Cellular Automata and lattice Boltzmann modeling of physical systems

Bastien Chopard

Abstract Cellular automata (CA) and lattice Boltzmann (LB) methods provide a natural modeling 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. The theoretical basis of the CA and LB approaches are introduced and their potential is illustrated for several applications in physics, biophysics, environmental sciences, traffic models and multiscale modeling.

1 Introduction

As we can observe everyday, nature is made of a large number of interacting parts, distributed over space and evolving in time. In many cases we are interested in describing a natural system at a scale which is much larger than its elementary constituents. Then, often, the behavior of these parts can be reduced to a set of rather simple rules, without affecting the behavior of the whole. For instance, when modeling a fluid flow at a macroscopic scale, one does not have to account for the detailed miscroscopic interactions between the atoms of that fluid. Instead we can assume the existence of abstract "fluid elements" interacting in such a way as to conserve mass and momentum.

Cellular Automata (CA) can be thought of as a mathematical abstraction of the physical world, an abstraction in which time is discrete and space is made of little blocks, or cells. These cells are organized as a regular lattice, in such a way as to fully cover the spatial domain of interest. In such an approach, spatio-temporal physical quantities are introduced as numerical values associated with each cell. These quantities are called the *states* of the cells. Formally, the definition of a CA also assumes that these sates can only take a finite number of discrete values. On

Bastien Chopard
University of Geneva, Switzerland, e-mail: Bastien.Chopard@unige.ch

the other hand, lattice Boltzmann (LB) methods are more flexible and allow cells to have real-valued quantities.

CA have been proposed in the late 1940s by J. von Neumann and S. Ulam [6] as an abstraction of a biological system, in order to study the algorithmic mechanisms leading to self-reproduction of living organisms. Since then, the CA approach has been applied to a wide range of problems and is an acknowledge modeling techniques [58, 65, 48, 9, 16, 32, 22, 61, 66]. Moreover it is still a quite active field of research: international conferences (e.g ACRI) and dedicated journals (e.g. J. of Cellular Automata) describe current developments.

LB methods developed later, in the 1990's, as an extension of the concept of lattice gas automata (LGA), are a family of CA designed to describe hydrodynamical processes in terms of a discrete kinetic model. LB is now recognized as an alternative way to simulate on a computer complex fluid flows or other complex systems such as reaction-diffusion and advection-diffusion phenomena, as well as wave propagation in complicated geometries [9, 63, 54, 56]. LB methods are also an important research topic and two international conferences (DSFD and ICMMES) are places to disseminate new results in this field.

A key conceptual ingredient of CA is that they are not meant to be a space-time discretization of the partial differential equation (PDE) representing a given physical process. Instead, CA implement a mesoscopic model of this process, in terms of behavioral rules mimicking the physical interactions and translating them into a fully discrete universe. Concretely, CA evolution rules are transition functions which change synchronously the state of each cell according to its value and those of the adjacent cells.

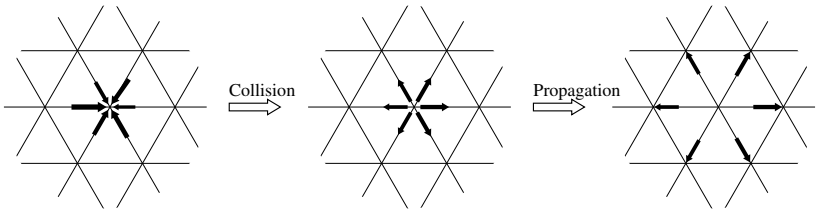
In other words, CA are based on an idealized, virtual microscopic version of the real world. Statistical physics teaches us that the macroscopic behavior of many systems depends very little on the details of the microsopic interactions between the elementary constituents. This suggests that, in view of modeling efficiency, one can consider a new, fictitious microscopic universe whose numerical implementation is easy and fast, as long as this fictitious system has the same macroscopic behavior as the real one. The recipe to achieve this goal is to build a model with the right conservation laws and symmetries. These properties are indeed those which are preserved at all scales of description.

The above principles make the design of CA models quite intuitive and natural. Rule-based interactions are often easier to understand and discuss than a PDE, especially for researchers outside mathematics or physics. Since the level of description of a CA model is mesoscopic, the rules are close to the underlying physical interaction and it is rather easy to add new features to a model and to describe systems for which no PDE apply. On the other hand, CA and LB models heavily rely on computer simulations to derive results.

The diagram in Fig. 1 sketches the solution process of a CA-LB approach compare to that of the more traditional PDE approach.

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

phenomenon → PDE→ discretization → computer solution



phenomenon → computer model

Fig. 1 The solution process, from problem to numerical results for either a PDE (top) or a CA (bottom) approach. In this figure we use fluid dynamics to illustrate the point. In an appropriate limit, fluid dynamics can be either described by the Navier-Stokes equations or by a discrete model of interacting particles. This corresponds to two different languages to represent the same physical problem.

2 Definition of a cellular automata

In order to give a definition of a cellular automaton, we first present a simple example. Although it is very basic, the rule we discuss here exhibits a surprisingly rich behavior. It has been proposed initially by Edward Fredkin in the 1970s [3] and is defined on a two-dimensional square lattice.

Each site of the lattice is a cell which is labeled by its position $\mathbf{r} = (i, j)$ where i and j are the row and column indices. A function $\psi(\mathbf{r},t)$ is associated with the lattice to describe the state of each cell $\mathbf{r}$ at iteration t. This quantity ψ can be either 0 or 1.

The cellular automata rule specifies how the states $\psi(\mathbf{r},t+1)$ are to be computed from the states at iteration t. We start from an initial condition at time $t = 0$ with a given configuration of the values $\psi\mathbf{r},t=0)$ on the lattice. The state at time $t = 1$ is obtained as follows

(1) Each site $\mathbf{r}$ computes the sum of the values $\psi(\mathbf{r}',0)$ 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 $t = 2,3,4,....$

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

$$\psi(i,j,t+1) = \psi(i+1,j,t) \oplus \psi(i-1,j,t) \oplus \psi(i,j+1,t) \oplus \psi(i,j-1,t) \qquad (1)$$

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

When this rule is iterated, very nice geometric patterns are observed, as shown in figure 2. This property of producing complex patterns starting from a simple rule is generic of many cellular automata rules. Here, 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.

Based on this example we now give a definition of a cellular automata. Formally a cellular automata is a tuple $(A, \Psi, R, \mathscr{N})$ where

(i) A is a regular lattice of cells covering a portion of a d-dimensional space.
(ii) $\Psi(\mathbf{r},t) = \{\Psi_1(\mathbf{r},t), \Psi_2(\mathbf{r},t), ..., \Psi_m(\mathbf{r},t)\}$ 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, $R = \{R_1, R_2, ..., R_m\}$, which specifies the time evolution of the states $\Psi(\mathbf{r},t)$ in the following way

$$\Psi_j(\mathbf{r}, t+\delta_t) = R_j(\Psi(\mathbf{r},t), \Psi(\mathbf{r}+\mathbf{v}_1,t), \Psi(\mathbf{r}+\mathbf{v}_2,t), ..., \Psi(\mathbf{r}+\mathbf{v}_q,t)) \qquad (2)$$

where $\mathbf{r}+\mathbf{v}_k$ designate the cells belonging to the neighborhood $\mathscr{N}$ of cell $\mathbf{r}$ and δ_t is the duration of one time step.

In the above definition, the rule R is identical for all sites and is applied simultaneously to each of them, leading to a synchronous dynamics. As the number of configurations of the neighborhood is finite, it is common to pre-compute all the values of R in a lookup table. Otherwise, an algebraic expression can be used and evaluated at each iteration, for each cell, as in eq. (1).

It is important to notice that the rule is *homogeneous*, that is it cannot depend explicitly on the cell position $\mathbf{r}$. However, spatial (or even temporal) inhomogeneities can be introduced anyway by having some $\Psi_j(\mathbf{r})$ 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. Finally, memory states can be included by simply copying the current state into a "past state" during the update.

The neighborhood $\mathscr{N}$ of each cell (i.e. the spatial region around each cell used to compute the next state) is usually made of its adjacent cells. It is often restricted to the nearest or next to nearest neighbors, to keep the complexity of the rule reasonable. For a two-dimensional cellular automaton, two neighborhoods are often considered, as illustrated in fig. 3: the *von Neumann neighborhood* which consists of a central cell (the one which is to be updated) and its four geographical neighbors North, West, South and East. The *Moore neighborhood* contains, in addition, the

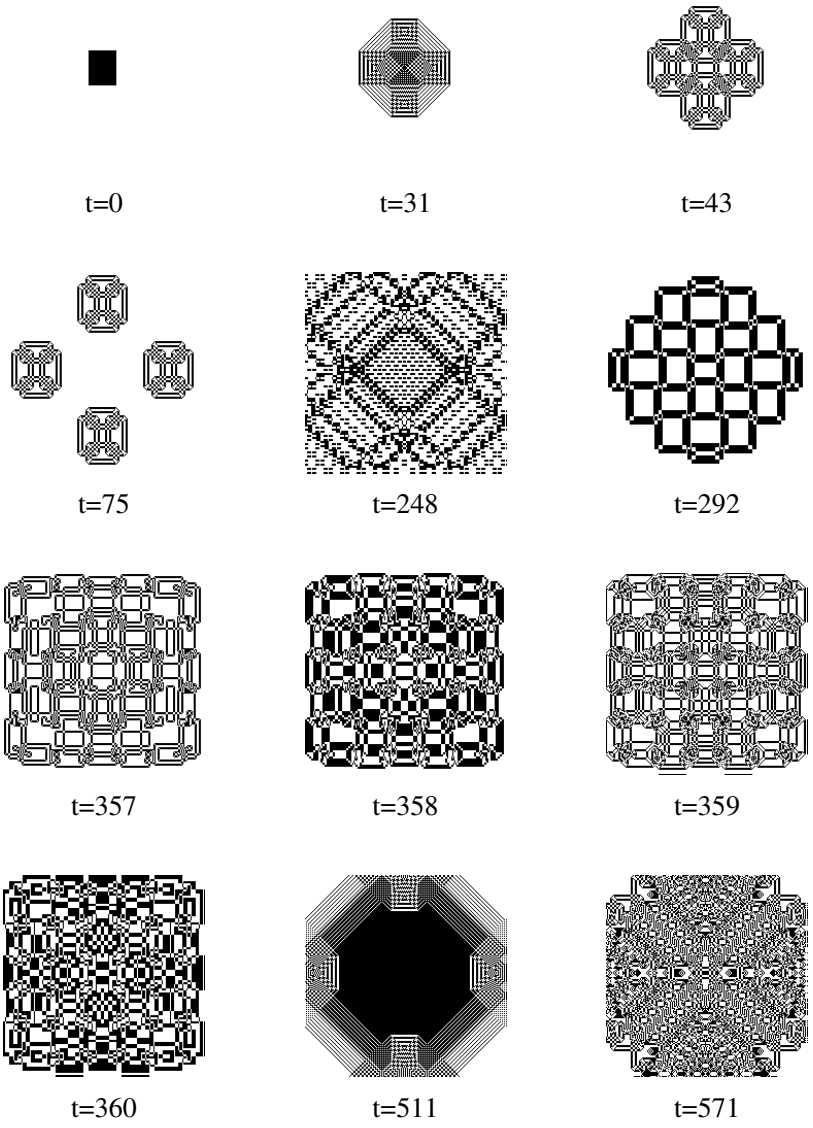


Fig. 2 Several snapshot of the parity rule on a 196×196 periodic lattice. The upper left image correspond to an initial configuration make up of a square of 27×34 cells in state 1.

second nearest neighbor North-East, North-West, South-East and South-East, that is a total of nine cells. See [9] for more details. In practice, when simulating a given cellular automata rule, one cannot deal with an infinite lattice. The system must be finite and have boundaries. Clearly, a site belonging to the lattice boundary does not

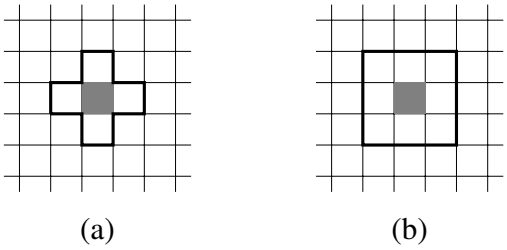


Fig. 3 *(a) Von Neumann and (b) Moore neighborhoods. The shaded region indicates the central cell which is updated according to the state of the cells located within the domain marked with the bold line.*

have the same neighborhood as other internal sites. In order to define the behavior of these sites, a different evolution rule can be considered, which sees the appropriate neighborhood. This means that the information of being or not a boundary cell is coded at the site, using a particular value of Ψ_j, for a chosen j. Depending on this information, a different rule is selected. Following this approach, it is possible to define several types of boundaries, all with a different behavior.

Instead of having a different rule at the limits of the system, another possibility is to extend the neighborhood for the sites at the boundary . For instance, a very common solution is to assume *periodic* (or cyclic) boundary conditions, that is one supposes that the lattice is embedded in a torus-like topology. In the case of a two-dimensional lattice, this means that the left and right sides are connected, and so are the upper and lower sides.

Other possible types of boundary conditions are illustrated in figure 4, for a one-dimensional lattice. We assume that the lattice is augmented by a set of virtual cells beyond its limits. *A fixed* boundary is defined so that the neighborhood is completed with cells having a pre-assigned value. An adiabatic boundary condition (or zero-gradient) is obtained by duplicating the value of the site to the extra virtual cells. A reflecting boundary amounts to copying the value of the other neighbor in the virtual cell.

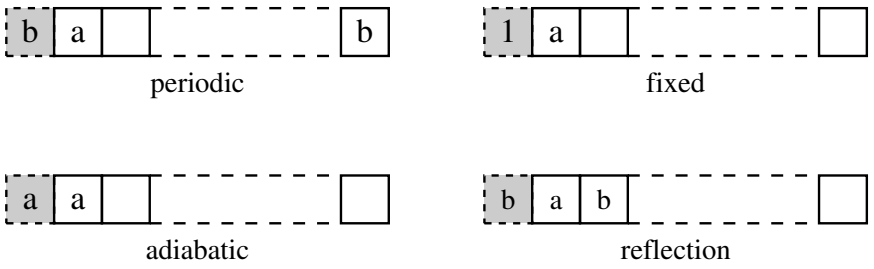


Fig. 4 *Various types of boundary conditions obtained by extending the neighborhood. The shaded block represents a virtual cell which is added at the extremity of the lattice (left extremity, here) to complete the neighborhood.*

According to our definition, a cellular automaton is deterministic. The rule is some well defined function and a given initial configuration will always evolve identically. However, it may be very convenient for some applications to have a certain degree of randomness in the rule. For instance, it may be desirable that a rule selects one outcome among several possible states, with a probability p. Cellular automata whose updating rule is driven by some external probabilities are called *probabilistic* cellular automata. On the other hand, those which strictly comply with the definition given above, are referred to as *deterministic* cellular automata.

3 Cellular Automata and complex systems

3.1 Game of life and Langton ant

Complex systems are now an important domain in sciences. They are systems made of many interacting constituents which often exhibit spatio-temporal patterns and collective behaviors.

A standard and successful methodology in research has been to isolate phenomena from each other and to study them independently. 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. Therefore, the whole 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. In this section, we introduce two CA, the so-called *game of life* and the *Langton's ant* model. Both illustrates interesting aspects of complex systems.

3.1.1 The Game of Life

In 1970, the mathematician John Conway proposed the now famous game of life [21] CA. The motivation was to find a simple rule leading complex behaviors in a system of fictitious one-cell organisms evolving in a fully discrete universe. The game of life rule is defined on a two-dimensional square lattice in which each spatial cell can be either occupied by a living organism (state one) or empty (state zero). The updating rule of the game of life is as follows: an empty cell surrounded by exactly three living cells gets alive; a living cell surrounded by less than two or more than three neighbors dies of isolation or overcrowdness. Here, the surrounding cells corresponds to a Moore neighborhood. composed of the four nearest cells (north, south, east and west), plus the four second nearest neighbors, along the diagonals. It turns out that the game of life automaton has an unexpectedly rich behavior. Complex structures emerge out of a primitive "soup" and evolve so as to develop

some new skills (see fig 5). For instance a particular spatial assembly of cells has

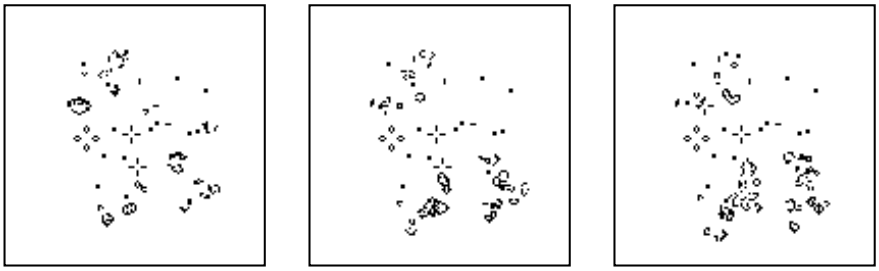


Fig. 5 *The game of life automaton. Black dots represents living cells whereas dead cells are white. The figure shows the evolution of a random initial configuration and the formation of spatial structures, with possibly some emerging functionalities.*

the property to move across the lattice. Such an object, called *glider*, can be seen as higher level organism because it is composed of several simple elementary cells. Its detailed structure is shown in fig. 6. Thus, by assembling in a clever way cells that

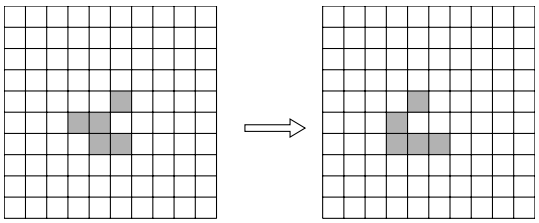


Fig. 6 *The detailed structure of a glider, over two consecutive iterations. A glider is an assembly of cells that has a higher functionality than its constituent, namely the capability to move in space by changing its internal structure in a periodic way.*

are unable to move, it is possible to produce, at a larger scale, a new capability. This is a signature of complex systems. Of course, more complex objects can be build, such a as for instance *glider guns* which are arrangements of cell producing gliders rhythmically.

The game of life is a cellular automata capable of *universal computations*: it is always possible to find an initial configuration of the cellular space reproducing the behavior of any electronic gate and, thus, to mimic any computation process. Although this observation has little practical interest, it is very important from a theoretical point of view since it assesses the ability of CAs to be a non restrictive computational technique. As an illustration of this fact the game of life has been used to compute prime numbers.

3.1.2 Langton's Ant

As we have just discussed, CAs exemplify the fact that a collective behavior can emerge out of the sum of many, simply interacting, components. Even if the basic and local interactions are perfectly known, it is possible that the global behavior obeys new laws that are not obviously extrapolated from the individual properties. The Langton's ant model further illustrate this aspect.

The ant rule is a cellular automata invented by Chris Langton [53] and Greg Turk which models the behavior of a hypothetical animal (ant) having a very simple algorithm of motion. The ant moves on a square lattice whose sites are either white or gray. When the ant enters a white cell, it turns 90 degrees to the left and paints the cell in gray. Similarly, if it enters a gray cell, it paints it in white and turn 90 degree to the right. This is illustrated in Fig. 7

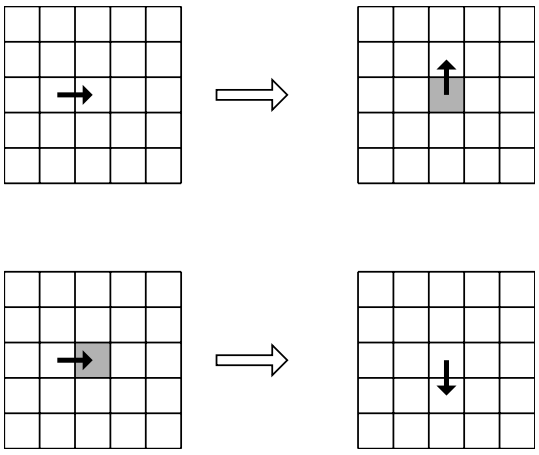


Fig. 7 *The Langton's ant rule.*

It turns out that the motion of this ant exhibits a very complex behavior. Suppose the ant starts in a completely white space. After a series of about 500 steps where it essentially keeps returning to its initial position, it enters a chaotic phase during which its motion is unpredictable. Then, after about 10000 steps of this very irregular motion, the ant suddenly performs a very regular motion which brings it far away from where it started.

Figure 8 illustrates the ant motion. The path the ant creates to escape the chaotic initial region has been called a highway [47]. Although this highway is oriented at 45 degrees with respect to the lattice direction, it is traveled by the ant in a way which makes very much think of a sewing machine: the pattern is a sequence of 104 steps which are repeated indefinitely.

The Langton ant is a good example of a cellular automata whose rule is very simple and yet generates a complex behavior which seems beyond our understanding. Somehow, this fact is typical of the cellular automata approach: although we do

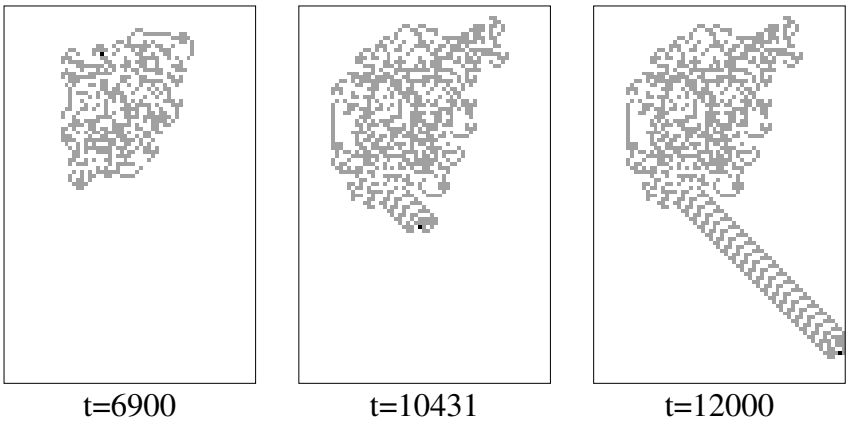


Fig. 8 *The Langton's ant rule. The motion of a single ant starts with a chaotic phase of about 10000 time steps, followed by the formation of a highway. The figure shows the state of each lattice cell (gray or white) and the ant position (marked by the black dot). In the initial condition all cells are white and the ant is located in the middle of the image.*

know everything about the fundamental laws governing a system (because we set up the rules ourselves!), we are often unable to explain its macroscopic behavior.

There is anyway a global property of the ant motion: the ant visits an unbounded region of space, *whatever* the initial space texture is (configuration of gray and white cells).

The proof (due to Bunimovitch and Troubetzkoy) goes as follows: supposed the region the ant visits is bounded. Then, it contains a finite number of cells. Since the number of iteration is infinite, there is a domain of cells that are visited infinitely often. Moreover, due to the rule of motion, a cell is either entered horizontally (we call it a H cell) or vertically (we call it a V cell). Since the ant turns by 90 degrees after each step, a H cell is surrounded by four V cells and conversely. As a consequence, the H and V cells tile the lattice in a fixed checkerboard pattern. Now, we consider the upper rightmost cell of the domain, that is a cell whose right and upper neighbor is not visited. This cell exists if the trajectory is bounded. If this cell is an H cell (and be so for ever), it has to be entered horizontally from left and exited vertically downward and, consequently be gray. However, after the ant has left, the cell is white and there is a contradiction. The same contradiction appears if the cell is a V cell. Therefore, the ant trajectory is not bounded.

Beyond the technical aspect of this proof, it is interesting to realize that its conclusion is based on symmetry properties of the rule. Although we are not able to predict the detailed motion of the ant analytically (the only way is to perform the simulation), we have learned something about the global behavior of the system: the ant goes to infinity. This observation illustrates the fact that, often, global features are related to symmetries more than to details.

Let us now consider the case where several ants coexists on the lattice. The rule defined in Fig. 7 is only valid for a single ant. We can be generalized for situations where up to four ants enter the same site at the same time, from different sides.

To describe mathematically the ant motion, we introduce $n_i(\mathbf{r},t)$, a Boolean variable representing the presence ($n_i = 1$) or the absence ($n_i = 0$) of an ant entering site $\mathbf{r}$ at time t along lattice direction $\mathbf{c}_i$, where $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$ and $\mathbf{c}_4$ stand for direction right, up, left and down, respectively. If the color $\mu(\mathbf{r},t)$ of the site is gray ($\mu = 0$), *all* entering ants turn 90 degrees to the right. On the other hand, if the site is white($\mu = 1$), they all turn 90 degrees to the left. The color of each cell is modified after one or more ants have gone through. Here, we chose to switch $\mu \to 1-\mu$ only when an odd number of ant are present.

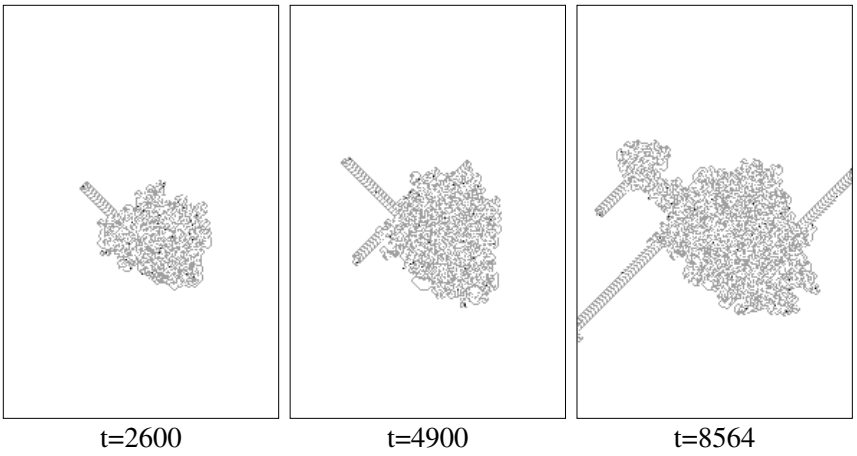


Fig. 9 *Motion of several Langton's ants. Gray and white indicate the colors of the cells at the current time. Ant locations are marked by the black dots. At the initial time, all cells are white and a few ants are randomly distributed in the central region, with random directions of motion. The first highway appears much earlier than when the ant is alone. In addition the highway can be used by other ants to travel much faster. However, the "highway builder" is usually prevented from continuing its construction as soon as it is reached by the following ants. For instance, the highway heading north-west after 4900 steps get destroyed. A new highway emerges later on from the rest, as we see from the snapshot at time* $t = 8564$.

When several ant travel simultaneously on the lattice, cooperative and destructive behaviors are observed. First, the erratic motion of several ants favors the formation of a local arrangement of colors allowing the creation of a highway. One has to wait much less time before the first highway appears. Second, once a highway is being created, other ants may use it to travel very fast (they do not have to follow the complicated pattern of the highway builder). In this way, the term "highway" is very appropriate. Third, a destructive effect occurs as the second ant gets to the highway builder. It breaks the pattern and several situations may be observed. For instance, both ants may enter a new chaotic motion; or the highway is traveled in the

other direction (note that the rule is time reversal invariant) and destroyed. Figure 9 illustrates the multi-ant behavior.

The problem of an unbounded trajectory pauses again with this generalized motion. The assumption of Bunimovitch-Troubetzkoy's proof no longer holds in this case because a cell may be both an H or a V cell. Indeed, two different ants may enter a same cell one vertically and the other horizontally. Actually, the theorem of an unbounded motion is wrong in several cases where two ants are present. Periodic motions may occur when the initial positions are well chosen.

For instance, when the relative location of the second ant with respect to the first one is $(\Delta x, \Delta y) = (2,3)$, the two ants returns to their initial position after 478 iterations of the rule (provided they started in an uniformly white substrate, with the same direction of motion). A very complicated periodic behavior is observed when $(\Delta x, \Delta y) = (1,24)$: the two ant start a chaotic-like motion for several thousands of steps. Then, one ant builds a highway and escape from the central region. After a while, the second ant finds the entrance of the highway and rapidly catches the first one. After the two ants meet, they start undoing their previous paths and return to their original position. This complete cycle takes about 30000 iterations.

More generally, it is found empirically that, when $\Delta x + \Delta y$ is odd and the ants enter their site with the same initial direction , the two-ant motion is likely to be periodic. However, this is not a rule and the configuration $(\Delta x, \Delta y) = (1,0)$ yields an unbounded motion, a diamond pattern of increasing diameter which is traveled in the same direction by the two ants.

It turns out that the periodic behavior of a two-ant configuration is not so surprising. The rule we defined is reversible in time, provided that there is never more than one ant at the same site. Time reversal symmetry means that if the direction of motion of all ants are reversed, they will move backward through their own sequence of steps, with an opposite direction of motion. Therefore, if at some point of their motion the two ants cross each other (on a lattice link, not on a site), the first ant will go through the past of the second one, and vice versa. They will return to the initial situation (the two ants being exchanged) and build a new pattern, symmetrical to the first one, due to the inversion of the directions of motion. The whole process then cycles for ever. Periodic trajectories are therefore related to the probability that the two ants will, at a some time, cross each other in a suitable way. The conditions for this to happen are fulfilled when the ants sit on a different sublattice (black or white sites on the checkerboard) and exit two adjacent sites against each other. This explain why a periodic motion is likely to occur when $\Delta x + \Delta y$ is odd.

An interesting conclusion is that, again, it is a symmetry of the rule (time reversal invariance) that allows us to draw conclusion on the global behavior. We do not know the details of periodic motions but we know that they are possible.

3.2 Cellular automata as simple dynamical systems

CA can also be seen as simple prototype of dynamical systems. In physics, the time evolution of physical quantities is often governed by nonlinear equations. Due to the nonlinearities, solution of these dynamical systems can be very complex. In particular, the solution of these equation can be strongly sensitive to the initial conditions, leading to what is called a chaotic behavior. Similar complications can occur in discrete dynamical systems. Models based on cellular automata provide an alternative approach to study the behavior of dynamical systems. By virtue of their discrete nature, the numerical studies are free of rounding approximations and thus lead to exact results. Also, exhaustive searches in the space of possible rule can be considered for simple CA.

Crudely speaking, two classes of problem can be posed. First, given a cellular automaton rule, predicts its properties. Second, find a cellular automaton rule that will have some prescribed properties. These two closely related problems are usually difficult to solve as we have seen on simple examples.

The simplest cellular automata rules are one-dimensional ones for which each site has only two possible states and the rule involves only the nearest-neighbors sites. They are easily programmable on a personal computer and offer a nice "toy model" to start the study of cellular automata.

A systematic study of these rules was undertaken by S. Wolfram in 1983 [64, 65]. Each cell at location r has, at a given time, two possible states $s(r) = 0$ or $s(r) = 1$. The state s at time $t+1$ depends only on the triplet $(s(r-1), s(r), s(r+1)$ at time t:

$$s(r,t+1) = \Phi(s(r-1,t), s(r,t), s(r+1,t)) \tag{3}$$

Thus to each possible values of the triplet $(s(r-1), s(r), s(r+1)$, one associates a value $\alpha_k = 0$ or 1 according to the following list:

$$\underbrace{111}_{\alpha_7} \quad \underbrace{110}_{\alpha_6} \quad \underbrace{101}_{\alpha_5} \quad \underbrace{100}_{\alpha_4} \quad \underbrace{011}_{\alpha_3} \quad \underbrace{010}_{\alpha_2} \quad \underbrace{001}_{\alpha_1} \quad \underbrace{000}_{\alpha_0} \tag{4}$$

Each possible cellular automata rule $\mathscr{R}$ is characterized by the values $\alpha_0, ..., \alpha_7$. There are clearly 256 possible choices. Each rule can be identified by an index $\mathscr{N}_{\mathscr{R}}$ computed as follows

$$\mathscr{N}_{\mathscr{R}} = \sum_{i=0}^{7} 2^{(i)} \alpha_i \tag{5}$$

which corresponds to the binary representation $\alpha_7\alpha_6\alpha_5\alpha_4\alpha_3\alpha_2\alpha_1\alpha_0$

Giving a rule and an initial state, one can study the time evolution of the system. Some results can be deduced analytically using algebraic techniques, but most of the conclusions follow from numerical iterations of the rules. One can start from a simple initial state (i.e. only one cell in the state 1) or with a typical random initial state. According to their behavior, the different rules have been grouped in four different classes.

(1) Class 1. These cellular automata evolve after a finite number of time steps from almost all initial states to a unique homogeneous state (all the sites have the same value). The set of exceptional initial configurations which behave differently is of measure zero when the number of cells N goes to infinity. A example is given by the rule 40 (see figure 10(a)). From the point of view of dynamical systems, these automata evolve towards a simple *limit point* in the phase space.

(2) Class 2. A pattern consisting of separated periodic regions is produced from almost all the initial states. The simple structures generated are either stable or periodic with small periods. An example is given by the rule 56 (see figure 10(b)) 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*.

(3) Class 3. These cellular automata evolve from almost all initial states to chaotic, aperiodic patterns. An example is given by the rule 18 (see figure 10(c)). 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*.

(4) Class 4. For these cellular automata, persistent complex structures are formed for a large class of initial states. An example is given by the rule 110 (see figure 10 (d)). The behavior of such cellular automata can generally be determined only by explicit simulation of their time evolution.

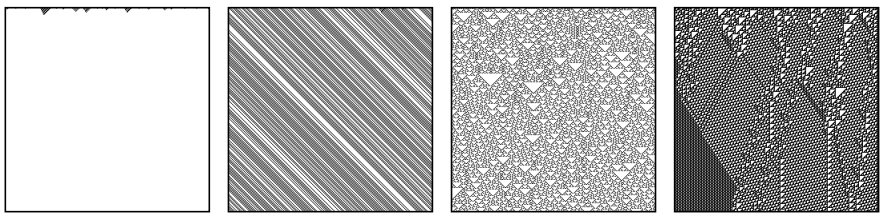


Fig. 10 *Example of the four Wolfram rules with a random initial configuration. Horizontal lines correspond to consecutive iterations. The initial state is the uppermost line. (a) Rule 40 belonging to class 1 reaches very quickly a fixed point (stable configuration). (b) Rule 56 of class 2 reaches a pattern composed of stripes which move from left to right. (c) Rule 18 is in class 3 and exhibits a self-similar pattern. (d) Rule 110 is an example of a class 4 cellular automaton. Its behavior is not predictable and as a consequence, we observe a rupture in the pattern, on the left part.*

The "toy rules" considered by Wolfram, although very simple in construction, are capable of very complex behavior. The validity of this classification is not restricted to the simple rules described above but is somehow generic for more complicated rules. For example, one can consider rules for which each cell can have k different states and involve a neighborhood of raduis ℓ (thus the rule depends on the values of $2\ell+1$ cells). In this case, the number of possible rules is $k^{(k^{(2\ell+1)})}$. Several cases have been studied in the literature and the different rules can be classified in one

of the four above classes. Many of the class 4 cellular automata (starting with $k = 2, \ell = 2$) have the property of computational universality and initial configurations can specify arbitrary algorithmic procedures.

However the above "phenomenological" classification suffers drawbacks, the most serious of which is its non-decidability. See [31] for more details.

3.3 Competition, cooperation, contamination

In this section we briefly describe a few CA rules that are very simple and mimic very natural interactions: competition between adjacent cells, cooperation between them or contamination of neighbors. All these ideas can be implemented within a discrete universe, with cells having only a few possible states. Other models of cooperation-competition are discussed in [20].

3.3.1 Cooperation models:

For instance, in a simple cooperation models, a cell may want to evolve by copying the behavior of the majority of its neighbors. If the possible states are 0 or 1, then clearly an all 0's configuration or an all 1's configuration are both stable. But what happens if the initial configuration contains cells that are 1 with probability p and 0 with probability $1-p$. It is likely that such a system will evolve to one of the two stable configuration. It would be very nice if the all 1's configuration is reached whenever $p > 1/2$ and the all 0's configuration would be the final stage of the case $p < 1/2$. Then, we would have build a system with only *local* calculation that is able to solve a global problem: deciding if the initial density of cells in state 1 is larger or smaller than 1/2. This problem is known as the *density task* and, in general, a simple majority rule is not able to give a reliable answer. See [52] for more details.

From the point of view of modeling physical systems, a slight variant of the majority rule produces interesting patterns. The *twisted majority rule* proposed by G. Vichniac [60] is defined on a two-dimensional lattice where each cell considers its Moore neighborhood. The evolution rule first computes the sum of the cells in state 1. This sum can be any value between 0 and 9. The new state $s(t+1)$ of each cell is then determined from this local sum, according to the following table

$$\begin{array}{ll} \text{sum}(t) & 0\,1\,2\,3\,4\,5\,6\,7\,8\,9 \\ s(t+1) & 0\,0\,0\,0\,1\,0\,1\,1\,1\,1 \end{array} \tag{6}$$

As opposed to the plain majority rule, here, the two middle entries of the table have been swapped. Therefore, when there is a slight majority of 1 around a cell, it turns to 0. Conversely, if there is a slight majority of 0, the cell becomes 1.

Surprisingly enough this rule describes the interface motion between two phases, as illustrated in fig. 11. It is observed that the normal velocity of the interface is pro-

portional to its local curvature, as required by several physical theories. Of course, due to its discrete and local nature, the rule cannot detect the curvature of the interface directly. However, as the the rule is iterated, local information is propagated to the nearest neighbors and the radius of curvature emerges as a collective effect.

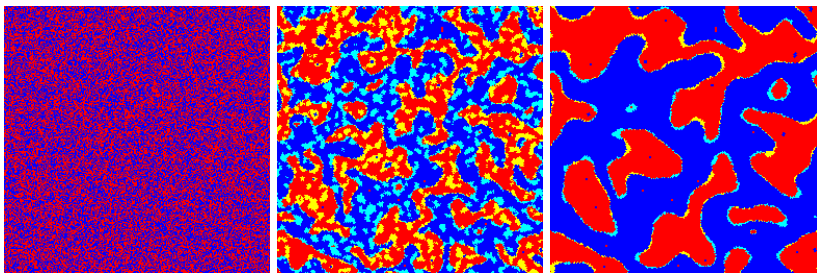


Fig. 11 Evolution of the twisted majority rule. The inherent "surface tension" present in the rule tends to separate the red phases $s = 1$ from the blue phase $s = 0$. The snapshots (a), (b) and (c) correspond to $t = 0$, $t = 72$ and $t = 270$ iterations, respectively. The other colors indicate how "capes" have been eroded and "bays" filled: light blue shows the blue regions that have been eroded during the last few iterations and yellow marks the red regions that have been filled.

3.3.2 Competition models:

In some sense, the twisted majority rule corresponds to a cooperative behavior between the cells. A quite different situation can be obtained if the cells obey a competitive dynamics. For instance we may imagine that the cells compete for some resources at the expense of their nearest neighbors. A winner is a cell of state 1 and a looser a cell of state 0. No two winner cells can be neighbor and any looser cell must have at least one winner neighbor (otherwise nothing would have prevented it to win).

This problem has a direct application in biology, to study cells differentiation. It has been observed in the development of the drosophila that about 25% of the cells forming the embryo are evolving to the state of neuroblast, while the remaining 75% does not. How can we explain this differentiation and the observed fraction since, at the beginning of the process all cells can be assumed equivalent? A possible mechanism [42] is that some competition takes place between the adjacent biological cells. In other word, each cell produces some substance S but the production rate is inhibited by the amount of S already present in the neighboring cells. Differentiation occurs when a cell reaches a level of S above a given threshold.

Following this interpretation we can consider the following CA model of competition. First, we consider a hexagonal lattice, which is a reasonable approximation of

the cell arrangement observed in the drosophila embryo. We assume that 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_{growth} 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_{anihil} 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.

The evolution stops (stationary process) when no $S = 1$ cell feels any more inhibition from its neighbors and when all $S = 0$ cells are inhibited by their neighborhood. Then, with our biological interpretation, cells with $S = 1$ are those which will differentiate.

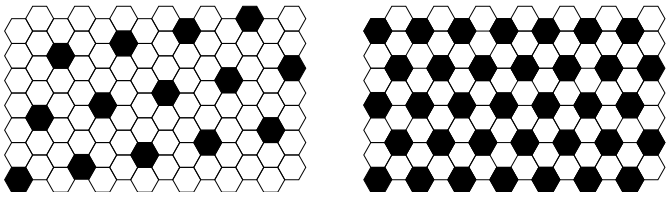


Fig. 12 The hexagonal lattice used for the competition-inhibition CA rule. Black cells are cells of state 1 (winners) and white cells are cells of state 0 (loosers). The two possible final states with a fully regular structure are illustrated with density 1/3 and 1/7 of winner, respectively.

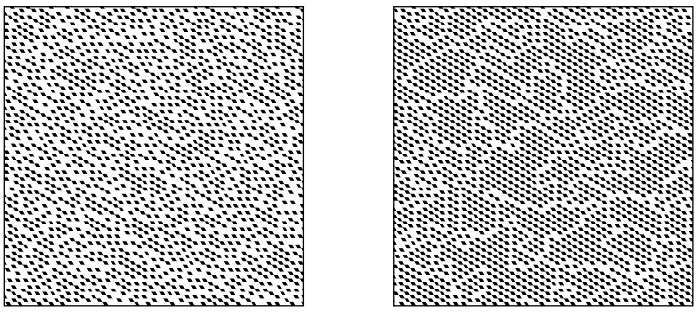
What is the expected fraction of these $S = 1$ cells in the final configuration? Clearly, from figure 12, the maximum value is 1/3. According to the inhibition condition we imposed, this is the close-packed situation on the hexagonal lattice. On the other hand, the minimal value is 1/7, corresponding to a situation where the lattice is partitioned in blocks with one active cell surrounded by 6 inhibited cells. In practice we do not expect any of these two limits to occur spontaneously after the automaton evolution. On the contrary, we observe clusters of close-packed active cells surrounded by defects, i.e. regions of low density of active cells.

As illustrated in Fig. 13, CA simulations show indeed that the final fraction s of active cells is a mix of the two limiting situations of figure 12

$$.23 \leq s \leq .24$$

almost irrespectively of the values chosen for p_{anihil} and p_{growth}.

This is exactly the value expected from the biological observations made on the drosophila embryo. Thus, cell differentiation can be explained by a geometrical competition without having to specify the inhibitory couplings between adjacent cell and the production rate (i.e. the values of p_{anihil} and p_{growth}): the result is quite robust against any possible choices.



(a) (b)

Fig. 13 Final configuration of the competition CA model. (a) A typical situation with about 23% of active cells, obtained with almost any value of p_{anihil} and p_{growth}. (b) Configuration obtained with $p_{anihil} = 1$ and $p_{growth} = .8$ and yielding a fraction of 28% of active cells; one clearly sees the close-packed regions and the defects.

3.3.3 Contamination models:

Finally, after cooperation and competition dynamics, we can also consider a contamination process. To make the model more interesting we consider cells with at least three possible states. Theses states are: the resting state, the excited state and the refractory state.

The resting state is a stable state of the system. But a resting state can respond to a local perturbation and become excited. Then, the excited state evolves to a refractory state where it no longer influences its neighbors and, finally, returns to the resting state.

The Greenberg–Hasting model is an example of a cellular automata model with a contamination mechanism. It is also called a model for an *excitable media* in the context of reactive systems and chemical waves.

The Greenberg–Hasting model can be defined as follows: the state $\psi(\mathbf{r},t)$ of site $\mathbf{r}$ at time t takes its value in the set $\{0,1,2,...,n-1\}$. The state $\psi = 0$ is the resting state. The states $\psi = 1,...,n/2$ (n is assumed to be even) correspond to excited states. The rest, $\psi = n/2+1,...,n-1$ are the refractory states. The cellular automata evolution rule is the following:

1. If $\psi(\mathbf{r},t)$ is excited or refractory, then $\psi(\mathbf{r},t+1) = \psi(\mathbf{r},t)+1 \bmod n$.
2. If $\psi(\mathbf{r},t) = 0$ (resting state) it remains so, unless there are at least k excited sites in the Moore neighborhood of site $\mathbf{r}$. In this case $\psi(\mathbf{r},t+1) = 1$.

The n states play the role of a clock: an excited state evolves through the sequence of all possible states until it returns to 0, which corresponds to a stable situation.

The behavior of this rule is quite sensitive to the value of n and the excitation threshold k. Figure 14 shows the evolution of this CA for a given set of the parameters n and k. The simulation is started with a uniform configuration of resting states, perturbed by some excited sites randomly distributed over the system. Note that if

the concentration of perturbation is low enough, excitation dies out rapidly and the system returns to the rest state. Increasing the number of perturbed states leads to the formation of traveling waves and self-sustained oscillations may appear in the form of ring or spiral waves.

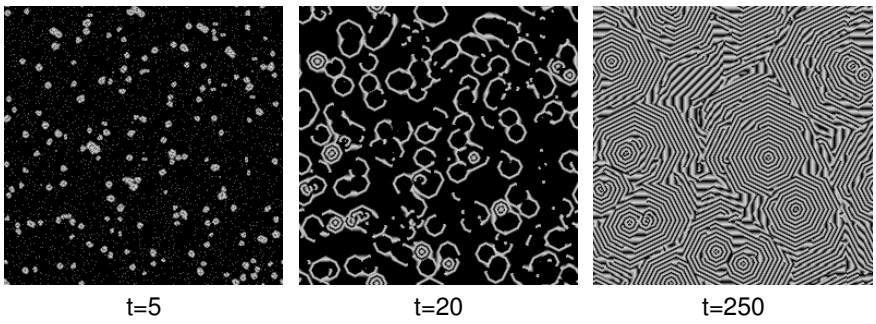


Fig. 14 Excitable medium: evolution of a configuration with 5% of excited states $\phi = 1$, and 95% of resting states (black), for $n = 8$ and $k = 3$.

The Greenberg–Hasting model has some similarity with the "tube-worms" rule proposed by Toffoli and Margolus [58]. This rule is intended to model the Belousov–Zhabotinsky reaction and is as follows. The state of each site is either 0 (refractory) or 1 (excited) and a local timer (whose value is 3, 2, 1 or 0) controls the refractory period. Each iteration of the rule can be expressed by the following sequence of operations: (i) where the timer is zero, the state is excited; (ii) the timer is decreased by 1 unless it is 0; (iii) a site becomes refractory whenever the timer is equal to 2; (iv) the timer is reset to 3 for the excited sites which have two, or more than four, excited sites in their Moore neighborhood.

Figure 15 shows a simulation of this automaton, starting from a random initial configuration of the timers and the excited states. We observe the formation of spiral pairs of excitations. Note that this rule is very sensitive to small modifications (in particular to the order of operations (i) to (iv)).

Another rule which is also similar to Greenberg-Hasting and Margolus-Toffoli tube-worm models is the so-called forest-fire model. This rule describes the propagation of a fire or, in a different context, may also be used to mimic contagion in case of an epidemic. Here we describe the case of a forest-fire rule. The forest-fire rule is a probabilistic CA defined on a two-dimensional square lattice. Initially, each site is occupied by either a tree, a burning tree or is empty. The state of the system is parallel updated according to the following rule: (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 neighbor becomes a burning tree with probability f (so as to mimic an effect of lightning). Figure 16 illustrates the behavior of this rule, in a two-dimensional situation.

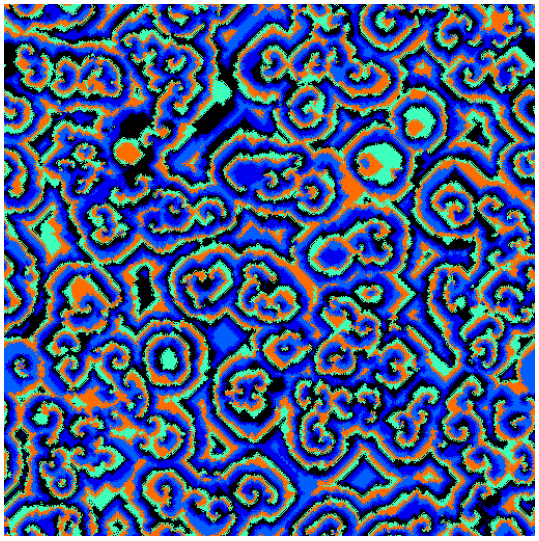


Fig. 15 *The tube-worms rule for an excitable media.*

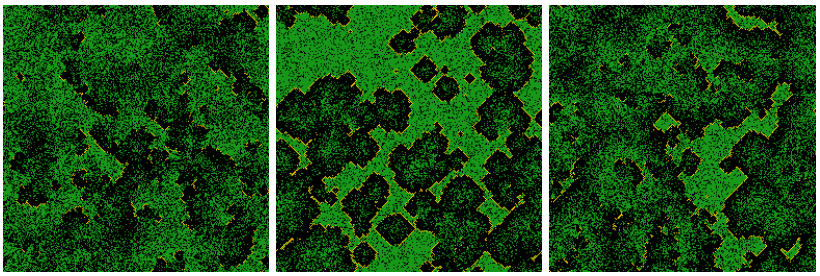


Fig. 16 The forest fire rule: green sites correspond to a grown tree, black pixels represent burned sites and the yellow color indicates a burning tree. The snapshots given here represents three situations after a few hundred iterations. The parameters of the rule are $p = 0.3$ and $f = 6 \times 10^{-5}$.

3.4 Traffic models

Cellular automata models for road traffic have received a great deal of interest during the past few years (see [68, 62, 14, 49, 45] for instance).

CA models for a single lane car motions are quite simple. The road is represented as a line of cells, each of them being occupied or not by a vehicle. All cars travel in the same direction (say to the right). Their positions are updated synchronously. During the motion, each car can be at rest or jump to the nearest neighbor site, along the direction of motion. The rule is simply that a car moves only if its destination cell is empty. This means that the drivers do not know whether the car in front will move or will be blocked by another car. Therefore, 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 s_{i-1} and s_{i+1}. The motion rule can be summarized by the following table, where all eight possible configurations $(s_{i-1}s_is_{i+1})_t \rightarrow (s_i)_{t+1}$ are given

$$\underbrace{(111)}_{1}\ \underbrace{(110)}_{0}\ \underbrace{(101)}_{1}\ \underbrace{(100)}_{1}\ \underbrace{(011)}_{1}\ \underbrace{(010)}_{0}\ \underbrace{(001)}_{0}\ \underbrace{(000)}_{0} \tag{7}$$

This cellular automaton rule turns out to be Wolfram rule 184 [64, 68]. It is illustrated in fig. 17

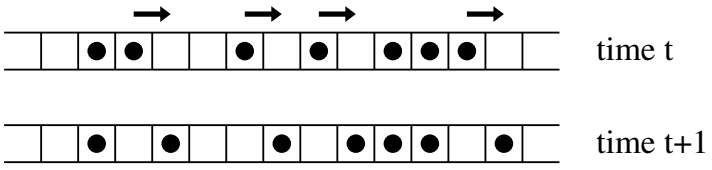


Fig. 17 *Illustration of the basic traffic rule: car with a free cell in front can move. The other ones stay at rest.*

This simple dynamics captures an interesting feature of real car motion: traffic congestion. Suppose we have a low car density ρ in the system, for instance something like

$$\ldots 00100000100100000010\ldots \tag{8}$$

This is a *free* traffic regime in which all the cars are able to move. The average velocity $< v >$ defined as the number of motions divided by the number of cars is then

$$< v_{free} >= 1 \tag{9}$$

On the other hand, in a high density configuration such as

$$\ldots 110101110101101110\ldots \tag{10}$$

only 6 cars over 12 will move and $< v >= 1/2$. This is a partially jammed regime.

In this case, since a car needs a hole to move to, we expect that the number of moving cars simply equals the number of empty cells [68]. Thus, the number of motions is $L(1-\rho)$, where L is the number of cells. Since the total number of car is ρL, the average velocity in the jammed phase is

$$< v_{jam} >= \frac{1-\rho}{\rho} \tag{11}$$

From the above relations we can compute the so-called fundamental flow diagram, i.e. the relation between the flow of cars $\rho < v >$ as a function of the car density ρ: for $\rho \leq 1/2$, we use the free regime expression and $\rho < v >= \rho$. For densities

$\rho > 1/2$, we use the jammed expression and $\rho < v >= 1-\rho$. The resulting diagram is shown in figure 18. As in real traffic, we observe that the flow of car reaches a maximum value before decreasing.

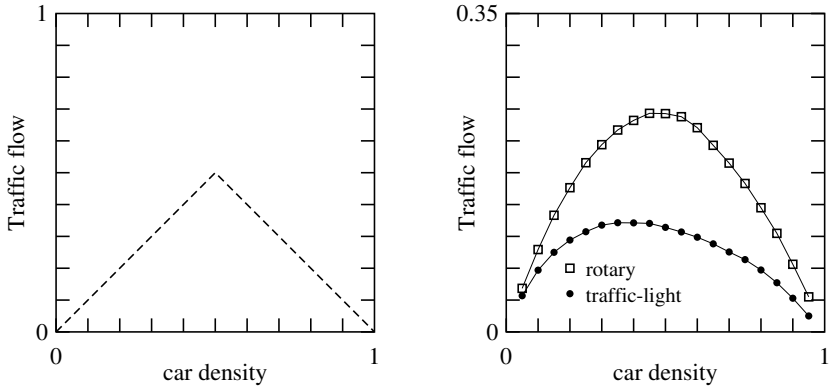


Fig. 18 Traffic flow diagram. Left: for the simple CA traffic rule. Right: for a urban CA traffic model, for a configuration of streets as shown in Fig. 19.

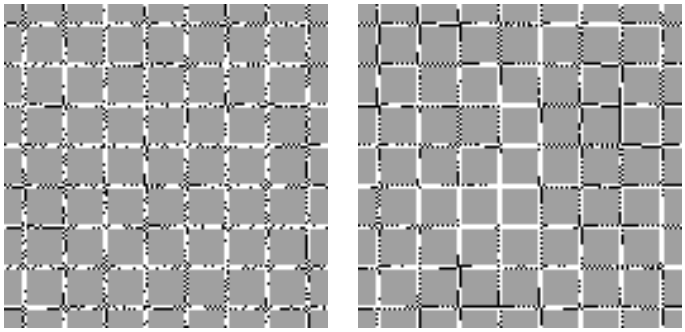
A richer version of the above CA traffic model is due to Nagel and Schreckenberg [46]. The cars may have several possible velocities $u = 0, 1, 2, ..., u_{max}$. Let u_i be the velocity of car i and d_i the distance, along the road, separating cars i and $i+1$. The updating rule is:

- The cars accelerate when possible: $u_i \rightarrow u'_i = u_i + 1$, if $u_i < u_{max}$.
- The cars slow down when required: $u'_i \rightarrow u''_i = d_i - 1$, if $u'_i \geq d_i$.
- The cars have a random behavior: $u''_i \rightarrow u'''_i = u''_i - 1$, with probability p_i if $u''_i > 0$.
- Finally the cars move u'''_i sites ahead.

This rule captures some important behaviors of real traffic on a highway: velocity fluctuations due to a non-deterministic behavior of the drivers, and "stop-and-go" waves observed in high density traffic regime.

Note that a street network can also be described using a CA. A possible approach is to couple several 1D CA model at each road intersection using a roundabout [14, 10]. This is illustrated in Fig. 19 for a Manhattan-like configuration of streets.

We refer the reader to recent literature for the new developments of this topic. See for instance [35, 36].



(a) (b)

Fig. 19 *Traffic configuration after 600 iterations, for a car density of 30%. Situation (a) corresponds to a situation where junctions are modeled as roundabouts, whereas image (b) mimics the presence of traffic lights. In the second case, car queues are more likely to form and the global mobility is less than in the first case, as shown in the right part of Fig. 18.*

4 A simple model for a gas of particles

The HPP rule is a simple example of an important class of cellular automata models: lattice gas automata (LGA). The basic ingredient of such models are point particles that move on a lattice, according to appropriate rules so as to mimic a fully discrete "molecular dynamics."

This model is mostly interesting for pedagogical reasons as it illustrates many important features of LGA and lattice Boltzmann (LB) models in a simple way. However, HPP is of little practical interest because its physical behavior has many flaws (see for instance Fig. 22) that are cured in more sophisticated models, such as the famous FHP model [19] which has been shown to reproduce the Navier-Stokes equations.

The HPP lattice gas automata is defined on a two-dimensional square lattice. Particles can move along the main directions of the lattice, as shown in figure 20. The model limits to 1 the number of particles entering a given site with a given direction of motion. This is the exclusion principle which is common in most LGA (LGA models without exclusion principle are called multiparticle models [9]).

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. For instance, if at iteration t site $\mathbf{r}$ has the following state $s(\mathbf{r},t) = (1011)$, it means that three particles are entering the site along direction 1,3 and 4, respectively.

The cellular automata rule describing the evolution of $s(\mathbf{r},t)$ is often split in two steps: collision and propagation (or streaming. The collision phase specifies how the particles entering the same site will interact and change their trajectories. During the propagation phase, the particles actually move to the nearest neighbor site they are traveling to. This decomposition into two phases is a quite convenient way to partition the space so that the collision rule is purely local.

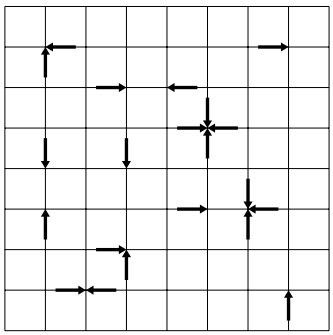


Fig. 20 Example of a configuration of HPP particles

Figure 21 illustrates the HPP rules. According to our Boolean representation of the particles at each site, the collision rules for the two-particle head on collisions are expressed as

$$(1010) \rightarrow (0101) \qquad (0101) \rightarrow (1010) \tag{12}$$

All other configurations are unchanged by the collision process.

After the collision, the propagation phase moves information to the nearest neighbors: the first bit of the state variable is shifted to the east neighbor cell, the second bit to the north and so on. This gives the new state of the system, at time $t+1$. Remember that both collision and propagation are applied simultaneously to all lattice site.

The aim of the HPP rule is to reproduce some aspect of the real interactions between particles, namely that momentum and particle number are conserved during a collision. From figure 21, it is easy checked that these properties are obeyed: a pair of zero momentum particles along a given direction is transformed into another pair of zero momentum along the perpendicular axis.

It is easy to express the HPP model in a mathematical form. For this purpose, the so-called occupation number $n_i(\mathbf{r},t)$ are introduced for each lattice site $\mathbf{r}$ and each time step t. The index i labels the lattice directions (or the possible velocities of the particles). In the HPP model, the lattice has four directions (north, west, south and east) and i runs from 1 to 4.

By definition and due to the exclusion principle, the n_i's are Boolean variables

$$n_i(\mathbf{r},t) = \begin{cases} 1 \text{ if a particle is entering site } \mathbf{r} \text{ at time } t \text{ along lattice direction } i \\ 0 \text{ otherwise} \end{cases}$$

From this definition it is clear that, for HPP, the n_i's are simply the components of the state s introduced above

$$s = (n_1, n_2, n_3, n_4)$$

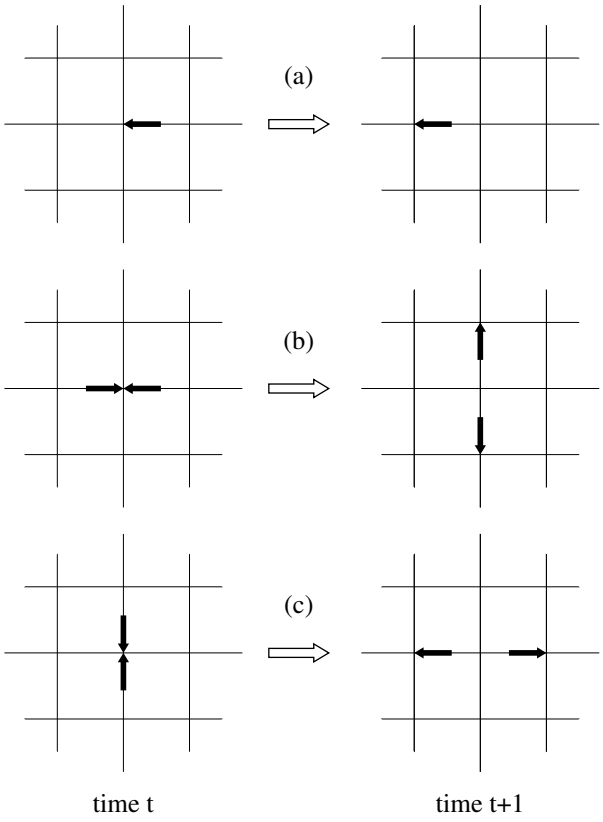


Fig. 21 The HPP rule: (a) a single particle has a ballistic motion until it experiences a collision; (b) and (c) the two non-trivial collisions of the HPP model: two particles experiencing a head on collision are deflected in the perpendicular direction. In the other situations, the motion is ballistic, that is the particles are transparent to each other when they cross the same site.

In a LGA model, the microdynamics can be naturally expressed in terms of the occupation numbers n_i as

$$n_i(\mathbf{r}+\mathbf{v}_i\delta_t, t+\delta_t) = n_i(\mathbf{r},t) + \Omega_i(n(\mathbf{r},t)) \tag{13}$$

where $\mathbf{v}_i$ is a vector denoting the speed of the particle in the ith lattice direction and δ_t is the duration of the time step. The function Ω is called the collision term and it describes the interaction of the particles which meet at the same time and same location.

Note that another way to express eq. (13) is through the so-called collision and propagation operators C and P

$$n(t+\delta_t) = PCn(t) \tag{14}$$

where $n(t)$ describe the set of values $n_i(\mathbf{r},t)$ for all i and $\mathbf{r}$. The quantities C and P act over the entire lattice. They are defined as

$$(Pn)_i(\mathbf{r}) = n_i(\mathbf{r}-\mathbf{v}_i\delta_t) \qquad (Cn)_i(\mathbf{r}) = n_i(\mathbf{r})+\Omega_i$$

More specifically, for the HPP model, it can be shown [9] that the collision and propagation phase can be expressed as

$$n_i(\mathbf{r}+\mathbf{v}_i\delta_t,t+\delta_t) = n_i - n_i n_{i+2}(1-n_{i+1})(1-n_{i+3}) + n_{i+1}n_{i+3}(1-n_i)(1-n_{i+2}) \tag{15}$$

In this equation, the values $i+m$ are wrap onto the values 1 to 4 and the right-hand term is computed at position $\mathbf{r}$ and time t. From this relation, it is easy to show that, for any values of n_i,

$$\sum_{i=1}^{4} n_i(\mathbf{r}+\mathbf{v}_i\delta_t,t+\delta_t) = \sum_{i=1}^{4} n_i(\mathbf{r},t) \tag{16}$$

which expresses the conservation of the number of particle during the collision and the propagation. Similarly, it can be shown ($\mathbf{v}_1 = -\mathbf{v}_3$ and $\mathbf{v}_2 = -\mathbf{v}_4$) that

$$\sum_{i=1}^{4} n_i(\mathbf{r}+\mathbf{v}_i\delta_t,t+\delta_t)\mathbf{v}_i = \sum_{i=1}^{4} n_i(\mathbf{r},t)\mathbf{v}_i \tag{17}$$

which reflects that momentum is conserved.

The behavior of the HPP model is illustrated in fig. 22. From this simulation it is clear that some spatially anisotropic behavior builds up during the time evolution of the rule. A square lattice is actually too poor to represent correctly a fluid system. The FHP model [19, 9], in essence similar to the HPP one, is based of a hexagonal lattice and also includes three-particle collision rules.

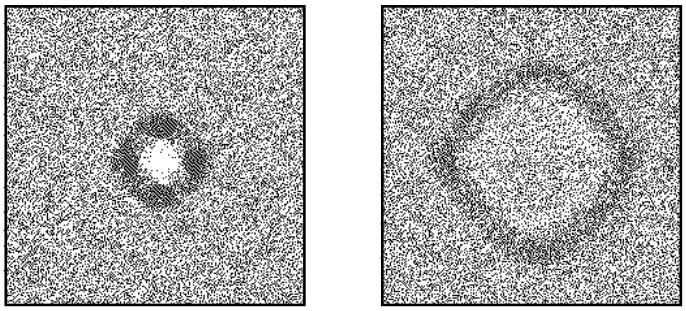


Fig. 22 Time evolution of a HPP gas with a density wave. (a) The initial state is a homogeneous gas with a higher density of particles in the middle region (dark area) (b) After several iterations, the initial perturbation propagates as a wave across the system. As can be observed, there is a clear lack of isotropy is this propagation.

5 Lattice Boltzmann models

Historically Lattice Boltzmann (LB) developed as an extension of the CA-fluids described in the previous section. Another approach is to derive LB models from a discretization of the classical continuous Boltzmann equation [26, 51].

Here we stick to the historical approach as it better illustrates the close relation between the numerical scheme and the underlying discrete physical model of interacting particles. The main conceptual difference between LGA and LB models is that in the latter the cell state is no longer Boolean variables n_i but a real-valued quantity f_i for each lattice directions i. Instead of describing the presence or absence of a particle, the interpretation of f_i is the density distribution function of particles traveling in lattice directions i.

From a practical point of view, the advantages of suppressing the Boolean constraint are several: less statistical noise, more numerical accuracy and, importantly, more flexibility to choose the lattice topology, the collision operator and boundary conditions. Thus, for many applications, the LB approach is preferred to the LGA one. On the other hand, LB models do not integrate local fluctuations that are naturally present in a LGA and that can have relevant physical effects [9].

Several textbooks exist which describe in great details the LB approach [7, 54, 9, 63, 56]. The method has been used extensively in the literature to simulate complex flows and other physical processes [9]. For hydrodynamics, the LB method is now recognized as a serious competitor to the more traditional approaches based on the computer solution of the Navier-Stokes partial differential equations. Among the advantages of the LB method over more traditional numerical schemes, we can mention its simplicity, its flexibility to describe complex flows, its local nature (no need to solve a Poisson equation). Another feature of the LB method is its extended range of validity when the Knudsen number is not negligible (e.g. in microflows) [2].

5.1 General principles

5.1.1 Definitions

The key quantities to define a LB model are the density distributions $f_i(\mathbf{r},t)$ and the "molecular velocities" $\mathbf{v}_i$, for $i = 0 \dots z$, where z is the lattice coordination number of the chosen lattice topology and $z+1$ is the number of discrete velocities. The quantity f_i then denotes the number of particles entering lattice site $\mathbf{r}$ at time t with discrete velocity $\mathbf{v}_i$. Note that $\mathbf{v}_i$ is a vector so that molecular velocities have a norm and a direction. For instance, a common choice of velocities in 2D problem is

$$\begin{aligned} &\mathbf{v}_0 = (0,0) \quad \mathbf{v}_1 = v(1,0) \quad \mathbf{v}_2 = v(1,1) \quad \mathbf{v}_3 = v(0,1) \quad \mathbf{v}_4 = v(-1,1) \\ &\mathbf{v}_5 = v(-1,0) \quad \mathbf{v}_6 = v(-1,-1) \quad \mathbf{v}_7 = v(0,-1) \quad \mathbf{v}_8 = v(1,-1) \end{aligned} \tag{18}$$

In these expressions, v is a velocity norm defined as $v = \delta_r/\delta_t$, with δ_r being the lattice spacing and δ_t the duration of the time step. Both δ_r and δ_t can be expressed in any desired unit system.

From the f_i's and the $\mathbf{v}_i$'s we can define the standard physical quantities such as particle density ρ, particle current $\rho\mathbf{u}$, by taking various moments of the distribution

$$\rho(\mathbf{r},t) = \sum_i f_i(\mathbf{r},t) \qquad \rho(\mathbf{r},t)\mathbf{u}(\mathbf{r},t) = \sum_i f_i(\mathbf{r},t)\mathbf{v}_i \tag{19}$$

The intuitive interpretation of these relations is obvious: the number of particle at point $\mathbf{r}$ and time t is the sum of all particles coming with all velocities; and the total momentum is the sum of momentum carried by each f_i.

In hydrodynamics, it is also important to define higher moments, such as the *momentum tensor*

$$\Pi_{\alpha\beta} = \sum_i f_i(\mathbf{r},t) v_{i\alpha} v_{i\beta}$$

where Greek subscript label spatial coordinates. The tensor $\Pi_{\alpha\beta}$ describes the amount of α-momentum transported along the β-axis.

Following our particle interpretation, we can say that, in a LB model, all particles entering the same site at the same time from different directions (i.e. particles with different molecular velocities $\mathbf{v}_i$) collide. As a consequence a new distribution of particles results. Then, during the next time step δ_t, the particles emerging from the collision move to a new lattice site, according to their new speed. Therefore, the dynamic of a LB model is the alternation of collision and propagation phases.

This is illustrated in figs. 23 and 24, for two different lattice topologies in two dimensions (hexagonal and square lattices). In accordance with figs. 23 and 24, the

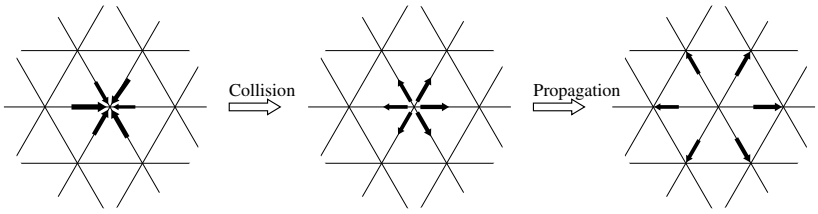


Fig. 23 Illustration of the collision and propagation phases in a LB model defined on a 2D hexagonal lattice, with 6 possible velocities. The arrows represent the particles, their directions correspond to the $\mathbf{v}_i$ and their length is proportional f_i.

LB dynamics can be written as a collision phase

$$f_i^{out}(\mathbf{r},t) = f_i^{in}(\mathbf{r},t) + \Omega_i\left(f^{in}(\mathbf{r},t)\right) \tag{20}$$

and a propagation[1] phase

[1] Propagation is often termed *streaming* in the literature.

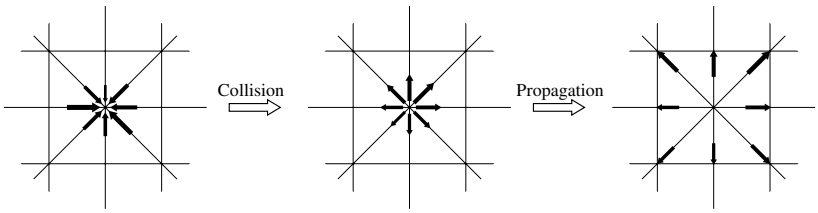


Fig. 24 Illustration of the collision and propagation phases in a LB model defined on a 2D square lattice with 8 velocities.

$$f_i^{in}(\mathbf{r}+\delta_t\mathbf{v}_i,t+\delta_t) = f_i^{out}(\mathbf{r},t) \tag{21}$$

where Ω, the collision term is a model specific function describing the outcome of the particle collision. When subscript i is omitted, Ω denote the all set of Ω_i. Below we will discuss its form. Here we add an upper-script *in* or *out* to the distribution function f_i to distinguish the pre-collision distribution f_i^{in} from the post-collision ones f_i^{out}. When no upper-script is used, we define that $f = f^{in}$. Therefore, by combining eqs (20) and (21), a LB model can also be expressed as

$$f_i(\mathbf{r}+\delta_t\mathbf{v}_i,t+\delta_t) = f_i(\mathbf{r},t) + \Omega_i\left(f(\mathbf{r},t)\right) \tag{22}$$

Conservation laws play an important role in building a LB model. When some physical quantities are known to be conserved in a given phenomena, this conservation must be reflected exactly by the dynamics of the corresponding LB equation. For instance, if the number of particle ρ is conserved in the collision process, we must have

$$\sum_{i=0}^{z} f_i^{out}(\mathbf{r},t) = \sum_{i=0}^{z} f_i^{in}(\mathbf{r},t)$$

for all $\mathbf{r}$ and all t.

From eq. (22), this means that the collision term must obey

$$\sum_{i=0}^{z} \Omega_i = 0 \tag{23}$$

Similarly, if momentum is also conserved (as in a fluid), we must have

$$\sum_{i=0}^{z} \mathbf{v}_i f_i^{out}(\mathbf{r},t) = \sum_{i=0}^{z} \mathbf{v}_i f_i^{in}(\mathbf{r},t)$$

and then

$$\sum_{i=0}^{z} \mathbf{v}_i \Omega_i = 0 \tag{24}$$

5.1.2 Lattice properties

As we have seen, propagation moves particles with velocity $\mathbf{v}_i$ from on lattice site to a neighbor one. As a consequence $\mathbf{r}+\delta_t\mathbf{v}_i$ must also be a point of the lattice. Thus there is a tight connection between the spatial lattice and the discrete set of molecular velocities.

In the LB framework, the choice of velocity $\mathbf{v}_i$ (and consequently the corresponding spatial lattice) is commonly labeled as DdQq, where d is the spatial dimension ($d=2$ for a two-dimensional problem) and q is the number of discrete velocities (or quantities).

When q is odd, it is assumed that the model includes a rest speed $\mathbf{v}_0=0$. Then velocities $\mathbf{v}_i$ are labeled from $i=0$ to $q-1$ and $q=z+1$, z being the lattice coordination number. When q is even, the model contains no rest speed and the velocities $\mathbf{v}_i$ are labeled from $i=1$ to $i=q$ and $q=z$. For instance, the D2Q9 lattice corresponds the velocities given in eq. (18) and illustrated in fig. 25. A D2Q8 lattice is the same, but without $\mathbf{v}_0$. To build a proper LB model, the $\mathbf{v}_i$ should be carefully chosen. In

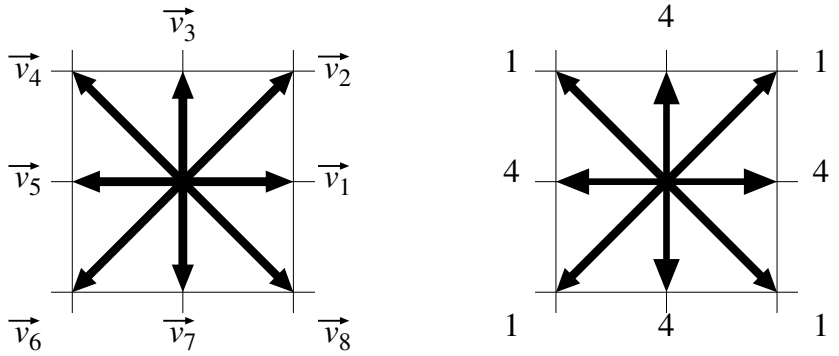


Fig. 25 The D2Q9 lattice with 9 velocities, corresponding to a square lattice, including diagonals. Note that $\mathbf{v}_0=(0,0)$ is not shown on the figure. The right panel shows the ratio of the weights associated with every direction: the diagaonal directions should have a weight four times smaller than the main directions in order to ensure the isotropy of the fourth order tensor.

addition to the fact that $\mathbf{r}+\delta_t\mathbf{v}_i$ must correspond to a lattice site, the molecular velocities must have enough symmetry and isotropy properties. In short, tensors built by summing the velocity components should have enough rotational invariance to represent the physical process under consideration. To achieve this goal, it is often necessary to add weights w_i to velocity vector $\mathbf{v}_i$, as suggested in the right panel of Fig. 25. In practice it is required that

$$\sum_i w_i = 1 \qquad \sum_i w_i \mathbf{v}_i = 0 \qquad \sum_i w_i v_{i\alpha} v_{i\beta} = c_s^2 \delta_{\alpha\beta} \tag{25}$$

where $\delta_{\alpha\beta}$ is the Kronecker symbol and c_s a coefficient to be determined. For a D2Q5 lattice $\mathbf{v}_0 = 0$, $\mathbf{v}_1 = (v,0)$, $\mathbf{v}_2 = (0,v)$, $\mathbf{v}_3 = (-v,0)$ and $\mathbf{v}_4 = (0,-v)$, the above set of conditions is easily fulfilled by choosing $w_1 = w_2 = w_3 = w_4 = (1-w_0)/4 > 0$ and $c_s^2 = v^2(1-w_0)/2$ because $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ are orthogonal 2D vectors.

Conditions (25) are sufficient to model diffusion processes or wave propagation [9]. However, to model non-thermal hydrodynamic flows, they are not enough and must be supplemented by conditions on the third, fourth and five order tensors that can be built out of the $\mathbf{v}_i$'s. These conditions reads

$$\begin{aligned}
\sum_i w_i v_{i\alpha} v_{i\beta} v_{i\gamma} &= 0 \\
\sum_i w_i v_{i\alpha} v_{i\beta} v_{i\gamma} v_{i\delta} &= c_s^4 \left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}\right) \\
\sum_i w_i v_{i\alpha} v_{i\beta} v_{i\gamma} v_{i\delta} v_{i\varepsilon} &= 0
\end{aligned} \tag{26}$$

For thermo-hydrodynamic models, even higher order conditions must be considered, forcing us to add more discrete velocities to the system [51]. We refer the reader to LB textbooks to better understand the origin and meaning of these isotropy conditions. Here we shall just accept them as a requirement on a proper choice of velocity sets. They can be satisfied for a D2Q9 model by taking

$$w_0 = 4/9 \qquad w_1 = w_3 = w_5 = w_7 = 1/9 \qquad w_2 = w_4 = w_6 = w_8 = 1/36$$

for which one gets $c_s^2/v^2 = 1/3$.

Three-dimensional models, such as the well known D3Q19 model, can also be constructed to satisfy eqs. (25) and (26). In the D3Q19 model, the velocity vectors are defined as

$$\begin{aligned}
\mathbf{v}_0 &= 0 \\
\mathbf{v}_1 = v(-1,0,0) \quad \mathbf{v}_2 &= v(0,-1,0) \quad \mathbf{v}_3 = v(0,0,-1) \\
\mathbf{v}_4 = v(-1,-1,0) \quad \mathbf{v}_5 &= v(-1,1,0) \quad \mathbf{v}_6 = v(-1,0,-1) \\
\mathbf{v}_7 = v(-1,0,1) \quad \mathbf{v}_8 &= v(0,-1,-1) \quad \mathbf{v}_9 = v(0,-1,1) \\
\mathbf{v}_{10} = v(1,0,0) \quad \mathbf{v}_{11} &= v(0,1,0) \quad \mathbf{v}_{12} = v(0,0,1) \\
\mathbf{v}_{13} = v(1,1,0) \quad \mathbf{v}_{14} &= v(1,-1,0) \quad \mathbf{v}_{15} = v(1,0,1) \\
\mathbf{v}_{16} = v(1,0,-1) \quad \mathbf{v}_{17} &= v(0,1,1) \quad \mathbf{v}_{15} = v(0,1,-1)
\end{aligned} \tag{27}$$

and the lattice properties are

$$c_s^2/v^2 = 1/3 \qquad w_0 = 1/3 \qquad w_{slow} = 1/18 \qquad w_{fast} = 1/36$$

where w_{slow} concerns the $\mathbf{v}_i$ of norm v and w_{fast} the $\mathbf{v}_i$ of norm $\sqrt{2}v$.

5.2 Lattice BGK models

We now return to the LB equation (22)

$$f_i(\mathbf{r}+\delta_t\mathbf{v}_i,t+\delta_t) = f_i(\mathbf{r},t) + \Omega_i\left(f(\mathbf{r},t)\right) \tag{28}$$

and consider a special family of collision terms: the so-called *single relaxation time* models, also termed lattice BGK models (LBGK) for its correspondence with the BGK form of the continuous Boltzmann equation [4].

Although more sophisticated models exist [17, 40, 8], the LBGK is still the most popular version of a LB model. It reads

$$f_i(\mathbf{r}+\delta_t\mathbf{v}_i,t+\delta_t) = f_i(\mathbf{r},t) + \frac{1}{\tau}\left(f_i^{eq} - f_i\right) \tag{29}$$

where f^{eq} is called the *local equilibrium* distribution; it is a given function which depends on the phenomena that we want to model (note that when we refer to all f_i or all f_i^{eq}, we drop the subscript i). The quantity τ is the so-called *relaxation time*. It is a parameter of the model which is actually related to the transport coefficient of the model: viscosity for a fluid model, diffusion constant in case of a diffusion model.

In eq. (29) it is important to note that the local equilibrium distribution f^{eq} depends on space and time only through the conserved quantities. This is a common assumption of statistical physics. In a hydrodynamic process, where both mass and momentum are conserved, f^{eq} will then be a function of ρ and $\mathbf{u}$.

Thus, in eq. (29), to compute $f_i(\mathbf{r}+\delta_t\mathbf{v}_i,t+\delta_t)$ from the $f_i(\mathbf{r},t)$ one first has to compute $\rho = \sum f_i$ and $\mathbf{u} = (1/\rho)\sum f_i\mathbf{v}_i$ before computing $f_i^{eq}(\rho,\mathbf{u})$. Then, only, f_i can be updated.

It is beyond the scope of this article to show the equivalence between the LB model and the differential equations representing the corresponding physical phenomena. This derivation requires rather heavy mathematical calculations and can be found in several textbooks. See for instance [9, 13, 39] for a derivation based on the so-called multiscale *Chapman-Enskog* formalism. Or, see [34] for a derivation based on the *asymptotic expansion*. Here we will simply give the important results, without demonstration.

5.2.1 LBGK Fluid models

A first central ingredient of LB models is to properly enforce the physical conservation laws in the collision term. Hydrodynamics is characterized by mass and momentum conservation which, in the differential equation language, are expressed by the *continuity* and *Navier-Stokes* equations.

From eq. (23) and (24), conservation laws impose conditions on f_i^{eq} when a LBGK model is considered, namely

$$\sum_i f_i^{eq} = \sum_i f_i = \rho \qquad \sum_i \mathbf{v}_i f_i^{eq} = \sum_i \mathbf{v}_i f_i = \rho \mathbf{u} \tag{30}$$

In addition, in order to recover a hydrodynamic behavior, one imposes that $\Pi^{eq}_{\alpha\beta}$, the second moment of f^{eq}, which is the non-dissipative part of the momentum tensor, has the standard Euler form

$$\Pi^{eq}_{\alpha\beta} = \sum_i f_i^{eq} v_{i\alpha} v_{i\beta} = p\delta_{\alpha\beta} + \rho u_\alpha u_\beta \tag{31}$$

where p is the pressure.

Using eq. (25) and (26) it is easy to show that the following expression for f^{eq} satisfies the conservation laws (30)

$$f_i^{eq} = f_i^{eq}(\rho, \mathbf{u}) = \rho w_i (1 + \frac{v_{i\alpha} u_\alpha}{c_s^2} + \frac{1}{2c_s^4} Q_{i\alpha\beta} u_\alpha u_\beta) \tag{32}$$

where the Q_i's are tensors whose spatial components are

$$Q_{i\alpha\beta} = v_{i\alpha} v_{i\beta} - c_s^2 \delta_{\alpha\beta} \tag{33}$$

Note that in this equation and in what follows, we use Einstein summation convention over repeated Greek indices

$$v_{i\alpha} u_\alpha = \sum_{\alpha=x,y,z} v_{i\alpha} u_\alpha$$

and

$$Q_{i\alpha\beta} u_\alpha u_\beta \equiv \sum_{\alpha,\beta\in\{x,y,z\}} Q_{i\alpha\beta} u_\alpha u_\beta$$

Note that f_i^{eq} can also be interpreted as a discretization of the Maxwell-Boltzmann distribution function of statistical physics.

We can also check that the second moment of eq. (32) gives the correct expression for the Euler momentum tensor (31), provided that the pressure is related to the density ρ through an ideal gas relation

$$p = \rho c_s^2$$

From this expression. we can interpret the lattice parameter c_s as the speed of sound.

The fact that, in a LB model, the pressure is directly obtained from thze density is an important observation. It means that in a LB fluid model, there is no need to solve a (non-local) Poisson equation for the pressure, as is the case when solving Navier-Stokes equations.

Using expression (32) for f^{eq}, the behavior of the LB model (29) can be analyzed mathematically with, for instance a Chapman-Enskog method. Several important results are obtained.

It is found that, to order δ_t^2 and δ_r^2, and for small Mach number ($\mathbf{u} \ll c_s$), the LB dynamics implies that ρ and $\mathbf{u}$ obey the continuity equation

$$\partial_t \rho + \partial_\alpha \rho u_\alpha = 0 \tag{34}$$

and the Navier-Stokes equation

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} \tag{35}$$

with a kinematic viscosity ν depending on the relaxation time τ as

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

The last question we want to address here is how to obtain the expression of f in terms of the hydrodynamic quantities. It is easy to obtain ρ and $\mathbf{u}$ from the f_i, using eq. (19). But the inverse relations, expressing f_i as a function of ρ and $\mathbf{u}$ is more difficult. There are $z+1$ variables f_i and only $1+d$ hydrodynamic quantities in d dimensions. However, in the hydrodynamic limit, it turns out that the derivatives of $\mathbf{u}$ are precisely the missing piece of information.

The first step is to split the density distributions f_i as

$$f_i = f_i^{eq} + f_i^{neq} \qquad \text{assuming } f_i^{neq} \ll f_i^{eq}$$

where f_i^{eq}, by its definition (32) is already a function of ρ and $\mathbf{u}$

$$f_i^{eq} = f_i^{eq}(\rho, \mathbf{u}) = \rho w_i (1 + \frac{\mathbf{v}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} Q_{i\alpha\beta} u_\alpha u_\beta) \tag{36}$$

The Chapman-Enskog expansion then gives

$$f_i^{neq} = -\delta_t \tau \frac{w_i}{c_s^2} \rho Q_{i\alpha\beta} \partial_\alpha u_\beta = -\delta_t \tau \frac{w_i}{c_s^2} \rho Q_{i\alpha\beta} S_{\alpha\beta} \tag{37}$$

where $S_{\alpha\beta} = (1/2)(\partial_\alpha u_\beta + \partial_\beta u_\alpha)$ is the so-called strain rate tensor. As we can see from this relation, the derivatives of $\mathbf{u}$ are part of the LB variables.

By taking the second moment of eq. (37) we obtain $\Pi_{\alpha\beta}^{neq}$, the non-equilibrium part of the momentum tensor. Due to the lattice properties (25) and (26),

$$\Pi_{\alpha\beta}^{neq} = \sum_i v_{i\alpha} v_{i\beta} f_i^{neq} = -2\delta_t \tau c_s^2 \rho S_{\alpha\beta} \tag{38}$$

Thus, the strain rate tensor can be directly obtained from $f^{neq} = f - f^{eq}$, without the need of computing finite differences.

Therefore, with eqs. (19), (36) and (37) we have established relation that allows us to translate the hydrodynamic quantities to LB quantities and vice-versa

$$\begin{pmatrix} \rho \\ \mathbf{u} \\ S_{\alpha\beta} \end{pmatrix} \leftrightarrow (f_i) \tag{39}$$

They are valid in the hydrodynamic regime, with $u \ll c_s$ and for $\delta_t \to 0$ and $\delta_r \to 0$.

In order to illustrate the LBGK method, Fig. 26 shows an example of a LB fluid simulation, the flow past an obstacle.

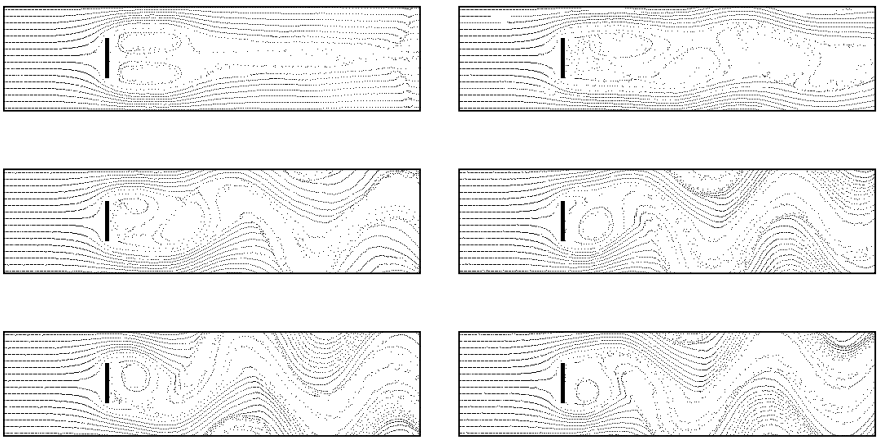


Fig. 26 Several stages (from left to right and top to bottom) of the evolution of a flow past an obstacle, using a D2Q9 LBGK fluid model.

5.3 Diffusion and reaction-diffusion LBGK models

It is actually very easy to devise a LBGK model to describe other physical processes. The basic equation remains the same, namely

$$f_i(\mathbf{r}+\delta_t \mathbf{v}_i, t+\delta_t) = f_i(\mathbf{r},t) + \frac{1}{\tau}\left(f_i^{eq} - f_i\right) \tag{40}$$

What changes is the expression for f^{eq} and also the isotropy requirements of the lattice. For instance, for a diffusion model, eq. (25) is sufficient because diffusion does not involve fourth order isotropy constraints. In 2D, a D2Q4 lattice (having $w_i = 1/4$, $c_s^2/v^2 = 1/2$) is enough to model a diffusion process and, in 3D, a D3Q6 lattice ($w_i = 1/6$, $c_s^2/v^2 = 1/2$) has enough isotropy.

5.3.1 Diffusion:

To build the local equilibrium distribution corresponding to a diffusion model, we notice that only the density $\rho = \sum f_i$ is conserved in the process. Then, it is found that

$$f_i^{eq} = w_i \rho \tag{41}$$

produces the diffusion equation

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

where

$$D = c_s^2 \delta_t \left(\tau - \frac{1}{2} \right)$$

is the diffusion coefficient. Also, it is found that the particle current $\mathbf{j} = -D\nabla\rho$ can be computed from the non-equilibrium part of the distribution function, $f^{neq} = f - f^{eq} = -\tau \delta_t w_i v_{i\alpha} \partial_\alpha \rho$ as

$$\mathbf{j} = \left(1 - \frac{1}{2\tau} \right) \sum_i \mathbf{v}_i f_i^{neq} = \left(1 - \frac{1}{2\tau} \right) \sum_i \mathbf{v}_i f_i$$

Advection-diffusion processes

$$\partial_t \rho + \partial_\alpha \rho u_\alpha = D \nabla^2 \rho$$

where $\mathbf{u}(\mathbf{r},t)$ is a given velocity field, can be modeled by adding an additional term to the local equilibrium (41)

$$f_i^{eq} = w_i \rho (1 + \frac{1}{c_s^2} \mathbf{u} \cdot \mathbf{v}_i) \tag{42}$$

If $\mathbf{u}$ is the solution of a fluid flow, it is appropriate to add a term $w_i \rho (1/(2c_s^4)) Q_{i\alpha\beta} u_\alpha u_\beta$ to eq. (42). See [15, 55, 11, 23, 50] for more details.

5.3.2 Reaction-diffusion:

In order to simulate a reaction-diffusion process

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

where R is any reaction term, the LB diffusion model can be modified as

$$f_i(\mathbf{r} + \delta_t \mathbf{v}_i, t + \delta_t) = f_i(\mathbf{r}, t) + \frac{1}{\tau}(f_i^{eq} - f_i) + \delta_t w_i R(\rho) \tag{43}$$

For instance, the reaction $A + B \rightarrow C$ where C is some inert product can be simulated by two sets of equations (43), one for species A and one for species B. The reaction term is chosen as $R = -k\rho_A \rho_B$. Fig. 27 shows the evolution of the concentrations of both species when they are initially mixed.

Another example of a reaction-diffusion process is show is fig. 28. See [12] for more details.

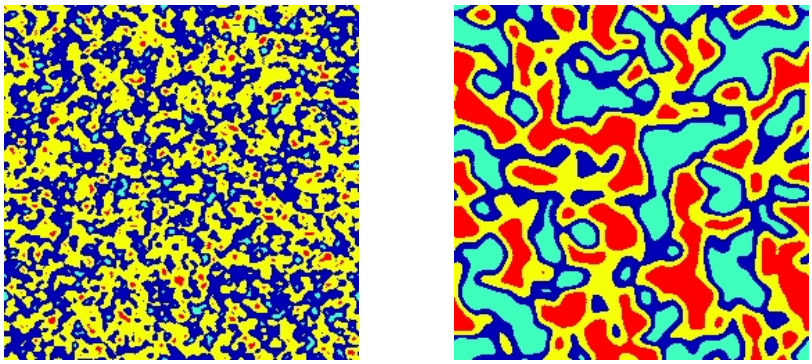


Fig. 27 A reaction-diffusion model simulating the $A+B \rightarrow C$ reaction. The left panel shows an early stage and the right panel a later stage of the reaction process. Red denotes regions very rich in A, yellow indicates regions where A dominates over B. Similarly, green regions are those very rich is B and blue those where B is slightly more abundant than A.

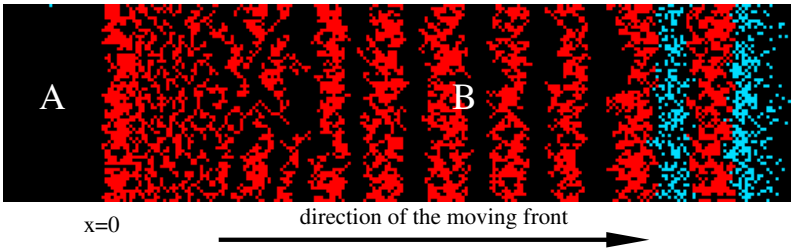


Fig. 28 A reaction-diffusion model simulating the formation of Liesegang bands in a $A+B \rightarrow C$ process, where C can precipitate and forms bands at positions that have interesting geometrical properties.

5.4 Wave propagation:

Finally, *wave propagation* can also be described by a LBGK model [9, 13]. In this case we have to choose $\tau = 1/2$ which ensure the time reversibility of the LB dynamics [9], a required symmetry of the wave equation.

Wave processes have two conserved quantities, $\rho = \sum f_i$ which can be any scalar quantity obeying a wave process, and its current $\mathbf{j} = \sum f_i \mathbf{v}_i$. However, as opposed to flow and diffusion models, the f_i are no longer positive quantities and they oscillate between a minimal negative value and a maximum positive one.

The appropriate form of the local equilibrium is found to be

$$f_i^{eq} = w_i \rho + w_i \frac{\mathbf{j} \cdot \mathbf{v}_i}{c_s^2}$$

$$f_0^{eq} = w_0 \rho \tag{44}$$

In the continuous limit, eq. (40) with eq. (44) yield

$$\begin{aligned} \partial_t \rho + \partial_\beta j_\beta &= 0 \\ \partial_t j_\alpha - c_s^2 \partial_\alpha \rho &= 0 \end{aligned} \tag{45}$$

When combined, these two equations give the wave equation

$$\partial_t^2 \rho - c_s^2 \nabla^2 \rho = 0 \tag{46}$$

The rest population f_0 allows us to adjust the speed of the wave from place to place by having the value of w_0 depend of spatial location $\mathbf{r}$ or time t.

As for the diffusion case, a second order isotropy is enough for the wave equation. Therefore D2Q5 and D3Q7 lattices are appropriate. For other topologies it is important to use the same weight $w_i = w$ for all non-zero velocities. It means that the second order isotropy condition now reads

$$\sum_{i \geq 1} v_{i\alpha} v_{i\beta} = z c_{max}^2 \delta_{\alpha\beta} \tag{47}$$

for some coefficient c_{max} and for z the lattice coordination number. For D2Q5 and D3Q7, it easy to show that $c_{max}^2 / v^2 = 2/z$.

When $w_i = w$, we conclude form the condition

$$1 = \sum_{i \geq 0} w_i = w_0 + zw$$

that

$$w = \frac{1 - w_0}{z} \tag{48}$$

Therefore

$$\sum_{i \geq 1} w_i v_{i\alpha} v_{i\beta} = (1 - w_0) c_{max}^2 \delta_{\alpha\beta} \tag{49}$$

and we obtain

$$c_s^2 = (1 - w_0) c_{max}^2 \tag{50}$$

A consequence of this relation is that we must impose $w_0 < 1$. Note that this way of adjusting c_s is only possible for processes which do not require fourth order isotropy. For a fluid model, there is however a way to adjust the speed of sound [1, 67, 13].

The numerical stability of the LBGK wave model is guaranteed because a quantity, termed *energy*,

$$E = \frac{w}{w_0} f_0^2 + \sum_{i \geq 1} f_i^2 \tag{51}$$

is conserved during the collision step

$$E^{out} = \frac{w}{w_0}\left(f_0^{out}\right)^2 + \sum_{i\geq 1}\left(f_i^{out}\right)^2 = \frac{w}{w_0}\left(f_0^{in}\right)^2 + \sum_{i\geq 1}\left(f_i^{in}\right)^2 = E^{in} \tag{52}$$

The proof of this condition also requires that $w_i = w$ for $i \neq 0$. When $(w/w_0) \geq 0$, the numerical stability of the model is guaranteed because, with E given, the f_i cannot diverge to $\pm\infty$.

With $0 \leq w_0 < 1$ we obtain from (50) that the speed of the wave is such that

$$0 < c_s \leq c_{max}$$

the maximum speed being achieved with $w_0 = 0$ i.e. without a rest population f_0. The refraction index is defined as

$$n = \frac{c_{max}}{c_s}$$

Using eqs. (48) and (50) when then obtain

$$w = \frac{1}{z}\frac{c_s^2}{c_{max}^2} = \frac{1}{n^2 z} \qquad w_0 = 1 - \frac{c_s^2}{c_{max}^2} = \frac{n^2-1}{n^2} \tag{53}$$

With the above value of w_i and the fact that $\tau = 1/2$, the LBGK wave model can also be written as

$$\begin{aligned} f_i^{out} &= \frac{2}{n^2 z}\rho + \frac{2}{z c_{max}^2}\mathbf{v}_i \cdot \mathbf{j} - f_i \\ f_0^{out} &= 2\frac{n^2-1}{n^2}\rho - f_0 \end{aligned} \tag{54}$$

Fig. 29 illustrates this model with a D2Q5 lattice. A plane wave is produced on the left side of the domain and propagates to the right where it penetrates in a lens shaped media with slower propagation speed.

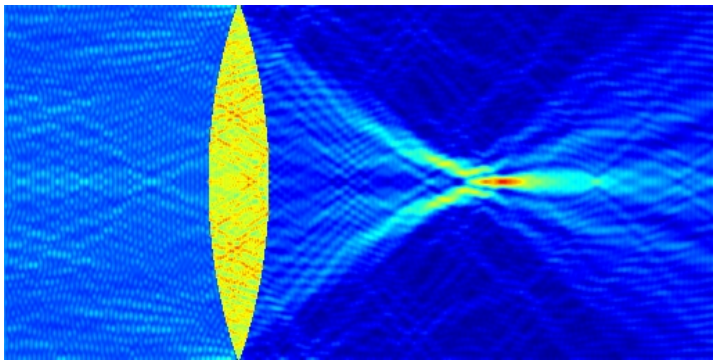


Fig. 29 LB model simulating the propagation of a wave in a lens. The colors represent the energy E.

Note that in [9, 13, 43] the above model is investigated for its capability to model deformable elastic solids. Fig. 30 illustrates an application of the wave model to describe a fracture process.

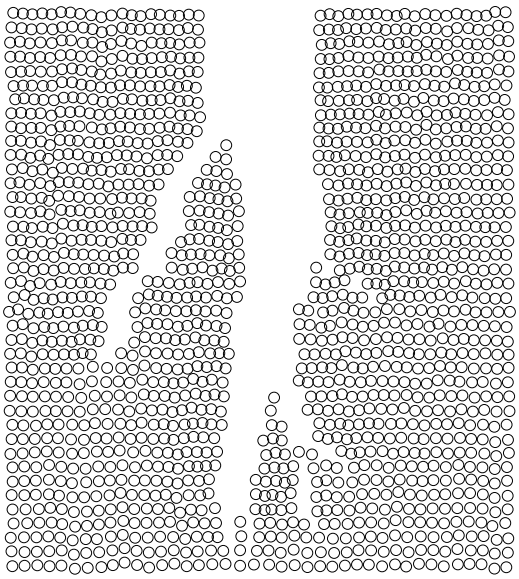


Fig. 30 Fracture process in a LB model for an elastic solid body.

Finally note that the LBGK wave model is also known in the literature as the Transmission Line Matrix model [28] and has been derived in several different contexts [59].

5.5 Boundary conditions

Boundary conditions are an important aspect of a LB model. It is not an easy question to properly specify the values of the distributions f_i at the limit of the computational domain. Clearly, to apply the collision phase, all f_i must be defined. But at a boundary cell, the propagation phase does not provide any information from outside the domain. This is illustrated in fig. 31. These unknown distributions must be specified according to the desired behavior of the system at the boundary.

Following this procedure, the time evolution of a LB model can then be represented by the following loop in a computer program

```
for t=0 to tmax
  boundary
  observation
```

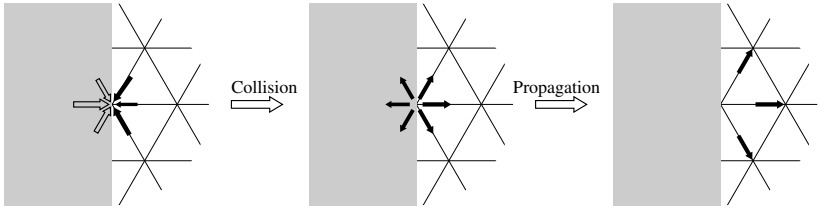


Fig. 31 *At the boundary of the domain, the density distribution coming from outside the system are not known. Specific boundary conditions must be used to define them.*

```
    collision
    propagation
  endfor
```

Note that here we have introduced a new operation, termed *observation* in the LB execution loop. This is were measurement can be done on the system. It is important to remember that f_i actually denotes f^{in} so that the theoretical results showing the correspondence between LB and physical quantities are only valid after the propagation step and before collision. Obviously, quantities such as ρ that are conserved during the collision can be measure after collision too. But, in hydrodynamics, this is not the case of the strain rate $S_{\alpha\beta}$.

In what follow we focus the discussion on simple hydrodynamical LB models. In practice, the way to impose a boundary condition is to use the correspondence expressed in eq. (39) to build the missing f_i from the desired values of the fluid variables at the boundary.

A standard situation is shown in Fig. 32, for a D2Q9 lattice. The density distributions f_2, f_3 and f_4 must be determined before applying the collision operator. Assuming that the y-axis is vertical and pointing upwards, we have the following relations (in lattice units where $v = 1$)

$$\begin{aligned} \rho &= f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 \\ \rho u_y &= f_2 + f_3 + f_4 - f_6 - f_7 - f_8 \end{aligned} \tag{55}$$

From these equations, we get

$$\rho - \rho u_y = f_0 + f_1 + f_5 + 2(f_6 + f_7 + f_8) \tag{56}$$

The right-hand side of this equation is fully know.

In case $\mathbf{u} = (u_x, u_y)$ is specified at the boundary (velocity boundary condition), we can compute ρ consistently [33]

$$\rho = \frac{f_0 + f_1 + f_5 + 2(f_6 + f_7 + f_8)}{1 - u_y}$$

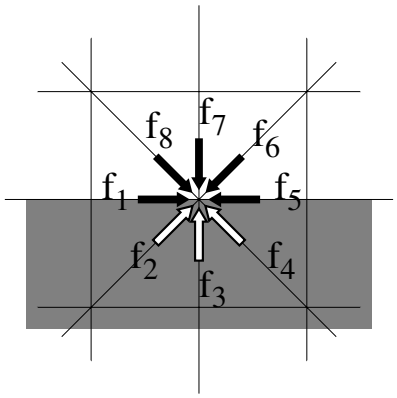


Fig. 32 For a flat wall at the lower boundary of a D2Q9 lattice, the distributions f_2, f_3 and f_4 are unknown and must be computed according to the desired boundary condition.

Once ρ and $\mathbf{u}$ are known, $f_i^{eq}(\rho,\mathbf{u})$ is also known for all i, due to eq. (36). Then, we can certainly compute $f_i^{neq} = f_i - f_i^{eq}$ for $i \in \{0,1,5,6,7,8\}$.

From these quantities we can now compute f_i^{neq} for the missing directions $i = 2,3,4$ by imposing that their non-equilibrium part obeys eq. (37), which tells us that $f_i^{neq} = f_{opp(i)}^{neq}$, where $j = opp(i)$ is the index such that $\mathbf{v}_j = -\mathbf{v}_i$.

Unfortunately, there is no guarantee that this choice of the non-equilibrium parts of the distributions is globally consistent. For instance we may well find that $\sum_i f_i^{neq} \neq 0$ which is inconsistent with eq. (55). The solution to this problem is a *regularization* step, in which the f_i^{neq} are redistributed over all directions. This step can be explained as follows: first we compute

$$\Pi_{\alpha\beta}^{neq} = \sum_i v_{i\alpha} v_{i\beta} f_i^{neq}$$

from the f_i^{neq} obtained above. Second, by combining eq. (37) and (38) we get

$$\begin{aligned} f_i^{neq} &= -\delta_t \tau \frac{w_i}{c_s^2} \rho Q_{i\alpha\beta} S_{\alpha\beta} \\ &= \frac{w_i}{2c_s^4} Q_{i\alpha\beta} \Pi_{\alpha\beta}^{neq} \end{aligned} \tag{57}$$

This equation allows us to recompute all f_i^{neq} ($i = 0,\ldots,z$) from the previous ones. All the f_i are then redefined to their regularized value

$$f_i = f_i^{eq} + f_i^{neq} \qquad i = 0,\ldots,z$$

This terminates the calculation of the boundary condition for $\mathbf{u}$ imposed at the wall and guarantees the proper values of ρ and $\mathbf{u}$ at the wall since $\sum_i w_i Q_{i\alpha\beta} = \sum_i w_i \mathbf{v}_i Q_{i\alpha\beta} = 0$

If, instead of $\mathbf{u}$, ρ is the prescribed quantity at the boundary (pressure boundary condition), we can determine the consistent flow speed u_y from eq. 56

$$u_y = 1 - \frac{f_0 + f_1 + f_5 + 2(f_6 + f_7 + f_8)}{\rho}$$

By choosing for instance $u_x = 0$ we then obtain f_i^{eq}, for all i. Then, the same regularization procedure as explained above can be used to compute all the f_i at the wall.

We refer the reader to [69, 41] for a detailed discussion of the above on-site boundary condition and to [5, 25, 38, 37] for a discussion of boundary conditions that are not located on lattice sites, or moving boundary conditions.

However, another simple solution to specify a boundary condition is to exploit the mesoscopic interpretation of the f_i as particles traveling with velocity $\mathbf{v}_i$. Following this idea, a very popular way to impose a boundary with zero velocity (no-slip condition) is to bounce back the particles from where they came. This is illustrated in fig. 33 (b). Using such a bounce back condition actually means a redefinition of

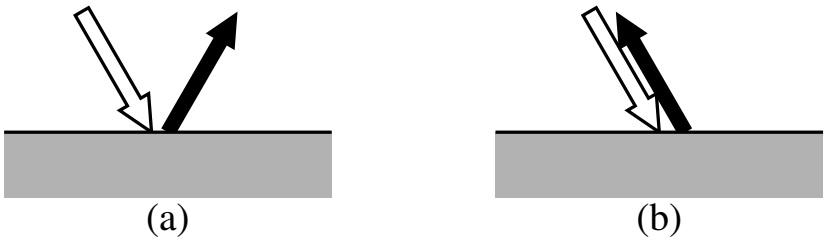


Fig. 33 Boundary conditions based on the particle interpretation of the LB method. The white arrows represent f^{in} and the black ones indicate f^{out}. (a) A free slip condition (specular reflection). (b) The bounce back rule to create a no-slip boundary condition on a wall.

the collision operator of the LB model on the boundary cells

$$f_i^{out} = f_{opp(i)}^{in}$$

where $opp(i)$ is the direction such that $\mathbf{v}_i = -\mathbf{v}_{opp(i)}$

In addition to the bounce back rule, periodic boundary condition can be used when appropriate. For instance, when simulating the flow in a straight tube, on can sometimes say that the particles leaving the tube through the outlet are re-injected in the inlet. In such a situation it is also convenient to add a body force $\mathbf{F}$ to create and maintain the flow in the tube. The LBGK fluid model can then be modified as follows

$$f_i(\mathbf{r} + \delta_t \mathbf{v}_i, t + \delta_t) = f_i(\mathbf{r}, t) + \frac{1}{\tau}\left(f_i^{eq} - f_i\right) + w_i \frac{\delta_t}{c_s^2} \rho \mathbf{v}_i \cdot \mathbf{F} \tag{58}$$

in order to reproduce a Navier-Stokes equation with external force

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F} \tag{59}$$

Note that this way of adding a body force on a LBGK fluid model is only possible is **F** is constant in space and time. For a more general body force, refer to [24].

5.6 Examples of LB modeling

The LB method has been used in many cases, from sediment transport to blood flow and clotting processes, to shallow water flow, just to mention a few applications. For the sake of illustration we show in this section the result of some simulations obtained with a LB model for complex flows, without detailed explanations. More examples and on-line animations can be found on the web[2,3] and in a vast body of literature on LB models.

By coupling an advection-diffusion LB model for a temperature field with a fluid LB model with gravity force, it is possible to model a thermal fluid in the Boussinesq approximation and, in particular, the well-known Rayleigh-Benard convection (instability that appears when a fluid is heated from below). This is illustrated in Fig. 34.

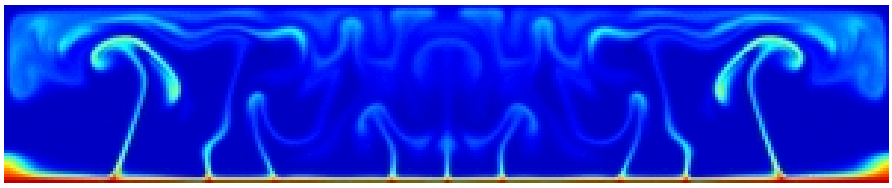


Fig. 34 A lattice Boltzmann simulation of a Rayleigh-Benard convection.

An interesting capability of the LB fluid models is to consider multicomponent flows. In short, two-fluid models are implemented with, in addition, a repulsive coupling between them. Several ways exist to create the separation of the components [44, 57]. Fig. 35 shows a simulation of the Rayleigh-Taylor instability which occurs when a heavy fluid layer is on top of a lighter one. Fig. 36 represents the evolution of droplets in suspension in another fluid subject to a shear flow.

Fig. 37 illustrates a two-component system (gas+liquid), in a porous media, with a buoyancy force field. Bubbles of gas are produced at the lower part of the system and they rise inside a porous media.

[2] http://cui.unige.ch/∼chopard/CA/Animations/root.html

[3] http://www.lbmethod.org

Fig. 35 Rayleigh-Taylor instability with two immiscible LB fluids. The blue fluid is more heavy than the black one.

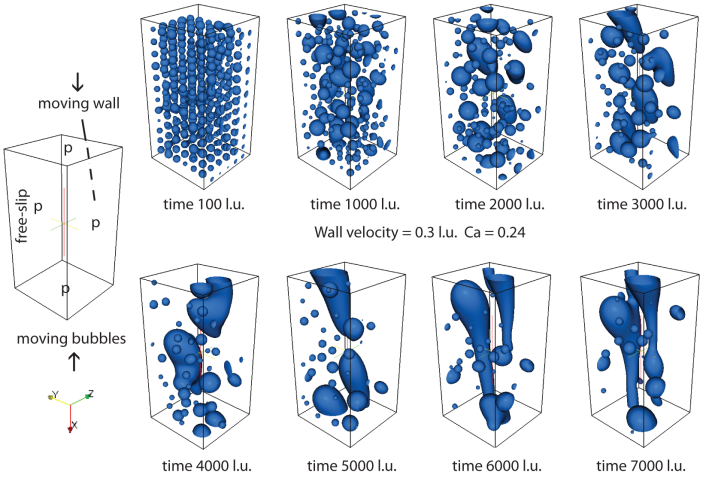


Fig. 36 Evolution of bubbles in a shear flow.

6 Conclusions

In this article we have presented the concepts underlying the CA and LB approaches. We have developed the ideas using a mesoscopic level of description. Physical systems are represented by idealized entities evolving on a discrete space-time universe. Implementing the correct conservation laws and symmetry in this virtual discrete

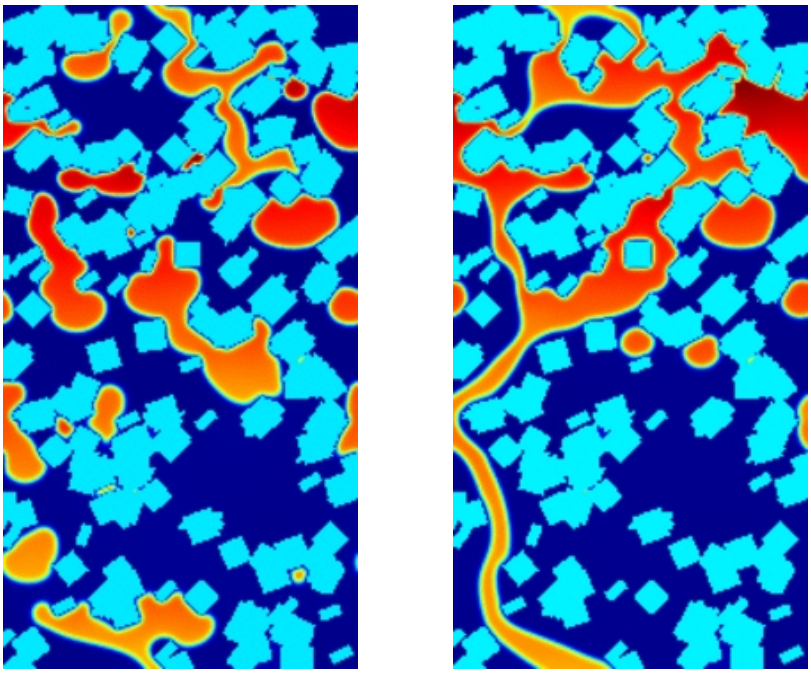


Fig. 37 Bubbles rising in a fluid inside a porous media.

mesoscopic universe ensures that the observed macroscopic behavior of the system is close enough to that of the corresponding physical system.

The clear relation that exists between the model and the real process makes CA and LB methods very intuitive. As such they offer a new language to describe physical systems (as illustrated in fig. 1) which has proven quite powerful in interdisciplinary research.

The simplicity and flexibility of the approach makes it very appropriate to describe many complex systems for which more traditional numerical methods are difficult to apply.

CA are by definition fully discrete numerical models. Usually they have only a few possible states per cell and they can be implemented very efficiently on low cost dedicated hardware. Since CA models only require integer values, their numerical implementation is exact. No numerical instabilities or truncation error affect the simulation.

LB model offer a higher level of abstraction and consider real-valued variables. They offer much more flexibility than CA to implement a given interaction rule. However, they can be numerically unstable. LB methods are now acknowledged as a powerful way to simulate hydrodynamics and specifically complex fluids in time dependent regimes. The reader interested to use the LB method can

learn more in the several textbooks mentioned earlier and by exploring the website www.lbmethod.org.

A current challenging issue in computational science is multiscale, multiscience modeling. Many real life problems cover a wide range of spatial and temporal scales and include different processes. Biomedical applications are a good example where physical processes (e.g. hemodynamics) interact with a much slower biological ones at a microsopic scale (e.g tissue modifications). Often, scales can be separated in the sense that the full system can be represented as several coupled submodels, each of them corresponding to a given scale and a given process.

Coupling different CA and LB models together is possible and has been investigated recently [30, 29] using the concept of CxA (*Complex Automata*) A CxA is a graph whose nodes are CA or LB models and the edges implement the coupling strategies. A software framework [27] offers a way to realize such a graph of submodels and to simulated challenging biomedical applications [18].

Acknowledgments

I thank Jonas Latt, Oresits Malaspinas, Andrea Parmigiani and Chris Huber for stimulating discussions and for providing many of the figures illustrating section 5.6.

References

1. F. J. Alexander, H. Chen, S. Chen, and G. D. Doolen. Lattice boltzmann model for compressible fluids. *Phys. Rev. A*, 46(4):1967–1970, Aug 1992.
2. S. Ansumali, I.V. Karlin, S. Arcidiacono, A. Abbas, and N.I. Prasianakis. Hydrodynamics beyond navier-stokes: Exact solution to the lattice boltzmann hierarchy. *Phys. Rev. Lett.*, 98:124502, 2007.
3. E. Banks. Information processing and transmission in cellular automata. Technical report, MIT, 1971. MAC TR-81.
4. P. Bhatnager, E.P. Gross, and M.K. Krook. A model for collision process in gases. *Phys. Rev.*, 94:511, 1954.
5. M. Bouzidi, M. Firdaouss, and P. Lallemand. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids,*, 13(11):3452–3459, 2001.
6. A.W. Burks. Von neumann's self-reproducing automata. In A.W. Burks, editor, *Essays on Cellular Automata*, pages 3–64. University of Illinois Press, 1970.
7. S. Chen and G.D. Doolen. Lattice Boltzmann methods for fluid flows. *Annu. Rev. Fluid Mech.*, 30:329, 1998.
8. S. S. Chikatamarla, S. Ansumali, and I.V. Karlin. Entropic Lattice Boltzmann models for hydrodynamics in three dimensions. *Phys. Rev. Lett.*, 97:010201, 2006.
9. B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, 1998.
10. B. Chopard and A. Dupuis. Cellular automata simulations of traffic: a model for the city of geneva. *Network & Spatial Economics*, 3:9–21, 2003.
11. B. Chopard, J.-L. Falcone, and J. Latt. The lattice boltzmann advection-diffusion model revisited. *European Physical Journal*, 2008. in press.

12. B. Chopard, P. Luthi, and M. Droz. Reaction-diffusion cellular automata model for the formation of Liesegang patterns. *Phys. Rev. Lett.*, 72(9):1384–1387, 1994.
13. B. Chopard, P. Luthi, A. Masselot, and A. Dupuis. Cellular automata and lattice boltzmann techniques: An approach to model and simulate complex systems. *Advances in Complex Systems*, 5(2):103–246, 2002. http://cui.unige.ch/∼chopard/FTP/CA/acs.pdf.
14. B. Chopard, P. O. Luthi, and P.-A. Queloz. Cellular automata model of car traffic in two-dimensional street networks. *J. Phys. A*, 29:2325–2336, 1996.
15. R. G. M. Van der Sman and M. H. Ernst. Convection-diffusion lattice boltzmann scheme for irregular lattices. *J. Comp. Phys.*, 160:766–782, 2000.
16. Andreas Deutsch and sabin Dormann. *Cellular Automaton Modeling of Biological Pattern Formation*. Birkäuser, 2005.
17. Dominique d'Humières, Irina Ginzburg, Manfred Krafczyk, Pierre Lallemand, and Li-Shi Luo. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Phil. Trans. R. Soc. A*, 360:437–451, 2002.
18. D. Evans, P.-V. Lawford, J. Gunn, D. Walker, D.-R. Hose, R.H. Smallwood, B. Chopard, M. Krafczyk, J. Bernsdorf, and A. Hoekstra. The application of multi-scale modelling to the process of development and prevention of stenosis in a stented coronary artery. *Phil. Trans. Roy. Soc.*, 2008. in press.
19. U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Phys. Rev. Lett.*, 56:1505, 1986.
20. S. Galam, B. Chopard, A. Masselot, and M. Droz. Competing species dynamics: Qualitative advantage versus geography. *Eur. Phys. J. B*, 4:529–531, 1998.
21. M. Gardner. The fantastic combinations of john conway's new solitaire game life. *Scientific American*, 220(4):120, 1970.
22. R. J. Gaylord and K. Nishidate. *Modeling Nature with Cellular Automata using Mathematica*. Springer-Verlag, 1996.
23. Irina Ginzburg. Equilibrium-type and link-type lattice boltzmann models for generic advection and anisotropic-dispersion equation. *Advances in Water Resources Pages*, 28(11):1171–1195, 2005.
24. Zhaoli Guo, Chguang Zheng, and Baochang Shi. Discrete lattice effects on forcing terms in the lattice boltzmann method. *Phys. Rev. E*, 65:046308, 2002.
25. Zhaoli Guo, Chuguang Zheng, and Baochang Shi. An extrapolation method for boundary conditions in lattice Boltzmann method. *Phys. Fluids*, 14:2007–2010, 2002.
26. X. He and L.-S. Luo. A priori derivation of the lattice Boltzmann equation. *Phys. Rev. E*, 55:R6333–R6336, 1997.
27. Jan Hegewald, Manfred Krafczyk, Jonas Tölke, Alfons Hoekstra, and Bastien Chopard. An agent-based coupling platform for complex automata. In M. Bubak et al., editor, *ICCS 2008*, volume LNCS 5102, pages 291–300. Springer-Verlag Berlin Heidelberg 2008, 2008.
28. W. J. R. Hoeffer. The transmission-line matrix method. theory and applications. *IEEE Trans. on Microwave Theory and Techniques*, MTT-33(10):882–893, October 1985.
29. A. Hoekstra, Falcone J.-L, A. Caiazzo, and B. Chopard. Multi-scale modeling with cellular automata: The complex automata approach. In H. Umeo et al., editor, *ACRI 2008*, volume LNCS 5191, pages 192–199. Springer-Verlag Berlin Heidelberg 2008, 2008.
30. A. Hoekstra, E. Lorenz, J.-L. Falcone, and B. Chopard. Towards a complex automata formalism for multuscale modeling. *Int. J. Multiscale Computational Engineering*, 5(6):491–502, 2008.
31. K. Culick II and F. Yu. Undecidability of ca classification scheme. *Complex System*, 2:177–190, 1988.
32. Andrew Ilachinski. *Cellular Automata: a discrete universe*. World Scientific, 2001.
33. Takaji Inamuro, Masato Yoshino, and Fumimaru Ogino. A non-slip boundary condition for lattice Boltzmann simulations. *Phys. Fluids*, 7(12):2928–2930, 1995.
34. M. Junk, A. Klar, and L.-S. Luo. Asymptotic analysis of the lattice boltzmann equation. *Journal of Computational Physics*, 210(2), 2005.
35. M. Kanai, K. Nishinari, and T. Tokihiro. Stochastic optimal velocity model and its long-lived metastability. *Phys. Rev. E*, 72:035102(R), 2005.

36. M. Kanai, K. Nishinari, and T. Tokihiro. Stochastic cellular automaton model for traffic flow. In S. El Yacoubi, B. Chopard, and S. Bandini, editors, *Cellular Automata: 7th ACRI conference*, volume 4173 of *LNCS*, pages 538–547. Springer, 2006.
37. P.-H. Kao and R.-J Yang. An investigation into curved and moving boundary treatments in the lattice Boltzmann method. *Journal of Computational Physics*, 227(11):5671–5690, 2008.
38. Pierre Lallemand and Li-Shi Luo. Lattice boltzmann method for moving boundaries. *Journal of Computational Physics*, 184(2):406–421, 2003.
39. Jonas Lätt. *Hydrodynamic Limit of Lattice Boltzmann equations*. PhD thesis, University of Geneva, Switzerland, 2007. http://www.unige.ch/cyberdocuments/theses2007/LattJ/meta.html.
40. Jonas Latt and Bastien Chopard. Lattice Boltzmann method with regularized non-equilibrium distribution functions. *Math. Comp. Sim.*, 72:165–168, 2006.
41. Jonas Latt, Bastien Chopard, Orestis Malaspinas, Michel Deville, and Andreas Michler. Straight velocity boundaries in the lattice Boltzmann method. *Phys. Rev. E*, 77:056703, 2008.
42. Pascal O. Luthi, Anette Preiss, Jeremy J. Ramsden, and B. Chopard. A cellular automaton model for neurogenesis in drosophila. *Physica D*, 118:151–160, 1998.
43. S. Marconi and B. Chopard. A lattice boltzmann model for a solid body. *Int. J. Mod. Phys. B*, 17(1/2):153–156, 2003.
44. N.S. Martis and H. Chen. Simulation of multicomponent fluids in complex 3d geometries by the lattice Boltzmann method. *Phys. Rev. E*, 53:743–749, 1996.
45. K. Nagel and H.J. Herrmann. Deterministic models for traffic jams. *Physica A*, 199:254, 1993.
46. K. Nagel and M. Schreckenberg. Cellular automaton model for freeway traffic. *J. Physique I (Paris)*, 2:2221, 1992.
47. J. Propp. Trajectory of generalized ants. *Math. Intelligencer*, 16(1):37–42, 1994.
48. D. Rothman and S. Zaleski. *Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics*. Collection Aléa. Cambridge University Press, 1997.
49. M. Schreckenberg, A. Schadschneider, K. Nagel, and N. Ito. Discrete stochastic models for traffic flow. *Phys. Rev. E*, 51:2939, 1995.
50. B. Servan-Camas and F.T.C. Tsai. Lattice Boltzmann method for two relaxation times for advection-diffusion equation:third order analysis and stability analysis. *Advance Water Resources*, 31:1113–1126, 2008.
51. Xiaowen Shan, Xue-Feng Yuan, and Hudong Chen. Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. *J. Fluid Mech.*, 550:413–441, 2006.
52. M. Sipper. *Evolution of Parallel Cellular Machines: The cellular programming approach*. Springer-Verlag, Berlin, 1997. Lecture notes in computer science, vol 1194.
53. I. Stewart. The ultimate in anty-particle. *Scientific American*, 270:88–91, July 1994.
54. Sauro Succi. *The Lattice Boltzmann Equation, For Fluid Dynamics and Beyond*. Oxford University Press, 2001.
55. Shinsuke Suga. Numerical schemes obtained from lattice boltzmann equations for advection diffusion equations. *Int. J. Mod. Phys. C*, 17(11):1563–1577, 2006.
56. M.C. Sukop and D.T. Thorne. *Lattice Boltzmann Modeling: an Introduction for Geoscientists and Engineers*. Springer, 2005.
57. M.R. Swift, E. Olrandini, W.R. Osborn, and J.M. Yeomans. Lattice boltzmann simulations of liquid-gas and binary fluid systems. *Phys. Rev. E*, 54:5041–5052, 1996.
58. T. Toffoli and N. Margolus. *Cellular Automata Machines: a New Environment for Modeling*. The MIT Press, 1987.
59. C. Vanneste, P. Sebbah, and D. Sornette. A wave automaton for time-dependent wave propagation in random media. *Europhys. Lett.*, 17:715, 1992.
60. G. Vichniac. Simulating physics with cellular automata. *Physica D*, 10:96–115, 1984.
61. Jörg R. Weimar. *Simulation with Cellular Automata*. Logos-Verlag, Berlin, 1998.
62. D.E. Wolf and collaborator, editors. *Traffic and Granular Flow '97*. Springer, to appear.
63. Dieter A. Wolf-Gladrow. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models: an Introduction*. Lecture Notes in Mathematics, 1725. Springer, Berlin, 2000.
64. S. Wolfram. *Theory and Application of Cellular Automata*. World Scientific, 1986.

65. S. Wolfram. *Cellular Automata and Complexity*. Addison-Wesley, Reading MA, 1994.
66. S. Wolfram. *A new kind of science*. WolframSciences, 2002.
67. Huidan Yu and Kaihua Zhao. Lattice boltzmann method for compressible flows with high mach numbers. *Phys. Rev. E*, 61(4):3867–3870, Apr 2000.
68. S. Yukawa, M. Kikuchi, and S. Tadaki. Dynamical phase transition in one-dimensional traffic flow model with blockage. *J. Phys. Soc. Jpn*, 63(10):3609–3618, 1994.
69. Q. Zou and X. He. On pressure and velocity boundary conditions fot he lattice boltzmann bgk model. *Phys. Fluids*, 7:2998, 1997.

Index