Introduction to Computational Physics

Cellular Automata and a Simple Gas Model

Andreas Adelmann

Paul Scherrer Institut, Villigen
E-mail: andreas.adelmann@psi.ch

https://moodle-app2.let.ethz.ch/course/view.php?id=18025

Plan for today

- Introduction
- Complex Systems & Cellular Automata
  - The Game of Life
  - The Langton Ant
- Dynamical Systems as Cellular Automata
  - One-Dimensional Cellular Automata
  - Time Evolution
  - Classes of Automata
  - Traffic Models
  - A Simple Model for a Gas of Particles

Comments I

- Cellular automata (CA) provide a natural modelling framework to describe and study many physical systems composed of interacting components.
- The reason of this success is the close relation between theses methods and a **mesoscopic abstraction** of many natural phenomena

Example: Some of the fluid models follow the Eulerian formulation of fluid mechanics. In these models, the fluid is considered as a continuum subject to the conservation laws for mass, momentum (Newton's law) and energy as well as the state equations connecting the macroscopic variables that define the thermodynamic states of the fluid: pressure P, density ρ and

Comments II

temperature T. The mass conservation, also called the continuity equation, is then given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1}$$

where $\mathbf{u}$ is the velocity of the fluid. The momentum conservation equation (Navier-Stokes), can be obtained by applying the Newton's second law to a volume element dV of the fluid. It can be written as

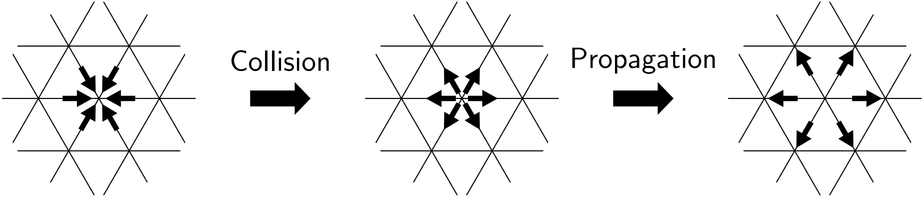
$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u}, \tag{2}$$

where we assume no external force and the fluid to be incompressible (density ρ is constant) for the moment. This equation together with the incompressible version of the continuity equation $\nabla \cdot \mathbf{u} = 0$ describes the dynamics of incompressible fluids.

Let's discuss solution methods

Comments III

Phenomena ⇒ **PDE** ⇒ **Discretisation** ⇒ **Computer Solution**



Phenomena ⇒ **Computer Solution**

6.1 Introduction I

- simple example, due to Edward Fredkin (1970s)
- exhibits a surprisingly rich behavior
- defined on a two-dimensional square lattice

Ingredients:

- each site of the lattice is a cell which is labeled by its position $\mathbf{r} = (i, j)$
- i and j are the row and column indices.
- a function $\psi(\mathbf{r}, t) \in \mathbb{B}$ is associated with the lattice to describe the state of each cell $\mathbf{r}$ at iteration t.

The cellular automata rule specifies

- how the states $\psi(\mathbf{r}, t+1)$ are to be computed from the states at iteration t.
- initial condition at time $t = 0$ with a
- given configuration of the values $\psi(\mathbf{r}, t = 0)$ on the lattice.

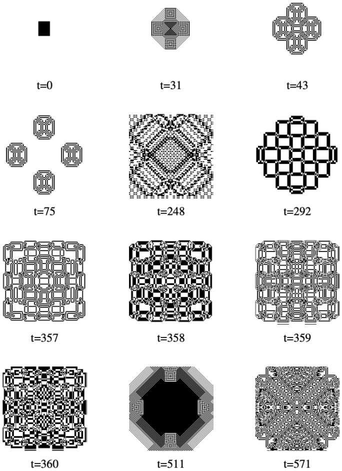
6.1 Introduction II

The state at time $t = 1$ is obtained as follows

1. Each site $\mathbf{r}$ computes the sum of the values $\psi(\mathbf{r}^{'}, t = 0)$ on the four nearest neighbor sites $\mathbf{r}^{'}$ at north, west, south and east. The system is supposed to be periodic in both i and j directions (like on a torus) so that this calculation is well defined for all sites.
2. If this sum is even, the new state $\psi(\mathbf{r}^{'}, t = 1)$ is 0 (white) and, else, it is 1 (black)

The same rule (steps 1 and 2) is repeated over to find the states at time $t = 2, 3, \ldots$.

6.1 Introduction III



6.1 Introduction IV

From a mathematical point of view, this parity rule can be expressed by the following relation

$$\psi(i,j,t+1) = \psi(i+1,j,t) \otimes \psi(i-1,j,t) \otimes \psi(i,j+1,t) \otimes \psi(i,j-1,t) \quad (3)$$

where the symbol $\otimes$ stands for the exclusive OR logical operation.
It is also the sum modulo 2: $1 \otimes 1 = 0 \otimes 0 = 0$, and $1 \otimes 0 = 0 \otimes 1 = 1$

- this property of generating complex patterns starting from a simple rule is generic of many cellular automata rules.
- complexity results from some spatial organization which builds up as the rule is iterated.
- the various contributions of successive iterations combine together in a specific way.
- the spatial patterns that are observed reflect how the terms are combined algebraically.

Definition Cellular Automata

Definition (Cellular Automata)

A cellular automata (CA) is a tuple $(\mathcal{L}, \psi, R, \mathcal{N})$ where

(i) $\mathcal{L}$ is a regular lattice of cells covering a portion of a d-dimensional space

(ii) $\psi(\mathbf{r}, t) = \{\psi_1(\mathbf{r}, t), \ldots, \psi_m(\mathbf{r}, t)\}$ is a set of m Boolean variables attached to each site $\mathbf{r}$ of the lattice and giving the local state of the cells at time t

(iii) R is a set of rules, $R = \{R_1, \ldots, R_m\}$, which specifies the time evolution of the states $\psi(\mathbf{r}, t)$ in the following way

$$\psi_j(\mathbf{r}, t + \delta_t) = R_j(\psi(\mathbf{r} + \mathbf{v}_1, t), \psi(\mathbf{r} + \mathbf{v}_2, t), \ldots, \psi(\mathbf{r} + \mathbf{v}_q, t)) \quad (4)$$

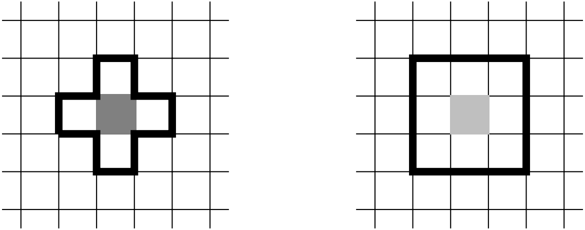
where $\mathbf{r} + \mathbf{v}_k$ designate the cells belonging to the neighbourhood $\mathcal{N}$ of cell $\mathbf{r}$.

We see that there are thus 2^{2^m} possible rules!

Remarks

- in the above definition, the rule R is identical for all sites
- R is applied simultaneously to all sites $\Rightarrow$ synchronous dynamics
- the number of configurations of the neighbourhood is finite, it is common to pre-compute all the values of R in a lookup table.
- alternative an algebraic expression can be used and evaluated at each iteration, for each cell, as in eq. (3).
- the rule is homogeneous, that is it does not depend explicitly on the cell position $\mathbf{r}$.
- spatial (or even temporal) inhomogeneities can be introduced by having some $\psi(\mathbf{r})$ systematically 1 in some given locations of the lattice to mark particular cells on which a different rule applies. Boundary cells are a typical example of spatial inhomogeneities.
- similarly, it is easy to alternate between two rules by having a bit which is 1 at even time steps and 0 at odd time steps.

Neighborhoods



- (left) Von Neumann neighborhood
- (right) Moore neighborhood
- The shaded region indicates the central cell which is updated according to the state of the cells located within the domain marked with the bold line

Boundary Conditions

periodic

fixed

adiabatic

reflection

Summary

A cellular automaton is a model that consists of

- regular grid of cells
- each one of a finite number of states (e.g. ± 1)
- grid can have any finite number of dimensions
- it is a discrete model
- describes discrete deterministic dynamics $t \to t + 1$,

That is to say that all three major components are discrete:

- the variables have discrete values (boolean, i.e. 0 and 1)
- sites on the lattice are discrete
- finite time steps: next step $t + 1$ is based on t.

6.2 Complex Systems & Cellular Automata I

- complex systems are everywhere in sciences and the daily live.
- systems are made of many interacting constituents
- exhibit spatio-temporal patterns
- collective behaviours

Need a methodology to cope:

- standard and successful methodology in research has been to isolate phenomena from each other and to study them independently $\Rightarrow$ divide and concur
- this leads to a deep understanding of the phenomena themselves but also leads to a compartmentalized view of nature.
- the real world is made of interacting processes and these interactions brings new phenomena that are not present in the individual constituents.

6.2 Complex Systems & Cellular Automata II

⇒ the whole system is more that the sum of its parts and new scientific tools and concepts may be required to analyze complex systems

⇒ CAs offer such a possibility by being themselves simple, fully discrete complex systems

6.2.1 The Game of Life I

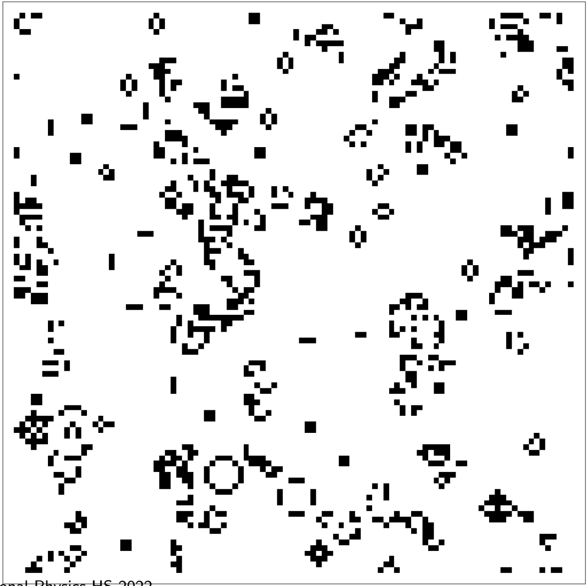
- in 1970, the mathematician John Conway proposed the now famous game of life
- the motivation was to find a simple rule leading complex behaviours in a system of fictitious one-cell organisms evolving in a fully discrete universe
- the surrounding cells corresponds to a Moore neighborhood
- complex structures emerge out of a primitive "soup" and evolve or die.

Let us consider a square lattice, and let n be the number of nearest and next-nearest neighbors that are 1. We shall then use the following rule:

- if $n < 2$: 0 dead because of isolation
- if $n = 2$: stay as before
- if $n = 3$: 1 birth
- if $n > 3$: 0 dead because of over overpopulation

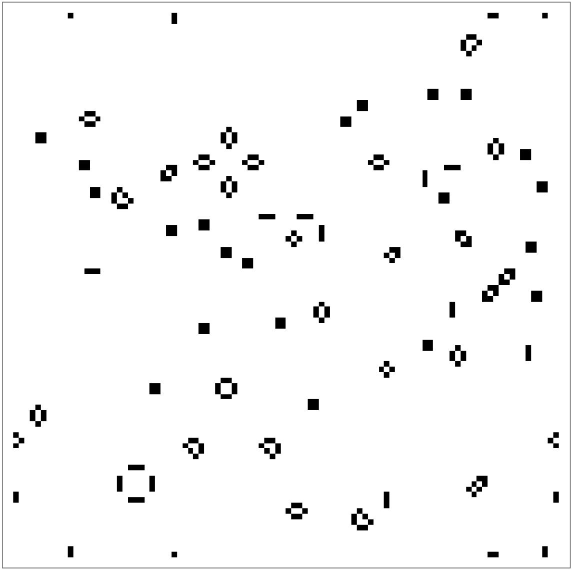
6.2.1 The Game of Life

$t = n$



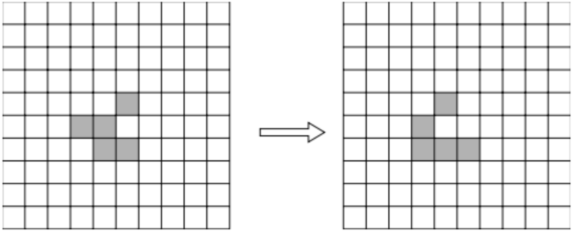
6.2.1 The Game of Life

$t = \infty$



6.2.1 The Game of Life

What does “evolves” mean?



A detailed structure of a glider, over two consecutive iterations:

- a glider is an assembly of cells that has a higher functionality
- capability to move in space by changing its internal structure

> - this is a signature of complex systems
> - more complex objects can be built, such a as for instance glider guns which are arrangements of cell producing gliders

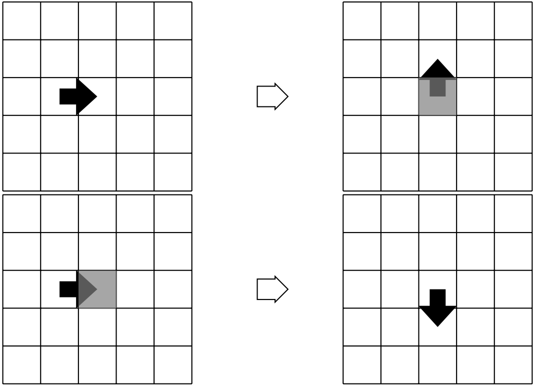
6.2.2 The Langton Ant I

- the ant rule is a CA invented by Chris Langton and Greg Turk
- models the behavior of a hypothetical animal having a very simple algorithm of motion

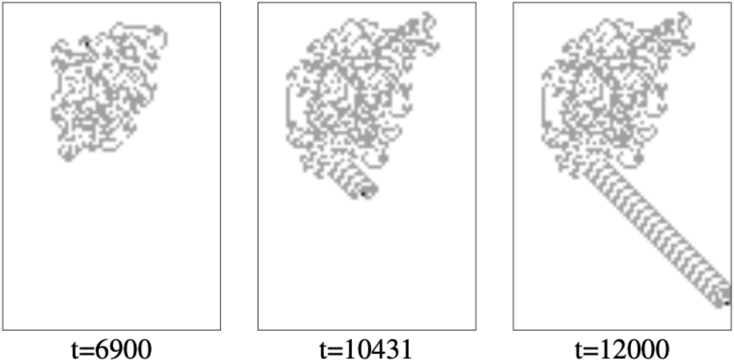
The ant moves on a square lattice whose sites are either white or gray.

1. when the ant enters a white cell, it turns 90 degrees to the left and paints the cell in gray
2. if it enters a gray cell, it paints it in white and turn 90 degree to the right

6.2.2 The Langton Ant II



6.2.2 The Langton Ant III



Observations:

1. chaotic phase of about 10000 time steps
2. formation of a highway
3. walking on highway

6.2.2 The Langton Ant IV

- $\Rightarrow$ we are not able to predict the detailed motion analytically
- $\Rightarrow$ the only way is to perform the simulation
- $\Rightarrow$ learned about the global behavior of the system: ant goto ∞
- $\Rightarrow$ observation: often, global features are related to symmetries more than to details
- $\Rightarrow$ the proof of the theorem is based on symmetries

6.3 Dynamical Systems as Cellular Automata

In general

- a cellular automaton is a model that consists of a regular grid $\mathcal{L}$
- cells, each one of a finite number of states (e.g. ± 1).
- the grid can be in any finite number of dimensions d.
- the model is discrete i.e. we are dealing with boolean variables on $\mathcal{L}$ form t to $t+1$.
- deterministic dynamics

For a fixed rule R_j and q inputs we get for $\psi_j(\mathbf{r}, t+\delta_t)$

$$\psi_j(\mathbf{r}, t+\delta_t) = R_j(\psi(\mathbf{r}+\mathbf{v}_1, t), \psi(\mathbf{r}+\mathbf{v}_2, t), \ldots, \psi(\mathbf{r}+\mathbf{v}_q, t)).$$

6.3.1 One-Dimensional Cellular Automata

- rule involves only the nearest-neighbors sites
- systematic study of these rules was undertaken by S. Wolfram in 1983.

For $q = 3$:

- the state ψ at time $t + 1$ depends only on the triplet $\psi(r-1), \psi(r), \psi(r+1)$ at time t:
- thus to each possible values of the triplet $\psi(r-1), \psi(r), \psi(r+1)$, one associates a value $\alpha_k = 0|1$ according to

	111	110	101	100	011	010	001	000
k:	7	6	5	4	3	2	1	0

Each possible cellular automata rule R is characterized by the values $\alpha_0 \dots \alpha_7$. There are clearly 2^{2^q} rules and with i.e. $q = 3 \Rightarrow 256$ possible choices. Each rule can be identified by an index c computed as follows

$$c = \sum_{k=0}^{2^q-1} 2^k \alpha_k$$

6.3.1 One-Dimensional Cellular Automata

Example $c = 101$

Let us consider the following rule ($c = 101$):

entries:	111	110	101	100	011	010	001	000
α:	0	1	1	0	0	1	0	1

Furthermore, we define

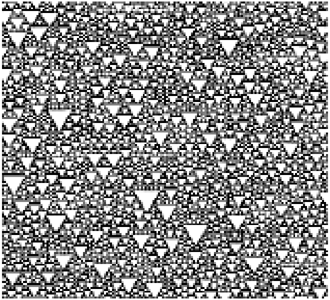
$$c = \sum_{n=0}^{2^k-1} 2^n \alpha_n$$

which for the presented rule is

$$c = 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 64 + 32 + 4 + 1 = 101$$

6.3.2 Time Evolution

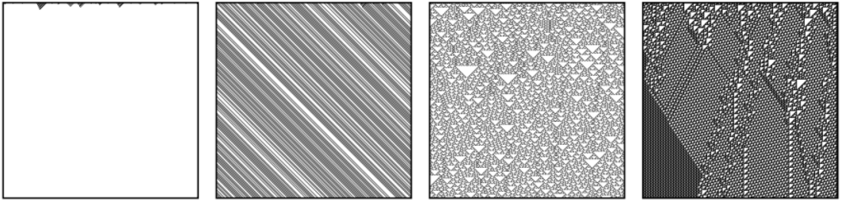
- given the very different character of the rules, the evolution patterns differ significantly
- in some cases special patterns start to appear while in other cases, the area goes blank
- different rules have been classified according to global behaviour



According to their behavior, the different rules have been classified.

6.3.3 Classes of Automata I

Wolfram classification



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

- Class 1 (fig a): Almost all initial patterns evolve quickly into a stable, homogeneous state, any randomness in the initial pattern disappears. The set of exceptional initial configurations which behave differently is of measure zero when the number of cells N goes to infinity. From the point of view of dynamical systems, these automata evolve towards a simple limit point in the phase space.

6.3.3 Classes of Automata II

Wolfram classification

- Class 2 (fig b): Almost all initial patterns evolve quickly into stable or oscillating structures. Some of the randomness in the initial pattern may be filtered out, but some remains. Local changes to the initial pattern tend to remain local. The simple structures generated are either stable or periodic with small periods. Here again, some particular initial states (set of measure zero) can lead to unbounded growth. The evolution of these automata is analogous to the evolution of some continuous dynamical systems to limit cycles.
- Class 3 (fig c): Nearly all initial patterns evolve in a pseudo-random, aperiodic or chaotic manner. Any stable structures that appear are quickly destroyed by the surrounding noise. Local changes to the initial pattern tend to spread indefinitely. Small changes in the initial conditions almost always lead to increasingly large changes in the later stages. The evolution of these automata is analogous to the evolution of some continuous dynamical systems to strange attractors.

6.3.3 Classes of Automata III

Wolfram classification

- Class 4 (fig d): Nearly all initial patterns evolve into structures that interact in complex and interesting ways. Eventually a class 4 may become a class 2 but the time necessary to reach that point is very large. An example is given by the rule 110.

The behavior of such cellular automata can generally be determined only by explicit simulation of their time evolution.

6.3.4 Traffic Models I

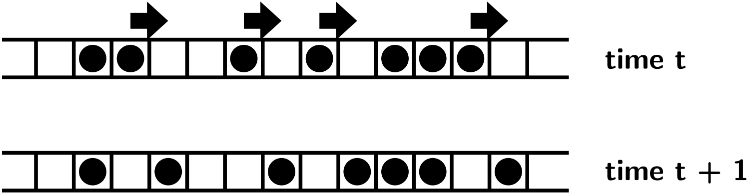
- traffic model for a single lane car motion
- all cars travel in the same direction (say to the right
- the road is represented as a line of cells
- each of them being occupied or not by a vehicle.
- their positions are updated synchronously.
- each car can be at rest or jump to the nearest neighbour site
- the rule is simply that a car moves only if its destination cell is empty
- the drivers do not know whether the car in front will move or will be blocked by another car

The state s_i of each cell at location i is entirely determined by the occupancy of the cell itself and that of its two nearest neighbors ψ_{i-1} and ψ_{i+1}. The motion rule is the following

$$(\psi_{i-1}, \psi_i, \psi_{i+1})_t \rightarrow (\psi_i)_{t+1}$$

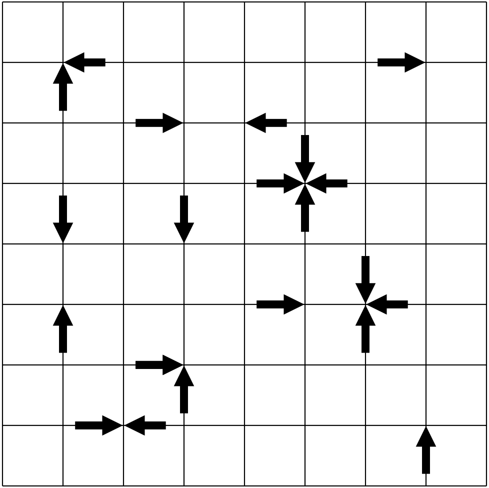
6.3.4 Traffic Models II

	111	110	101	100	011	010	001	000
c = 184 :	1	0	1	1	1	0	0	0



6.3.5 A Simple Model for a Gas of Particles I

HPP model, named after Hardy, Pomeau and De Pazis



6.3.5 A Simple Model for a Gas of Particles II

HPP model, named after Hardy, Pomeau and De Pazis

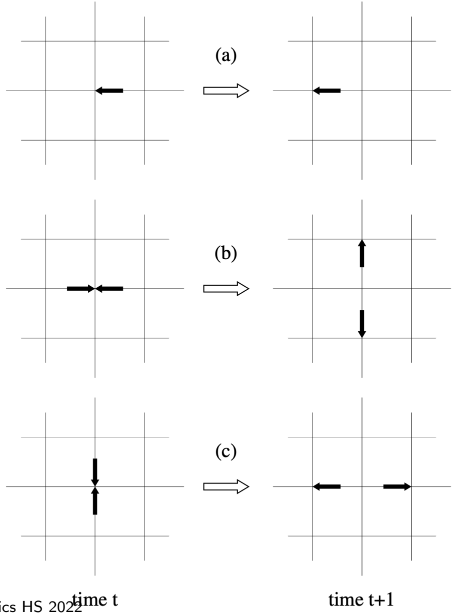
- the HPP lattice gas automaton is defined on a 2d square lattice
- particles can move along the main directions of the lattice
- in our model the number of particles entering a given site with a given direction of motion is 1
- with at most one particle per site and direction, four bits of information at each site are enough to describe the system during its evolution.

If at iteration t site r has the following state $\psi(r, t) = (1011) \Rightarrow$ three particles are entering the site along direction 1,3 and 4, respectively.
The CA rule describing the evolution of $\psi(r, t)$ is often split in two steps:

1. collision
2. propagation (or streaming).

6.3.5 A Simple Model for a Gas of Particles III

HPP model, named after Hardy, Pomeau and De Pazis



S Wolfram: Announcing the Rule 30 Prizes

https://writings.stephenwolfram.com/2019/10/announcing-the-rule-30-prizes/