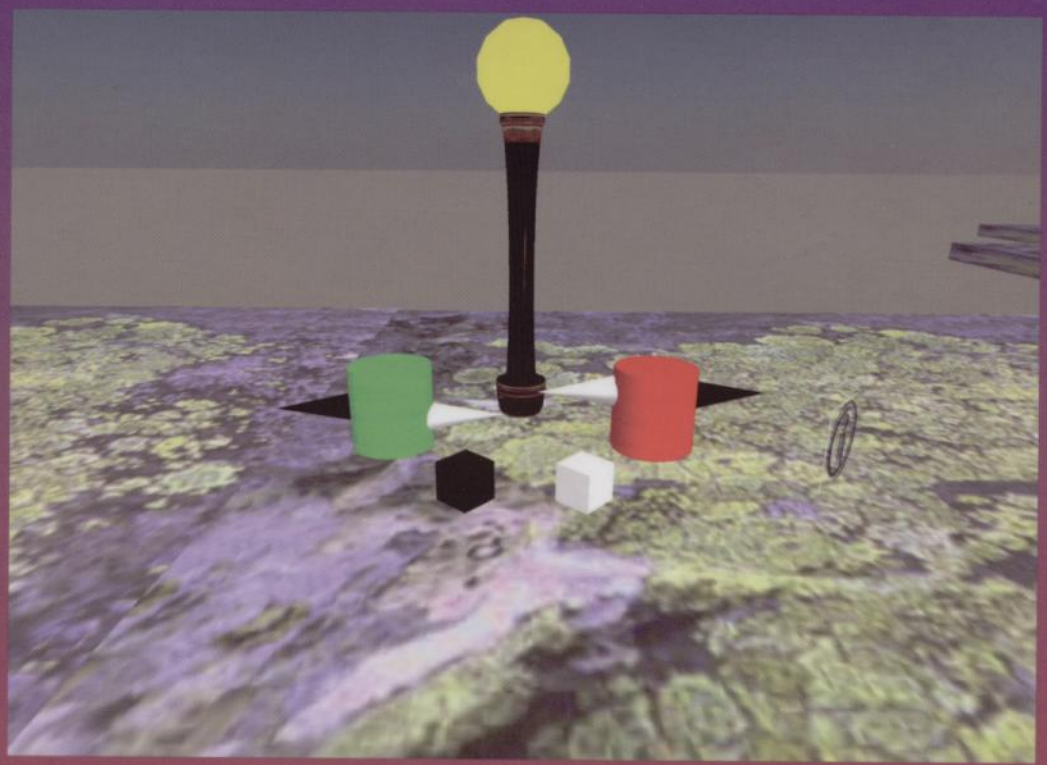
CHAPMAN & HALL/CRC COMPUTER and INFORMATION SCIENCE SERIES

Handbook of Dynamic System Modeling



Edited by

Paul A. Fishwick

Chapman & Hall/CRC
Taylor & Francis Group

21

Modeling Dynamic Systems with Cellular Automata

Peter M.A. Sloot
University of Amsterdam

Alfons G. Hoekstra
University of Amsterdam

21.1 Introduction

In modeling dynamic systems, one of the first questions to be answered is whether the involved processes can be viewed to be discrete in state, time, space, or continuous. The model choice should be robust with respect to the chosen space–time–state framework. Table 21.1 gives a selective overview of the various modeling approaches.

In this chapter, we focus on complete discrete model systems: cellular automata (CAs). CAs are decentralized spatially extended systems consisting of large numbers of simple and identical components with local connectivity. Such systems have the potential to perform complex computations with a high degree of efficiency and robustness as well as to model the behavior of complex systems from nature. CAs have been studied extensively in the natural sciences, mathematics, and computer science. They have been considered as mathematical objects about which formal properties can be proved and have been used as parallel computing devices, both for high-speed simulation of scientific models and for computational

TABLE 21.1 Mathematical and numerical modeling approaches to spatio-temporal processes.

Model/Variable	State	Space	Time
PDEs	C	C	C
Integro-difference equations	C	C	D
Coupled ODEs	C	D	C
Interacting particle systems	D	D	C
Coupled map lattices, systems of difference equations, LBE models	C	D	D
Cellular automata and lattice gas automata	D	D	D

PDE, partial differential equation; ODE, ordinary differential equation; LBE, lattice Boltzmann equation. For more details see Berec (2002) and Deutsch and Dormann (2004).

tasks such as image processing. CAs have also been used as abstract models for studying "emergent" cooperative or collective behavior in complex systems (e.g., Sloot, 2001b). In addition, CAs have been successfully applied to the simulation of a large variety of dynamical systems such as biological processes including pattern formation, earthquakes, urban growth, galaxy formation, and most notably in studying fluid dynamics. Their implicit spatial locality allows for very efficient high-performance implementations and incorporation into advanced programming environments. For a selection of the numerous papers in all of these areas, see, e.g., Bandini (2002), Burks (1970), Deutsch and Dormann (2004), Farmer et al. (1984), Forrest (1990), Frisch et al. (1986), Ganguly et al. (2003), Gutowitz (1990), Jesshope et al. (1994), Kaandorp et al. (1996), Mitchell (1998), Naumov (2004), Sloot (1999), Sloot and Hoekstra (2001), Sloot et al. (1997, 2001c, 2002, 2004), and Wolfram (1986a, 1986b, 2002).

In this chapter, we will give some background on CA modeling and simulation of dynamical systems with an emphasis on simulating fluid dynamics.

21.2 A Bit of History

In 1948, on the occasion of the Hixon Symposium at Caltech, John von Neumann gave a lecture entitled "The General and Logical Theory of Automata" (von Neumann, 1951, 1966), where he introduced his thoughts on universal, self-reproducing machines, trying to develop an abstract model of self-reproduction in biology, a topic that had emerged from investigations in cybernetics (Wolfram, 2002, 876 ff). von Neumann himself said to have been inspired by Stanislaw Ulam (1952, 1962) and Turing's theory of universal automata, which dates back another 10 years (Turing, 1936). Some scientists regard the paper by Wiener and Rosenblueth (1946) as the start of the field (Wolf-Gladrow, 2000), or mathematical work that was done in the early 1930s in Russia.

So we see that the roots of CA may be traced back to biological modeling, computer science, and (numerical) mathematics. From the early days of von Neumann and Ulam up to the recent book of Wolfram, CAs have attracted researchers from a wide variety of disciplines. It has been subjected to rigorous mathematical and physical analysis for the last 50 years, and its application has been proposed and explored in almost all branches of science. A large number of research papers are published every year. Specialized conferences, such as Sloot et al. (2004), Automata (2005), and NKS (2005), and special issues of various journals on CA have been initiated in the last decades. Several universities started offering courses on CA. The reason behind the popularity of CA can be traced to their simplicity and to the enormous potential they hold in modeling complex systems, in spite of their simplicity. Or in the words of R. May: "We would all be better off if more people realized that simple dynamical systems do not necessarily lead to simple dynamical behavior" (May, 1976). This has led to some very remarkable claims and predictions by renowned researchers about the potential of CAs. In this respect, we came across the following statements that are worth mentioning:

> *The entire universe is being computed on a computer, possibly a cellular automaton.*
>
> Konrad Zuse, as he referred to this as "Rechnender Raum" (Zuse, 1967, 1982)

> *I am convinced that CA, in one form or another will eventually be found lurking at the very heart of how the universe really works*
>
> Andrew Ilachinski (2001)

> *The view of the Universe as a cellular automaton provides the (same) perspective, (i.e.,) that reality ultimately is a pattern of information.*
>
> Ray Kurzweil (2002)

> *I have come to view [my discovery] as one of the more important single discoveries in the whole history of theoretical science*
>
> Stephan Wolfram: Talking about his CA work in his NKS book (Wolfram, 2002)

The remainder of this chapter is organized as follows, we start with a formal description of CA and some of their spatio-temporal properties and we will briefly discuss their capacity to model dynamical systems. Next we will focus on the use of CAs to model fluid flow, starting from Lattice Gas CAs up to recent developments in the related Lattice Boltzmann Method for fluid flow.

21.3 Cellular Automata to Model Dynamical Systems

In general, a CA is specified by the following four characteristics:

- *A discrete lattice* $\mathcal{L}$: This is the discrete lattice of cells (nodes and sites) upon which the CA dynamics unfolds. $\mathcal{L} \subset \mathcal{R}_{\bullet}^{\mathcal{D}}$ consists of a set of cells that homogeneously cover a $\mathcal{D}$-dimensional Euclidian space. $\mathcal{L}$ can have any dimension "$\mathcal{D}$" (normally 1, 2, or 3), with well-defined boundary conditions.
- *A finite state space*: Each cell can assume only one of a finite number of different values: $\sigma_{i\in\mathcal{L}}(t) \in \Sigma \equiv \{0, 1, 2, \cdots, k-1\}$, where σ_i is the value of the ith cell at time t, and Σ is usually taken to be the set of integers modulo k, $\mathcal{Z}_k$ (formally any finite commutative ring will do). For a finite lattice of $\mathcal{N}$ cells, the total number of global states is also finite and given by $k^{\mathcal{N}}$.
- *Boundary conditions*: Boundary conditions play an important role in CA dynamics. Although CA are defined on infinite large lattices, computer simulations impose finite sets. Common boundary conditions are *periodic* (i.e., the lattice is repeated periodically in each direction, in effect wrapping the boundaries onto each other in each direction), *reflecting* (i.e., boundary values are reflected back into the lattice), and *fixed* (i.e., the boundary values have a prescribed fixed value).
- *Dynamical update and transition Rule* ϕ:
  ϕ: $\underbrace{\Sigma \times \Sigma \times \cdots \times \Sigma}_{n} \rightarrow \Sigma$, where n is the number of cells that defines the "neighborhood" of a given cell i. With $\mathcal{S}_i$ to be the sublattice neighborhood about cell i, the transition rule is given by $\sigma_i(t+1) = \phi(\sigma_j(t) \in \mathcal{S}_i)$.

The spatial arrangement of the cells is specified by the nearest neighbor connection links, obtained by joining pairs of cells. State transitions are local in both space and time. Individual cells evolve iteratively according to a fixed (often deterministic) function of the current state of that cell and its neighboring cells. One iteration step of the dynamical evolution is achieved after synchronous (i.e., simultaneous in time) application of the rule ϕ to each cell in the lattice $\mathcal{L}$.

21.4 One-Dimensional CAs

The general form of a one-dimensional (1D) CA rule ϕ with an arbitrary range 'r' is given by $\sigma_i(t+1) = \phi(\sigma_{i-r}(t), \ldots, \sigma_i(t), \ldots, \sigma_{i+r}(t))$; with ϕ: $\Sigma^{2r+1} \rightarrow \Sigma$, where $\sigma_j \in \{0, 1, \ldots, k-1\}$, and ϕ is explicitly defined by assigning values to each of the k^{2r+1} possible $(2r+1)$-tuples of possible configurations for a given sublattice neighborhood $\mathcal{S}_i$. From this we see that we have a total of $k^{k^{2r+1}}$ possible rules in a 1D CA. So for a binary state 1D CA, with nearest neighbor interaction, there are 256 (2^{2^3}) possible rules.

The boundary conditions imposed in a 1D CA can be

- *Periodic*: When the left boundary cell is kept identical to the most right (normal) cell, and the right boundary cell is kept identical to the most left (normal) cell.
- *Reflecting*: When the left boundary cell is kept identical to the most left (normal) cell, and the right boundary cell is kept identical to the most right (normal) cell.
- *Fixed*: When the boundary cells are set to a fixed value.

The system dynamics of the CA is determined by the local transition rule ϕ, which can be *spatial homogeneous* (independent of cell position), or *inhomogeneous* (for instance in the case of fixed boundary conditions shown in Figure (21.1). Furthermore, the update can be *time dependent* or *time independent*,

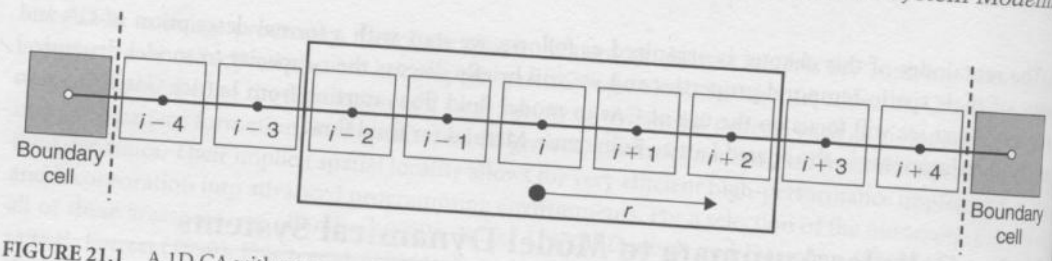


FIGURE 21.1 A 1D CA with range $r = 2$.

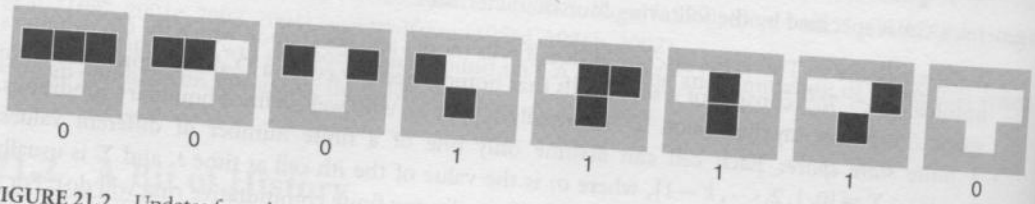


FIGURE 21.2 Updates for rule 30. The upper three blocks denote the cell that must be update, together with its left and right neighbors, and the lower block shows the outcome of applying rule 30. Black denotes for a state 1 and white for a state 0.

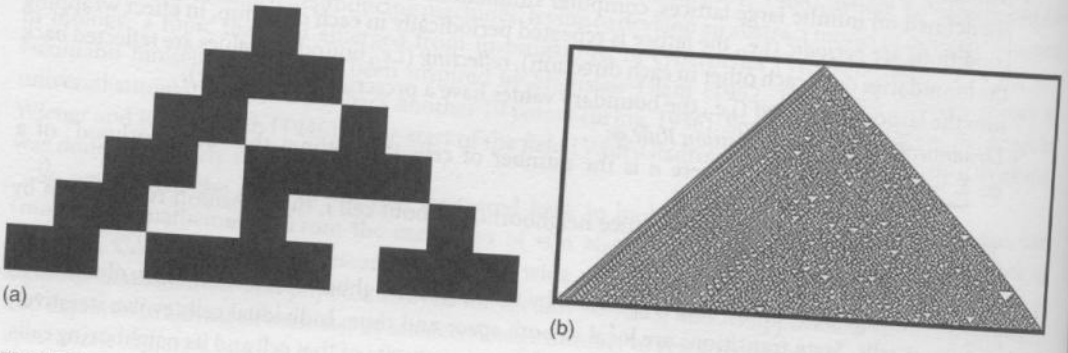


FIGURE 21.3 (a) Space–time diagram of Wolfram's elementary ($k = 2, r = 1$) CA rule 30 starting from a single seed. The sequence of updates "time" runs from top to bottom. (b) same rule for 300 updates on the 1D CA.

synchronous or *asynchronous*, *deterministic* or *stochastic*. This gives the modeler a plethora of possibilities to capture the dynamics of the system to be modeled. In the paragraph on lattice gas CAs, we will see more examples of how to design and apply such update rules. For the binary state 1D CA with nearest neighbor interaction, Wolfram introduced a convenient rule-code $\mathcal{R}$ for these important CAs, which uniquely identifies the update mechanism:

$$\mathcal{R}[\phi] = 2^7 \cdot \phi_{1,1,1} + 2^6 \cdot \phi_{1,1,0} + 2^5 \cdot \phi_{1,0,1} + 2^4 \cdot \phi_{1,0,0} + 2^3 \cdot \phi_{0,1,1} + 2^2 \cdot \phi_{0,1,0} + 2^1 \cdot \phi_{0,0,1} + 2^0 . \phi_{0,0,0}$$

The number ϕ is an eight-bits number that encodes all possible 256 rules for this 1D CA and $\phi_{i,j,k}$ are the bits of the binary notation of ϕ. The rule should be read as follows. The outcome of the rule is determined by the current state of a cell (0 or 1) and the current state of the left and right neighbor of the cell. The binary number that is formed by concatenating the state of the left neighbor, the cell itself, and the right neighbor is a number between 0 and 7. The outcome of rule $\mathcal{R}[\phi]$ is then the bit of the binary representation of ϕ at the position of the number encoded by the input states. So, for example, $\mathcal{R}[30] = 00011110$, since the decimal value of $00011110 = 30$. The update rules in terms of black (=1) and white (=0) blocks are graphically depicted in Figure 21.2.

The first five updates (*spatial homogeneous, time independent, synchronous, and deterministic)* of this $\mathcal{R}[30]$, starting from a single seed is shown in Figure 21.3.

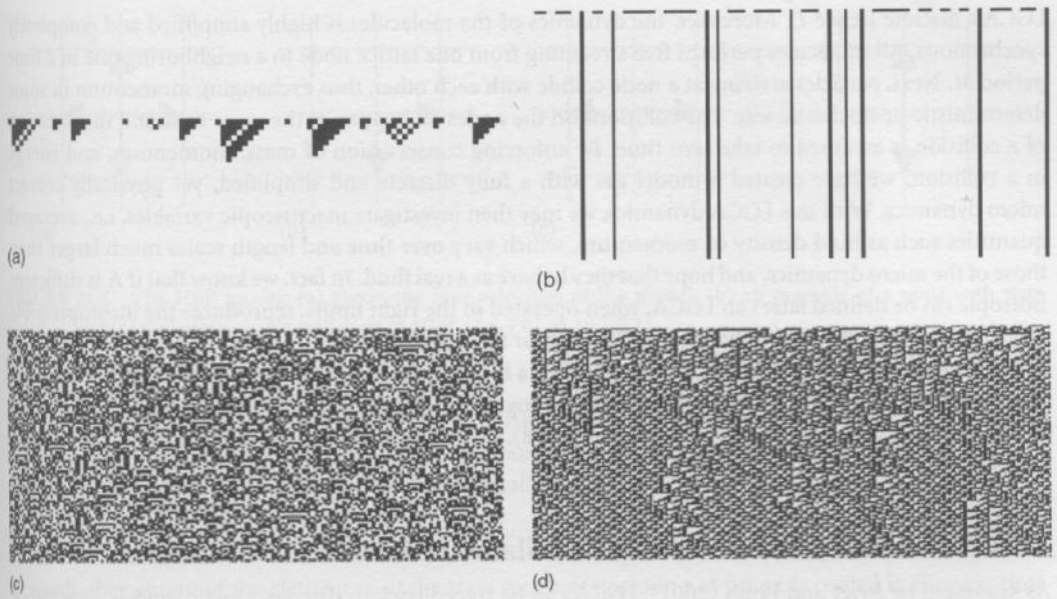


FIGURE 21.4 (a) Class 1 (e.g., Rules 0, 8, 128, 136, 160, 168): evolution leads to a homogeneous state, in which all cells eventually attain the same value (continues analog: attractive fixed limit point), shown is rule 168. (b) Class 2 (e.g., Rules 4, 37, 56, 73): evolution leads to inhomogeneous state; either simple stable states or periodic and separated structures (continues analog: limit cycle), shown is rule 4. (c) Class 3 (e.g., Rules 18, 45, 105, 126): evolution leads to chaotic nonperiodic patterns (continues analog: strange attractor), shown is rule 105. (d) Class 4 (e.g., rules 30, 110): evolution leads to complex, localized propagating structures (no continuous analog), shown is rule 110.

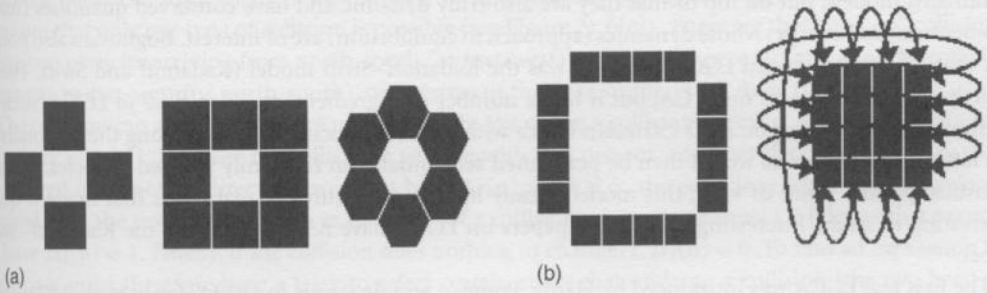


FIGURE 21.5 (a) von Neumann, Moore, and Hexagonal neighborhoods. (b) Fixed and periodic boundary conditions.

In his seminal paper on CA classification, Wolfram (1986a) identified four classes of CAs, linking them to analogs in continuous system dynamics (see Figure 21.4).

The behavior in two-dimensional (2D) CAs is much more complex and less well understood. In this case, we need again to define the update neighborhood and the boundary conditions; this is shown in Figure 21.5.

21.5 Lattice Gas Cellular Automata Models of Fluid Dynamics

Perhaps the most successful practical application of CAs as computing devices has been in the field of fluid dynamics. Coined lattice gas cellular automata (LGCA), this class of CA mimics a fully discretized fictitious fluid. Both the positions and velocities of the fluid's "molecules" are discrete, and tightly coupled to the

LGCA's discrete lattice $\mathcal{L}$. Moreover, the dynamics of the molecules is highly simplified and completely synchronous. All molecules perform free streaming from one lattice node to a neighboring one in a time period δt. Next, particles arriving at a node collide with each other, thus exchanging momentum in some deterministic or stochastic way. The collisions on the nodes all happen at the same time and the duration of a collision is assumed to take zero time. By enforcing conservation of mass, momentum, and energy in a collision, we have created a model gas with a fully discrete and simplified, yet physically correct micro dynamics. With this LGCA dynamics, we may then investigate macroscopic variables, i.e., averaged quantities such as fluid density or momentum, which vary over time and length scales much larger than those of the micro dynamics, and hope that they behave as a real fluid. In fact, we know that if Λ is sufficient isotropic (to be defined later) an LGCA, when operated in the right limits, reproduces the incompressible Navier–Stokes equations and therefore is a model for fluid dynamics.

The most complete account of LGCA (including a highly useful "guide to further reading") is the book by Rivet and Boon (2001). Other influential monographs on LGCA are Rothman and Zaleski (1997), Wolf-Gladrow (2000), and Chopard and Droz (1998). Finally, Boghosian (1999) provides a nice overview of lattice gases and cellular automata.

21.5.1 The Road to Lattice Gas Cellular Automata

As suggested by Rivet and Boon (2001), LGCA can be traced back to discrete kinetic models, in which a gas is modeled as a collection of particles with continuous position and time variables, but with a (small) discrete set of velocities. Such discrete kinetic models were studied intensively starting in the sixties of the previous century. LGCA would then be one step further down the road to minimalist models, in which also space and time are discrete. Indeed, in 1972 the point of departure for Hardy and Pomeau, who 1 year later introduced the first real LGCA, was the discrete velocity Maxwell model (Hardy and Pomeau, 1972). In contrast, Boghosian (1999) suggests a connection between early minimalist discrete models in statistical physics (such as the Ising spin, Creutz, and Kawasaki models) in the sense that LGCAs are comparable minimalist models, but on top of that they are also truly dynamic and have conserved quantities (mass, momentum, and energy) whose dynamics (approach to equilibrium) are of interest. Boghosian also points out that a first step toward LGCA probably was the Kadanoff–Swift model (Kadanoff and Swift, 1968). Strictly speaking this was not a CA, but it had a number of ingredients that are close to LGCA, such as fictitious particles living on a 2D Cartesian lattice with discrete velocities oriented along the diagonals of the lattice. The dynamics would then be performed sequentially on randomly selected particles. From a statistical physics point of view, this model already had many features of real fluids that made it quite interesting to study. Interestingly, the basic papers on LGCA have never referenced the Kadanoff–Swift paper.

The first real LGCA was introduced by Hardy, Pomeau and de Pazzis in 1973 (Hardy et al., 1973), and its hydrodynamics were studied in detail in Hardy et al. (1976). The HPP model, as we now call it, is a real CA. It has an underlying 2D Cartesian lattice. On each node, particles with unit mass are defined that can have one one of the four discrete velocities $\mathbf{c}_i$, $i \in \{1, 2, 3, 4\}$ (see also Figure 21.6(a)):

$$\mathbf{c}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \mathbf{c}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \quad \mathbf{c}_3 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}; \quad \mathbf{c}_4 = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \tag{21.1}$$

At each node only one particle can have a velocity $\mathbf{c}_i$. This exclusion principle allows us to denote the state of a node as a binary 4-vector $\mathbf{n}$. A value "TRUE" (or 1) of element i of the state vector (denoted as n_i) encodes for the presence of a particle with velocity $\mathbf{c}_i$, and a value "FALSE" (or 0) denotes the absence of a particle with velocity $\mathbf{c}_i$. So, the vector (1,0,0,1) would represent a state as shown in Figure 21.6(b). The number of different states per node in HPP is 16. Owing to the exclusion principle, the maximum number of particles on a node is 4. The dynamics is very straightforward. First, incoming particles at a node collide, and next particles perform a free streaming, in which they move from their node to the neighboring node in the direction of their velocity. We assume that the time for this streaming $\delta t = 1$, so

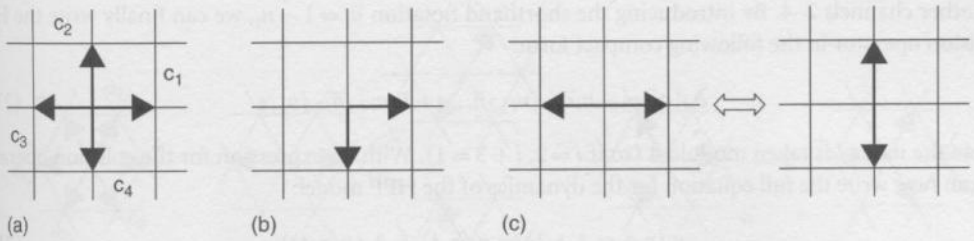


FIGURE 21.6 The HPP model: (a) shows the four velocities of the model, (b) an example of a node with state (1,0,0,1), and (c) the only possible collisions in HPP, (1,0,1,0) ⇔ (0,1,0,1).

that with the velocities as defined in Eq. (21.1), particles move exactly from one node to a neighboring node during the streaming phase. Assigning a position vector $\mathbf{r}$ to the state vector of the node at $\mathbf{r}$, and a time stamp t, the streaming step can be mathematically expressed as

$$n_i(\mathbf{r} + \mathbf{c}_i, t + 1) = n_i(\mathbf{r}, t) \tag{21.2}$$

In words, the content of the element n_i of the state vector at position $\mathbf{r}$ at time t is copied in the next time step to the neighbor at position $\mathbf{r} + \mathbf{c}_i$. If $n_i(\mathbf{r}, t) = 0$, it just means that at $t + 1$ $n_i(\mathbf{r} + \mathbf{c}_i, t + 1)$ is correctly set to 0.[1]

The collision is also very straightforward. In a collision, the velocities of the particles are redistributed. During a collision we must conserve mass, momentum, and energy. Since for all particles the magnitude of velocity and their mass is 1, their kinetic energy is always 1/2. Therefore, mass conservation immediately implies energy conservation.[2] Mass conservation means preservation of the number of particles in the collision. The momentum of a particle is just $\mathbf{c}_i$. So, during a collision we must preserve the total momentum on a node, i.e., $n_1\mathbf{c}_1 + n_2\mathbf{c}_2 + n_3\mathbf{c}_3 + n_4\mathbf{c}_4$. By systematically going through all possible 16 states, it turns out that only *one* type of collision is possible (see Figure 21.6[c]). These are the "head-on" collisions, where two particles arriving from north-south (or east-west) are scattered over 90° and after the collision propagate to east-west (or north-south), or in terms of the state vector, (1,0,1,0) ⇔ (0,1,0,1).

The collision can also be described more formally. We define a collision operator $\Delta_i(\mathbf{n})$, which can take the values $\{-1,0,1\}$. If before the collision a particle with $\mathbf{c}_i$ is present, and after the collision this particle is scattered into another direction, we must have $\Delta_i(\mathbf{n}) = -1$ (i.e., the particle is removed from velocity channel i). In the reverse case, when as the result of a collision, an empty channel i is filled with a particle, we have $\Delta_i(\mathbf{n}) = 1$. Finally, if the collision does nothing to channel i, $\Delta_i(\mathbf{n}) = 0$. To find an expression for $\Delta_i(\mathbf{n})$, we must therefore have a trick to select certain states that undergo a collision (the two head-on cases) and then assign the correct value to $\Delta_i(\mathbf{n})$. Here we take advantage of the binary notation that we introduced using the symbols "0" and "1." In the collision operator, we assume these symbols are in fact the integer numbers 0 or 1 and we compute with them. So, to select say the state (1,0,1,0) we could formulate a logical expression as n_1 AND (NOT n_2) AND n_3 AND (NOT n_4), which returns TRUE if the state (1,0,1,0) is present at a node and FALSE in all other cases. We could also write $n_1(1 - n_2)n_3(1 - n_4)$ and fill in the number 1 and 0 depending on the state. This expression will evaluate to 1 for the state (1,0,1,0) and to 0 in all other cases. In the case of a precollision state (1,0,1,0), we know that the result of the collision must be (0,1,0,1), so for, e.g., channel 1 we must have $\Delta_1(\mathbf{n}) = -1$. This we can achieve by writing $\Delta_1(\mathbf{n}) = -n_1(1 - n_2)n_3(1 - n_4)$. However, this is not the complete expression as we must also accommodate the reverse situation, i.e., that the precollision state is (0,1,0,1), in which case a particle will appear in channel 1, and $\Delta_1(\mathbf{n}) = 1$. We achieve this by adding another term, i.e., $(1 - n_1)n_2(1 - n_3)n_4$ resulting in the full expression $\Delta_1(\mathbf{n}) = -n_1(1 - n_2)n_3(1 - n_4) + (1 - n_1)n_2(1 - n_3)n_4$, and likewise for

[1] Physicists would say that in this case a "hole" is streaming from $\mathbf{r}$ to $\mathbf{r} + \mathbf{c}_i$.

[2] For this reason, the HPP model (and other "homokinetic" models) have no thermal effects.

the other channels 2–4. By introducing the shorthand notation $\overline{n}_i = 1 - n_i$, we can finally write the HPP collision operator in the following compact form:

$$\Delta_i(\mathbf{n}) = -n_i\overline{n}_{i+1}n_{i+2}\overline{n}_{i+3} + \overline{n}_i n_{i+1}\overline{n}_{i+2}n_{i+3} \tag{21.3}$$

where the index i is taken modulo 4 (so if $i=2$, $i+3=1$). With an expression for the collision operator, we can now write the full equation for the dynamics of the HPP model:

$$n_i(\mathbf{r} + \mathbf{c}_i, t+1) = n_i(\mathbf{r}, t) + \Delta_i(\mathbf{n}(\mathbf{r}, t)) \tag{21.4}$$

In fact, Eq. (21.4) expresses the micro dynamics for all LGCAs. However, for each specific model, the total number and definitions of the $\mathbf{c}_i$ may be different, and the details of the collision operator are different.

We can now formulate the following CA rule for LGCA.

```
for each node in the Lattice do
    1. Perform a collision step, i.e., redistribute the state
    vector n such that n_i := n_i + Δ_i (n)
    2. Perform a streaming step, i.e.,
        for all i
            copy n_i to n_i at position = my_position + c_i.
        update the time t := t + 1
```

Having defined the structure, collision operator, and the dynamics as expressed in the CA rule, we are now in the position to execute the HPP LGCA. Next, we must define observables. The total number of particles and total momentum on a node are obtained by summing n_i and $n_i\mathbf{c}_i$ over all i, respectively. However, these instantaneous observables are very noisy, they fluctuate strongly as a function of time and position. Although these fluctuations contain a wealth of interesting physical information (Rivet and Boon, 2001), we want to observe smooth hydrodynamic fields such as the density or momentum of the fluid. To achieve this we must first take ensemble averages of n_i, yielding $f_i = <n_i>$. The f_i values are real numbers between 0 and 1 and should be interpreted as the probability to find a particle with velocity $\mathbf{c}_i$. In an LGCA simulation, we can compute the ensemble average by, e.g., taking spatial or temporal averages of the $n_i(\mathbf{r},t)$. We can now define the fluid density ρ and fluid velocity $\mathbf{u}$ as follows:

$$\begin{aligned} \rho(\mathbf{r}, t) &= \sum_{i=1}^{b} f_i(\mathbf{r}, t) \\ \rho(\mathbf{r}, t)\mathbf{u}(\mathbf{r}, t) &= \sum_{i=1}^{b} \mathbf{c}_i f_i(\mathbf{r}, t) \end{aligned} \tag{21.5}$$

where b is the total number of velocity vectors (for HPP $b=4$). With these definitions, and the full machinery of statistical mechanics and kinetic theory, one can work out the equations that govern ρ and $\mathbf{u}$. Although the resulting equations for HPP have a strong resemblance to the Navier–Stokes equations that govern macroscopic fluid flow, there is a major flaw. It turns out that the resulting macroscopic equations are not isotropic, meaning that the flow properties depend on the orientation with respect to the underlying lattice. This problem can be traced back to isotropy properties of the underlying HPP lattice and its four discrete velocities.[3] This anisotropy is of course unacceptable for a model of fluid dynamics, and therefore HPP did not catch the attention of people interested in doing fluid dynamics.

In 1986 the big breakthrough came for LGCA. Frish et al. (1986) introduced the first LGCA that produces isotropic macroscopic equations for the density and velocity, reproducing the Navier–Stokes equations for an incompressible fluid. The main innovation in this FHP model was to change the underlying lattice

[3]Technically, the HPP model has a crystallographic isotropy of order 3, which is too low to obtain isotropic macroscopic equations (for details on this issue, see Rivet and Boon [2001]).

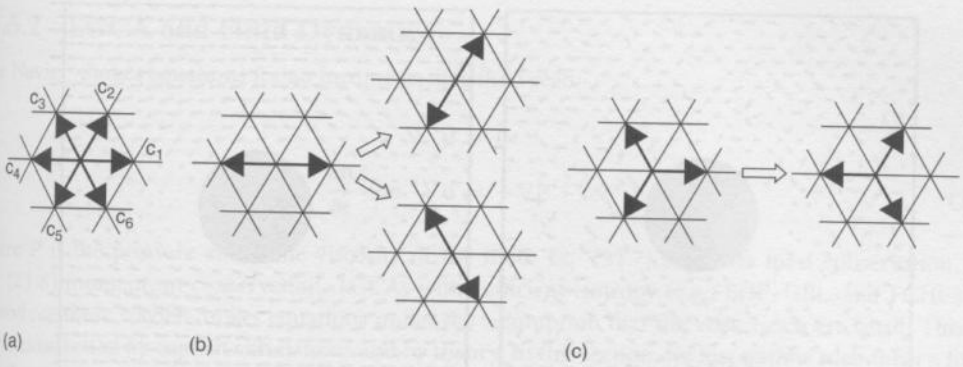


FIGURE 21.7 The FHP model: (a) shows the six velocities of the model, (b) the head on collision, and (c) the triplet collision.

to a 2D triangular lattice, and define six discrete velocities along the six directions of the lattice (see Figure (21.7[a]). This lattice has sufficient isotropy[4] to reproduce isotropic Navier–Stokes equations for an incompressible fluid. In this version of the FHP model (coined FHP-I),[5] the magnitude of the six discrete velocities are the same and equal to 1. Therefore, mass conservation and energy conservation are equivalent and FPH-I is also an a-thermal model. The micro dynamics are governed by Eq. (21.4), but we need to adapt the collision operator. In FHP-I, we only consider two types of collisions (see Figure 21.7[b] and Figure 21.7[c]). The first type is the head-on collision (as in HPP). However, in this case two possible postcollision states are possible (rotated $+60°$ or $-60°$ with respect to the incoming direction). FHP-I randomly selects one of those two postcollision states with a probability of 0.5. Because of this, FHP-I is no longer a deterministic CA, but has become probabilistic. The second type of collision is the triplet state, where three particles arrive with mutual angles of 120°. The postcollision state is the same triplet rotated over 60°. Using the same procedure as for HPP, we can express the collision operator for FHP-I as follows

$$\begin{aligned}\Delta_i^{\text{FHP-I}}(\mathbf{n}) = &- n_i\overline{n}_{i+1}\overline{n}_{i+2}n_{i+3}\overline{n}_{i+4}\overline{n}_{i+5}\\ &+ \xi\overline{n}_i n_{i+1}\overline{n}_{i+2}\overline{n}_{i+3}n_{i+4}\overline{n}_{i+5}\\ &+ (1-\xi)\overline{n}_i\overline{n}_{i+1}n_{i+2}\overline{n}_{i+3}\overline{n}_{i+4}n_{i+5}\\ &- n_i\overline{n}_{i+1}n_{i+2}\overline{n}_{i+3}n_{i+4}\overline{n}_{i+5}\\ &+ \overline{n}_i n_{i+1}\overline{n}_{i+2}n_{i+3}\overline{n}_{i+4}n_{i+5}\end{aligned} \tag{21.6}$$

where ξ is a Bernouilli random variable (i.e., it randomly takes the values 0 or 1) with mean 0.5. On each time step and at each lattice node ξ is evaluated. On the right-hand side, the first three terms represent the head-on collisions and the last two terms the triplet collision. The variable i is now taken modulo 6.

We can now proceed to execute the FPH-I LGCA and compute observables using Eq. (21.5), where $b=6$. In Figure 21.8, an example is shown, demonstrating the need to perform ensemble averaging before computing the observables. In Figure 21.8(a) we show the results of a single iteration of FHP-I, in fact we have assumed that $f_i = n_i$ (i.e., no ensemble averaging is performed). Clearly, the resulting flow field is very noisy. To observe smooth flow lines one should really compute $f_i = <n_i>$. Because the flow is static, we compute the ensemble average f_i by averaging the Boolean variables n_i over a large number of FHP iterations. The resulting flow velocities are shown in Figure 21.8(b).

[4]It has crystallographic isotropy of order 5, so sufficient for the required fourth-order isotropy needed for isotropic large-scale dynamics.

[5]A number of extensions to FHP exist, including the so-called rest particles (i.e., particles on a node with zero velocity, $\mathbf{c}_0 = \{0,0\}$), and with extended collision operators, taking into account all possible collisions (including the rest particle). These extensions will not be further treated here, but see Rivet and Boon (2001).

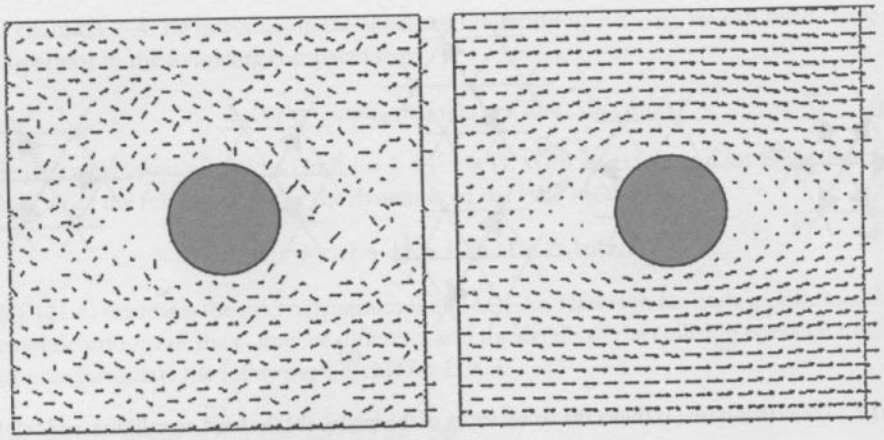


FIGURE 21.8 (a) FHP simulation of flow around a cylinder, the result of a single iteration of the LGCA is shown (i.e., no averaging). The arrows are the flow velocities, the length is proportional to the absolute velocity. The simulations were done on a 32×64 lattice, the cylinder has a diameter of 8 lattice spacing, only a 32×32 portion of the lattice is shown; periodic boundary conditions in all directions apply. (b) As in (a), the velocities are shown after averaging over 1000 LGCA iterations.

The year 1986 marked the beginning of an enormous research activity on LGCAs. Chapter 11 (guide for further reading) from Rivet and Boon (2001) provides numerous references to this literature. Here we will only touch upon a few interesting developments of practical interest. First, we mention the extension to three-dimensional (3D) models, clearly a necessity for a serious model of hydrodynamics. It turns out that a 3D LGCA is far from trivial. In fact, the LGCA community rapidly realized that all 3D regular (Bravais) lattices do not have sufficient isotropy ("the magic didn't work," as Rivet and Boon put it). However, by taking a detour into four-dimensional (4D) space and projecting back to three dimensions, d'Humières, Lallemand, and Frisch were able to build a 3D LGCA with all required isotropy properties (d'Humières et al., 1986). This model is based on a 4D face-centered hyper cube (FCHC), and has as many as 24 velocity channels per node. The sheer amount of possible states in this model ($2^{24} = 16{,}777{,}261$) and the number of possible collisions (18,736 or 10,805, depending on assumption put on the model) make it completely impossible to write down an explicit equation for the collision operator, and one must resort to a more general approach. Moreover, an efficient implementation of such a complicated collision operator requires new algorithmic strategies (see, e.g., Hénon, 1987), clever lookup table strategies (see, e.g., Rivet and Boon, 2001), or a combination of both.

An example of a thermal LGCA (i.e., a model where energy conservation is nontrivial) is the 2D model proposed by (Grosfils et al. 1992). This GBL model, like FHP, is defined on the 2D triangular lattice, but now has 19 velocities. It has one rest particle, six particles connected to nearest neighbors ($\|\mathbf{c}_i\| = 1$), six connected to next-nearest neighbors ($\|\mathbf{c}_i\| = \sqrt{3}$), and six connected to next-next-nearest neighbors ($\|\mathbf{c}_i\| = 2$). The GBL model has $2^{19} = 524{,}288$ possible states, of which 517,750 can undergo collisions that change the state while preserving mass, momentum, and energy. Again, the complexity of the collision operator requires efficient implementations (see, e.g., Dubbeldam et al., 1999). GBL is a true thermo-hydrodynamic model for 2D fluid dynamics.

LGCAs are easily extended to multiple species models by coloring the particles. Besides the normal collision rules, one also demands a color conservation. After collisions, colors are then randomly redistributed, while preserving total color. In this way nonreacting mixed fluids can be modeled. Moreover, by adding reactions one can create reaction–diffusion LGCAs (see, e.g., Boon et al., 1996) or by adding an interaction term between differently colored particles, one can model multiphase immiscible fluids (see, e.g., Rothman and Zaleski, 1997).

21.5.2 LGCA and Fluid Dynamics

The Navier–Stokes equations for an incompressible fluid read

$$\nabla \cdot \mathbf{u} = 0 \tag{21.7}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} \tag{21.8}$$

where P is the pressure and ν the viscosity of the fluid. Eq. (21.7) expresses mass conservation, and Eq. (21.8) momentum conservation. LGCAs with sufficient isotropy (e.g., FHP, GBL, and FCHC) can reproduce these Navier–Stokes equations under the assumption that the velocities $\mathbf{u}$ are small. This can be demonstrated by explicit simulations and by theory. In this section, we just outline such theory, for all details we refer to Frisch et al. (1987), Rivet and Boon (2001), Rothman and Zaleski (1997), Chopard and Droz (1998), and Wolf-Gladrow (2000).

The starting point is the LGCA micro dynamics (see Eq. [21.4]). The mass and momentum conservation of the collision operator can be expressed as

$$\sum_{i=1}^{b} \Delta_i(\mathbf{n}(\mathbf{r}, t)) = 0 \tag{21.9}$$

$$\sum_{i=1}^{b} \mathbf{c}_i \Delta_i(\mathbf{n}(\mathbf{r}, t)) = 0 \tag{21.10}$$

One can ask if the evolution equation (21.4) is also valid for the averaged particle densities f_i. It turns out that this is true, but only under the Boltzmann molecular chaos assumption, which states that particles that collide are not correlated before and after collisions, or, that for any number of particles k, $\langle n_1 n_2 \dots n_k \rangle = \langle n_1 \rangle \langle n_2 \rangle \cdots \langle n_k \rangle$. In this case, one can show that $\langle \Delta_i(\mathbf{n}) \rangle = \Delta_i(\mathbf{f})$, where $\mathbf{f}$ is the vector containing all f_i. By averaging Eq. (21.4) and applying the molecular chaos assumption we find

$$f_i(\mathbf{x} + \mathbf{c}_i, t + 1) - f_i(\mathbf{x}, t) = \Delta_i(\mathbf{f}(\mathbf{x}, t)) \tag{21.11}$$

A first-order Taylor expansion of $f_i(\mathbf{x} + \mathbf{c}_i, t + 1)$, substituted into Eq. (21.11), results in

$$\partial_t f_i(\mathbf{x}, t) + \partial_\alpha \mathbf{c}_{i\alpha} f_i(\mathbf{x}, t) = \Delta_i(\mathbf{N}(\mathbf{x}, t)) \tag{21.12}$$

Note that the shorthand ∂_t means $\partial/\partial t$; the subscript α denotes the α-component of a D-dimensional vector, where D is the dimension of the LGCA lattice; and we assume the Einstein summation convention over repeated Greek indices (e.g., in two dimensions $\partial_\alpha \mathbf{c}_{i\alpha} f_i(\mathbf{x}, t) = \partial_x \mathbf{c}_{ix} f_i(\mathbf{x}, t) + \partial_y \mathbf{c}_{iy} f_i(\mathbf{x}, t)$). Next, we sum Eq. (21.12) over the index i and apply Eq. (21.5), Eq. (21.9), and Eq. (21.10), thus arriving at $\partial_t \rho + \partial_\alpha \rho \mathbf{u} = 0$, or

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \tag{21.13}$$

which is just the equation of continuity that expresses conservation of mass in a compressible fluid. One can also first multiply Eq. (21.12) with $\mathbf{c}_i$ and then summate over the index i. In this case, we arrive at

$$\partial_t \rho \mathbf{u} + \partial_\beta \Pi_{\alpha\beta} = 0 \tag{21.14}$$

with

$$\Pi_{\alpha\beta} = \sum \mathbf{c}_{i\alpha} \mathbf{c}_{i\beta} f_i \tag{21.15}$$

The quantity $\Pi_{\alpha\beta}$ must be interpreted as the flow of the α-component of the momentum into the β-direction; $\Pi_{\alpha\beta}$ is the momentum density flux tensor. To proceed, one must be able to find expressions

for the particle densities f_i. This is a highly technical matter that is described in detail in, e.g., Frisch et al. (1987) or Rivet and Boon (2001). The bottom line is that one first calculates the particle densities for an LGCA in equilibrium, f_i^0, and then substitute them into Eq. (21.15). This results in an equation that is almost similar to the Euler equation, i.e., the expression of conservation of momentum for an inviscid fluid. Next, one proceeds by taking into account small deviations from equilibrium, resulting in viscous effects. Then, after a lengthy derivation one is able to derive the particle densities, substitute everything into Eq. (21.15) and derive the full expression for the momentum conservation of the LGCA, which very closely resembles the Navier–Stokes equations for an incompressible fluid. The viscosity and sound speed of the LGCA are determined by its exact nature (i.e., the lattice, the discrete velocities, and the exact definition of the collision operator).

We must stress that the derivations in this section are very loose, in the sense that we ignored many important details. For instance, the Taylor expansion, which resulted in Eq. (21.12), was only accurate up to first order. In fact one can show that an LGCA obeys the Navier–Stokes equations up to second order. Also, we introduced very loosely the concept of equilibrium distributions, and small deviations from equilibrium that give rise to viscous effects. By using a very powerful technique, known as the Chapmann–Enskog expansion, one is able to solve Eq. (21.11) and derive expressions for mass and momentum conservation of an LGCA, which turn out to be almost equal to the equations for a real, incompressible fluid.

To be complete, we note that the derivation of the Navier–Stokes equations for the LGCA is correct in the limit of small Mach and small Knudsen numbers. The first restriction means that the flow velocities must be much smaller than the sound speed of the LGCA, and the second limit demands that the mean free path of the particles must be much smaller than some macroscopic dimensions of the LGCA, i.e., the particle density cannot be too small.

Finally, we must mention one last technical detail. As stated above, the momentum conservation equations of the LGCA are almost equal to the Navier–Stokes equations of a real fluid. The difference lies in a factor $g(\rho)$ in the advection term (the $\mathbf{u} \cdot \nabla \mathbf{u}$ term in Eq. [21.8]), which leads to the breakdown of Galilean invariance. In the low velocity limit, however, this is not a real problem, because the fluid becomes incompressible and $g(\rho)$ a constant. A rescaling of the velocity and time with $g(\rho)$ allows to fully recover the exact Navier–Stokes equations for an incompressible fluid.

21.5.3 Simulating an LGCA

Although LGCA simulations can benefit from generic CA environments, there are a few typical aspects of LGCA that will be discussed here. First, most 2D LGCAs are defined on triangular lattices. Such lattice should be represented by some 2D array. To denote each node in the triangular lattice with coordinates (x,y) by integer numbers, we multiply the coordinates by a factor $(2, 2/\sqrt{3})$. To avoid awkward diamond-shaped grids representation, the streaming step is different for even and odd parity of the lattice (see Figure 21.9). This same mapping can be used for the 19 velocity GBL model.

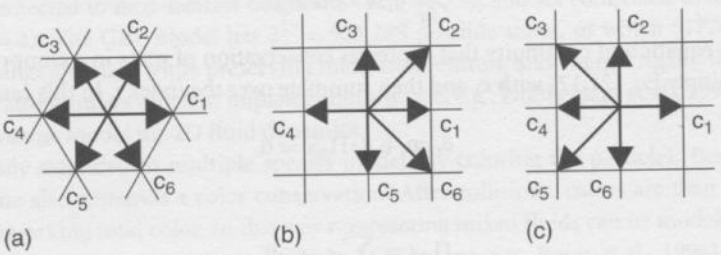


FIGURE 21.9 The mapping of the original triangular lattice (a) to a square domain. The mapping and streaming is based on the parity of the lattice. On even lines we propagate using (b) and, for odd lines we propagate using (c).

The state vector per node of an LGCA is a b-vector containing bits. So, the total LGCA lattice can be stored with $N \times b$ bits with N being the total number of nodes in the lattice. The streaming step can be accomplished by using two different lattice grids. The second lattice is used to calculate the new state at the next time step from the first lattice. After updating the lattice, we swap the pointers to the lattices. If memory size is a problem (e.g., large 3D problems), then it would be possible to do the streaming step in-place, at the cost of accessing each lattice point several times instead of just once.

The collision can be handled in two ways. If the collision is not very complicated (as with HPP or FHP-I), then it can be implemented as logical operations on the state bits. For instance, to implement the HPP collision operator (Eq. [21.3]), one could first define a selection operator S_i that returns TRUE if a head-on collision state is present. For instance, for channel 1 the selection operator S_1 would be

$$S_1 = (n_1 \text{ AND (NOT } n_2) \text{ AND } n_3 \text{ AND (NOT } n_4)) \text{ OR ((NOT } n_1) \text{ AND } n_2 \text{ AND (NOT } n_3) \text{ AND } n_4).$$

If S_1 returns 1, a collision occurs, and the state n_1 must flip (from 1 to 0 or vice versa). If S_1 returns 0, the state n_1 must stay the same. The postcollision state ($n_1 + \Delta_1(\mathbf{n})$) can be obtained by applying the exclusive OR operation on n_1 with S_1, i.e., $n_1 + \Delta_1(\mathbf{n}) = n_1$ XOR S_1.

This postcollision value is then streamed using the procedures as sketched above.

Another approach is to use lookup tables. If the collision becomes too complex to explicitly write down the Boolean expression (as with FCHC or GBL) this is the only possibility. However, one may also want to resort to lookup tables for HPP or FHP as this may be faster. We give an example of the use of lookup tables in the implementation of the GBL model (for details see Dubbeldam et al., 1999). More discussions on the lookup tables can be found in Rivet and Boon (2001). The first step is to group all 2^{19} states of GBL in equivalence classes with the same total mass, momentum, and energy. The collision then amounts to randomly selecting a state from the equivalence class to which the input state belongs. It is clear that the input state is also among them (meaning no collision when selected as output state), but since most equivalence classes are quite large this has little influence. For the 19-bits GBL model we create a collision table of 2^{19} indexes, followed by the equivalence classes (see Figure 21.10). Every index of an element in a class points to the start of the class (for instance, in Figure 21.5, 138, 273, and 41,024 all point to $2^{19} + X$). The left 12 bits are used to indicate the number of collision outcomes. If the number is zero, then the outcome is equal to the input state. Otherwise, the value of the right 20 bits is an index pointing to the start point of an array of possible postcollision states, of which we choose one at random (using the information on the size of the equivalence class).

To have solid boundaries or solid objects in the flow, one must be able to set boundary conditions. In LGCA this is almost trivial using the bounce-back rule, where particles that hit the boundary are reversed and sent back into the direction they came from. To start an LGCA simulation the Boolean field must be initialized. Usually one knows some initial values of the macroscopic fields (the density and the fluid velocity). Based on these values, the initial Boolean field must be computed. This is done by using the equilibrium distribution f_i^0 that is explicitly known as a function of ρ and $\mathbf{u}$ in the limit of small $\mathbf{u}$ (see Rivet and Boon, 2001, Chapter 4). This equilibrium distribution gives the average of the Boolean field in equilibrium (i.e., the probability that a particle is present in channel i), so, the actual field is computed from the equilibrium distribution using a random number generator that delivers random numbers between 0.0 and 1.0. If the random number is smaller than f_i^0, then n_i is initialized to 1, otherwise it is set to 0.

With the initialization and the boundary conditions in place, the LGCA simulation can be started. If the flow is driven by some pressure gradient, one can apply a body force that, after each collision, effectively adds some momentum to the nodes. This was done in the simulations presented in Figure 21.8. Finally, to extract the wanted macroscopic fields, an ensemble average must be computed. This is typically done using time- or space-averaging, or a combination of both. In Figure 21.8(b) time averaging was applied.

LGCA simulations have been executed on every type of computers, from desktop PCs to massively parallel supercomputers, and even dedicated CA and LGCA machines (including programmable FPGA hardware). On current state-of-the-art computers, one can easily simulate quite large 2D and 3D LGCAs (see Dubbeldam et al., 1999) for an example of running the GBL model on a parallel computer).

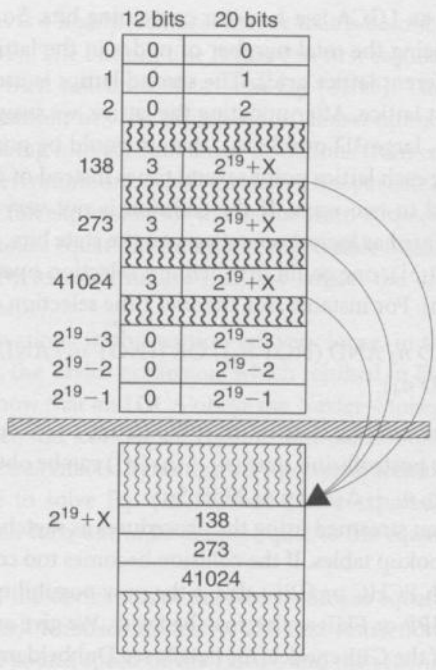


FIGURE 21.10 Collision table. The first 2^{19} indices are divided into 12 and 20 bits. The left 12 bits denote the number of collision outcomes, the rightmost 20 bits denote an index number from where the collision outcomes are stored in the table. These collision outcomes start from index 2^{19}. The figure shows an example for an equivalence class of size 3.

21.5.4 Some Applications

Lattice gases have been and are still being applied[6] to challenging problems of multiphase fluids in complex (e.g., porous) flow domains. These typically include fluid mixtures and colloids, reaction diffusions systems, immiscible fluids and phase separation, multiphase flows, fluids with free interfaces, magneto hydrodynamics, nonideal fluids, etc. Because of the ease to define boundary conditions using the bounce back rule, all such complex fluids have been studied in porous media. For details we refer to the LGCA books and references therein (Rivet and Boon, 2001; Rothman and Zaleski, 1997; Chopard and Droz, 1998).

21.5.5 Lattice Boltzmann Method

Immediately after the discovery of LGCA as a model for hydrodynamics in 1986, it was criticized on three points. LGCA have noisy dynamics, lack Galilean invariance, and have an exponential complexity of the collision operator. The noisy dynamics is clearly illustrated in Figure 21.8 and the lack of Galilean invariance was discussed above. Adding more velocities in an LGCA leads to increasingly more complex collision operators, exponentially in the number of particles (remember the numbers for GBL and FCHC). Therefore, another model, the lattice Boltzmann method (LBM), was introduced. This method is reviewed in detail in Succi (2001).

The basic idea is that one should not model the individual particles n_i, but immediately the particle densities f_i, i.e., one iterates the lattice Boltzmann equation (Eq. [21.11]). This means that particle densities are streamed from cell to cell, and particle densities collide, immediately solving the problem of noisy

[6] Although it must be said that the Lattice Boltzmann method, which will be mentioned in the next section, currently is the preferred method over LGCAs, although in special situations LGCAs are still to be preferred.

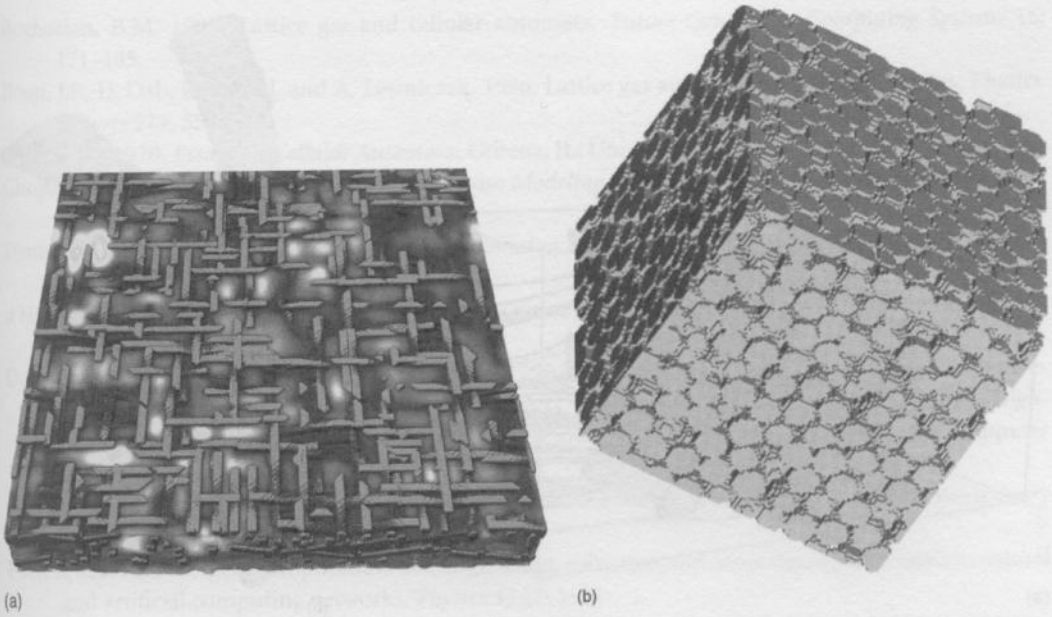


FIGURE 21.11 (a) Flow in a porous fiber mat; and (b) a porous medium composed of a dense random packing of spheres.

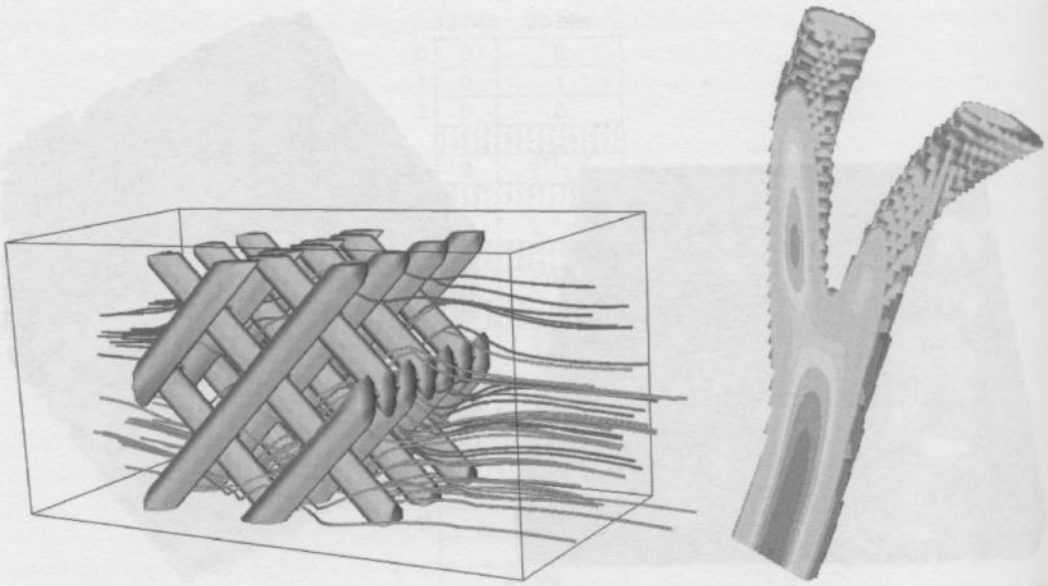
dynamics. However, in a strict sense, we no longer have a CA with a Boolean state vector (in fact, the state is now a vector of real numbers, so the state space per node is infinite). However, we can view LBM as a generalized CA. By a clever choice of the equilibrium distributions f_i^0, the model becomes isotropic and Galilean invariant, thus solving the second problem of LGCA. Finally, a very simple collision operator can be introduced. This so-called BGK collision operator models the collisions as a single-time relaxation τ, toward equilibrium, i.e.,

$$\Delta_i^{\mathrm{BGK}}(\mathbf{N}) = -\frac{1}{\tau}\left(f_i - f_i^0\right) \tag{21.16}$$

Eq (21.11) and Eq. (21.16) together with a definition of the equilibrium distributions result in the Lattice-BGK (L-BGK) model. The L-BGK model leads to correct hydrodynamic behavior in two and three dimensions. The L-BGK not only applies to the triangular lattice, but also correctly works for other lattices, e.g., 2- or 3D cubical lattices with nearest and next-nearest neighbor interactions. The LBM and especially the L-BGK have found widespread use in simulations of highly complex fluid dynamical problems, including turbulence, multiphase flows, spinal decomposition, etc. To get a flavor we refer to Succi (2001). Below we present a few examples of successful L-BGK applications. All these L-BGK simulations were routinely executed on parallel computers, using a highly efficient parallel implementation of the L-BGK method (Kandhai et al., 1998).

The L-BGK method has played a significant role in studying flow in porous media. In Figure 21.11(a), we show an example of flow in a porous medium that consists of randomly placed fibers. This fiber mat is a model system for paper. Through L-BGK simulations, one can compute the permeability of such fiber mats, as a function of the porosity (Kopenen, et al., 1998). The results showed a remarkable agreement with experimental results. Since the permeability could be studied over a very large range of volume fractions, much larger than accessible through experiments, the authors were able to check the quality of theoretical approximations for this particular problem.

Another example of successful L-BGK simulations of flow in porous media was the transient dispersion in homogeneous porous media. Figure 21.11(b) shows an example of a medium of a random distribution



(a) (b)

FIGURE 21.12 (a) Flow in a static mixer reactor; and (b) flow in the lower abdominal aortic bifurcation at peak systole.

of densely packed spheres. Here the flow around the spheres is computed, as well as advection-diffusion of tracer particles in the flow, and diffusion of the tracer particles through a microporous structure of the spheres. From these simulations, the authors could derive transient tracer dispersion curves, which showed a very good agreement with propagators that were measured using nuclear magnetic resonance (Kandhai et al., 2002a, 2002b).

L-BGK simulations have been compared extensively with traditional computational fluid dynamics (CFD) codes. As an example, we mention flow through a static mixer reactor (see Figure 21.12[a]). The flow fields as computed with L-BGK were in very good agreement with results stemming from a commercial CFD package (FLUENT). The computed fluid flux through the system, as a function of the applied pressure gradient, was in very good agreement with experimental data (Kandhai et al., 1999). It turned out that preparing the computational grid for L-BGK is trivial (just a Cartesian grid), whereas creating a body-fitted finite element grid for this geometry is very tedious.

As a final example we mention the application of L-BGK for simulation of blood flow. In Figure 21.12(b), we show the results of L-BGK simulations of time periodic systolic flow through the bifurcation of the lower abdominal aorta. The geometry is taken from MRI images of real patients. The results are in good agreement with previously published studies of systolic flow in region of the arterial tree (Artoli et al., 2005).

References

Artoli, A.M.M., A.G. Hoekstra, and P.M.A. Sloot. 2005. Mesoscopic simulations of systolic flow in the human abdominal aorta. *Journal of Biomechanics* 39: 873–884.

Automata. 2005. *11th Workshop on Cellular Automata*, Gdansk, 3–5 September 2005 http://iftia9.univ.gda.pl/~CA2005/

Bandini, S. 2002. Special issue on cellular automata. *Future Generation Computing Systems* 18: 871–1004.

Berec, L. 2002. Techniques of spatially explicit individual-based models: Construction, simulation and mean-field analysis. *Ecological Modelling* 150: 55–81.

Boghosian, B.M. 1999. Lattice gas and cellular automata. *Future Generation Computing Systems* 16: 171–185.

Boon, J.P., D. Dab, R. Kapral, and A. Lawniczak. 1996. Lattice gas automata for reactive systems. *Physics. Reports* 273: 556–647.

Burks, A.W. 1970. *Essays on Cellular Automata*. Urbana, IL: University of Illinois Press.

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

Deutsch, A. and S. Dormann. 2004. *Cellular Automaton Modeling of Biological Pattern Formation*. Basel: Birkhauser.

d'Humières, D., Lallemand, P., and Frisch, U. 1986. Lattice gas models for 3-D hydrodynamics. *Europhysics Letters* 2: 291–297.

Dubbeldam, D., A.G. Hoekstra, and P.M.A. Sloot. 1999. Computational aspects of multi-species lattice-gas automata. In P.M.A. Sloot, M.T. Bubak, A.G. Hoekstra, and L.O. Hertzberger (Eds.), *High-Performance Computing and Networking*, Amsterdam, The Netherlands, Lecture Notes in Computer Science, 1593: 339–349. Berlin: Springer-Verlag.

Farmer, D., T. Toffoli, and S. Wolfram. 1984. *Cellular Automata: Proceedings of an Interdisciplinary Workshop*. Amsterdam: North-Holland.

Forrest, S. 1990. Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks. *Physica D* 42: 1–11.

Frisch, U., D. d'Humières, B. Hasslacher, P. Lallemand, P. Pomeau, and J.-P. Rivet. 1987. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems* 1: 649–707.

Frisch, U., B. Hasslacher, and Y. Pomeau. 1986. Lattice-gas automata for the Navier–Stokes Equation. *Physical Review Letters* 56: 1505–1508.

Ganguly, N., B.K. Sikdar, A. Deutsch, G. Canright, and P.P. Chaudhuri. 2003. A survey on cellular automata. Technical Report, Centre for High Performance Computing, Dresden University of Technology. http://www.cs.unibo.it/bison/publications/CAsurvey.pdf

Grosfils, P., J.P. Boon, and P. Lallemand. 1992. Spontaneous fluctuation correlations in thermal lattice gas automata. *Physical Review Letters* 68: 1077–1080.

Gutowitz, H.A. 1990. *Cellular Automata*. Cambridge, MA: MIT Press.

Hardy, P., O. de Pazzis, and Y. Pomeau. 1976. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Physical Review A* 13: 1949–1961.

Hardy, P. and Y. Pomeau. 1972. Thermodynamics and hydrodynamics for a modeled fluid. *Journal of Mathematical Physics* 13: 1042–1051.

Hardy, P., Y. Pomeau, and O. de Pazzis. 1973. Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions. *Journal of Mathematical Physics* 14: 1746–1759.

Hénon, M. 1987. Isometric collision rules for the 4-D FCHC lattice gas. *Complex Systems* 1: 475–494.

Ilachinski, A. 2001. *Cellular Automata, a Discrete Universe*. Singapore: World Scientific.

Jesshope, C., V. Jossifov, and W. Wilhelmi. 1994. *International Workshop on Parallel Processing by Cellular Automata and Arrays* (Parcella'94). Berlin: Akademie Verlag.

Kaandorp, J.A., C.P. Lowe, D. Frenkel, and P.M.A. Sloot. 1996. The effect of nutrient diffusion and flow on coral morphology. *Physical Review Letters* 77: 2328–2331.

Kadanoff, L.P. and J. Swift. 1968. Transport coefficients near the critical point: A master-equation approach. *Physical Review* 165: 310–322.

Kandhai, B.D., D. Hlushkou, A.G. Hoekstra, P.M.A. Sloot, H. van As, and U. Tallarek. 2002a. Influence of stagnant zones on transient and asymptotic dispersion in macroscopically homogeneous porous media. *Physical Review Letters* 88: 234501.

Kandhai, B.D., A. Koponen, A.G. Hoekstra, M. Kataja, J. Timonen, and P.M.A. Sloot. 1998. Lattice Boltzmann hydrodynamics on parallel systems. Computer Physics Communications 111: 14–26.

Kandhai, B.D., U. Tallarek, D. Hlushkou, A.G. Hoekstra, P.M.A. Sloot, and H. van As. 2002b. Numerical simulation and measurement of liquid hold-up in biporous media containing discrete stagnant zones. *Philosophical Transactions: Mathematical, Physical & Engineering Sciences* 360: 521–534.

Kandhai, B.D., D.J.-E. Vidal, A.G. Hoekstra, H.C.J. Hoefsloot, P. Iedema, and P.M.A. Sloot. 1999. Lattice-Boltzmann and finite element simulations of fluid flow in a SMRX mixer. *International Journal for Numerical Methods Fluids* 31: 1019–1033.

Koponen, A., D.B. Kandhai, E. Héllen, M. Alava, A.G. Hoekstra, M. Kataja, K. Niskanen, P.M.A. Sloot, and J. Timonen. 1998. Permeability of three-dimensional random fibre webs. *Physical Review Letters* 80: 716–719.

Kurzweil, R. 2002. Reflections on Stephen Wolfram's 'A New Kind of Science'. http://www.kurzweilai.net/meme/frame.html?main=/articles/art0464.html

May, R.M. 1976. Simple mathematical models with very complicated dynamics. *Nature* 261: 459–467.

Mitchell, M. 1998. Computation in cellular automata: A selected review. In T. Gramss, S. Bornholdt, M. Gross, M. Mitchell, and T. Pellizzari (Eds.), *Nonstandard Computation*, pp. 95–140. Weinheim: VCH Verlagsgesellschaft.

Naumov, L. 2004. CAMEL—Cellular Automata Modeling Environment & Library. Cellular Automata. 6th International Conference on Cellular Automata for Research and Industry. *Lecture Notes in Computer Science*, 3305: 735–744. Heidelberg: Springer.

NKS Conference. 2005. http://www.cs.indiana.edu/~dgerman/2005midwestNKSconference/index.html

Rivet, J.-P. and J.P. Boon. 2001. *Lattice Gas Hydrodynamics*. Cambridge: Cambridge University Press.

Rothman, D.H. and S. Zaleski. 1997. *Lattice-Gas Cellular Automata, Simple Models of Complex Hydrodynamics*. Cambridge: Cambridge University Press.

Sloot, P.M.A. 1999. High performance simulation with cellular automata. In N. Piskunov (Ed.), Proceedings of HiPer'99, pp. 169–207. Computer Centre University of Tromso, Tromso, Sweden.

Sloot, P.M.A., F. Chen, and C.A. Boucher. 2002. Cellular automata model of drug therapy for HIV infection. *Lecture Notes in Computer Science* 2493: 282–293.

Sloot, P.M.A., B. Chopard, and A.G. Hoekstra. 2004. Cellular Automata: 6th International Conference on Cellular Automata for Research and Industry, ACRI 2004, Amsterdam, The Netherlands, October 2004. *Lecture Notes in Computer Science*, 3305. Heidelberg: Springer.

Sloot, P.M.A. and A.G. Hoekstra. 2001. Cellular automata as a mesoscopic approach to model and simulate complex systems. *Lecture Notes in Computer Science*, 2074 (II): 518–527. Heidelberg: Springer.

Sloot, P.M.A., J.A. Kaandorp, A.G. Hoekstra, and B.J. Overeinder. 2001a. Distributed cellular automata: Large scale simulation of natural Phenomena. In A.Y. Zomaya, F. Ercal, and S. Olariu (Eds.), *Solutions to Parallel and Distributed Computing Problems: Lessons from Biological Sciences*, pp. 1–46. Wiley: New York.

Sloot, P.M.A., B.J. Overeinder, and A. Schoneveld. 2001b. Self-organized criticality in simulated correlated systems. *Computer Physics Communications* 142: 76–81.

Sloot, P.M.A., A. Schoneveld, J.F. de Ronde, and J.A. Kaandorp. 1997. Large scale simulations of complex systems Part I: Conceptual framework. SFI Working Paper: 97-07-070, *Santa Fe Institute for Complex Studies*.

Succi, S. 2001. *The Lattice Boltzmann Equation, for Fluid Dynamics and Beyond*. Oxford: Clarendon Press.

Turing, A.M. 1936. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, Ser. 2, Vol. 42; and extract On computable numbers, with an application to the entscheidungsproblem. A correction. *Proceedings of the London Mathematical Society*, Ser. 2, Vol. 43, 1937.

Ulam, S. 1952. Random processes and transformations. *Proceedings of the International Congress on Mathematics* 2: 264–275.

Ulam, S. 1962. On some mathematical problems connected with patterns of growth of figures. *Proceedings of Symposia in Applied Mathematics*, Vol. 14, pp. 215–224. Providence: American Mathematical Society.

von Neumann, J. 1951. The general and logical theory of automata. In L.A. Jeffress (Ed.), *Cerebral Mechanisms and Behavior: The Hixon Symposium*, pp. 1–41. Wiley: New York.

von Neumann, J. 1966. *Theory of Self-Reproducing Automata*. Urbana, IL: University of Illinois Press.

Wiener, N. and Rosenbleuth, A. 1946. The mathematical formulation of the problem of conduction of impulses in a network of connected excitable elements, specifically in cardiac muscle. *Archivos del Instituto de Cardiologica de Mexico* 16: 202–265.

Wolf-Gladrow, D.A. 2000. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models.* Heidelberg: Springer

Wolfram, S. 1986a. *Theory and Applications of Cellular Automata.* Singapore: World Scientific.

Wolfram, S. 1986b. *Journal of Statistical Physics* 45: 471–526.

Wolfram, S. 2002. *A New Kind of Science.* Champaign, IL: Wolfram Media.

Zuse, K. 1982. The computing universe. *International Journal of Theoretical Physics* 21: 589–600.

Zuse, K. 1967. Rechnender Raum. *Elektronische Datenverarbeitung* 8: 336–344. See also: http://www.idsia.ch/~juergen/digitalphysics.html