Chapter 5

Cellular Automata

Literature

- J. VON NEUMANN, *The Theory of Self-reproducing Automata*, in: *Essays of Cellular Automata*, ed.: A.W. BURKS, University of Illinois Press, Urbana (1970).
- J. SCHMELZER, G. RÖPKE, and R. MAHNKE, *Aggregation Phenomena in Complex Systems: Principles and Applications*, J. Wiley, 1999 + References.
- D. STAUFFER and AMONON AHARONY, *Perkolationstheorie*, VCH, 1995.
- S. WOLFRAM, *Universality and Complexity in Cellular Automata* in *Cellular Automata*, ed.: FARMER *et al.* Proceedings of an Interdisplinary Workshop, Physica D **10** (1984).
- S. WOLFRAM, *Theory and Application of Cellular Automata*, World Scientific, Singapore (1986).

5.1 Dynamics of Complex Systems

The ubiquitous strategy in the description of complex physical systems consists in modeling the system by disassembling it into small and tractable parts (key elements) which may interact in one way or other. Such a simplified procedure has been the key success factor of natural sciences for several centuries but it is nonetheless an approximation. Complex systems which have been addressed successfully by disintegration are

- ISING, HEISENBERG model for magnetism,

- VAN DER WAALS model for fluids,
- Spring models for lattice vibrations,
- LANDAU theory for phase transitions.

An alternative approach are cellular automata (CA). They have been invoked to study

- Chemical reactions,
- Phase separation,
- Self criticality,
- Avalanches,
- Models to study chaos in physical systems.

Cellular automata are mathematical models of discrete dynamical systems whose behavior is completely specified in terms of a local relation. A cellular automaton can be thought of as a stylized universe. Space is represented by a uniform grid, with each cell containing a few bits of data; time advances in discrete steps and the laws of the 'universe' are expressed in, say, a small look-up table, through which at each step each cell computes its new state from that of its close neighbors. Thus, the system's laws are *local* and *uniform*.

The first cellular automaton was conceived by the mathematician J. VON NEUMANN in the late 1940's. He constructed an automaton which qualified as an universal computer (property of *universality*: the computer can emulate any desirable function of any other machine by the use of a set of logical rules, and, particularly, it can reproduce itself); it was very complex and consisted of two hundred thousand cells in any of 29 states. J.H. CONWAY suspected that a cellular automaton with universal computing capabilities might be simpler. The key to this simplicity would be the rules that dictated survival, birth, and death. CONWAY's cellular automaton had only two states: a cell would be either filled or empty, 'alive' or 'dead'. . . , and so the *Game of Life* began.

Game of Life

It is a simple 2D-system which is supposed to describe the basic processes of living systems. It describes the deterministic changes within a 2D-array with time under the restriction that the state of one cell depends only on the states of the surrounding eight neighbors. CONWAY had the following objectives:

- He wanted to make sure that no simple pattern would obviously grow without limit. It should not be easy to prove that any simple pattern grows for ever.
- He wanted to ensure, nevertheless, that some simple pattern do grow widely. There should be patterns that look like they might grow forever.
- There should be simple patterns that evolve for a long time before stabilizing. A pattern is stabilized by either vanishing completely or by producing a constellation of stable objects.

There are three rules:

- A dead cell with exactly three living neighbors becomes a living cell (birth).
- A living cell with two or three living neighbors stays alive (survival).
- In all other cases, a cell dies or remains dead (overcrowding or loneliness).

The game is played on a 2D grid of square cells (like a chessboard) extending infinitely in every direction or on a finite grid with with periodic boundary conditions. A cell can be alive or dead and a living cell is represented by a marker. A dead cell is empty. Each cell on the grid has a neighborhood consisting of the eight cells in every direction (including diagonals).

Algorithm 10 MATLAB Code: Game of Life

```
L = 20;
A = unidrnd(2,[L,L])-1;
            % initial configuration
i1 = [2:L, 1]; i2 = [L, 1:L-1];
            % n.n. tableaux with pbc
while (1)
  N = A(i1,:) + A(i2,:) + A(:,i1) + A(:,i2) + ...
  A(i1,i1) + A(i1,i2) +A(i2,i1) + A(i2,i2);
  A = N==3 | (N==2 & A==1);
  spy(A);
  pause(0.25);
end
```

5.2 One Dimensional Cellular Automata

Cellular automata are useful to analyze and understand the laws that govern complex phenomena. In natural sciences, non-linear phenomena are often described by reaction-diffusion equations. Non-linear space-time dynamics of interacting particles, chemical or biological systems can generate numerous local and non local effects far from equilibrium such as steady-state multiplicity, oscillations like limit cycles, propagating fronts, target patterns, spiral waves, pulses as well as stationary spatial patterns. In 1D these processes can be described generally by

$$\frac{\partial}{\partial t}\rho(r,t) = f[\rho(r,t)] + \frac{\partial^2}{\partial r^2}\rho(r,t),$$

where f stands for the reaction function. They are suitable for continuous systems with not too many degrees of freedom.

An alternate or complementary approach in discrete space-time is provided by 1D cellular automata. They are the most simple case: a 1D array of cells which take on the values zero and one. In each time step the values of each cell are modified according to some simple rules, the *neighborhood relations* or *evolution rules* and the *update rules*.

A simple example:

- States: zero and one.
- Neighborhood: the two neighbor cells: NCN with C the cell under observation.
- Update rules: we have three relevant cells which can be in two possible states. Thus, we need $2^3 = 8$ rules:

$$\begin{array}{ccccc}
0 & 0 & 0 & \to & 0 \\
0 & 0 & 1 & \to & 1 \\
0 & 1 & 0 & \to & 1 \\
0 & 1 & 1 & \to & 0
\end{array}
\qquad\qquad
\begin{array}{ccccc}
1 & 0 & 0 & \to & 1 \\
1 & 0 & 1 & \to & 1 \\
1 & 1 & 0 & \to & 0 \\
1 & 1 & 1 & \to & 0.
\end{array}$$

We start with a configuration of only one cell having the value one and find the following time series:

$$\begin{array}{lccccccccccccccc}
\text{Time } 0: & . & . & . & . & . & . & . & 1 & . & . & . & . & . & . & . \\
\text{Time } 1: & . & . & . & . & . & . & 1 & 1 & 1 & . & . & . & . & . & . \\
\text{Time } 2: & . & . & . & . & . & 1 & . & . & . & 1 & . & . & . & . & . \\
\text{Time } 3: & . & . & . & . & 1 & 1 & 1 & . & 1 & 1 & 1 & . & . & . & . \\
\text{Time } 4: & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . \\
\text{Time } 5: & . & . & 1 & 1 & 1 & . & 1 & 1 & 1 & . & 1 & 1 & 1 & . & . \\
\text{Time } 6: & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & .
\end{array}$$

with the zeros replaced by dots. This very primitive CA shows already signs of *self organization*, *i.e.:* certain patterns evolve in the time series. The result corresponds closely to the result shown in Fig. 5.1 for the (01111000) rule.

S. WOLFRAM studied these linear CAs quite extensively and introduced the following fundamental rules for 1D-CAs:

1. The CA consists of discrete lattice points,
2. The CA develops in discrete time steps,
3. Each cell can take on a finite number of states (values),
4. The value of each cell develops according to one and the same updating rule,
5. This rule depends solely upon the states of the nearest neighbor cells.

Furthermore, he was able to establish the existence of four classes:

- **Class 1:** After a finite number of steps, a homogeneous general final state is reached (all cells are either dead or alive).
- **Class 2:** Initially simple local patterns evolve, they transform sometimes in vertical stripes. Frozen configurations are found where initial activity ceased and stable structures reigned.
- **Class 3:** The states distribute seemingly without rules but now and then typical patterns can be observed.
- **Class 4:** This class displays behavior which is not disordered, but complex, and some times long-lived. CAs of this class are capable of propagating information and this class contains universal automata. The processes depend heavily on the initial conditions.

A CA can be viewed as an idealization of the reaction-diffusion equation for discrete time where the state ρ is mapped on a finite set of possible values only. CAs are in general deterministic, but stochastic forces can be included as well (*stochastic cellular automata*). Such systems can exhibit phase transitions as function of the noise level. Updating rules are employed either sequentially or in parallel.

A very simple binary CA consists of a one dimensional lattice of cells. The cell at site i has a value $a_i(t)$ at time t which is either zero or one. A possible local updating rule is

$$a_i(t+1) = [a_{i-1}(t) + a_{i+1}(t)] \,\mathrm{mod} 2.$$

Starting from a particular initial state $\mathbf{a}(t)$ the system develops in time.

Deterministic 1D Local Binary CA

The most general deterministic 1D local binary CA is defined by the mapping

$$a_i(t+1) = F\left[a_{i-r}(t), \ldots, a_i(t), \ldots, a_{i+r}(t)\right], \quad a_i \in \{0,1\},$$

with r being the range of the local generation rule. The boundary conditions are still at our disposal. Let us study the most simple CA, namely for $r = 1$, in some detail. The evolution rule reads in this case

$$a_i(t+1) = F\left[a_{i-1}(t), a_i(t), a_{i+1}(t)\right], \quad a_i \in \{0,1\}.$$

Since there are $2^3 = 8$ possible input values, we may construct in general $2^8 = 256$ different CA rules specified by

$$\begin{aligned}
F(0,0,0) &= f_0 \\
F(0,0,1) &= f_1 \\
F(0,1,0) &= f_2 \\
F(0,1,1) &= f_3 \\
F(1,0,0) &= f_4 \\
F(1,0,1) &= f_5 \\
F(1,1,0) &= f_6 \\
F(1,1,1) &= f_7.
\end{aligned}$$

The vector $(f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7)$ defines the local rule. Three examples are given below.

Algorithm 11 MATLAB Code: 1D Binary CA

```
N = 601;
M = 300;
A = zeros(1,N); A((N+1)/2) = 1;
B = zeros(M,N); B(1,:) = A;
rule = [0 1 1 1 1 1 1 0];
ii = [1: N]';
ind = [(ii-1) ii (ii+1)];
ind(1,1) = N; ind(N,3) = 1;
for l = 2: M
  A = rule(bi2de(A(ind))+1);
  B(l,:) = A;
end
spy(B);
```

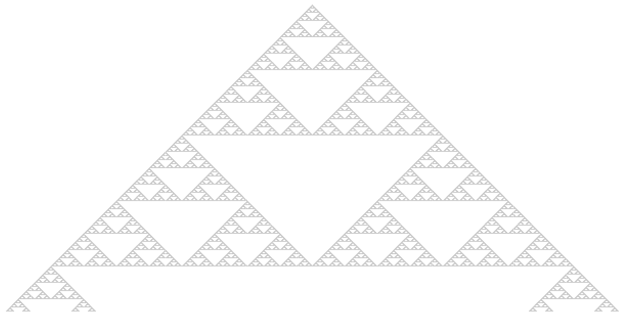

Figure 5.1: Cellular automata for rule (01111000).

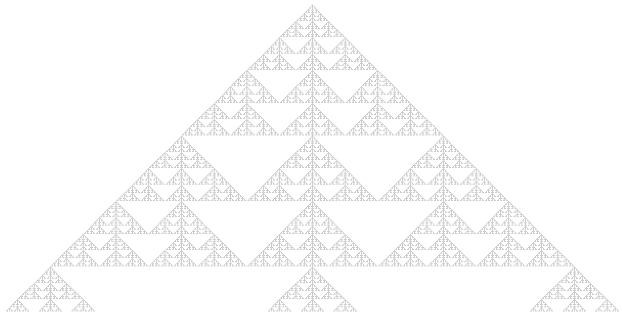

Figure 5.2: Cellular automata for rule (01101001).

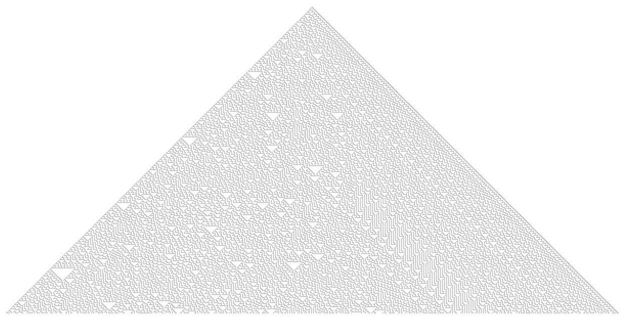

Figure 5.3: Cellular automata for rule (01101010) with chaotic behavior on the lhs and fairly regular structures on the rhs.

Figures 5.1 and 5.2 display typical fractal structures, which will be discussed later on. They also show signs of *self organization*. Obviously, Fig. 5.3 is the most complex. It is a standard example for a chaotic structure created by deterministic local CA. The analysis of the structure of the generated objects is called complexity problem. KOLMOGOROV coined the notion complexity as measure for the expense or length of all inner structures, which are necessary to generate, construct or describe the object.

5.3 Two-dimensional Cellular Automata

The generalization to 2D CAs is straight forward. The evolution rule for nearest neighbor coupling reads in this case

$$a_{i,j}(t+1) = F\left[a_{i,j-1}(t), a_{i,j}(t), a_{i,j+1}(t), a_{i-1,j}(t), a_{i+1,j}(t)\right].$$

Since there are 4 nearest neighbors on a square lattice, there are $2^{2^4} = 65536$ possible update rules. The number of rules grows rapidly with the number of neighbors and with the spatial dimension. PACKARD and WOLFRAM proposed to resort to random sampling with the expectation that the selected rules are representative.

5.4 Applications

5.4.1 Traffic Flow Studied by CA Models

Recent simulations of traffic flow based on CA have gained considerable importance. Here we will show that rather simple stochastic Cellular Automata can reproduce features of real traffic including jamming transition from low density laminar flow to high density congested flow, where stop-and-go waves are dominant.

Modern versions of CA for traffic flow are called particle hopping models. Studies of traffic flow based on CA trace back to the year 1956 but they received wide attention in 1992 with papers by SCHRECKENBERG. The 1D road is represented by a string of cells, which are either occupied by exactly one car or empty. If all cars are updated simultaneously (parallel update), then the particle hopping model is formally analogous to a 1D CA described before. NAGEL and SCHRECKENBERG introduced the following CA for 1D traffic with pbc (circular lane) called stochastic traffic cellular automata (STCA). In this model, the space coordinate i of the road, the time t and the velocity $v_i(t)$ are discrete variables. The length of the road is L. Each particle (the total number N is fixed) can have an integer velocity v between 0 and some maximum velocity v_{max}. A configuration is characterized by the position of the particles (cars) and their respective velocities. Each cell has a value v between -1 and v_{max}, where $v = -1$ represents an empty cell (no car), while other values indicate the presence of a car with velocity v.

The update rule consists of four consecutive steps:

1. *Acceleration*: If the velocity v is lower than $v_{\max}$ and if the distance to the next car (# of empty sites ahead) is larger than $v+1$, the speed is increased by one ($v \to v+1$).

2. *Slowing down*: Let d be the distance to the next car ahead. If $d \leq v$, then the speed is reduced to $d-1$ ($v \to d-1$) (No accidents!).

3. *Stochastic breaking*: With probability p_{sd} the velocity of a moving vehicle is reduced by one ($v \to v-1$).

4. *Propagation*: Each car proceeds by the value of its velocity v.

The dynamics of the traffic flow is determined by the density

$$\rho = \frac{\#\text{cars}}{\#\text{sites}},$$

by the initial configuration, and by the probability p.

The traffic flow consists of undisturbed motion with more or less constant velocity (laminar flow), at low densities. We find car clusters (small jams) at high densities, which are formed randomly due to fluctuations of velocity. One can also observe the phenomena 'congestion from nowhere'.

Non-interacting Cars

We start out with very low traffic density, which corresponds to isolated cars. The motion describes a random walk in which with probability $q = 1 - p_{sd}$ the speed increases by one and with probability $p_{sd} = 1 - q$ the car retains its speed. In N steps the mean number of acceleration steps is Nq. Starting with velocity zero, it takes on average $N = \frac{v_{\max}}{q}$ steps until the car reaches the maximum speed allowed. After this 'equilibration' phase in which a car reached the maximum speed the behavior is fairly simple. Starting from $v = v_{\max}$ it retains $v_{\max}$ with probability q, while the speed is reduced to $v_{\max} - 1$ with probability p_{sd}. In case it starts from velocity $v_{\max} - 1$ it increases its speed to $v_{\max}$ with probability q and keeps the velocity $v_{\max} - 1$ with probability p_{sd}. It can, however, never reduce its speed to $v_{\max} - 2$ or lower, once it has reached $v \geq v_{\max} - 1$. Hence, after the equilibration phase, the probability for $v_{\max}$ is $1 - p_{sd}$ and the probability for $v_{\max} - 1$ is p_{sd}. The average speed for 'non-interacting' cars is therefore

$$\langle v \rangle = v_{\max}(1 - p_{sd}) + (v_{\max} - 1)p_{sd} = v_{\max} - p_{sd}.$$

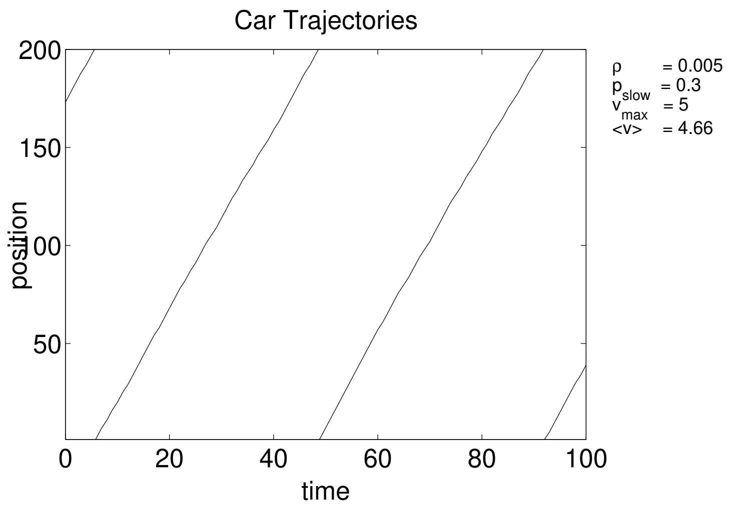

Figure 5.4: One car is on the road. It slows down stochastically with $p = 0.3$ and has a maximum speed $v = 5$. The average speed $\langle v \rangle = 4.66$ agrees with the general formula $\langle v \rangle = v_{\max} - p$.

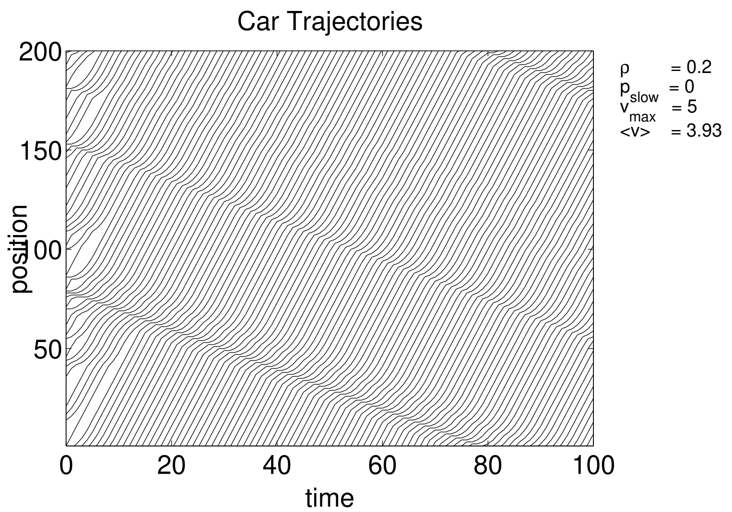

Figure 5.5: Parameters are $\rho = 0.2$, $p_{sd} = 0$, and $v_{\max} = 5$. The average speed is reduced to $\langle v \rangle = 3.93$ due to the presence of other cars. Apart from occasional minor slowing downs, the traffic is still laminar.

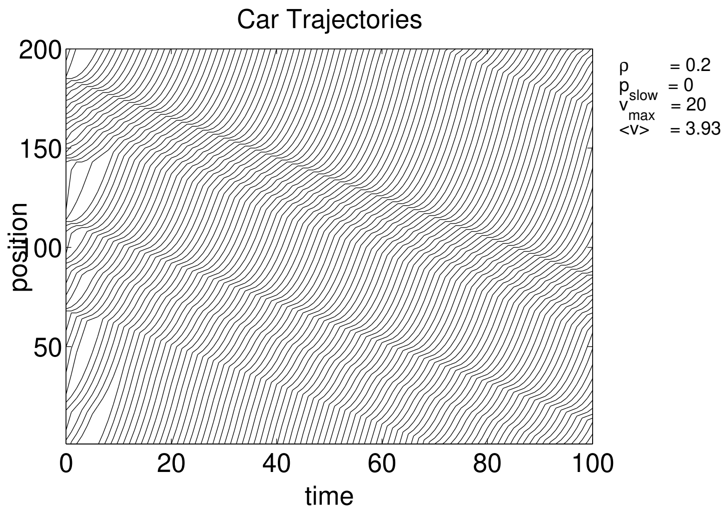

Figure 5.6: Parameters are $\rho = 0.2$, $p_{sd} = 0$, and $v_{\max} = 20$. Although the maximally allowed speed has been increased significantly, the traffic flow looks very similar to that of the previous figure. It is laminar with an average speed of $\langle v \rangle = 3.93$.

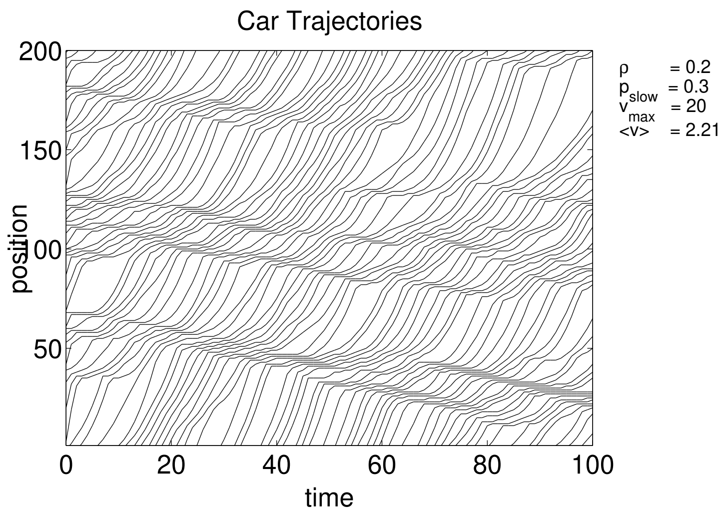

Figure 5.7: Parameters are $\rho = 0.2$, $p_{sd} = 0.3$, and $v_{\max} = 20$. The stochastic slowing down leads to an erratic traffic flow with congestion and a reduced average speed of $\langle v \rangle = 2.21$.

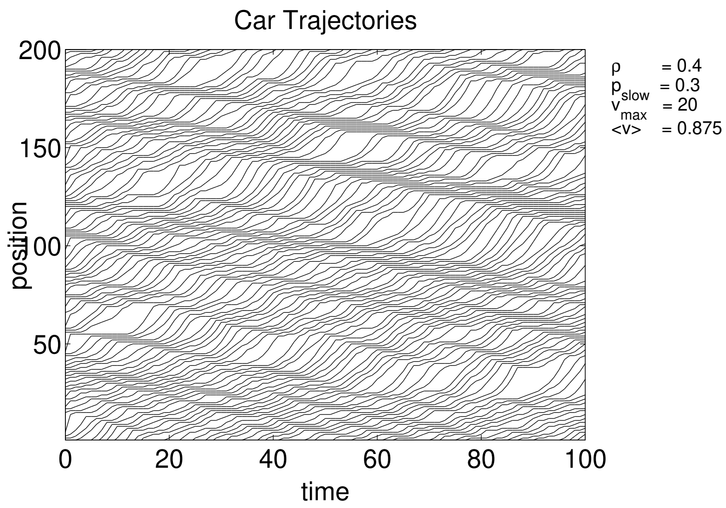

Figure 5.8: Parameters are $\rho = 0.4$, $p_{sd} = 0.3$, and $v_{\max} = 20$. A further increase of the traffic density leads to 'stop-and-go' waves and the average speed is merely $\langle v \rangle = 0.88$.

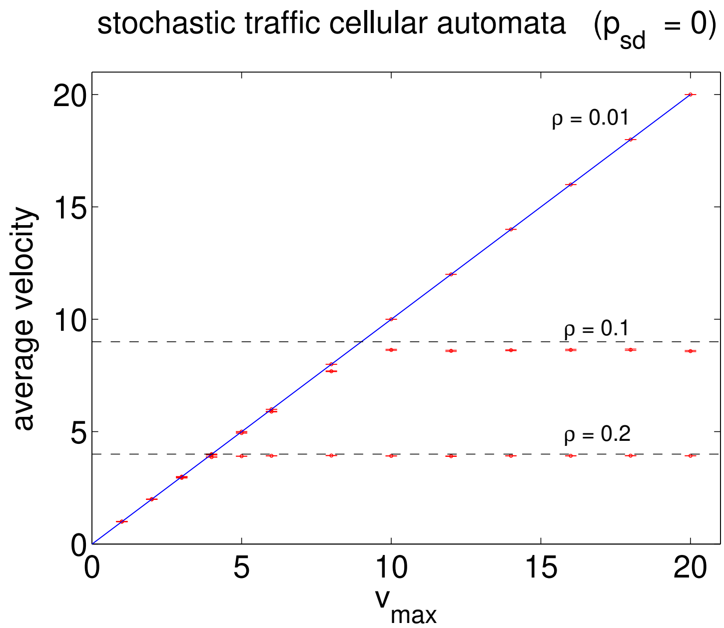

Figure 5.9: Average speed versus $v_{\max}$ for different densities without stochastic slowing down ($p_{sd} = 0$). The solid line shows the theoretical result for non-interacting cars.

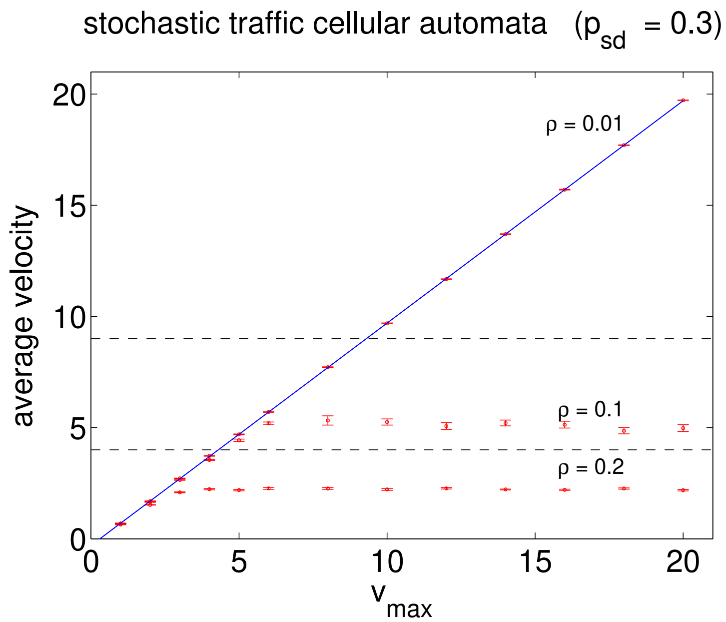

Figure 5.10: Average speed versus v_{max} for different densities with stochastic slowing down ($p_{sd} = 0.3$). The solid line shows the theoretical result for non-interacting cars.

Figure 5.9 depicts the average velocity as a functions of v_{max} in the absence of stochastic slowing down. In the low density limit the results agree with those for non-interacting cars. Since there is no stochastic slowing down, the cars drive with maximum speed. Also for higher densities the cars drive with the maximum speed allowed as long as v_{max} is small. Beyond a certain value v^*_{max} the average speed levels off at about v^*_{max}. With increasing density the critical velocity v^*_{max} decreases. The behavior can be understood easily. If N_c cars are arranged equidistantly along the road of length L, then the distance between cars is $d = \frac{L}{N_c} = \frac{1}{\rho}$. According to the STCA rules, the speed is limited to $d - 1$, which is the number of empty cells between successive cars. Hence,

$$v^*_{\text{max}} \simeq d - 1 = \frac{1-\rho}{\rho}.$$

In figure 5.10 stochastic slowing down is included. We observe a similar overall behavior, merely the critical velocities are reduced.

5.4.2 Forest Fires

We use a strongly simplified model to simulate forest fires. Starting from a two-dimensional simple cubic arrangement of cells with periodic boundary conditions, we place trees with occupation probability p in each cell.

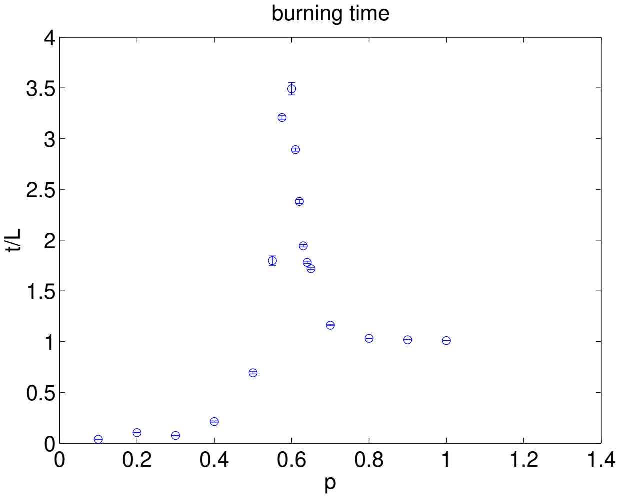

Figure 5.11: Dependence of the burning time on probability p. System size 200×200

The interpretation is either literally: occupied cells contain trees and empty cells do not, or occupied cells contain trees which catch fire and empty cells are such that even if there is a tree it does not catch fire when neighboring trees are burning.

The rules of the CA are as follows.

- Initialize the forest $f(x, y)$.
  - Vital tree $f = 1$
  - Empty cell $f = 4$.
  - Burning tree $f = 2$.
  - Burnt-out tree $f = 3$.
- Light the fire in the center
- Update rule
  - A vital tree ($f = 1$) catches fire if any of its nearest neighbors burns $f = 2$.
  - Trees burn one time step, afterward they are burnt-out $f = 3$.

System parameters are the system size and the probability for the presence of an inflammable tree p.

Algorithm 12 MATLAB Code: Forest Fire

```
p = 0.6;
L = 100;
i1 = [L, 1:L-1]; i2 = [2:L, 1];
            % n.n. tableaux
forest = (rand(L,L)<p);
            % randomly arranged trees
forest(forest == 0) = 4; % empty cells
forest(L/2,L/2) = 2;
            % start fire in center
fig = image(forest,'erasemode','xor');
colormap([.1 .7 .1; 1 0 0; 0 0 0; 1 1 1]);
            % green,red,black, white
while sum(forest(:)==2) > 0
  ind = find(forest==2);
            % indices of burning trees
  forest((forest==1) & ...
    (forest(i1,:)==2|forest(i2,:)==2|...
    forest(:,i1)==2|forest(:,i2)==2)) = 2;
  forest(ind) = 3;
  set(fig,'cdata',forest);
  drawnow;
end
```

5.5 Self-organized Criticality

Self-organized criticality is a phenomenon which can be observed in a variety of natural systems: earthquakes, traffic flow, forest fires, river currents, sand piles, avalanches, etc. It is a general feature of dynamical systems. We will study the flow of a river by the following simple model:

- One-dimensional arrangement of L blocks, connected by springs to its nearest neighbors.
- There is a friction between the blocks and the underlying plane.
- The flow is driven by an external force.
- The block slips, if the force on a block exceeds the maximum static friction.
- The slip of a block changes the forces on its nearest neighbors, which results in further slips and the chain moves.

Question: If we place a glass of water on a block in the middle, is it possible to pull in such a way that the water in the glass would never be spilled?

The answer is NO. If the length of the chain L becomes very large, the probability of spilling tends to unity. This fact reflects the essence of self-organized criticality.

In a more formal way, the motion of the chain in the steady state is characterized by an average velocity v_{av}. The quantity of interest, however, is the distribution of the kinetic energy of a block

$$P(\varepsilon) = P(|E - E_{\text{av}}| > \varepsilon).$$

Two scenarios are conceivable:

A) The energy fluctuations are characterized by a fixed parameter ε_0 independent of the number of blocks L and we have typically

$$P(\varepsilon) \propto e^{-\varepsilon/\varepsilon_0},$$

where ε_0 is a cutoff independent of L. The exponential law corresponds to a smooth motion, if ε_0 is sufficiently small.

B) The sequences of blocks at rest form clusters of growing size. When the force at the edge of a cluster exceeds the maximum static friction, the motion of the whole cluster is triggered. A block, involved in this motion, performs a big jump giving rise to a considerable fluctuation of the kinetic energy. If the size distribution of the clusters has no characteristic length except L, there are no cutoff parameters in the system in the limit of large L. Therefore, the energy distribution for large L can be written in the form

$$P(\varepsilon) \propto \varepsilon^{-\tau},$$

where τ is a constant. The power law implies unlimited large fluctuations in the course of motion.

The concept of self-organized criticality claims the necessity of the second kind of behavior.

We are now ready to explain both parts of the term "self-organized criticality". The word "criticality" comes from the theory of second-order phase transitions. The critical state in thermodynamics is associated with self-similar fluctuations of observable values. Self-similarity means the absence of a characteristic scale or infinite correlation lengths. This means, in turn, that correlation functions obey power laws at the phase transition. To reach the critical point, one should tune one of parameters, e.g. the temperature, with high accuracy. In contrast to thermodynamics, dissipative dynamical systems drive themselves to the critical state, or "self-organize" themselves, automatically, without any fine-tuning of parameters.

The paradigm of self-organized critical systems is a pile of sand. Let us drop grains of sand on a horizontal plane at randomly chosen places, one grain at a time. At some point in time, the average slope of the pile reaches a steady state corresponding to the angle that cannot be exceeded no matter how long we carry on adding sand. The stationary state of the sandpile is not completely uniform since variations of the local slopes are possible. If we add a grain of sand, which causes the local slope to exceed the critical angle, an avalanche is triggered. We encounter a model, that is quite similar to our spring-block model. The regular input of sand represents the external force of the previous example. The addition of particles leads to avalanches similar to the clusters of moving blocks. As a result, we obtain an extremely irregular output of sand or energy having no characteristic intervals between peaks and therefore obeying the power-law distribution.

To examine these observations mathematically, BAK, TANG, and WIESENFELD proposed a CA model. The sandpile model on the two-dimensional square lattice can be defined as follows: Consider a rectangular lattice of linear dimension L. Each site i of the lattice is characterized by an integer z_i, the number of particles or the local height at this site. One drops a grain of sand on a site i chosen at random, thereby increasing its height by one:

$$z_i \to z_i + 1.$$

If this new height exceeds a maximum stable value, say 4, then the column of sand at site i becomes unstable and topples. The height z_i decreases by 4 and each of the four nearest neighbors j of the site i receives one particle:

$$z_i \to z_i - 4, \quad z_j \to z_j + 1, \qquad j : \text{n.n. of site } i.$$

(Here, 'n.n' means nearest neighbor.) If a toppling occurs at the edge of the lattice, the toppled site gives one particle to each of three neighbors while one grain drops out of the system.

To watch the evolution of the sandpile in time, we assume that one adds a particle to a stable configuration at each discrete point in time. If the height z_i reaches 5 somewhere, there is a toppling wherein 4 particles are transferred from the unstable site to its neighbors. The transferred particles may cause instabilities among new sites. The toppling of the latter perturbs next neighbors and a chain reaction propagates up to the moment when all sites get stable again. One assumes the updating to be done concurrently, with all sites updated simultaneously. The relaxation processes are assumed to be quick enough to be completed by the next discrete time. A collection of s distinct sites relaxed during an interval between two successive discrete moments of time forms an avalanche of size s. If a toppling at a given site causes instabilities at all nearest neighbors, the initial site receives 4 particles back and gets unstable after the next updating and therefore topples again. Typically, almost all sites inside a large avalanche undergo multiple topplings. The total amount of topplings in the given time interval is the mass m of the avalanche. The duration t of an avalanche is the number of updatings for the relaxation process to complete. The formulated rules describe a CA which is useful for computer simulations. The very first investigations of sand piles displayed clear power-law dependencies for distributions of all basic characteristics of the model:

$$\begin{aligned} D(s) &\propto s^{-\tau} \\ D(m) &\propto m^{-\kappa} \\ D(t) &\propto t^{-\alpha} \end{aligned}$$

and provided rough estimates for the critical exponents τ, κ, α in the size, mass, and duration distributions.