Chapter 1

Reaction–diffusion, excitation and computation

This book is built on case studies. So is this chapter. Despite its mosaic structure this chapter nevertheless shows a logical interrelation of sections. We are mainly concerned with computations in reaction–diffusion and excitable media. Such computations are unconventional. This is why we start with a brief discussion of unconventional computing. Most computational techniques discussed in this book rely on wave dynamics in nonlinear media. Therefore we devote one of the chapter’s sections to this subject. Various methods for modelling reaction–diffusion and excitable media are discussed here both to ease understanding of the theoretical algorithms and their practical implementations. We have allocated a special section to cellular automata because they represent an intuitive link between real nonlinear media and mathematical models of massively parallel computing architectures. At the end of the chapter we provide several examples of computational problems solved in nonlinear active media to give readers a sense of the subject.

1.1 Unconventional computing

What is unconventional computing? This is a matter of time scales, personal preferences and orthodoxy in education. As Toffoli writes:

\[ \text{a computing scheme that today is viewed as unconventional may well be so because its time hasn’t come yet—or is already gone.} \quad [593] \]

We would possibly associate unconventional with the advanced, however this is not customary at the moment. Unconventional computing, as is commonly agreed, includes:

- computing in stirred chemical reactors (mass-transfer and kinetic-based computing) and thin-layer chemical reactors (this is discussed widely in this book, so we do not give any references at the moment);
• assembly-based computing, particularly DNA computing (see, e.g. [147, 35, 495, 381, 544, 342, 255, 335, 249, 149, 173]) and, partially, microtubule-based-computing [287, 110];
• computing in neural networks (see e.g. [285, 328, 391, 154, 45]);
• computing in cellular automata (the present book certainly deals with this subject, however see also [591, 566, 274, 176, 254]);
• quantum computing (see e.g. [399, 109, 642, 380]);
• genetic algorithms and evolutionary programming (see e.g. [260, 364, 462, 153, 237, 455]); and
• swarm optimization, the solution of optimization problems using the principles of social insect behaviour (see e.g. [97, 552, 553, 96, 139, 191, 252, 272])

Some of the entries are sometimes classified as the subjects of natural computing; others are considered to belong to the physics of computation.

We first build a computer that hides the underlying spatial structure of Nature, and then we try to find the best way to contort our spatial computation to fit into that mold! . . . and so our most efficient computers and algorithms will eventually evolve toward the underlying spatial ‘hardware’ of Nature. [424]

These words by Margolus [424] sufficiently characterize the basic ideas of unconventional, particularly spatial computing.

Molecular computing and chemical computing are the two most popular plots in the field of natural computing (see e.g. [142, 143, 144, 145, 146, 150, 256, 542, 149, 339]). DNA computing [35] gives us an example of recent successes in molecular computing: DNA computing has become a synonym for molecular computing [74], thus replacing the more well-known term ‘molecular electronics’. A DNA computer solves the Hamiltonian path problem by producing a molecule, which encodes the path, if such a path exists. A Hamiltonian path is not the only problem solvable in a DNA computer [495, 381, 544, 342, 255, 249]. Ideally, any problem can be solved in this molecular computer. Shortly after pioneer Adleman published his work [35] it was demonstrated that a universal Turing machine can be built from DNA processors (see also [75]), i.e. DNA processors are simulation universal.

Berry and Boudul’s chemical abstract machine [82, 83, 319] is a brilliant example of how a chemical paradigm of the interactions between molecules can be utilized in concurrent computations, e.g. in algebraic process calculi. The state of a concurrent system is seen as a stirred solution in which the molecules of reagents interact with each other according to certain rules [83]. This idea has evolved from an abstract machine. The machine employs a gamma formalism, which is based on a set of transformations. The transformations consume the elements of a multiset and produce new elements by specified rules, developed for parallel programming [62]. The process is parallel because all pairs of molecules
Unconventional computing

Molecules can interact at once. Molecules are represented by multisets and the reaction rules are rewritings of multisets. Thus, a specific machine is designed by defining the molecular algebra and the rules of interaction between the molecules. The basic elements of the machine are ions in which the interactions are determined by their valences. Therefore, we can talk about a reaction rule when a molecule is formed from ions: heating when a molecule is split into several ions and cooling when a compound molecule is assembled from several other molecules [83].

The ideas of a chemical abstract machine are also reflected in the abstract chemistry developed from the theory of the self-maintenance of systems, autopoiesis [620] and \( \lambda \)-calculus [137]. In abstract chemistry molecules are viewed as computational processes supplemented with a minimal reaction kinetics. [238]

Fontana and Buss [137] derived several levels of \( \lambda \)-calculus models of chemistry to build a link between abstraction and dynamics [238]. In the models a physical molecule corresponds to a symbolic representation of an operator. The operator’s action represents the molecules’ behaviour. A chemical reaction is seen as evaluation of this functional application. So, a collision between molecules is interpreted as a production rule or a relation between the abstract molecules [238], in the typed calculi chemical model we can additionally represent the specifics of an interaction by a type discipline [238].

When did physics-based natural computing start? Computing via analog means was known long before digital computing. It can be dated back to 1876 when Lord Kelvin published his first results on analog-related computers (see [601]; cited in [512]).

It may be possible to conceive that nature generates a computable function of a real variable directly and not necessarily by approximation as in the traditional approach. [512]

The differential analyser is a typical example of a general purpose analog computer. It generates functions of time; the functions are measured in volts. The analyser consists of the following blocks [512].

- An integrator (one input and one output): for the input \( u(t) \) and the initial condition \( e(a) \) at time step \( t = a \) we have \( \int_{t}^{a} u(t) \, dt + e(a) \) on the output.
- An adder: two inputs \( u(t) \) and \( v(t) \) and one output \( u(t) + v(t) \).
- A constant multiplier, which produces \( k \cdot u(t) \).
- A variable multiplier, which produces \( u(t) \cdot v(t) \).

The ideas of analog computing evolved into field computing, i.e. computation over a continuum of continuously variable quantities [408], where the basic operations are field transformations rather than discrete operators [407, 408]. Neural networks, in general, and retina, in particular, may be classified as subclasses of field computers, for which silicon chip realizations are now widespread (see e.g. [456]).
Reaction–diffusion, excitation and computation

In general, any physical system can play the role of a computer, if the system’s behaviour and reaction to perturbations are appropriately interpreted. Thus, for example,

*any physical system whose dynamics in phase space is dominated by a substantial number of locally stable states to which it is attracted can therefore be regarded as a general content-addressable memory.* [314]

Amongst other things the physics of computation deals with

- the physical meaning of information: *physical action measures amount of computation* [595];
- the thermodynamics of computation: *the digital computer may be thought of as an engine that dissipates energy in order to perform mathematical work* [78];
- the relation of the minimal action principle to computation abilities; do you think that *a system’s natural trajectory is the one that will hog most computational resources* [301]?

However, one of the most attractive principles of physics-based computing, which is utilized in this book, is highlighted by Margolus [424]:

*Because physical information can travel at only finite velocity, portions of our computation that need to communicate quickly must be physically located close together.*

1.2 Waves in reaction–diffusion and excitable media

The majority of physical, chemical, biological and sociological processes can be described in terms of propagating fronts of one nature or another. The spread of genes in population dynamics, heat transfer, the growth of microbial, plant and animal colonies (and even cities) and the dynamics of excitation in nervous systems are just a few arbitrarily picked examples of phenomena with propagating fronts. Calcium waves in a cytoplasm of frog oocytes, waves in a mixture supporting RNA replication and aggregation in colonies of *Dictyostelium discoideum* are instances of propagating fronts in reaction–diffusion and excitable media [72, 612, 580]. Let us briefly discuss three examples of spatial wave dynamics from different fields of science.

- In a point-wide stimulation of a cerebral cortex a spreading EEG depression is observed. The suppression of electrical activity propagates from the stimulation point as a concentric wave called Leão’s wave. The wave propagates on the cerebral cortex, due to K⁺ movements between intra- and extra-cellular spaces [432]. Similar waves are observed in laboratory experiments with isolated retina [114, 432].
• Auto-waves are phenomena typically observed in morphogenesis. They include proximal–distal waves of cell re-orientations and peristaltic annulations in *Hydrozoa*, cortical peristaltic waves in *Nematoda*, waves of cell contraction in embryos, contact cell polarizations, calcium waves accompanying cell division in *Paramecium*, mitotic activity waves in retinal differentiation of crayfish [77, 107, 547, 503, 283, 481].

• Circular patterns, wave-like in nature, are generated in bacterial cultures infected by phage [323]. Spiral spatial waves are very typical for host–parasite systems [91]. Almost all population systems, especially those with predator–prey relations exhibit wave patterns in their development [37, 465, 351, 379, 86].

Most types of propagating fronts, including those propagating in excitable media, can be analysed in terms of reaction–diffusion systems [212].

The first works on wave phenomena in excitable and chemical media were published at the beginning of this century by Mayer [437] and Luther (see [558, 559] for details). Their publications dealt with phenomenological results on the arhythmic properties of jellyfish muscle and the propagation of a wavefront in a tube containing a chemical liquid [559]. Robert Luther produced a propagating front in a permanganate oxidation of oxalic acid, where a purple solution becomes uncoloured at the moving boundary [212]. However, it was some time before the ideas were developed further, particularly in theoretical works. Thus, the 1946 paper by Wiener and Rosenblueth [640] is usually cited as the basic reference for mathematical models of excitable media, but in fact the first mathematical work, published in 1937, on the subject belongs to Kolmogorov, Petrovsky and Piscounoff [360], who analysed wave motion in excitable media and obtained a reaction–diffusion equation. Fisher independently published a similar equation in the same year. Then a quantitative mathematical model of wave propagation in a nerve was presented in 1952 by Hodgkin and Huxley [308]. Hodgkin and Huxley’s analytical studies of a travelling excitation in nerve membranes [308] were developed further by FitzHugh [231]. Actually, most of the basic features of wave motion in excitable media had already been discovered mathematically by the late 1930s. These features include the notion of a threshold or a necessary condition for the formation of a travelling wave: a sufficiently large region of a medium must be excited to start the excitation wave dynamic [545, 360] (see also [573]).

The simplest kind of a chemical wave emerges when

*reactants are converted into products as the front propagates through the reaction mixture*. [212]

Background information on reaction–diffusion and pattern formation in chemical, physical and biological systems can be found in [441, 160, 212, 357]. We consider just a few examples for illustrative purposes.

Turing’s paper [607] showed that stable patterns are produced in reaction–diffusion systems, where two reacting substances diffuse with different rates. This
idea has been successfully applied to analyses of pattern formation on mollusc shells [441, 443] that gives us a bright example of a biological application of reaction–diffusion theory. Meinhardt and Klinger [442, 443, 444] developed a model of pattern formation in a mollusc shell where pigment deposits oscillate and travelling waves of pigment production are formed, when pigment-producing cells trigger their neighbours. It is assumed that the patterns develop during the growth of the shell, which is a \((1 + 1)\)-dimensional process. The one-dimensional pulses of pigment production may collide with one another, annihilate, reflect or produce new travelling waves. As a result a wave dynamic which is typical for mollusc-shell patterns is generated. Patterns similar to those of *Strigilla* and *Palmadusta diluculum* can be produced [442, 443, 444]. The reaction–diffusion model itself is based on the interaction between an autocatalytic activator \(a\) and its inhibitor \(h\), concentrations of which evolve in a one-dimensional space \(x\) as follows:

\[
\frac{\partial a}{\partial t} = \frac{\rho a^2}{h} - \mu a + D_a \frac{\partial^2 a}{\partial x^2} + \rho_0
\]

and

\[
\frac{\partial h}{\partial t} = \rho a^2 - \nu h + D_h \frac{\partial^2 h}{\partial x^2} + \rho_1
\]

where \(\rho\) is a source density, \(\mu\) is the activator lifetime, \(\nu\) is the inhibitor lifetime, and \(\rho_0\) and \(\rho_1\) are source densities for the activator and inhibitor. Sometimes, we talk about the depletion of a substrate, instead of the production of an inhibitor, which may be consumed in the autocatalytic production of an activator [444].

The parameters for reaction and diffusion determine the exact structure of the developing pattern. A discrete cellular-automaton-based model of mollusc shell-pattern formation is developed in [500]. Various patterns that arise from collisions of classical waves, chemical waves and mobile localization are discussed there.

The nontrivial behaviour of nonlinear chemical systems can be explained by their bistability. Bistability means that two stable steady states coexist. For example, in a stirred tank a chemical system undergoes a transition from one stable state to another as a result of perturbations [212]. In reaction–diffusion systems

*fronts typically propagate with a constant velocity and wave form,*

*converting reactants that lie ahead into products which are left behind.*

*The bulk of the chemical reaction occurs within a narrow reaction zone.* [212]

The Belousov–Zhabotinsky-reaction-based medium [664, 669] is the most well-known example of an active nonlinear medium [644]. The classical reaction incorporates a one-electron redox catalyst, an organic substrate (that can be brominated and oxidized) and a bromate ion in either the form NaBrO3 or KBrO3. The reagents are dissolved in sulphuric or nitric acid. Ferroin is the most typical catalyst in classical reactions, and ruthenium is a standard catalyst in lightsensitive reactions. Malonic acid is used as an organic substrate; Belousov himself
used citric acid. Oxidized and reduced forms of the catalysts have different
colours, for example, oxidized ferroin has a blue colour and reduced ferroin has a
red colour.

A very typical recipe for a Belousov–Zhabotinsky reaction in a stirred reactor
includes 0.3 M malonic acid, 0.1 M NaBrO$_3$, chloride-free ferroin (or MnSO$_4$),
diluted in 1 M sulphuric acid. Namely, we dilute, in sulphuric acid, malonic
acid, the catalyst and the bromate. Another, more detailed recipe, is as follows:
0.2–0.36 M BrO$_3^-$, 0.03–0.2 M malonic acid, 0.21–0.51 M H$_2$SO$_4$, 0.001–0.1 M
catalyst (we can choose a catalyst from Fe(phen)$_3$SO$_4$, Ru(bipy)$_3$SO$_4$, Ce(SO$_4$)$_2$
and Mn(SO$_4$)$\cdot$4H$_2$O) and 0.09 M bromomalonic acid [475].

What happens in the reaction mixture liquid? In the standard version of
the reaction the following sequence of events is repeated [226, 47]. BrO$_3^-$ is
transformed into bromic acid HBrO$_2$. This transformation process is coupled
with oxidizing ferroin to give ferriin. At this stage the colour of the mixture
changes because ferroin is red and ferriin is blue. Ferroin can be recovered from
ferriin with the help of malonic acid. Bromide ions are produced in the ferriin-to-
ferroin transformation. These bromide ions inhibit the production of the bromic
acid. More generally we can state that the essentials of the Belousov–Zhabotinsky
reaction are:

- the oxidation of BrO$_3^-$ leads to the production of HOBr and Br$_2$, which
  brominate the organic substrate
- the autocatalytic generation of HBrO$_2$ with parallel oxidation of metal
catalyst
- the oxidation of the organic substrate by the catalyst to regenerate Br$^-$.

These three stages are cyclically repeated [212].

The following schematic description [501] of the Belousov–Zhabotinsky reac-
tion is adopted for the well-known Oregonator model [350] (see also [670]):

$$
A + W \xrightarrow{k_1} U + P \\
U + W \xrightarrow{k_2} 2P \\
A + U \xrightarrow{k_3} 2U + 2V \\
2U \xrightarrow{k_4} A + P \\
B + V \xrightarrow{k_5} hW
$$

where $A$ is the concentration of BrO$_3^-$, $B$ the concentration of bromomalonic
acid, $P$ the concentration of HOBr, and $U$ the concentration of HBrO$_2$. $V$
is the concentration of the catalyst (for example, Ce$^{4+}$ or Ru(bpy)$_3^{2+}$), $W$ is
the concentration of Br$^-$. The rate constants range from $k_1$ to $k_5$ and $h$ is a
stoichiometric factor, which counts the number of moles of Br$^-$ released in the
reduction of 1 mole of a catalyst [501]. The Oregonator model for light-sensitive modification of the Belousov–Zhabotinsky reaction may be formulated in the following system of two partial differential equations [579, 531]:

\[
\frac{\partial u}{\partial t} = \epsilon^{-1} \left( u - u^2 - (f v + h) \frac{u - q}{u + q} \right) + D_u \nabla^2 u \\
\frac{\partial v}{\partial t} = u - v + D_v \nabla^2 v
\]

where \( u \) is the concentration of HBrO2, \( v \) is the concentration of the catalyst, \( h \) is used to show the contribution of the catalyst produced as a result of illumination, \( \epsilon, f \) and \( g \) are small real constants, and \( D_u \) and \( D_v \) are diffusive coefficients.

The Brusselator is yet another well-accepted model for reaction–diffusion systems:

\[
\frac{\partial u}{\partial t} = a - (b + 1)u + u^2 v + D_u \nabla^2 u \\
\frac{\partial v}{\partial t} = bu - u^2 v + D_v \nabla^2 v.
\]

The concentrations of the two chemical species are represented by the variables \( u \) and \( v \). The control parameters are \( a \) and \( b \), and \( D_u \) and \( D_v \) are the diffusive coefficients. Various dissipative structures including spiral waves can be formed as the system evolves according to the initial parameters.

Experiments with the Belousov–Zhabotinsky reaction can be done either in stirred or non-stirred, thin-layer reactors. In the stirred reactor periodic changes between oxidized and reduced states happen concurrently in all sites in the medium. Therefore the whole solution changes colour from blue to red and back to blue if ferroin is the catalyst [227, 543]. If cerium is used as the catalyst the solution oscillates between colourless and yellow. It oscillates between colourless and red if we use manganese ions. The oscillations start after some induction interval. A typical oscillation period is approximately 1 min; however, it strongly depends on the temperature and other factors.

To make an unstirred reactor prepared solutions are poured into a Petri dish to make a layer 1–3 mm thick. To generate circular waves one can immerse Ag wire (100 \( \mu \)m diameter) in the solution [475]. The wave is detectable by its blue front, i.e. a blue colour indicates ferriin, the oxidized state of the catalyst. It spreads into a resting, red coloured solution where the catalyst is resting in its reduced ferroin form [472]. The wavelength is usually around 1 mm. The waves spread with the velocity 1 mm min\(^{-1}\).

What is going on in a non-stirred Belousov–Zhabotinsky reactor? We spread the solution as a thin layer and disturb it with a silver wire, if the catalyst is ferroin or just illuminate part of the solution, if ruthenium is used as a catalyst. The disturbance leads to the generation of a blue oxidation front that propagates on the red background of reduced ferroin.
Two types of waves are observed, and sometimes they coexist, in Belousov–Zhabotinsky solutions [47]:

- trigger waves (they have constant shape and low speed, 1–4 mm min$^{-1}$)
- phase waves (waves with a velocity much higher than that of the trigger waves); they may be similar to phase-shifted oscillations in chains of weakly coupled pendula.

We could possibly consider a continuum where trigger waves occupy one end and phase waves the other [47]. Diffusion phase waves are somewhere between the poles.

Fronts in a Belousov–Zhabotinsky medium are described by a quadratic reaction–diffusion equation, which is similar to the Fisher–Kolmogorov equation [212]. As previously discussed, the medium supports trigger waves—the propagation pulses that move through the resting reaction solution; and the system returns to the resting state behind the wave. So, the Belousov–Zhabotinsky medium is an excitable medium.

In excitable chemical media wave propagation occurs because of coupling between diffusion and autocatalytic reactions. When autocatalytic species are produced in one micro-volume of the medium they diffuse to the neighbouring micro-volumes and they trigger an autocatalytic reaction there. If an excitable system has spatial extent it will respond to perturbations exceeding the excitation threshold by producing excitation waves [212, 341].

Excitation waves travelling in excitable media are often called auto-waves because they propagate at the expense of energy stored in the medium [368, 322]. Even a superficial comparison of auto-waves with classical waves shows entirely different behaviour for the excitation waves [365, 368]:

- In contrast with classical waves auto-waves do not conserve energy. Rather they propagate at the expense of the energy of the chemical medium; however, they do conserve amplitude and waveform.
- Auto-waves, in general, are not reflected from a medium’s boundaries or inhomogeneities; however, they may annihilate when they collide with one another. It has been demonstrated in [671, 213] that waves in a Belousov–Zhabotinsky medium can be reflected as the result of collisions with rectangular inhomogeneities depending on wave speed and angle of incidence.
- Auto-waves do not exhibit interference.
- Both classical and excitation waves obey Huygens’ principle, therefore they diffract.

If an excitation wave is broken, for example due to defects in the medium, the broken wave may either contract or be transformed into a spiral wave. The dynamics of two- and three-dimensional spiral waves has been well investigated in [349, 350, 453].
An attractive setup for investigating excitation waves on substrates with a discrete topology may be implemented in a so-called heterogeneous active Belousov–Zhabotinsky medium. The medium consists of a reaction zone with an immobilized catalyst, usually ferroin fixed in silicahydrogel substrate, covered by a feeding solution without a catalyst [396]. One can also immobilize the catalyst on cation exchange resin beads [435]; in such a setup we can also obtain a pseudo-cellular topology in the reactor space.

What is the essence of excitation?

A system is excitable if it possesses a stable fixed point and responds to perturbation in such a manner that if a perturbation magnitude exceeds a certain threshold the system moves in a phase space along some closed trajectory until it returns to the fixed point. While moving along its trajectory the system is not sensitive to any perturbations and it is may be called refractory. [341]

A grass fire, particularly on large time and space scales, is a typical example of an excitable medium. If we set a path of dry grass on fire the fire will spread. Ideally, a fire front is circular, hence it may lose its regularity due to inhomogeneity of grass cover and landscape. The burnt zones cannot be repeatedly fired. However, over the years the grass grows again and the territory is ready to be set on fire again. The following analogies are remarkable.

- Setting on fire is a perturbation action.
- A fire-front is an excitation front.
- A burnt zone is the part of medium in a refractory state.
- Growth of a grass is the recovery of a substrate.

The grass-fire example demonstrates four essential features of every excitable medium:

(i) There is a spatially homogeneous resting state which is stable to small perturbations.
(ii) When a certain number of a medium’s sites are perturbed above a certain threshold self-propagating waves are generated.
(iii) The waves travel in the medium without damping.
(iv) The medium returns to its resting state after propagation of the waves.

In this book we do not deal with reaction and diffusion only in chemical species. Therefore, it is reasonable to mention the Fisher equation, which deals with the spread of a mutant gene in the population:

\[
\frac{\partial p}{\partial t} = D \frac{\partial^2 p}{\partial x^2} + kp(1 - p)
\]

where \( p \) is the frequency of the mutant gene. We should also appreciate that not only chemical and biological media are subject to excitation. Many other
systems exhibit excitation dynamics. We can demonstrate this using a system studied by Muratov [474]. The example clearly exposes the underlying mechanics of excitation. This is an elastic medium with friction exhibiting stick–slip motion. The Burridge–Knopoff model is under consideration here. The model is represented by a one-dimensional array of blocks resting on a frictional surface. The blocks are connected by springs, with spring constant $k_c$, one by one. Thus they form a chain. Each block is also connected by a spring, with spring constant $k_p$, to a loader plate. The blocks are pulled by a loader plate moving with a constant velocity. Each block is linked by springs to its left- and right-hand neighbours and to the loader plate, therefore the following total force

$$F_i = k_c(x_{i+1} + x_{i-1} - 2x_i) - k_p x_i - f_i$$

acts on the $i$th block, where $x_i$ is a displacement of the block $i$, and $f_i$ is a friction force [474]. When the force applied to a block exceeds the static friction force the block starts to slip and the friction force drops immediately to some value below the static friction. When the blocks stick the friction returns to its static value. The accumulated stress accelerates the block at the onset of motion. The released potential energy is enough to support travelling waves in the chain of blocks. In this model the static friction plays the role of excitation threshold while the springs connecting the blocks represent local coupling [474].

1.3 Cellular automata

Space, time and state of the space site are the basic constituents of any model. Each element may be either discrete or continuous. Partial differential equations are a classical instance in which all elements are continuous. We have coupled map lattices when space and time are discrete and the state is continuous. Cellular automata give us an example in which all three elements—time, space and state—are discrete.

The following words were recorded just at the beginning of the Second World War:

*Suppose one has an infinite regular system of lattice points in $E^n$, each capable of existing in various states $S_1, \ldots, S_k$. Each lattice point has a well defined system of $m$ neighbours, and it is assumed that the state of each point at time $t + 1$ is uniquely determined by the states of all its neighbours at time $t$. Assuming that at time $t$ only a finite set of points are active, one wants to know how the activation will spread.* [616]

This quotation from Stanislaw Ulam’s book [616] gives us not only the first evidence of cellular-automata theory but also quite accurately defines a cellular automaton itself.

*A cellular automaton is a local regular graph, usually an orthogonal lattice, each node of which takes discrete states and updates its states*
Figure 1.1. Classical neighbourhood of a two-dimensional cellular automaton: (A) von Neumann neighbourhood; (B) Moore neighbourhood. The central cell is shown by ⋆. Other elements of the cell ⋆’s neighbourhood are shown by ■s. Cells not belonging to the neighbourhood of cell ⋆ are shown by □s.

in a discrete time; all nodes of the graph update their states in parallel using the same state transition rule.

The adjective ‘cellular’ possibly reflects the original representation of such automata in matrix terms, as Ulam and, at least initially, von Neumann did; or Neuman’s early drawing of ruled squares. Biological cells, as is quite often wrongly assumed, never remained behind the cellular-automata concept. Sometimes one can meet cellular automata under such names as tesselations and tesselation structures, homogeneous structures, mosaic automata, uniform automata networks and cellular arrays.

A cellular automaton is a tuple \( \langle L, Q, u, f \rangle \) of a many-dimensional array \( L \) of uniform finite automata, finite state set \( Q \), local connection template, or automaton’s neighbourhood, \( u, u : L \rightarrow L^k \), \( k \) is a positive integer, and automaton transition function \( f, f : Q^k \rightarrow Q \). Automata of the array update their states simultaneously and in discrete time.

Each automaton \( x \) of the array \( L \) calculates its next state by the rule: \( x^{t+1} = f(u(x)^t) \), where \( x^{t+1} \) is the state of automaton \( x \) at time step \( t+1 \) and \( u(x)^t \) is the state of automaton’s neighbourhood \( u(x) \) at time step \( t \).

Automata constituting the array \( L \) are called cells. Cells are locally interconnected. The exact architectonics of local links between the cells determine the structure of the neighbourhood. In most situations, to define the neighbourhood \( u(x) \) of a cell \( x \) it is enough to determine the cells, cell \( x \)'s neighbours, being at a distance no more than \( r \), where \( r \) is a positive integer, in a metric \( M \):

\[
u(x) = \{ y \in L : |x - y|_M \leq r \}.
\]

For \( M = L_1 \) we obtain the so-called von Neumann (cruciform or four-cell, because \( k = 4 \)) neighbourhood (figure 1.1(A)), and the so-called Moore (complete or eight-cell, because \( k = 8 \)) neighbourhood (figure 1.1(B)) for
Cellular automata

There are no restrictions on the type of neighbourhood, therefore quite a variety of them can be defined. If the neighbourhood is not simply organized it can be defined as a list of neighbours: \( u(x) = (y_1, \ldots, y_k) \), \( k \) is the size of neighbourhood. The state of the neighbourhood at time step \( t \) can be written as \( u(x)^t = (y_1^t, \ldots, y_k^t) \).

The cell state transition rule can be represented in many different ways, for example, as a set of IF-THEN rules, a formula or a lookup table.

The configuration of a cellular automaton is a mapping \( c : L \rightarrow Q \), which assigns a state from \( Q \) to each cell of the array \( L \). A global transition function is the mapping \( G : Q^L \rightarrow Q^L \), which transforms any configuration \( c \) to another configuration \( c' \), \( c' = G(c) \) by applying local transition function \( f \) to the neighbourhood of every cell of the array \( L \). In this book we mostly deal with deterministic automata. A cellular automaton is called deterministic if for any configuration \( c \) there is exactly one configuration \( c' \) that \( G(c) = c' \).

Given initial configuration \( c^0 \) the evolution of a cellular automaton is represented by a sequence of the automaton’s configurations:

\[
    c^0 \rightarrow c^1 \rightarrow \ldots \rightarrow c^t \rightarrow \ldots \rightarrow c^t+p.
\]

Eventually the cellular automaton falls into a cycle with period \( p \), \( p > 1 \), or a fixed point of its evolution, \( p = 1 \).

The global dynamics of finite cellular automata can be visualized in a graph of global transitions. Every node of the graph corresponds to a unique configuration of the cellular automaton and the arcs connecting the nodes represent global transitions between the configurations.

Here is a small example. Given a one-dimensional cellular automaton with five nodes. The lattice is concaved in a torus, i.e. the automaton has periodic boundaries. Every cell of the automaton takes two states, 0 and 1, updates its state depending on the states of its left- and right-hand neighbours. Indexing the cells of the lattice by \( i, i = 1, \ldots, 5 \), we can write cell \( x_i \)’s neighbourhood as \( u(x_i) = (x_{i-1}, x_{i+1}) \); in this particular example a cell is not included in its neighbourhood.

Let a cell take the state 1 if only one of its neighbours is in state 1, independently of the state of the cell itself. Otherwise, the cell takes the state 0. The cell state transition rule \( x_i^{t+1} = f(x_{i-1}^t, x_{i+1}^t) \) can be represented as \( x_i^{t+1} = (x_{i-1}^t + x_{i+1}^t) \mod 2 \), if states are arithmetical, or \( x_i^{t+1} = x_{i-1}^t \land x_{i+1}^t \lor x_{i-1}^t \land x_{i+1}^t \), if the states have Boolean values. We could also represent the rule as

\[
    x_i^{t+1} = \begin{cases} 
        1 & \text{if } x_{i-1}^t = 1 \text{ and } x_{i+1}^t = 0 \text{ or } x_{i-1}^t = 0 \text{ and } x_{i+1}^t = 1 \\ 
        0 & \text{otherwise.} 
    \end{cases}
\]

Finally, we can describe the rule via a lookup table, where every entry has the format \( x_{i-1}^t x_{i+1}^t \rightarrow x_i^{t+1} \):

\[
    \begin{array}{cccc}
        00 & \rightarrow & 0 & 01 & \rightarrow & 1 & 10 & \rightarrow & 1 & 11 & \rightarrow & 0.
    \end{array}
\]
The automaton consists of five cells, and every cell takes two states, therefore the global transition graph has 32 nodes. The graph consists of the six components of connectivity as shown in figure 1.2(A).

Increasing the number of cells in a cellular-automaton lattice one can obtain impressive results in spacetime (see, e.g., figure 1.3) and in the global integral dynamic (see, e.g., figure 1.2(B)) of the cellular automaton.

The development of cellular automata was certainly facilitated by early results from the beginning of computation [611], the formal modelling of neural networks [438] and the widespread application of automata theory in the natural sciences [641]. For the last 30 years cellular automata have grown from
entertainment toys for intellectuals to an interdisciplinary field that contributes to almost all sciences, from pure mathematics and logic, to physics and chemistry, engineering and biology, psychology and geology. As Toffoli stated:

*Cellular automata are dynamical systems that play in discrete mathematics a role comparable to that partial differential equations in the mathematics of the continuum.* [595]

This can be paraphrased as

*cellular automata are parallel models of computation, discrete dynamic systems and models of natural processes that replace traditional mathematics in natural sciences and substitute nature in contemporary mathematics.*

The half-century of struggle it has taken for cellular automata to become a recognized discipline reflects, on a small scale, the time it has taken humanity to recognize the discreteness of the surrounding world. This started with Leeuwenhoek’s discoveries of the cellular structure of living matter [388], was continued in the identification of the discreteness of ‘grey matter’, when Santiago Ramon y Cajal convinced the world that neurons are separated by synaptic spaces and not continuously connected to each other as Camillio Golgi believed, and reached its apogee in Neumann’s self-organizing automata [479], Zuse’s *discrete universe* [676], Toffoli’s *programmable matter* [592, 593, 594] and Minsky’s *cellular vacuum* [457, 458]. For a period of time the theory of cellular automata flourished significantly and dozens of quite distinct classes of cellular automata emerged.
The definition of a cellular automaton given earlier concerns deterministic and synchronous cellular automata with a stationary architecture of local connections. There are also such attractive classes as asynchronous cellular automata, stochastic cellular automata and structurally dynamical cellular automata.

In asynchronous cellular automata cells do not update their states at the same time, state-switching may be triggered by internal or external events, sometimes stochasticity is involved (see e.g. [477, 116, 8, 152, 403, 551, 278, 499, 574]). We must mention that no proper asynchronous cellular automaton has ever existed. Either such automata are probabilistic or cells use some external input to control the delay of state-switching.

Stochastic cellular automata rely on the probabilistic update of cell state: a cell will choose one of the states with a predetermined conditional probability (see e.g. [189, 411, 277, 392, 9, 338, 84, 398]).

Structurally dynamic cellular automata are a lovely class. Cellular automata and automata networks were the seeds from which a spacetime theory on the Planck scale in [534, 535] emerged. In Requardt constructions the strength of the link between two neighbouring cells of a cellular automaton depends on the intensity of the interaction between the cells, where the interaction is considered to occur via transportation of quanta between the cells. The strength of each link is changed in such a manner to balance the intensity of the energy exchange between the cells. Considering the model in the context of reaction and diffusion we can see the strength of the link as an analogue of dynamical diffusive coupling. These representations have evolved from ‘structurally dynamic’ cellular automata [318], the most promising class of cellular automata. The notion of structurally dynamic automata implies that the lattices evolve dynamically: the links connecting cells evolve together with the cells. A link, as usual, may take two states: is and is not. The states of the links are updated depending on the states of adjacent cells and nearby links. Some examples of the identification of these automata can be found in [9]. Advanced discussion of dynamical cellular automata is given in [292], where models assume that the local connectivity of the cells changes as the automaton evolves depending on the cell states. In such an interpretation, a link is analogous to a synapse, in which the strength depends on the ‘excitation flow’ between axon and dendritic terminals.

Instead of re-telling numerous results from cellular-automata theory and applications I prefer to refer readers to some introductory and advanced texts.

Thus, Egan’s permutation city [208] explores the idea of compactly encoding human beings and their worlds in cellular-automata non-constructible configurations—Gardens of Eden. This is the only example of good science fiction dealing directly with cellular automata. As to non-fiction readers should start with von Neumann’s self-organization [479] and Zuse’s discrete universe [675] as texts that should be necessarily read through.

Finally, classical results on one-dimensional cellular automata can be found in Wolfram’s collection of papers [647]. Most results in [647] are still valid and certainly will be for a long time.
The book [648] by Wünsche and Lesser deals mainly with a graph representation of the spacetime dynamics of one-dimensional binary cellular automata. Really wonderful examples of global transition graphs are to be found there. Some sections of the book must be paid attention, particularly those devoted to glider dynamics in cellular automata.

The monographs by Toffoli and Margolus [591] and by Weimar [635] discuss simulation of physical phenomena in cellular automata. They both present useful techniques of cellular-automata modelling together with analysis of the efficiency and correctness of cellular-automata simulators.

Supervised evolution of cellular-automata machines is the main subject of Sipper’s monograph [566]. Anyone eager to design parallel processors using genetic algorithms which appeal to evolution should consider this book.

There is one chapter in the book [9] that raises the question of designing molecular computers but it focuses entirely on the identification problem and considers the essence of the problem rather simplistically.

Aladyev’s non-standard account of his personal results in cellular-automata theory, particularly the algebras of local transition functions and simulation of biological processes [41] is quite interesting, at least in a historical perspective. Ideologically similar, but entirely different in implementation, is the project realized by Voorhees [627].

Preston and Duff’s text [506] deals with cellular-automata algorithms and processors for image-processing. Clearly, image-processing has always been close to cellular-automata architectures. A systolic screen [175] is a very typical example, where a digitized image is represented in a rectangular array of pixels, the array in its turn is projected onto the same size array of simple processors, where there is a one-to-one correspondence between the pixel and the processor. When the processor updates its state the pixel also switches its state.

A mathematical theory of cellular automata, which also includes problems of decidability and mutual simulations between automata classes, is derived in [373].

And last but not least, if you like Mathematica you could be happy to analyse the programs from Gaylord and Nishidate’s book [257]. Quite useful examples of reaction–diffusion and excitable media are considered there, you can even find implementations of sand-piles and traffic flows.

"My question is, ‘Can physics be simulated by a universal computer?’… I would like to have the elements of this computer locally interconnected, and therefore sort of think about cellular automata as an example… we might change the idea that space is continuous to the idea that space perhaps is a simple lattice and everything is discrete… and that time jumps discontinuously. [224]"

Feynman’s words may form a bridge between the current section and the next one.
1.4 Modelling reaction–diffusion and excitable media

Mathematical models of excitable media usually employ partial differential equations of a reaction–diffusion type. Some examples can be found in [366, 679, 401, 214, 489, 678, 324]. Use of a finite difference method of numerical integration is a typical move when we deal with solutions of reaction–diffusion equations. Let us tackle a two-dimensional excitable medium described by a couple of reaction–diffusion equations:

\[
\begin{align*}
\frac{\partial a}{\partial t} &= f(a, b) + D_a \left( \frac{\partial^2 a}{\partial x^2} + \frac{\partial^2 a}{\partial y^2} \right) \\
\frac{\partial b}{\partial t} &= g(a, b) + D_b \left( \frac{\partial^2 b}{\partial x^2} + \frac{\partial^2 b}{\partial y^2} \right)
\end{align*}
\]

where \(a\) and \(b\) are the concentrations of the species, \(D_a\) and \(D_b\) are the diffusion coefficients of the species. The functions \(f(\cdot)\) and \(g(\cdot)\) describe the local kinetics of the species. When there is no general analytic technique we should appeal to numerical methods. A classical finite difference method of numerical integration is based on a system of difference equations:

\[
\begin{align*}
a_{i,j}^{t+\Delta t} &= f(a_{i,j}^t, b_{i,j}^t) \Delta t + D_a \frac{\Delta t}{(\Delta x)^2} (a_{i-1,j}^t + a_{i+1,j}^t + a_{i,j-1}^t + a_{i,j+1}^t - 4a_{i,j}^t) \\
b_{i,j}^{t+\Delta t} &= g(a_{i,j}^t, b_{i,j}^t) \Delta t + D_b \frac{\Delta t}{(\Delta x)^2} (b_{i-1,j}^t + b_{i+1,j}^t + b_{i,j-1}^t + b_{i,j+1}^t - 4b_{i,j}^t)
\end{align*}
\]

where indices \(i\) and \(j\) mean coordinates \((i \Delta x, j \Delta y)\) at the orthogonal lattice with step \(\Delta x\). The multipliers \(D_a (\frac{\Delta t}{(\Delta x)^2})\) and \(D_b (\frac{\Delta t}{(\Delta x)^2})\) are usually taken to be close to \(\frac{1}{4}\). Not surprisingly, most patterns emerging in reaction–diffusion systems relate to time and space scales. Therefore, numerically obtained patterns may be incorrect, compared to analytical solutions, if no numerical diffusion anisotropy is taken into account and no calibrating is provided [188]. The following neighbourhood templates, or discrete representations of Laplacians, seem to be correct when used in a numerical integration [188]:

- **One-dimensional system:**
  \[u_{i,j}^{t+\delta t} = u_{i,j}^t + \frac{1}{6} (u_{i-1,j}^t + u_{i+1,j}^t - 2u_{i,j}^t) + f(u_{i,j}^t) \Delta t\]

- **Two-dimensional system:**
  \[u_{i,j}^{t+\delta t} = u_{i,j}^t + \frac{1}{36} (u_{i-1,j-1}^t + u_{i+1,j+1}^t + u_{i,j-1}^t + u_{i,j+1}^t - 4u_{i,j}^t) + \frac{1}{36} (u_{i-1,j+1}^t + u_{i+1,j-1}^t + u_{i,j+1}^t + u_{i,j-1}^t - 4u_{i,j}^t) + f(u_{i,j}^t) \Delta t\]
Three-dimensional system:

\[
\begin{align*}
\Delta u_{i,j,z}^{t+1} &= u_{i,j,z}^{t+1} + \frac{1}{18}(u_{i-1,j,z}^{t+1} + u_{i+1,j,z}^{t+1} + u_{i,j-1,z}^{t+1} + u_{i,j+1,z}^{t+1} \\
&+ u_{i,j,z-1}^{t+1} + u_{i,j,z+1}^{t+1} - 6u_{i,j,z}^{t+1}) \\
&+ \frac{1}{36}(u_{i-1,j-1,z}^{t+1} + u_{i+1,j-1,z}^{t+1} + u_{i-1,j+1,z}^{t+1} + u_{i+1,j+1,z}^{t+1} \\
&+ u_{i,j-1,z+1}^{t+1} + u_{i,j+1,z+1}^{t+1} + u_{i-1,j,z-1}^{t+1} + u_{i-1,j,z+1}^{t+1} + u_{i,j-1,z-1}^{t+1} + u_{i,j-1,z+1}^{t+1} - 12u_{i,j,z}^{t+1}) \\
&+ f(u_{i,j,z}^{t+1})\Delta t.
\end{align*}
\]

The space and time steps should satisfy the scaling relation \( \frac{D\Delta t}{\Delta x^2} = \frac{1}{3} [188]. \)

Several examples of recent results on excitation dynamics include the motion equations of excitation derived in [210], Galilean transformations in reaction–diffusion equations which allow the consideration of auto-wave regimes in two- and three-dimensional excitable media [170] and rigorous results on the three-dimensional simulation of excitation in cardiac tissue [46].

We do not need to have a deep imagination to draw an analogy between the discrete representations of Laplacians and cellular-automata neighbourhoods; and, in fact, between differences and a cell state transition function. How good are cellular-automata models for reaction–diffusion systems?

Historically, cellular-automata models are of the same age as conventional continuous models. Thus, an abstract model of excitation in a continuous medium, namely propagation of impulses in a fibre with refractivity can be traced back to [259, 604]. The models of the excitation dynamics in continuous media [259], designed in the early 1960s, deal with the propagation of the impulses in a ring of refractive fibres, and the pacemaker model where the medium, the elements of which have different periods of spontaneous excitation, becomes excited at the frequency of its fastest element. One of the early two-dimensional models dealt mainly with the synchronization of oscillatory networks [402] and models of heart pathology based on wave propagation in a two-dimensional excitable medium with defects [60]. The first ever cellular-automata model of an excitable medium was developed by Balakhovsky [61] in 1965. This model was promptly elaborated in great detail by Reshod’ko [536] and Zykov [678]. And already by 1966 Krinsky [366, 367] had described the spiral properties of two-dimensional excitation waves.

Recall that the behaviour of an excitable element [275, 640, 341] may be seen to be determined by three intervals: \([-R, \theta], [\theta, U]\) and \([U, 1]\). The element relaxes back to the stable state 0 (and outputs 0 as well) if it is perturbed by a value from the interval \([-R, \theta]\). The element is ‘excited’, and its output equals 1, if its perturbed by a value from \([\theta, U]\). The value \(\theta\) is called a threshold of excitation, or excitation threshold. Outside the excitation interval, i.e. in the interval \([U, 1]\), the element takes a refractory state where it is absolutely insensitive to input perturbations. In [341] we can find models which take
excitability as originating from the interaction between two chemical species, sometimes called the propagator and controller variables.

Feldman \textit{et al} [221, 222] used cellular automata on a randomized lattice to simulate two-dimensional excitable media. They established a relationship between the excitation time scale, the diffusion constant of the medium and the corresponding characteristics of a cellular-automata model and derived a general equation describing wavefront propagation. Expanding the finite-cell state up to the real interval yields ‘cellular neural nets’ which can also be used to investigate various regimes of excitation dynamics [329]. An excellent approximation of continuous models by discrete ones has also been demonstrated in a series of papers [494, 643, 48, 159, 428, 429, 427, 261, 262, 263, 264, 550, 273, 554, 215, 377]. It is shown that almost all possible spatial phenomena and effects of curvature and dispersion on wave propagation, e.g. wobbling cores, scroll rings, linked pairs of scroll rings and ripples, obtained in continuous models can also be described, to some degree of accuracy, in far more tractable discrete or cellular-automata models.

Cellular-automata models of reaction–diffusion media exhibit the same spectrum of coherent patterns as numerical simulations of reaction–diffusion equations: travelling waves, spiral waves, stable spots, strips and labyrinths [425, 200].

A strong agreement between the numerical solutions of partial differential equations and cellular-automata models is demonstrated in [130]. The accuracy and performance of three-dimensional cellular automata for the simulation of continuous excitable media is demonstrated in [300].

One realistic cellular-automata model for simulating the spread of excitation during ventricular fibrillation under a wide range of conditions is designed in [344]. The model reproduces ‘electrocardiograms’ similar to real data in their time series.

In the case of a Belousov–Zhabotinsky reaction, probabilistic cellular automata—in the form of networks of stochastic oscillators—allow the simulation of turbulent stirring and of a stochastic Oregonator [619].

Let us now tackle some peculiarities of cellular-automata-based simulation.

Cellular-automata models of reaction–diffusion systems phenomenologically correspond to the solution of partial differential equations, as do most of our models presented in this book, qualitatively [163, 632] or quantitatively [634, 635].

Here is an example. In [218] a cellular-automata model is used to investigate the stability of vortex rotation. In this model each element of a two-dimensional array takes a finite number of states: 0 (resting state), 1 (excited state), 2, \ldots, \rho_a\text{ (absolutely refractory states)}, and \rho_a + 1, \ldots, \rho \text{ (relative refractory state). The values } \rho_a \text{ and } \rho - \rho_a \text{ express the duration of the absolute refractory and relative refractory periods. Every cell } x \text{ of the array updates its state in the following}
manner:

\[ x_{t+1}^i = \begin{cases} 
  x_t^i + 1 & \text{if } x_t^i \in [1, \rho_a] \text{ or } (x_t^i \in (\rho_a, \rho) \text{ and } \sigma(x_t^i) \leq \theta(x_t^i)) \\
  1 & \text{if } (x_t^i = 0 \text{ or } x_t^i \in (\rho_a, \rho)) \text{ and } \sigma(x_t^i) > \theta(x_t^i) \\
  0 & \text{if } (x_t^i = 0 \text{ or } x_t^i = \rho) \text{ and } \sigma(x_t^i) \leq \theta(x_t^i) 
\end{cases} \]

where \( \sigma(x_t^i) = \sum_{y \in u(x)} w_{xy}(y_t) \) is a weighted sum of states of cell \( x \)'s neighbours. In the model the threshold \( \theta(x_t^i) \) of cell excitation is a function of cell state. We can use any kind of weighting of the neighbourhood. For example, a linear dependence seems to be good enough. If the neighbourhood \( u(x) \) looks like \( u(x) = \{ y : |x - y| \leq r \} - x \) then we can define the weight \( w_{xy} \), for example, as \( w_{xy} = r - |x - y| + 1 \), where \( r \) is the neighbourhood radius. The radius \( r \) determines the range of velocities in the dispersion curve; however, large values of \( r \) certainly slow down the simulation [218].

In the previous model we noticed that the excitation threshold depends on the current values of the medium site. This is because the threshold may be determined by a decreasing linear function with domain \((\rho_a, \rho)\), the closer cell state is to the resting state the lower the excitation threshold. Using such a model we can investigate most of the properties of excitable media, e.g. the dependence of velocity on wavefront curvature [677], observed in laboratory experiments and in numerical models of partial differential equations (see [218] and [261, 262]).

Anisotropy of wave propagation is visible in lattices with eight-cell neighbourhoods and even more recognizable in lattices with four-cell neighbourhoods. Thus in the eight-connected lattice the wavefront velocity in the direction of the cell diagonal is \( \sqrt{2} \) higher than in the direction of the cell edge [426]. How do we cure the anisotropy? The problem of the discretization of a diffusion equation was addressed by the image-processing community long before this happened in physics: various types of local neighbourhoods and time delays were exploited (see [141] for relevant references).

One of the easiest approaches is simply to delete some lattice nodes at random; we need to increase the neighbourhood size in this case. It is shown that a square lattice, where a node is deleted with the probability 0.5, and a hexagonal lattice, where a node is deleted with the probability 0.3, are sufficient to obtain isotropic wave propagation [377].

Another approach to randomization is discussed in [426]. There some type of dynamical randomization of the cell neighbourhood is applied. Some neighbours of every cell are omitted at the stage when the cell counts the number of excited neighbours to calculate its own next state. With large neighbourhoods and several gradations of refractory states we obtain very attractive wavefronts [426].

A systematic investigation of the different kinds of neighbourhood templates, or masks, is provided in [633]. Several neighbourhood templates are under consideration there:

- Efimov–Fast weighted templates, where a neighbour \( y \) contributes to the sum
of ‘excitation’ of cell \(x\) with weight \(w(y, x) = \lfloor ce^{-\frac{d(x,y)^2}{2b}} \rfloor\), where \(c\) and \(b\) are constants and \(d(x, y)\) is the distance between the cells \(x\) and \(y\):

- the classical cruciform template;
- the square template: \(w(x_{ij}, x_{lm}) = \frac{1}{(2r+1)^2}\) if \(l, m \leq r + \frac{1}{2}\);
- the circular template: \(w(x_{ij}, x_{lm}) = \frac{1}{2\pi r^2}\) if \(l^2 + m^2 \leq r^2\); and
- the Gaussian template: \(w(x_{ij}, x_{lm}) = \frac{1}{2\sqrt{\pi} b} e^{-\frac{l^2 + m^2}{4b}}\)

where \(i, j, l\) and \(m\) are the cell indices of a two-dimensional lattice. Weimar et al. [633] demonstrate that the velocity–curvature relation \(n = v - DK\) between the normal velocity of the wave \(n\), wave velocity \(v\), diffusion coefficient \(D\) of an excitable variable and wavefront curvature \(K\) holds quite well for most types of weighted neighbourhood templates.

In cellular-automata models of reaction–diffusion systems the diffusive part, propagation and scattering, is implemented via modified finite difference schemes and the reactive part is usually realized probabilistically [634]. Moreover, the diffusive part may utilize the technique of a moving average in which the discrete Laplacian is calculated.

Weimar et al. [632] offer a classification for cellular-automata models of excitable media. Three generations of models are put forward. The first generation includes cellular automata with nearest neighbours, four-cell and eight-cell neighbourhoods, and very few cell states. First-generation models are mostly used in our book. The second generation consists of automata with larger, and usually weighted, neighbourhood templates and a large set of cell states. The models in this class can deal with those aspects of wave dynamics, which are not tackled well by the first-generation models [218, 261, 263, 425, 426]: the curvature effect (velocity depends on wavefront curvature), the dispersion relation (takes into account the gradations of recovery, i.e. relative refractory periods) and the spatial isotropy (smoothness of the wavefront shape—this factor is cured by extending the cell neighbourhood). The curvature effect is achieved by increasing the cell neighbourhood size. The dispersion relation is simulated by including several recovery states in the cell state set; a cell in the recovery state has a higher excitation threshold than a cell in the resting state. And anisotropy is avoided by randomization of either lattice or neighbourhood. The third generation goes back to classical numerical algorithms and combines the accuracy of difference schemes with the computational efficiency of cellular automata [632]. The combination of techniques used in the third-generation models resembles Barkley’s technique [67], which will be discussed later.

Coupled map lattice models, in contrast to cellular automata, employ continuous states but still retain discrete time and discrete space [341]. Every node of a coupled map lattice updates its state by the rule:

\[ x^{t+1} = f(x^t) + \sum_{y \in n(x)} g(y^t) \]
where \( f(\cdot) \) is the function responsible for cell state transition or the local dynamic of a cell, when no coupling with neighbours exists (autonomous evolution); \( g(\cdot) \) is a linear or nonlinear function of the states of cell \( x \)'s neighbours [341]. The spacetime dynamics of coupled map lattice excitation models is very rich and varies from chaotic to expanding spatial domains [341]. However, we could not find anything new in the dynamics of coupled map lattices that has not already been qualitatively described in cellular-automata models.

In 1991 Barkley offered a simple hence efficient hybrid of numerical methods and cellular automata [67]. His technique explicitly employs reaction–diffusion equations and allows continuous adjustment of spacetime resolutions [67]. Now the method is used widely.

Let us discuss the Barkley model [67]. It employs a two-variable system of reaction–diffusion equations, which simulates the dynamics of a excitable medium:

\[
\frac{\partial u}{\partial t} = f(u, v) + \nabla^2 u \quad \text{and} \quad \frac{\partial v}{\partial t} = g(u, v).
\]

The functions \( f(\cdot) \) and \( g(\cdot) \) represent the dynamics of the variables \( u \) and \( v \), respectively. The component \( v \) is assumed not to diffuse. This may be appropriate when one of the reagents is immobilized. The functions are described as follows:

\[
f(u, v) = \epsilon^{-1} u(1 - u)[u - u_{th}(v)] \quad \text{and} \quad g(u, v) = u - v
\]

\[u_{th}(v) = \frac{a + b}{a}, \epsilon \text{ is very small, } a \text{ and } b \text{ are parameters.}\]

Sometimes, the variables \( u \) and \( v \) are called the excitation and recovery variables. In the excitation region, the time scales of the variable \( u \) are very fast, therefore either very small or very large time steps should be used to prevent instabilities [67]. Moreover, a simulation must take into account the fact that the recovery region of the system is very simple; we can think of an exponential decay of the variable \( v \). As in lattice gases and cellular-automata models of reaction and diffusion two essential steps are considered: the local dynamic and diffusion. The local dynamic of the variables is simulated as follows:

\[
\text{if } u^t < \delta \quad u^{t+1} \leftarrow 0, \quad v^{t+1} \leftarrow (1 - \Delta t) v^t \\
\text{else } u_{th} \leftarrow (v^t + b)/a, \quad v^{t+1} \leftarrow v^t + \Delta t (u^t - v^t), \quad u^{t+1} \leftarrow F(u^t, u_{th}).
\]

The values of the variables at the \( t \)th time step are shown as \( u^t \) and \( v^t \); \( \Delta t \) is the time increment. A special region of the \( (u, v) \) phase space, where no excitation occurs and a site in the excitable medium simply returns to its resting value after a small perturbation, is bounded by \( \delta \). Function \( F \) is responsible for the excitation dynamic. Barkley [67] discusses several ways to represent this function. Thus, for example, for large time steps we have

\[
\text{if } u^t \leq u_{th} \quad u^{t+1} \leftarrow u^t + (\Delta t/\epsilon) u^{t+1} (1 - u^t)(u^t - u_{th}) \\
\text{else } u^{t+1} \leftarrow u^t + (\Delta t/\epsilon) u^t (1 - u^{t+1})(u^t - u_{th}).
\]
For small $\Delta t/\epsilon$ we can use the following explicit form of the function $F$:

$$F(u', u_{th}) = u' + (\Delta t/\epsilon)u'(1 - u')(u' - u_{th}) + O(\Delta t^2)$$

while for large $\Delta t/\epsilon$, which is quite similar to classical cellular automata, we obtain

$$F(u', u_{th}) = \begin{cases} 
0 & \text{if } u' < u_{th} \\
 u_{th} & \text{if } u' = u_{th} \\
 1 & \text{if } u' > u_{th}.
\end{cases}$$

If we do not worry about a slight slowdown in computation speed we can use the following approximation of the Laplacian to simulate the diffusion of the excitation component $u$:

$$h^2 \nabla^2 u_{i,j} = u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}.$$ 

Several techniques to speed up the simulation are discussed in [67]. The method is tested on three-dimensional excitable media in [195]. In the three-dimensional model a reactive step of the simulation in [195] is similar to that in the two-dimensional model [67] while the diffusive process requires some attention; i.e. the seven-point finite-difference scheme is adopted for modelling three-dimensional diffusion:

$$h^2 \nabla^2 u_{i,j,z} = u_{i+1,j,z} + u_{i-1,j,z} + u_{i,j+1,z} + u_{i,j-1,z} + u_{i,j,z+1} + u_{i,j,z-1} - 6u_{i,j,z}.$$ 

Neither difference schemes nor cellular-automata models nor even a combination of them take into account the behaviour of a single molecule or a particle. Lattice-gas automata do.

Lattice-gas models imply that discrete particles spread on the integer lattice in integer time and undergo discrete collisions with each other, with a finite number of possible outcomes. A lattice-gas model is invented in [290, 246, 247] to simulate fluid dynamics efficiently (see also fluid flow in cellular automata in [646]). Since then lattice-gas automata have been employed in a myriad of simulations from multiphase flow and the Dirac equation [450] to the Schrödinger [92] equation and reaction–diffusion systems [98]. Recently, Hasslacher and Meyer [292] have produced a remarkable example of a dynamically evolving model, where they combine a lattice gas with structurally dynamic cellular automata.

How is the lattice-gas model organized? If there is no reaction between the gas molecules only two instead of three phases, which will be discussed later, must be simulated. The first phase is an advection: every particle is transferred to a lattice node depending on its previous position and momentum. The second phase is scattering: the particles at the lattice nodes are scattered by deterministic, probabilistic or quantum-mechanical rules [292].

The term lattice chemistry is derived in [163, 98]. This combines hydrodynamics with chemistry and is expressed in terms of lattice-gas automata.
More generally, lattice chemistry offers a microscopic approach to the dynamics of spatially distributed reacting systems where hydro-dynamics and statistical mechanics are also exploited.

Lattice chemistry obviously operates on two- and three-dimensional lattices, where molecules are seen as point particles, which travel or undergo displacements along the links connecting lattice nodes, in the lattice. The particles travel with discrete velocities; i.e. they hop from one node to a neighbouring one, determined by the particle’s velocity vector \([163, 98]\). The total number of particles at a node is usually restricted by lattice connectivity. This is an exclusion principle. There are also lattice chemistry models without exclusion \([98]\); they accept that any number of particles can reside at a lattice node, thus particles can simply diffuse on the lattice and react with one another \([134]\).

Given a lattice with particles an evolution is implemented by application of an evolution operator \(E\). The evolution operator is equivalent to the cellular-automaton global transition rule. Typically, the evolution operator \(E\) is represented by a composition of three basic operations \([163]\):

\[
E = C \circ R \circ P
\]

with propagation operator \(P\), velocity randomization operator (collision operator) \(R\) and chemical transformation, or reaction, operator \(C\). The operators \(P\) and \(R\) are analogues of the advection and scattering operators in classical lattice-gas models \([246, 247]\).

When the propagation operator \(P\) is applied each particle jumps from one node to one of its neighbouring nodes in a manner determined by its velocity vector.

The collision operator \(R\) randomizes the velocities of the particles at every node. The velocities are randomly shifted. This simulates elastic collisions between the particles and molecules of an implicit solvent. Velocity randomization can be defined via state transition probability matrices.

As a result of applying the chemical transformation operator \(C\) the particles react with one another. Particles are created and annihilated at every node: \(\alpha X \rightarrow \beta X\). There is no conservation of particles and their local momenta. The difference between \(R\) and \(C\) is that \(R\) conserves the number of particles while \(C\) does not.

If there is no reaction all particles execute discrete random walks because only operators \(P\) and \(R\) are applied. This is a diffusion.

Which real computers are suitable for the discussed models? Amongst many possibilities we would like to mention two types of massively parallel computing devices.

The first one is a connection machine \([302]\). Its architecture resembles a globally connected automata network, where millions of very primitive processors communicate via the spread of diffusive waves of packets. Each packet includes data and a recipient address. The packets are accepted by those
processors with the appropriate address while other packets are simply sent away to other processors.

Cellular-automata machines [591] are the second example. A cellular-automata machine is a multiprocessor board with a large graphical memory and a graphic accelerator. Such an architecture does not imitate cellular-automaton architecture but allows a reasonably fast run and, in particular, displays the evolution of large cellular-automata arrays. Recent versions of the cellular-automata machine, as well as recent versions of the connection machine, have become more similar to widespread multiprocessor systems such as, for example, SPARC, for the sole reason that end-users were not prepared to program cellular-automata processors and quite a few problems with straightforward data parallelism were known at that time.

1.5 Computing in nonlinear active media

Most of the problems with natural spatial parallelism can be efficiently solved in nonlinear active media. A problem has a natural parallelism if it consists of many subdomains that can be, at least up to some limit, analysed independently of each other and the results of analyses are combined.

A problem is spatially parallel if sites of the problem space can be updated in parallel, i.e. distant sites can be analysed independently at the same time step. [593]

The so-called programmable matter paradigm gives a commonsense approach which may be used in the design of unconventional computers for problems with natural spatial parallelism. Programmable matter architectures are local and uniform. They are finely grained and massively parallel. The computational tasks that can be solved on programmable matter computers are mainly spatial tasks with nonlinear interactions between a large number of elements, which results in the emergence of macroscopic phenomenology from microscopic mechanisms [593]. So, in fact, the dynamic of the computer simulates, or even directly corresponds to, the dynamic of the simulated problem. We, therefore, have the mapping

\[
\text{spatial parallel problem} \rightarrow \text{programmable matter architecture}.
\]

Both the connection machine [302] and the cellular-automata machine [591] are programmable matters [593]. In this section we consider several examples of wet-ware realizations of programmable matter. When thinking about the reaction–diffusion and excitable computers, as well as all prototypes of real chemical processors discussed in this book we must remember that no competition between wet-ware and conventional silicon hardware can even be imagined at this stage. As Toffoli reasonably notes:
programmable matter architectures play a complementary rather than competitive role vis-à-vis traditional architectures; they are tuned, as it were, to a different band of the computational spectrum. [593]

However, cellular-automata models of reaction–diffusion and excitable computers, if implemented in silicon, may certainly overrun not only classical serial but also existing parallel architectures.

Kuhnert was probably the first to design a laboratory prototype of a reaction–diffusion chemical processor. He developed a recipe for a light-sensitive modification of the thin-layer Belousov–Zhabotinsky reaction and showed how the reaction medium can be used in image-processing [375, 376]. The idea was developed further by Agladze, Krinsky, Rambidi, Aliev and their collaborators [376, 368, 514, 515, 516, 517, 519, 520, 522, 523, 524, 525, 526, 527]. The reaction–diffusion approach to image-processing has also been implemented in mathematical models [50, 674, 507, 508, 155]. Thus, for example, Alvarez and Esclarin [50] use the diffusion component of the reaction–diffusion partial differential equations to reduce the noise in data images; this is an analogue of averaging or homogenization of the image. That is, in the Alvarez and Esclarin [50] model diffusion is used to filter noise and the reaction is employed to enhance contrast.

In general, image-processing in a Belousov–Zhabotinsky medium may be implemented as follows [376, 368, 514, 515, 516, 517, 519, 520, 522, 523, 524, 525, 526, 527]. An image is projected onto a reaction layer. As a result of such a projection a non-uniform oscillation phase distribution is generated in the medium. The distribution represents image features, i.e. encoded data. After that image-processing is implemented because of the interaction between the phase and diffusion waves.

What exactly is happening in excitable chemical media when an image is projected onto a thin-layer reactor? When an image is projected onto a photosensitive excitable chemical medium, the micro-volumes, which become illuminated above some threshold, become excited. Excitation waves start to spread in the medium and interact with each other. By changing the parameters of wave generation and propagation it is possible to implement a wide range of image-processing operations, for example, the restoration of broken contours, edge detection and contrasting [376, 368].

Here we consider several examples of image-processing in Belousov–Zhabotinsky thin-layer chemical reactors. They include the detection of closed curves and the restoration of a broken contour [368, 518] as well as image contouring, enhancement and detection of certain image features [528].

An algorithm for the detection of the closeness of a curve works as follows [368]. We project a curve onto an excitable medium. The sites corresponding to points of the curve become non-excitable or ever-resting. We make the curve sites ever-resting by physical intervention in a medium’s characteristics. Then we excite an arbitrary site of the medium far from the projected curve. If the
curve is not closed the excitation fills the whole lattice. If the curve is closed the sites inside the curve remain at rest. Obviously, we need to ‘scan’ all medium sites to check whether the whole medium is excited or not. The recognition of closed curves from open curves has also been implemented in a two-dimensional array of cellular neural networks, i.e. resistively coupled Chua circuits, using the propagation of excitation waves [496].

The next solution is even more tricky.

A broken contour is projected onto the excitable medium. We want to restore a contour’s continuity. Sites belonging to the contour are excited and an excitation wave propagates in the medium. The broken edges of the contour, represented by excitation waves, expand and finally merge. As a result we have some kind of inflated but restored shape, see figure 1.4 [518]. Its thickness may be later reduced, either by conventional procedures or by back propagating the excitation waves and stopping them at a certain time. Again, you see, the solution is not in the exact form we wanted [518].

The two previous examples are rather playful instances of very simple operations. Could we do something more sophisticated with complex images. A series of works published by Rambidi et al [514, 515, 516, 517, 519, 520, 522, 523, 524, 525, 526, 528] gives us a positive answer.

The typical setup of a Belousov–Zhabotinsky thin-layer processor includes a chemical reactor and a computer-controlled video projector. A light-sensitive catalyst Ru(bpy)$_3$Cl$_2$ is immobilized on a thin, approximately 0.2 mm, layer of a silica gel, placed at the bottom of a reactor. The gel film is covered by a 0.5–1.5 mm thin non-stirred layer of reaction liquid with the following concentration of reagents: KBrO$_3$ (0.3 M), H$_2$SO$_4$ (0.3–0.6 M), malonic acid (0.2 M) and KBr (0.05 M). Image data are ‘loaded’ in the chemical processor by light radiation, i.e. by projecting an image onto the thin layer surface. As indicated in [528] it is very handy to immobilize a ferroin catalyst, in a low concentration, on the gel film surface, after immobilization of the ruthenium catalyst; this would increase the optical density of the processed images.

At the beginning of an experiment a medium is illuminated by a white light to reboot the chemical processor initially. This causes an increase in the concentration of Ru$^{3+}$. The medium therefore becomes light coloured. Then the data image is projected onto the medium. The concentration of Ru$^{3+}$ in those parts of the medium, which correspond to white fragments of the image, becomes
higher than in other parts of the medium. At the same time, the concentration of Ru\(^{2+}\) increases and the concentration of Ru\(^{3+}\) decreases in darkened parts of the medium. Due to the particulars of the reaction an oscillating process starts in the medium. This leads to an increase in Ru\(^{3+}\) in previously darkened areas of the medium and a decrease in the reagent in the previously white (illuminated) area. As a result, the image, represented in a reagent concentration, becomes negative. An opposite evolution happens in the medium when we project a negative image. Several snapshots of the evolution of a black and white image in a Belousov–Zhabotinsky medium are shown in figure 1.5.

By varying the contrast of the original image, projected onto a Belousov–Zhabotinsky reactor, we could obtain either the external contour of the image (figure 1.6(A)) or a typical contour of the image (figure 1.6(B)). If a negative image is processed, the basic shape of the image is enhanced (figure 1.6(C)). The projection of a positive image onto the medium accentuates particular features of the image (figure 1.6(D)) [528].

Could we ‘re-program’ Belousov–Zhabotinsky processors or control the computation process? In general, no. However, there are some hints which may allow us to modify the properties of the media during the computation. Thus, for example, we could apply an electromagnetic field.

A spacetime electric field gradient affects the transport of ionic components. Therefore the field gradients may influence the wavefront dynamic. A negative
field accelerates the wave. In positive electrical fields the propagating front may slow down or even stopps. Moreover, the wave may start moving backwards if the
field is strong enough [421, 422]. When the field is switched off the front recovers
and continues its propagation, probably with changed parameters. Varying the
electric field applied to the chemical medium we can achieve transitions between
several regimes of the reagents’ spacetime dynamic [473, 213]. So, the electric
field can be used for data inputs, control of computation process or even dynamic
re-programming of Belousov–Zhabotinsky reaction-based computing devices.

Reaction–diffusion media can handle image-processing. Do they cope with
image recognition? Yes, at least at a very primitive stage. Chemical processors
may recognize patterns because they react differently to different input patterns
and similarly to similar ones. This is demonstrated in laboratory experiments on
patterns recognition by a small network of chemical reactors [384]. Each reactor
is a continuous flow stirred tank. Its states are bistable. The reactors are mass-
coupled with each other. The iodate-arsenous acid reaction is used (shown for the
case when the iodate is in stoichiometric excess) [384]:

\[
2\text{IO}_3^- + 5\text{H}_3\text{AsO}_3^- + 2\text{H}^+ \rightarrow \text{I}_2 + 5\text{H}_3\text{AsO}_4^- + \text{H}_2\text{O}.
\]

Each reactor of the network takes binary states of low and high concentrations
of reagents. The states of the reactors are visualized by adding starch: the high
iodine state is blue and the low iodine state is colourless. The state of a reactor is
a pixel of the analysed pattern.

Each reactor in the network is fed with an identical flow of reagents. Such a
flow rate is taken to mean that the reactor is in a bistable state of either high or low
iodine concentration. The chemical processor is programmed by modifying the
network connectivity strength. The reactor network of this chemical processor is
programmed by choosing such a flow rate between the reactors that store binary
pattern which correspond to the network’s stable state attractor [384]. The stored
patterns can be evoked by applying similar initial configurations.

In mass-coupling the coupled reactors must be spatially close to each other.
Instead of mass-coupling we can also employ electrical coupling between the
reactors of the network [309].

Results from pattern recognition have been developed into logical functions
via chemical kinetics [306, 307] and chemical implementation of finite automata
and neural networks (see, for example, [304, 305]). These works will be discussed
in the main body of the book, so we omit detailed discussion here.

The Babloyantz–Sepulcher algorithm [56] for finding the shortest obstacle-
free path using an oscillatory two-dimensional medium is another brilliant
example of pioneering results on reaction–diffusion computing. The model
consists of a two-dimensional lattice of oscillators [56]. The behaviour of each
oscillator is represented by the variables \(x\) and \(y\) and is described by either the
kinematic equations of the Brusselator:

\[
\dot{x} = a - (b + 1)x + x^2y \quad \text{and} \quad \dot{y} = bx - x^2y
\]
or the Maginu scheme

\[
\dot{x} = x - y - \frac{x^3}{3} \quad \text{and} \quad \dot{y} = \frac{x - ky}{b}
\]

where \(a\) and \(b\) are abstract quantities and \(k\) is the threshold for bifurcation.

It is reasonable to describe the whole oscillator network by the following classical equations:

\[
\begin{align*}
\dot{x}_{i,j} &= f(x_{i,j}, y_{i,j}) + c(x_{i+1,j} + x_{i,j+1} + x_{i-1,j} + x_{i,j-1} - 4x_{i,j}) \\
\dot{y}_{i,j} &= g(x_{i,j}, y_{i,j}) + c(y_{i+1,j} + y_{i,j+1} + y_{i-1,j} + y_{i,j-1} - 4y_{i,j})
\end{align*}
\]

where \(f(\cdot)\) and \(g(\cdot)\) are functions that describe the local kinetics of the oscillatory elements, and \(c\) is a constant coupling coefficient between the oscillators.

Initially, all oscillators are in an unstable rest state. To start the wave we perturb the lattice by changing the state of a few oscillators of the lattice. An oscillation front is generated by the perturbed elements. The front propagates in the medium. A wave propagates behind the front. Incoming, or backpropagating, waves are also formed. This leads to periodical generation of circular waves by the once perturbed region of the lattice. In [56] front propagation in a continuous medium is simulated by approximating the kinetics of the oscillator lattice by the Ginzburg–Landau equation for the complex amplitude \(w\):

\[
\frac{\partial w}{\partial t} = (1 + i\omega_0)w - (1 + i\beta)|w|^2w + D\nabla^2w
\]

where \(\omega_0\) is the initial frequency, \(\beta\) the dispersion coefficient and \(D\) the diffusive coefficient [56].

The main point of the oscillatory network is that if such a system is perturbed the front propagates from the point of perturbation. The front velocities indicate from where the front came and where the initial perturbation is. This idea will be exploited throughout this book. As to the Babloyantz–Sepulcher experiments [56] the following problem is solved: a robot travels in a room with obstacles and an oscillatory network guides the robot toward a target. So the nonlinear medium finds the shortest collision-free path. We associate the physical space of the problem with a network of oscillators, which corresponds, in turn, to the Ginzburg–Landau equation. The equation shows the propagation of waves that travel in the opposite direction to the propagating front. Obstacles are represented by absent oscillators or empty nodes on the grid. The target is assumed to be a signalling centre or pacemaker. Once the target region is perturbed the wavefronts propagate in the lattice. Eventually they reach the robot. The travelling waves start to propagate behind the front and toward the target. As a result the periodic propagation of the waves allows the robot to find a pacemaker centre and, therefore, to hit the target [56].
1.6 Reaction–diffusion media with smart molecules

Chemical media are not the only ones described by the reaction–diffusion equations and thus there are other candidates for the material base of reaction–diffusion algorithms. Sometimes even crowds, especially in extreme situations of panic or excitement, act according to the laws of physics [297], partly due to the absence of self-control, anonymity, suggestibility and the presence of a collective mind [564, 184, 345]. When moving intelligent entities employ mainly self-propulsion, the control of individual motion is by local interaction with other individuals and randomness of environment [625, 297, 298].

Attractive analogies between reaction–diffusion, morphogenesis and pattern formation in social insect societies are provided in Bonabeau et al [94, 95]. They show that the distribution of dead ant bodies in ant cemeteries—the distribution itself is governed by pheromone concentrations—is well described by a system of reaction–diffusion equations. The dynamics of opinion formation is another field where nonlinear physics is flourishing [336, 502]. The new generation of physics-based models of opinion formation considers the spacetime dynamics of individual opinions; thus, e.g., Plewczyński [502] describes social change in a collective with a lattice topology, and that of individuals by a nonlinear Schrödinger equation by analogy respectively with a superfluid and a weakly interacting Bose gas in an external potential.

Despite the complex nature of human and animal communication some primitive aspects of their collective behaviour can be described by mathematical models of chemical communications and still bear great predicting power [454]. Dictyostelium discoideum forms an impressive example of such primitive models. There the collective behaviour of myriads of simple organisms is phenomenologically similar to the processes occurring in a Belousov–Zhabotinsky reaction, including the wave motions (see e.g. [420, 194, 471]). Another good example deals with an activation in an ant colony. Ant colonies show global oscillations of activity levels, i.e. changes in the number of active workers. The activity spreads concentrically from the points of a disturbance [571].

In populations of cells or insects individuals may release chemical mediators. These mediators diffuse in the medium. The chemical composition of the mediators represents an activity pattern of the population. In some assumptions, such activity patterns may be considered to be a representation of public opinion or distributed knowledge in the system [454]. Moreover, we can design a reaction-like interaction between doxastic states of each individual in the population and investigate the spacetime dynamics of the population in stirred and non-stirred reaction mixtures [29, 30].

The molecular dynamic applied to societies of social insects may produce quite original computation algorithms in collectives of dumb agents. Computing with social insects forms the new field of swarm computing and ant-colony optimization. The subject is extensively discussed in [97]. Specific applications...
in graph optimization and telecommunication networks may be found in [139, 191, 252, 22, 552, 192, 100, 415]. Development of the social insect paradigm in collective robotics, particularly in collective sorting and transport, can be found in [76, 446, 371, 447].