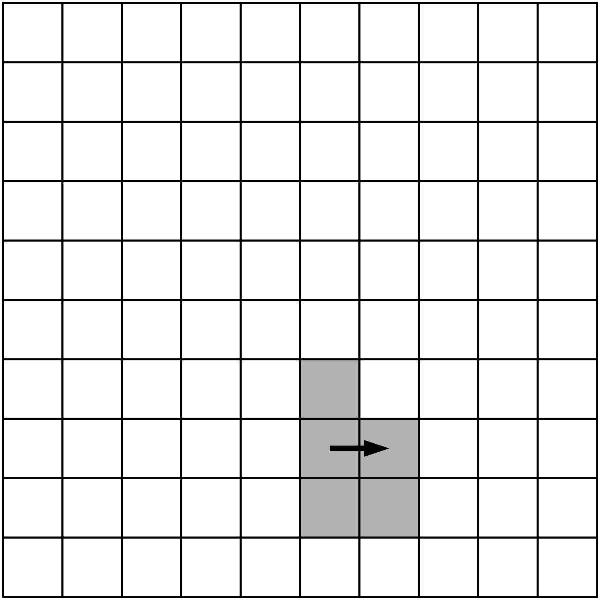
# Some evolution steps



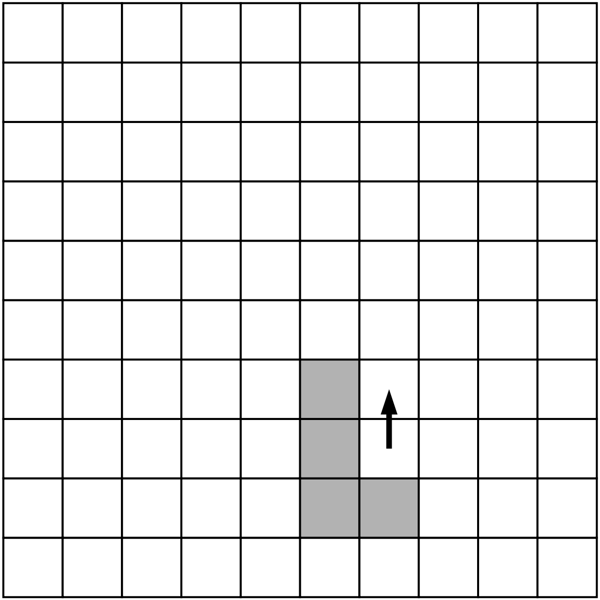
# Some evolution steps



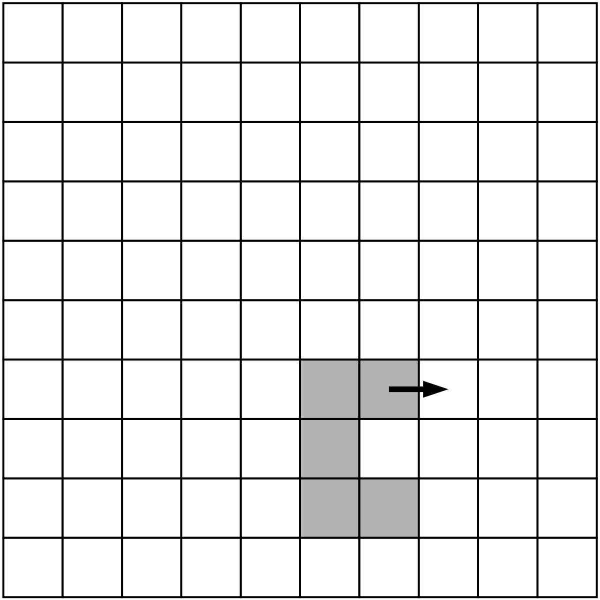
# Some evolution steps



# Some evolution steps



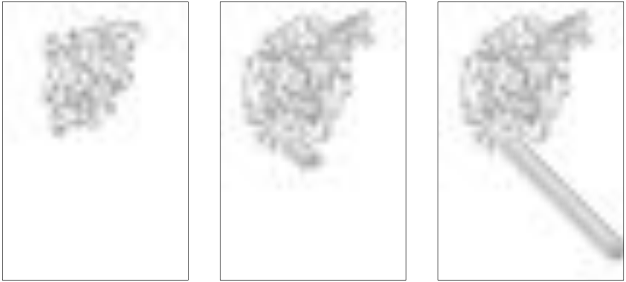
# Some evolution steps



## Where does the ant go in the long run

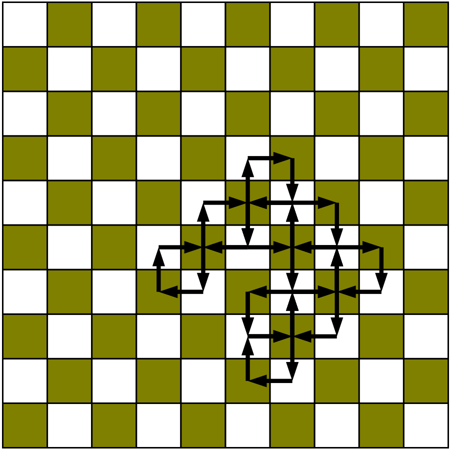
- Animation...

# Where does the ant go in the long run



- t=6900 t=10431 t=12000

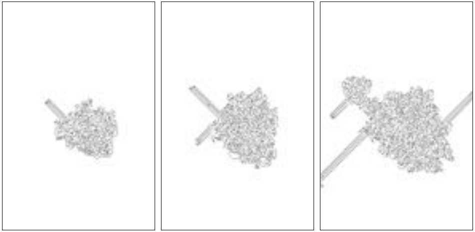
# The ants always escape to infinity



for any initial coloration of the cells

# What about many ants ?

- Adapt the "change of color" rule
- Cooperative and destructive effects
- The trajectory can be bounded or not
- Past/futur symmetry explains periodic motion

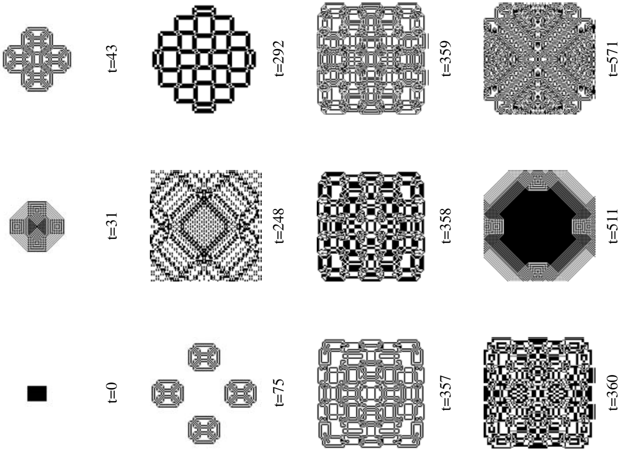


t=2600 t=4900 t=8564

# Impact on the scientific methodolgy

- The laws are perfectly known
- But we cannot predict the details of the movements (when does a highway appears)
- Microscopic knolwdge is not enough to predict the macroscopic behavior
- Then, the only solution is the **observe** the behavior
- The only information we have on the trajectory are the reflect of the symmetries of the rule

# Prediction means to compute faster than reality

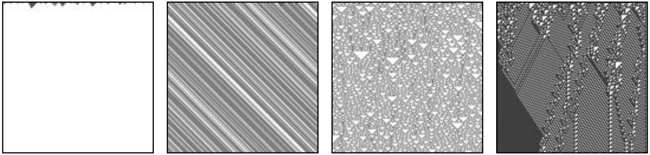


# Prediction means to compute faster than reality


(a) (b) (c)

# Wolfram's rules

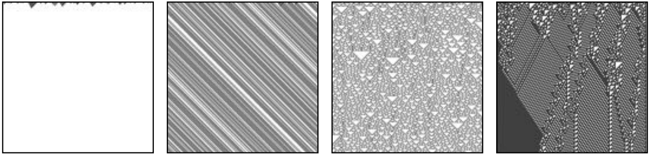
256 one-dimensional, 3 neighbors Cellular Automata :



(a) (b) (c) (d)

Coombs, Stephen 2009, The Geometry and Pigmentation of Seashells

# Wolfram's rules : complexity classes

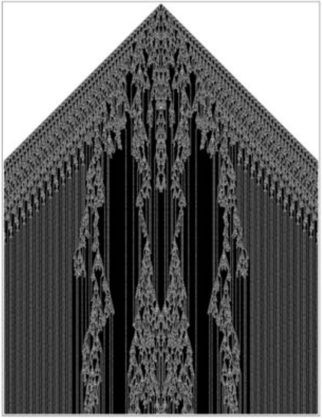


(a) (b) (c) (d)

- **Class I** Reaches a fixed point
- **Class II** Reaches a limit cycle
- **Class III** self-similar, chaotic attractor
- **Class IV** unpredicable persistent structures, irreducible, universal computer

Note : it is **undecidable** whether a rule belongs or not to a given class.

# Wolfram's rules : 1D, 5 neighbors



## Other simple rules

- time-tunnel

$$\begin{aligned} Sum(t) &= C(t) + N(t) + S(t) + E(t) + W(t) \\ C(t+1) &= \begin{cases} C(t-1) & \text{if } Sum(t) \in \{0,5\} \\ 1 - C(t-1) & \text{if } Sum(t) \in \{1,2,3,4\} \end{cases} \end{aligned}$$

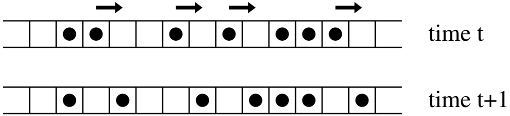
- random

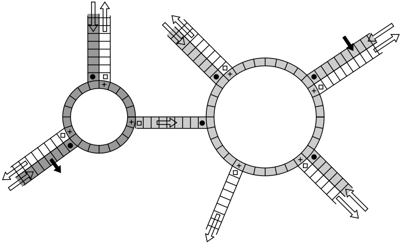
$$C(t+1) = (S(t).and.E(t)).xor.W(t).xor.N(t).xor.C(t)$$

- string : a one-dimensional spring-bead system

# Traffic Models

A vehicle can move only when the downstream cell is free.



# Flow diagram

The car density at time $t$ on a road segment of length $L$ is defined as

$$\rho(t) = \frac{N(t)}{L}$$

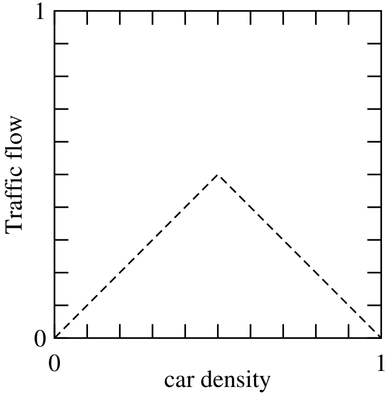
where $N$ is the no of cars along $L$
The average velocity $< v >$ at time $t$ on this segment is defined as

$$< v >= \frac{M(t)}{N(t)}$$

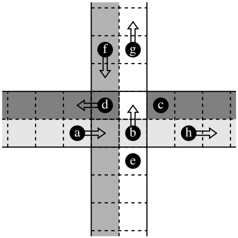
where $M(t)$ is the number of car moving at time $t$
The traffic flow $j$ is defined as

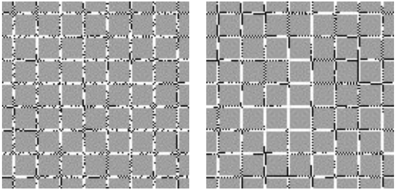
$$j = \rho < v >$$

# Flow diagram of rule 184

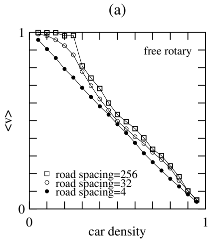


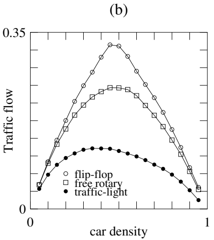
# Traffic in a Manhattan-like city



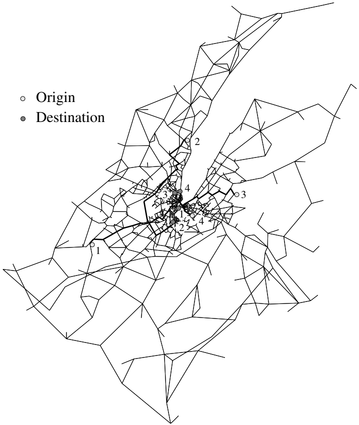
(a) (b)



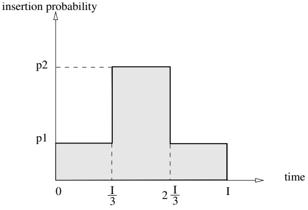



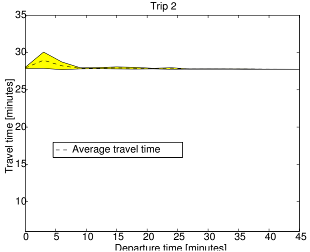
# Case of the city of Geneva

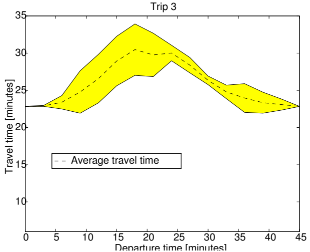
- 1066 junctions
- 3145 road segments
- 560886 road cells
- 85055 cars



# Travel time during the rush hour

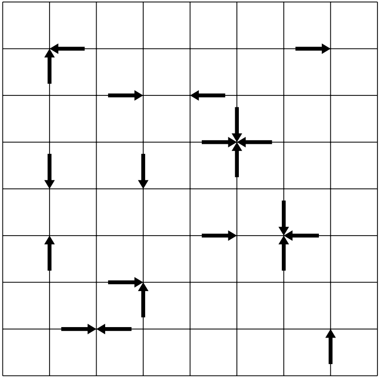


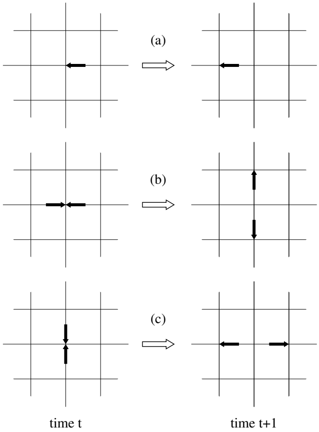
# Lattice gases

Fully discrete molecular dynamics



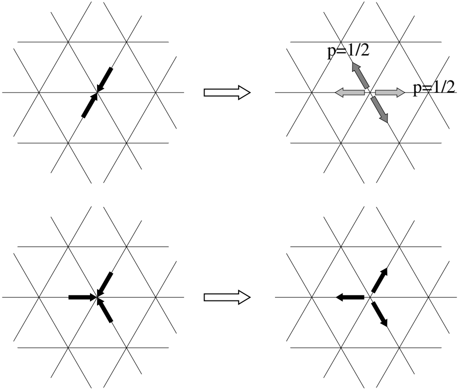
# Example : HPP model collision rules

- HPP : Hardy, Pomeau, de Pazzis, 1971 : kinetic theory of point particles on the D2Q4 lattice
- FHP : Frisch, Hasslacher and Pomeau, 1986 : first LGA reproducing a (almost) correct hydrodynamic behavior (Navier-Stokes eq.)



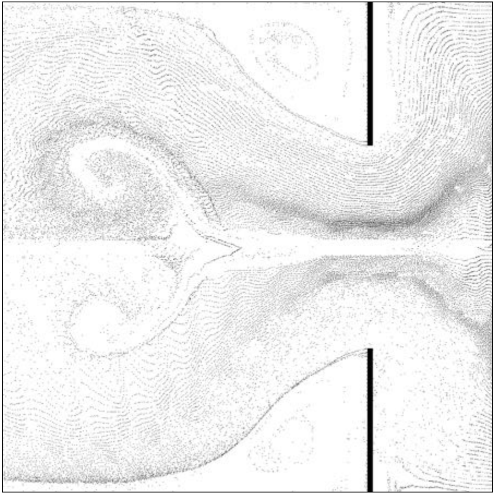

Exact mass and momentum conservation : that is what really matters for a fluid ! ! !

# FHP model




Stochastic rule with Conservation of mass and momentum.

# Flow past an obstacle (FHP)



## Why can such a simple model work ?

- At a macroscopic scale, the detail of the interaction does not matter so much
- Only conservation laws and symmetries are important
- We can invent our own fluid, especially one adapted to computer simulation

## Demos

- Pressure/density wave : aniotropy
- Reversibility
- Spurious invariants : momentum along each line and column, checkerboard invariant
- Diffusion, DLA, reaction-diffusion models
- Snow transport by wind

# Lattice Boltzmann (LB) models

- Lattice Gases implement an exact dynamics
- But they require large simulations, statistical averages and have little freedom to adjust problem parameters
- In the early 1990s, the discrete Boltzmann equation describing the average dynamics of a lattice Gas was re-interpreted (with improvements) as a flow solver
- $\rightarrow$ Lattice Boltzmann models

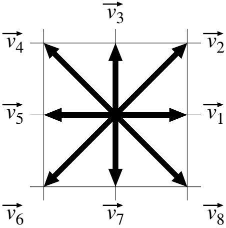
# The lattice Boltzmann (LB) method : the historical way

- Historically, LB was born from **Lattice Gases**, discrete kinetic models of colliding particles
- Now the LB method is often derived by a discretization procedure (in velocity, space and time variables) of the standard Boltzmann equation

$$\partial_t f(v,r,t) + v \cdot \partial_r f(v,r,t) = \Omega(f)$$

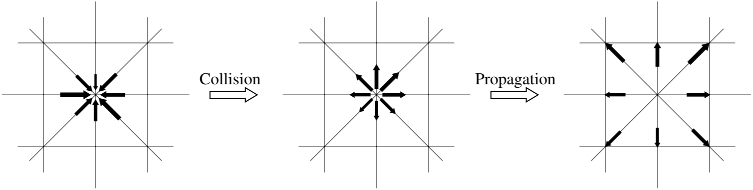
- where $f(v,r,t)$ is the density distribution of particles at location $r$, time $t$, with velocity $v$.

# The Lattice Boltzmann scheme : definitions



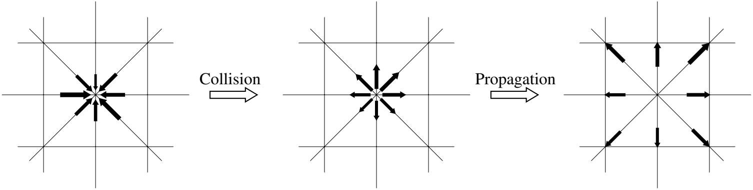

- Possible particle velocities : $\mathbf{v}_i$, $i = 0, 1, ..., q-1$
- Lattice spacing : $\Delta x$, time step : $\Delta t$, $|v_i| = \Delta x / \Delta t$.

# The Lattice Boltzmann scheme : definitions



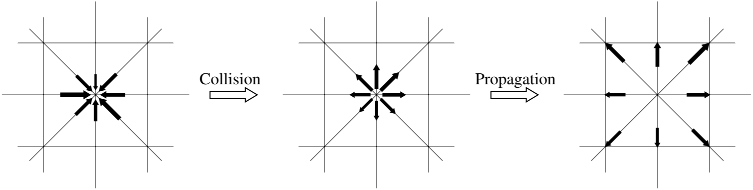

- Possible particle velocities : $\mathbf{v}_i$, $i = 0, 1, ..., q-1$
- Lattice spacing : $\Delta x$, time step : $\Delta t$, $|v_i| = \Delta x / \Delta t$.
- $f_i^{in}(\mathbf{r}, t)$ is the density of particle entering site $\mathbf{r}$ with velocity $\mathbf{v}_i$, at time $t$.

# The Lattice Boltzmann scheme : definitions



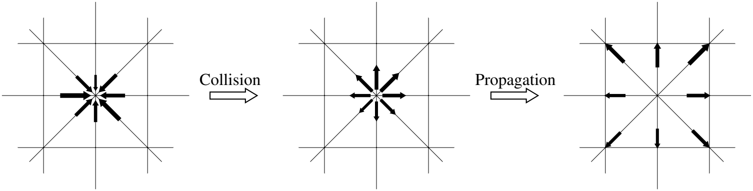

- Possible particle velocities : $\mathbf{v}_i$, $i = 0, 1, ..., q-1$
- Lattice spacing : $\Delta x$, time step : $\Delta t$, $|v_i| = \Delta x / \Delta t$.
- $f_i^{in}(\mathbf{r}, t)$ is the density of particle entering site $\mathbf{r}$ with velocity $\mathbf{v}_i$, at time $t$.
- Density : $\rho(\mathbf{r}, t) = \sum_i f_i^{in}$ ;

# The Lattice Boltzmann scheme : definitions



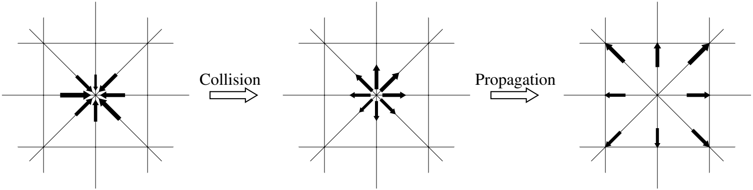

- Possible particle velocities : $\mathbf{v}_i$, $i = 0, 1, ..., q-1$
- Lattice spacing : $\Delta x$, time step : $\Delta t$, $|v_i| = \Delta x / \Delta t$.
- $f_i^{in}(\mathbf{r}, t)$ is the density of particle entering site $\mathbf{r}$ with velocity $\mathbf{v}_i$, at time $t$.
- Density : $\rho(\mathbf{r}, t) = \sum_i f_i^{in}$ ;
- Velocity : $\rho \mathbf{u} = \sum_i f_i^{in} \mathbf{v}_i$

# The Lattice Boltzmann scheme : definitions




- Possible particle velocities : $\mathbf{v}_i$, $i = 0, 1, ..., q-1$
- Lattice spacing : $\Delta x$, time step : $\Delta t$, $|v_i| = \Delta x / \Delta t$.
- $f_i^{in}(\mathbf{r}, t)$ is the density of particle entering site $\mathbf{r}$ with velocity $\mathbf{v}_i$, at time $t$.
- Density : $\rho(\mathbf{r}, t) = \sum_i f_i^{in}$ ;
- Velocity : $\rho\mathbf{u} = \sum_i f_i^{in}\mathbf{v}_i$
- Momentum tensor $\Pi_{\alpha\beta} = \sum_i f_i^{in}(\mathbf{r}, t) v_{i\alpha} v_{i\beta}$

# The Lattice Boltzmann scheme : dynamics




- Collision : $f_i^{out} = f_i^{in} + \Omega_i(f)$
- Propagation : $f_i^{in}(\mathbf{r} + \Delta t \mathbf{v}_i, t + \Delta t) = f_i^{out}(\mathbf{r}, t)$

**Collision and Propagation :**

$$f_i(\mathbf{r} + \Delta t \mathbf{v}_i, t + \tau) = f_i(\mathbf{r}, t) + \Omega_i(f) \tag{1}$$

where $f = f^{in}$

# The single relaxation time LB scheme (BGK)

The collision term $\Omega_i$ is a relaxation towards a prescribed **local equilibrium** distribution

$$\Omega_i(f) = \frac{1}{\tau}(f_i^{eq}(\rho, \mathbf{u}) - f_i) \tag{2}$$

where

$$f_i^{eq} = \rho w_i (1 + \frac{\mathbf{v}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{c_s^4} Q_{i\alpha\beta} u_\alpha u_\beta) \tag{3}$$

contains the desired physics (here hydrodynamics) and $Q_{i\alpha\beta}$ is

$$Q_{i\alpha\beta} = v_{i\alpha} v_{i\beta} - c_s^2 \delta_{\alpha\beta}$$

$\tau$ is a constant called the **relaxation time**

# Choice of the $\mathbf{v}_i$ and lattice weight $w_i$

The “microscopic” velocities $\mathbf{v}_i$ must be such that there exists constants $w_i$ and $c_s^2$ so that :

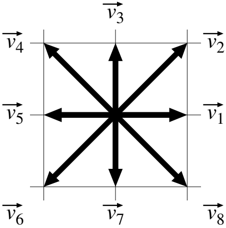
$$\begin{aligned}
\sum_i w_i &= 1 \\
\sum_i w_i \mathbf{v}_i &= 0 \\
\sum_i w_i v_{i\alpha} v_{i\beta} &= c_s^2 \delta_{\alpha\beta} \\
\sum_i w_i v_{i\alpha} v_{i\beta} v_{i\gamma} &= 0 \\
\sum_i w_i v_{i\alpha} v_{i\beta} v_{i\gamma} v_{i\delta} &= c_s^4 \left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right) \\
\sum_i w_i v_{i\alpha} v_{i\beta} v_{i\gamma} v_{i\delta} v_{i\epsilon} &= 0 \qquad (4)
\end{aligned}$$

# Lattice Geometries D$d$Q$q$

$d$ is the space dimension and $q$ the number of microscopic velocities

- D2Q9 : 2D, square lattice with diagonals and rest particles.
- D3Q19 : 3D, with rest particles

have enough symmetries.




$$w_0 = 4/9 \qquad w_1 = w_3 = w_5 = w_7 = 1/9$$

$$w_2 = w_4 = w_6 = w_8 = 1/36$$

# Continuous limit

Up to order $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta t^2)$, and provied that $Ma << 1$, the LB eq.

$$f_i(\mathbf{r} + \Delta t \mathbf{v}_i, t + \tau) = f_i(\mathbf{r}, t) + \frac{1}{\tau}(f_i^{eq} - f_i) \tag{5}$$

is equivalent to Navier-Stokes equations

$$\begin{cases} \partial_t \rho + \partial_\alpha \rho u_\alpha = 0 \\ \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} \end{cases} \tag{6}$$

for $\rho = \sum_i f_i$ and $\rho\mathbf{u} = \sum_i f_i \mathbf{u}$.

## Properties :

Viscosity :

$$\nu = c_s^2 \Delta t (\tau - 1/2)$$

Pressure :

$$p = \rho c_s^2$$

Thus, LB-fluids are compressible

# Relations between the $f_i$'s and the hydrodynamic quantites

| Hydrodynamic quantities from the $f_i$ | $f_i$ from the hydrodynamic quantities |
|---|---|
| $\rho = \sum_i f_i$<br>$\rho\mathbf{u} = \sum_i f_i \mathbf{v}_i$<br>$\Pi_{\alpha\beta} = \sum_i v_{i\alpha} v_{i\beta} f_i$ | $f = f^{eq} + f^{neq}$<br>$f_i^{eq} = \rho w_i (1 + \frac{\mathbf{v}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} Q_{i\alpha\beta} u_\alpha u_\beta)$<br>$f_i^{neq} = -\Delta t \tau \frac{w_i}{c_s^2} Q_{i\alpha\beta} \rho S_{\alpha\beta}$ |

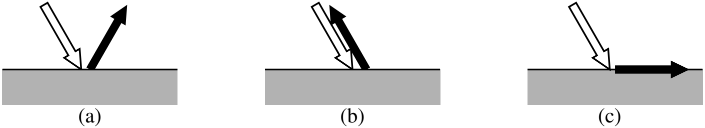
where

$$S_{\alpha\beta} = (1/2)(\partial_\alpha u_\beta + \partial_\beta u_\alpha)$$

and

$$Q_{i\alpha\beta} = v_{i\alpha} v_{i\beta} - c_s^2 \delta_{\alpha\beta}$$

# Boundary conditions

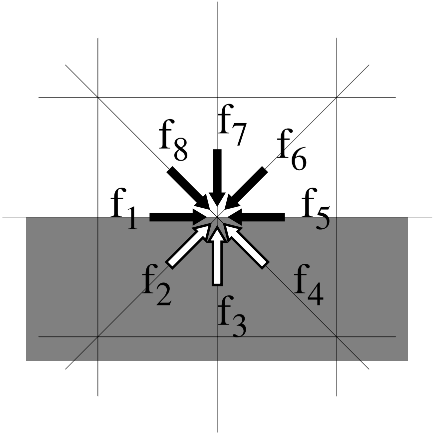



(a) Specular reflection, (b) bounce back condition and (c) trapping wall condition

The **Bounce Back rule** implements a no-slip condition. It is the most common choice :

$$f_i^{out} = f_{-i}^{in}$$

# Boundary conditions : beyond bounce-back




Compute the missing population so as to have the desired physical properties

# Pros and cons on the LB method

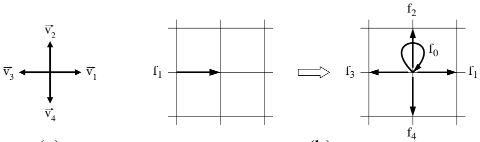
+ Closer to physics than to mathematics
+ Quite flexible to new developments, intuitive, multiphysics
+ Complicated geometries, cartesian grids
+ no need to solve a Poisson equation
+ Parallelization

- Recent methods
- No efficient unstructured grids
- Intrinsically a time dependent solver
- Not always so easy
- Still some work to have a fully consistent thermo-hydrodynamical model.

# More advantages...

- Streaming is exact
- Non-linearity is local
- Numerical viscosity is negative
- Extended range of validity for larger Knudsen numbers
- Palabos open source LB software (http ://www.palabos.org)

# Wave equation




(a) (b)

$$f_i(\mathbf{r} + \tau \mathbf{v}_i, t + \tau) = f_i(\mathbf{r}, t) + 2(f_i^{eq} - f_i) \tag{7}$$

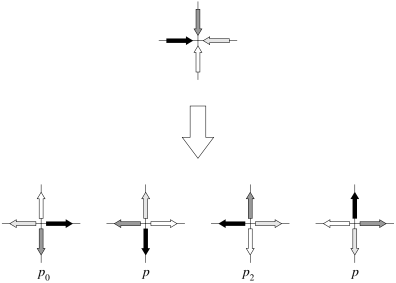
$$f_i^{eq} = a\rho + b\mathbf{u} \cdot \mathbf{v}_i$$

Conservation of $\rho$, its current $\mathbf{u}$ and time reversibility. Note that $\sum f_i^2$ is also conserved.

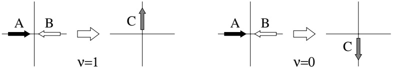
This is equivalent to

$$\partial_t^2 \rho + c^2 \nabla^2 \rho = 0$$

# CA for Reaction-Diffusion processes




Diffusion




Reaction

# LB Reaction-Diffusion

$$f_i(\mathbf{r} + \Delta t \mathbf{v}_i, t + \tau) = f_i(\mathbf{r}, t) + \omega(f_i^{eq} - f_i) + \frac{\Delta t}{2d} R \tag{8}$$

with $R$ the reaction term (for instance $R = -k\rho^2$). and

$$f^{eq} = \frac{1}{2d}\rho$$

This is equivalent to

$$\partial_t \rho = D\nabla^2 \rho + R$$

# Demos

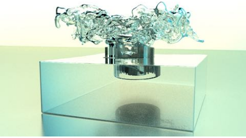
http ://cui.unige.ch/∼chopard/CA/Animation/root.html

# Palabos : an Open-Source solver (UNIGE)

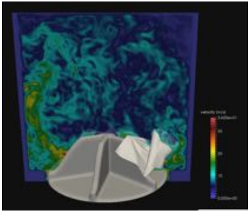
Multiphysics, same code from laptop to massively parallel computer : (www.palabos.org)

**Droplet**

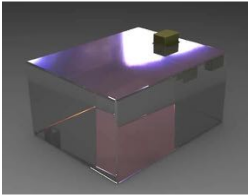
**Pumps**



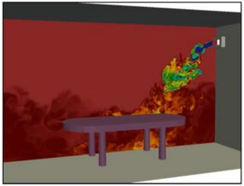
**Washing machines**



**Energy converter**

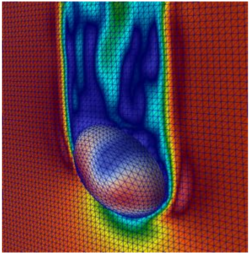


**Air conditioning**

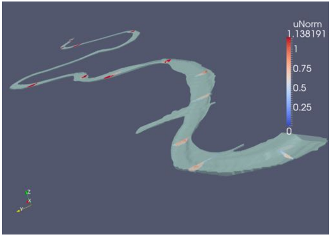


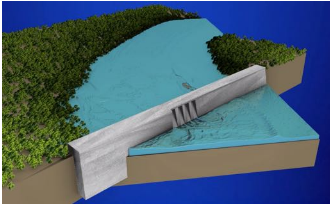
Fixed air-conditioner

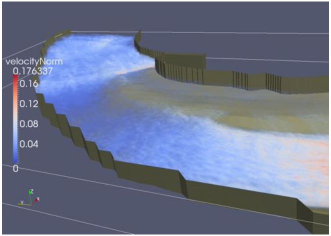
**sedimentation**



# Simulation of river Rhone in Geneva



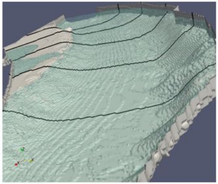






Draining of the river Rhône, with riverbed erosion.

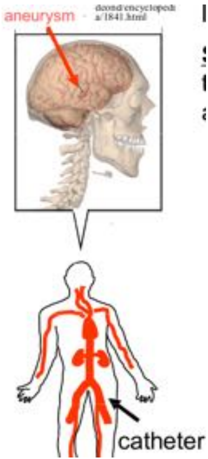
Water level is shown in semi-transparent light-blue color.

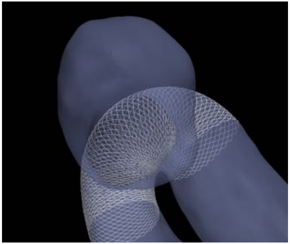
palabos.org

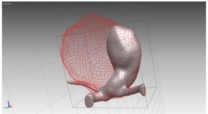


# How to treat cerebral aneurysms : flow diverters

thrombus





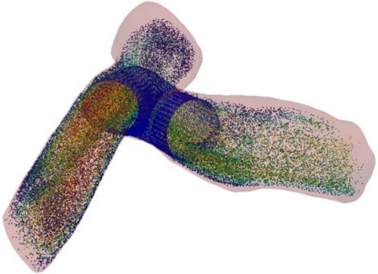
- The stent **reduces bloodflow** in the aneurysm
- Clotting is induced in the aneurysm

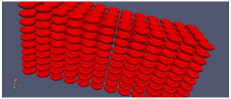
Our goal is to elucidate the mechanisms leading to thrombus formation from biological knowledge and numerical modeling

# Fully resolved simulation with a flow diverter

*Pipeline* flow diverter from EV3-COVIDIEN

| $\Delta x$ | $\Delta t$ | diameter | # fluid nodes | Re |
|---|---|---|---|---|
| 25 $\mu$m | 1 $\mu$s | 3.7 mm | 40 millions | $\approx 300$ |

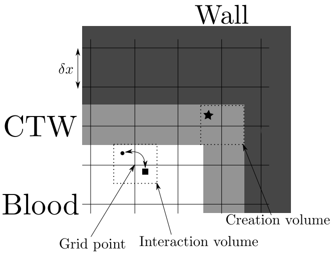




CPU time : 10 days (on 120 Westmere Intel cores)

# Spatio-temporal Thrombosis Model

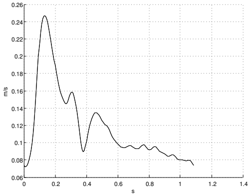
- Low shear : creation of TF, then thrombin from endothelial cells
- Fibrinogen and anti-thrombin are in suspension, brought by fresh blood
- thrombin+fibrinogen $\rightarrow$ fibrin (=clot)
- thrombin+anti-thrombin $\rightarrow$ 0
- Platelets attach to the fibrin, compact the clot and allow re-endothelialization
- Clot stops to grow when all thrombin molecules have been consumed



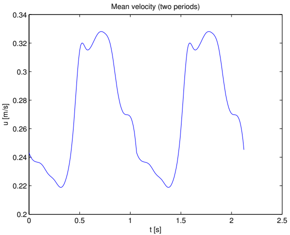

Need clever multiscale solutions for the numerical implementation

# Thrombosis Model

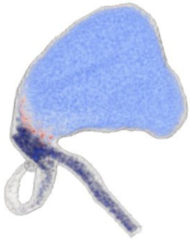
Pulsatile versus steady flow




# Simulation of the thrombus in giant aneurysm



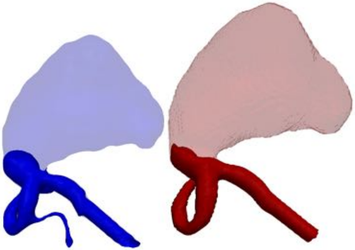

- movie
- accelerated for 2200 heart cycles

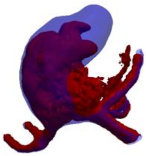


| $\nu$ | $\rho$ | inlet diam. | aneurysm size | inlet flow |
|---|---|---|---|---|
| 3.7e-6 $m^2/s$ | 1080 $kg/m^3$ | 0.8 $mm$ | 8 $cm$ | $4 \times 10^{-6}$ $m^3/s$ |

# Validation with a patient

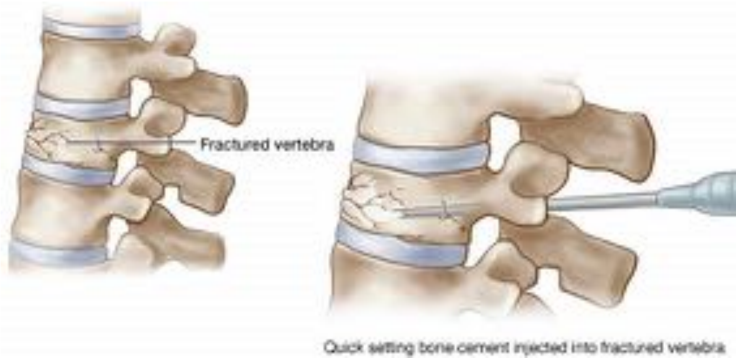
Blue : patient Red : simulation



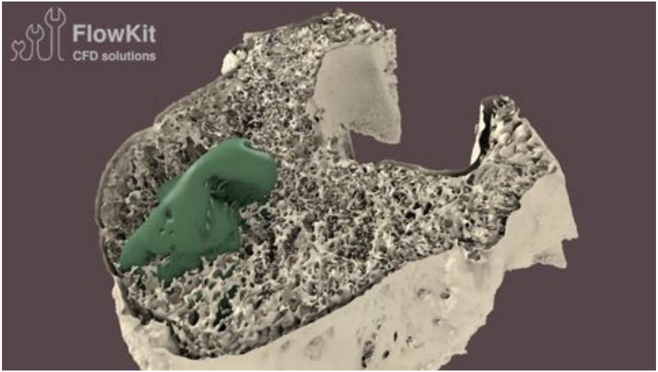


Another case

# Vertebroplasty



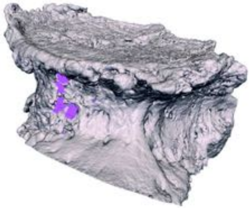
# Palabos Simulation

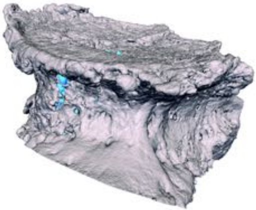


# Experiment versus simulation

After 6 ml

After 7 ml

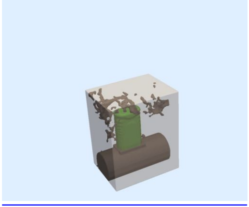


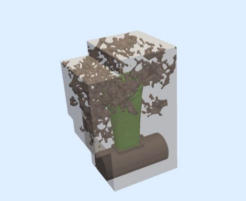


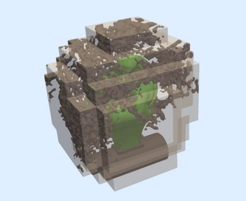

Good agreement within experimental errors

# Dynamical load balancing on Palabos

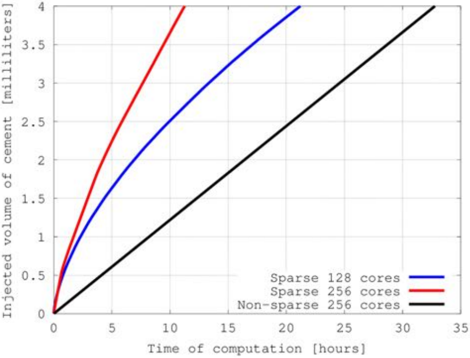
Domains reallocation at regular time intervals







Performance with and without data migration



# Exercices

- Play with a python code producing a 2D flow around a sphere **(d2q9.py)**. For instance, change the Reynolds number RE
- Play with a python code modeling the movement of bacteria in a field of nutrients **(bacteria.py)**. Try to add a source and diffusion of nutrients, and the change in concentration when eaten by the bacteria

http://cui.unige.ch/~chopard/FTP/USI/

# Acknowledgments

- Jonas Latt
- Yann Thorimbert
- Orestis Malaspinas