BUILDING EFFICIENT COMPUTATIONAL CELLULAR AUTOMATA MODELS OF COMPLEX SYSTEMS: BACKGROUND, APPLICATIONS, RESULTS, SOFTWARE, AND PATHOLOGIES

JIŘÍ KROC[*,¶], FRANCISCO JIMÉNEZ-MORALES[†,‖], J. L. GUISADO[‡,**], MARÍA CARMEN LEMOS[†,††] and JAKUB TKÁČ[§,‡‡]

[*]*Biomedical Center, Faculty of Medicine in Pilsen, Charles University, Alej Svobody 76, Pilsen 323 00, The Czech Republic*

[†]*Department of Condensed Matter Physics, University of Seville, Avda. Reina Mercedes, S/N, Seville 41012, Spain*

[‡]*Department of Computer Architecture and Technology, University of Seville, Avda. Reina Mercedes, S/N, Seville 41012, Spain*

[§]*Faculty of Electrical Engineering, Czech Technical University, Technická 2, Prague 166 27, The Czech Republic*

[¶]*dr.j.kroc@gmail.com*
[‖]*jimenez@us.es*
[**]*jlguisado@us.es*
[††]*lemos@us.es*
[‡‡]*tkacjaku@fel.cvut.cz*

Received 17 July 2018
Revised 13 August 2019
Accepted 11 October 2019
Published 12 December 2019

Cellular automaton models of complex systems (CSs) are gaining greater popularity; simultaneously, they have proven the capability to solve real scientific and engineering applications. To enable everybody a quick penetration into the core of this type of modeling, three real applications of cellular automaton models, including selected open source software codes, are studied: laser dynamics, dynamic recrystallization (DRX) and surface catalytic reactions.

The paper is written in a way that it enables any researcher to reach the cutting edge knowledge of the design principles of cellular automata (CA) models of the observed phenomena in any scientific field. The whole sequence of design steps is demonstrated: definition of the model using topology and local (transition) rule of a cellular automaton, achieved results, comparison to real experiments, calibration, pathological observations, flow diagrams, software, and discussions. Additionally, the whole paper demonstrates the extreme expressiveness and flexibility of massively parallel computational approaches compared to other

[¶]Corresponding author.

computational approaches. The paper consists of the introductory parts that are explaining CSs, self-organization and emergence, entropy, and CA. This allows readers to realize that there is a large variability in definitions and solutions of this class of models.

Keywords: Cellular automata; complex systems; modeling.

1. Introduction

Complex systems (CSs) are a large variety of systems arising in many different areas of natural and social sciences, characterized by being formed by a collection of simple components which interact locally and exhibit emergent global behaviors or features that result from the actions of its parts rather than being imposed by a central controller [3, 5]. Examples from different fields are the immune system (medicine), ant colonies (biology), reaction-diffusion systems (chemistry) or laser systems (physics). CSs are difficult to model with the usual mathematical tools of differential equations, because interactions between their components change states of these components and, in turn, the collection of states of the components changes the interactions between them [27]. Therefore, other mathematical tools must be employed to predict their behavior. One of the most useful mathematical tools to model CSs are cellular automata (CA).

CA are a class of mathematical systems with the following characteristics: spatial and temporal discrete character, local interaction and synchronous parallel dynamical evolution. CA can be used to successfully model CSs overcoming the difficulty posed by differential equations because they follow an algorithmic approach, allowing to model the evolution of each individual component of the system. Among various models used to simulate CSs, such as agent-based, lattice-Boltzmann or multiscale models, CA stand out for their simplicity, locality and parallelism. CA were developed during the early 40s and they often serve as a prototypical model of a CS [26, 29]. One the biggest advantages of CA models is the possibility to associate (un)known low-level, constituting processes with the global response of the system through simulations. This inherent modularity of local, low-level processes, which are interacting locally, applied within CSs enables one to associate otherwise highly unpredictable dependencies between low-level processes and the global response of the system. We can simply switch on/off processes, change their intensity and frequency, and modulate them in many ways. This feature of CA models enables us to study real CSs in silico representations in a very efficient way.

Therefore, CA have been used to build models of CSs of very different types [8, 26], for example in: fluid dynamics [16], reaction-diffusion processes [2], magnetization in solids [4], economy [44], geology [45], etc. Currently they are being extensively used in particular in fields such as geography (in applications such as urban development planning [15], future development of cities [1], and land use [36]) pedestrian or vehicular traffic [43, 52], and bio-medicine (with applications such as physiological modeling, for example for cancer [39], or epidemic modeling [7]).

The main purpose of this paper is to provide an introduction on CA modeling to everyone. The paper provides a concise background in design of CSs models on a small, carefully selected set of examples that are originating in the fields of physics and chemistry (e.g., photons in laser, dynamic recrystallization (DRX) of polycrystals, and chemicals in catalysis). Two different instances of software that simulates self-organization are provided in the paper along with links to the open source codes (DRX [55] and splitting a domain on equally sized sub-domains [40] (not discussed)). The software is aimed to those who would like to start to program their own CS-code but need some good examples of it. Videos describing functioning of the software accompany the source codes [41, 56].

2. Mathematical and Computational Background

This section provides a quick and concise introduction into the CA modeling of CSs for newcomers.

2.1. *Complex Systems modeling: selected applications*

CSs modeling represents a reliable, bottom-up build tool (often the only tool) capable to describe complicated and naturally observed phenomena that occur at the cutting edge of many scientific disciplines. First, it is very difficult to reach a deep understanding of the CS-models design. One of the best ways is to provide several carefully selected examples enabling an easy penetration into the core of CS-modeling. Last, a very important part of every reliable CS-modeling process is the quantification of CSs properties of those models: it is often achieved by the use of entropy measures.

In this paper are studied three models originating in different scientific areas, which are covered in Secs. 5–7:

(i) Laser dynamics model simulates the creation of a laser beam from the interaction of molecules inside the laser device material and laser photons through physical processes: stimulated emission, decay of electrons and photons, pumping of energy and creation of a photon noise. Experimentally observed laser characteristics and operational modes and its dependence on parameter values are computationally reproduced: pumping threshold, laser oscillatory and constant behavior, see Sec. 5.

(ii) DRX model simulates the deformation of low stacking fault polycrystalline metals under elevated temperatures. Single and multiple peak deformation curves and their dependence on strain rate, temperature, and ratio of the initial grain-size to the steady-state one are simulated using an interplay between feeding system by energy due to its mechanical deformation and erasing energy by recrystallization due to growth of new undeformed grains, see Sec. 6. Additionally, possible pathological behaviors of the model are demonstrated. Open-source software [54] and video [56] available.

(iii) Chemical reactions model can simulate the catalytic oxidation of CO on a metal surface from interactions among atoms and molecules which are adsorbed on the surface of the catalyst. Different physical and chemical processes take place: adsorption, reaction, desorption, and diffusion. Oscillatory regimes, that have been experimentally observed, such as chaotic or quasi-periodic behavior, are computationally reproduced in dependence on values of kinetic coefficients. Also the spatio-temporal patterns arising on the surface reaction can be simulated, see Sec. 7.

2.2. *Entropy measures*

In general, CSs continue to be poorly understood due to their complicatedness and our inability to describe them in their totality by using available mathematical tools. Entropy measures demonstrated themselves as very useful tools enabling us to get insight in, to quantify, and to distinguish various states and regimes that are operating within CSs [9, 11]. Despite their imperfectness, entropy measures are enabling to open windows going deep inside of CSs, to measure their basic states and properties, and to overcome a lack of theoretical predictive tools.

There is existing a whole range of entropies that can be used to describe CSs: original entropy used in statistical physics based on distribution of microstates developed by Boltzmann, information entropy asessing information content in the data streams (messages), approximate, fractal, permutation, Kolmogorov–Sinai, multiscale, multivariate, sample, alphabetic entropies, and many others [11].

An often used entropy measure is the Shannon entropy $S(\alpha)$ [47], which is sometimes called information entropy, for some fixed parameter α defined by the equation

$$S(\alpha) = -\sum_{i=1}^{m} p_i(\alpha) \cdot \ln\ p_i(\alpha). \tag{1}$$

Shannon entropy of the distribution of values of a variable $X(t)$ takes the whole range of values covered by the variable $X(t)$ and divides into m equally spaced bins. The probabilities p_i (later called as well frequencies f_i) defined by the relative number of cases where $X(t)$ is located inside every particular non-void bin are computed. There is no fixed criterion to select the value of m but the level up to which you want to discriminate the different behaviors. In our simulations m has the same value for each set of parameters α. If the values taken by $X(t)$ throughout the time interval are approximately constant, then all the bins except one are void and entropy will tend to zero. The value of $S(\alpha)$ increases as the probability distribution of the values of $X(t)$ is wider, as happens when it oscillates in time. The maximum value of the entropy would be obtained from an equiprobable distribution.

Therefore, the Shannon entropy measures the dispersion on the distribution of values taken by the variable or response of the CS that we are studying for a given set of parameters α. That is why it can be very useful to classify the occurrence of

different behaviors of the CS. Importantly, each entropy computed according to Eq. (1) is just a rough estimate of the CS-state, it does not capture all subtle details undergoing within the studied system and hence it can miss some important low-level processes operating there. It is always a good idea to test several types of entropies and choose the one which is the best for a given task. There are many more developed entropies not mentioned in this paper.

2.3. *Cellular automata*

John von Neumann [59] and Stanislaw Ulam [58] developed the concept of CA during the World War II as an alternative to the currently used computer paradigm (single processor unit and memory). CAs are based on the idea of the massively parallel computer design (multiple simple processor units [local rules] and associated, tiny memory units spread across a lattice), which was impossible to implement using vacuum tubes due to their low reliability and operational life at that time. CAs are defined and many of their vital applications demonstrated in [26, 29].

Each cellular automaton is defined as a four-tuple (L, N, S, f) which means: L is lattice, N is neighborhood, S is state, and f is transition function/rule. Lattice is defined in n-th dimensional space. We use two-dimensional (2D) lattice L, see Fig. 1. Periodic boundary conditions mimicking unlimited medium are often used. The neighborhood N in 2D can be von Neumann (North, West, South, and East), Moore (N, NW, W, SW, S, SE, E, NE) with diameter equal to one or greater, or others,

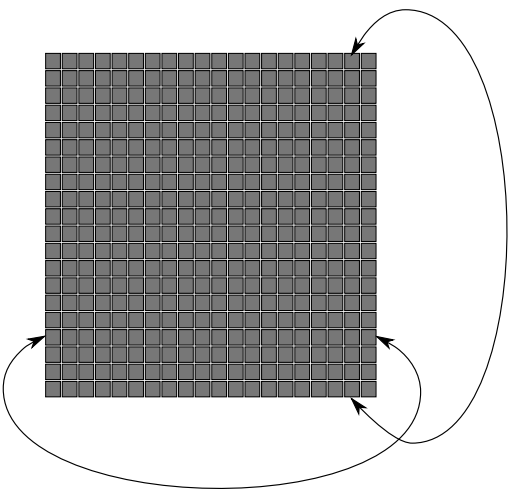

Fig. 1. Within this paper, each CA-lattice in two-dimensions is composed of a square grid of its constituting elements called cells. A cell is the smallest spatial unit of each automaton. The periodic boundary conditions mean that the top and bottom edges of the lattice are "glued" together, the left and right edges are "glued" together as well. An unlimited medium having the shape of the toroid made from cells is created. Each cell update its variables (e.g., photons, grain segments, chemicals) according to a rule independently from all cells except its neighbors.

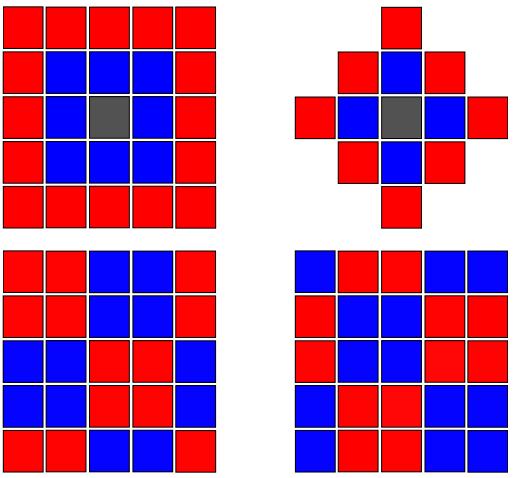

Fig. 2. (Color online) The central light gray cells within Moore and von Neumann neighborhoods are the updated ones. Three types of neighborhoods: (up-left) Moore of radius $R = 1$ [blue], and $R = 2$ [blue + red]; (up-right) von Neumann of radius $R = 1$ [blue], and $R = 2$ [blue + red]; (down-left) Margolus — odd time steps; and (down-right) Margolus — even time steps. All chemical reactions are happening within each block of the Margolus neighborhood separately. Therefore, Margolus neighborhood [57] conserves the total mass of all atoms present within the simulation by subsequent alternation of two types of neighborhoods.

see Fig. 2. State S is assumed to belong to the integer interval $(0, I_{\max})$ but can be as well using real numbers. Finally, f represents the transition function governing the evolution of the automaton.

The simulation of each CA (a massively parallel computer) is done using an engine that is simulating parallel evaluation of all cells within the lattice on a sequential computer (single processor unit and memory), see Fig. 1. It is done by using two alternating arrays (the old and new one, same as in Fig. 1) that are storing all variables for each cell. The simulation must be done using two arrays because otherwise old and newly computed values become mixed as the array is visited and reevaluated cell by cell in rows or columns. One array simulations produces unpredictable mess in the simulations.

Within two-array representation, all cells are evaluated sequentially using values taken from the old array, neighborhood and transition function. New values are stored in the array marked as the new one. It is absolutely necessary to keep in mind that data flow in CA-simulations is always centripetal! We are changing the central cell only! When we change values in neighboring cells of the evaluated cell, the simulation becomes highly unpredictable. It can be done by scheduling but it is a highly advanced approach not suitable for newcomers. Actually, such errors, changing value(s) in neighboring cells are very difficult to localize; it is recommended to do a thorough listing of the flow of the algorithm in such cases. When the evaluation of the whole lattice finishes, arrays are swapped, data reported, and visualized, and the simulation proceeds again in the evaluation of all cells.

3. General Information about Design of Cellular Automata Models of Natural Phenomena

A broader view on the presented CA-models and the ideas along with the basic concepts hidden behind their design is the essential starting information for everybody who would like to design his own CA-model of the observed phenomena in any scientific field. The whole text of this paper is designed in the way to provide such insights from various points of view. This is the main reason why different CA-models are presented in different ways.

In order to define a generic CA model [18, 32], we must specify the properties of its different components:

(i) What are the individual components and what is represented by their different states.
(ii) The space in which the simulation is carried out. It represents a discretization of the real space. Several components must be defined: the dimension, the type of lattice, the boundary conditions and the number of cells contained by the system.
(iii) The neighborhood or the set of cells that interacts among them.
(iv) The transition rules which represent the processes working at the individual level in the CS and are perhaps the most important ingredient in the definition of a CA model in order to reproduce the behavior of the real system. They define the way the individual components of the system interact with each other by specifying the state of each cell at time step $t + 1$ depending on its state and the state of the cells included in its neighborhood at time step t. All the cells of the CA are updated in parallel. We must define rules that represent the real interactions between the components of the system. In order to define each rule, we can choose between different possible types of rules (deterministic, probabilistic, totalistic, etc.) and specify exactly each particular rule.

The general structure of a CA algorithm is described in Algorithm 1. After initializing the system, the set of cells is updated synchronously following each of the

Algorithm 1. Pseudo-code diagram for a generic CA model.

```
Initialize system
Input data
for time step = 1 to maximum time step do
  for each cell in the array do
    Apply Transition Rules
  end for
  Make Measurements
end for
Final calculations
Output results
```

Table 1. The structure of all three CA-models is compared in one place for an easy comparison: cell variables, space dimension, type of the used neighborhood, transition rule, and output macroscopic measures.

	Laser	Recrystallization	Chemical reaction
Cells	Photon electron	Grain orientation Grain boundary velocity Dislocation density	CO(ads) O(ads) Empty cell
Space	$d = 2$	$d = 2$	$d = 2$
Neighborhood	Moore	Moore	Margoulus
CA rules	Pumping Stimulated Emission Photon decay Electron decay	Grain nucleation Movement of GBs Hardening Recrystallization	Adsorption Reaction Desorption Diffusion
Measures	Laser intensity Population Inversion Stability curve	Stress–Strain curves Mean Grain diameter evolution Nucleation rate evolution	CO(ads) concentration O(ads) concentration Oscillatory Behavior

transition rules. In each time step data is being collected either the average value of a quantity or the value of a specific cell.

The usual approach is to build a model as simple as possible, keeping only the essential ingredients that can still capture the behavior in which we are interested, but not simpler, because then that behavior would not be reproduced. Table 1 shows a summary of the three models that will be discussed in the following sections: the state of the cells, the neighborhood, the space of simulation, the transition rules. The last row of the table shows the properties to be measured or the behavior expected to be reproduced.

4. Efficient Implementation of CA Models on Parallel Computers

One of the advantages of CA models is its inherently parallel nature: the temporal evolution of its components takes place independently and the communication between them is restricted to a local environment. Thanks to this, CA are very suitable for implementation on high-performance parallel computing systems, taking full advantage of their efficiency, since the flow of communication between processors can be kept low.

A classical CA model is formed by a regular lattice in which cells evolve by applying evolution rules taking into account information only from the cell itself and a number of neighboring cells. The strong parallel nature of this kind of model is based on data parallelism: the whole CA space can be divided into different partitions that are executed on different processors.

Two different factors can limit the performance of this type of parallel implementation of a classical CA model running on a parallel computer. The first one is the fact that the model is only partially uncoupled, because data from neighboring cells

must be exchanged after each time step. The second limiting factor is the synchronous character of the CA model. That means that different partitions of the CA cannot be run independently on different processors for more than one time step.

The level at which these two conditions are met depends on the architecture of the parallel computer and on the particular CA model. In clusters (distributed-memory parallel computers) the bottleneck comes mainly from the first factor (partially uncoupled character of the model) and is due to the communication between cluster nodes, that must be carried out by explicitly exchanging messages and is orders of magnitude slower than computation on a node. In order to deal with this issue and get a good performance, the condition that a high computation/communication ratio is sustained must be met. That means that the time spent in communication of states of adjacent cells between different processors must be sufficiently smaller than the time required for computation of the states of cells included in each partition. That depends on the amount of data from neighboring cells needed for the application of evolution rules on a cell. CA models in which this amount is low can be good candidates to be efficiently parallelized in a cluster. There are many practical models in which this can be met, for example, models for 3D simulation of multicellular tissue growth [60], urban dynamics simulation about land uses [35], 2D dendritic growth on solidification [30], or the laser model described in this work (see Sec. 5.5). In the case of a heterogeneous cluster with nodes having different performance capabilities, the second limiting factor (synchronous character of the CA model) can also limit the performance because fast nodes must wait for slower nodes to finish after each time step before proceeding, but it can be met by using a dynamic load balancing procedure, as demonstrated, for example, in [17].

In the case of GPUs and shared-memory parallel computers (such as multicore CPUs), which consist of a number of tightly-coupled processors, communication between processing elements is carried out by read/write operations in a shared memory. Those operations are much faster than exchanging messages through the network, so the computation/communication ratio is not a bottleneck. In GPUs the bottleneck is due to the second factor (synchronous character of the CA). That means the time loop of the algorithm cannot be parallelized. The parallelism of the GPU can only be used to execute in parallel the computations of one time step. Therefore, a CA model with a modest lattice size can leave unused part of the computing elements of a large GPU. GPUs with many processing elements need CA models with a large lattice to make use of all the potential parallelism of the GPU. In addition, care must be taken in GPUs with the random number generation for probabilistic CA models. Many of the most efficient random number generators of good quality are designed to be run on CPUs and make fundamental assumptions about processor architecture and performance that make them not appropriate for use in GPUs. However, there are good random number generators that have been shown to be appropriate to be parallelized efficiently for GPUs, such as the Wallace Gaussian Generator or the hybrid Tausworthe Generator combined with the Box-Muller Transform [28], or a combination of a Tausworthe Generator and the

Linear Congruential Generator [14]. In spite of these limitations, in most cases GPUs are ideal platforms to accelerate CA models, thanks to their very large number of processing elements and the fast communication between them. A thorough study of the performance increases that can be expected from their parallelization for a wide range of CA parameters (such as lattice and neighborhood sizes, number of states, complexity of the state transition rules, number of generations, etc.) is given in [13]. A practical example involving the laser CA model described in this work is given in [37].

In the case of multicore CPUs the limitation is usually the small number of cores (processing elements) of the CPU, that typically range from several cores to several tens of cores at best, so that only a limited amount of parallelism can be used to accelerate the execution of the algorithm.

5. Cellular Automata Model of Laser Dynamics

The laser is one of the prototypical examples of CSs that show self-organization and emergent properties due to the local interactions among simple components. A diagram with the basic elements of a laser system and its operation [48] is shown in Fig. 3. The laser cavity contains the laser active medium with a collection of atoms, molecules, ions or a semiconductor crystal. The electrons of the laser active medium are excited into higher quantum-mechanical energy levels with a pumping process (using some external electrical, optical or chemical energy source) to induce a population inversion, that is, to make a high energy level more populated by electrons than a low one. If a photon with the right energy value crosses the laser active medium it can induce the stimulated emission process in one of its atoms or molecules with population inversion. As a result, its electron decays to a lower energy level and a new photon with the same properties as the original one is created. The new photon can induce a new stimulated emission, so that a cascade process can be produced. The photons are usually reflected back and forth into the laser active medium with a couple of mirrors. One of them has a reflectivity lower than 100% so that some photons can escape and produce the laser output beam.

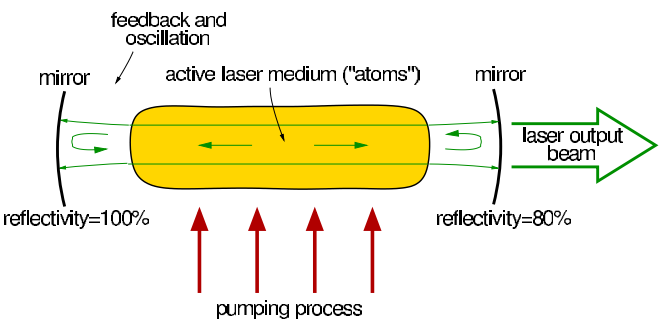

Fig. 3. Basic elements and mechanism of operation of a laser system.

The laser is a complex system in which the simple components are atoms or molecules of the active medium of the laser cavity and the radiation photons they produce. Different emerging properties are induced by self-organization: the emission of a coherent radiation beam when the energy that is pumped into the system is above a critical threshold value and the temporal patterns of the radiation intensity, which after an initial transient can be either constant or oscillatory depending on the laser parameters.

Laser dynamics has been traditionally modeled through a macroscopic view based on differential equations, the laser rate equations, that describe the global properties of the system [48]. The works [18, 22, 25] showed that it is possible to model it using a CA model that describe the evolution of the individual components of the system. The interesting point is that with this model the emergent properties of the laser system appear spontaneously as a result of the self-organization of its basic components, thus capturing the essence of the laser behavior as a CS.

5.1. *Identification of basic processes involved in phenomenon, at microscopic level*

(a) Individual components. In the case of the laser, the components are the atoms or molecules in the laser active medium and the laser light photons. A variable is associated to each component for each cell in the CA. In the laser model, two independent variables are used: the first one, a_{ij}, represents the state of the electron in cell ij (located at row i and column j) at time t: when $a_{ij} = 0$ the electron is in the ground state and when $a_{ij} = 1$ the electron is in the upper laser state. The second one, $c_{ij} \in 0, 1, 2, \ldots, M$, represents the number of laser photons in cell ij at time t, which is bounded by an upper value M that must be chosen large enough to avoid the saturation of the system. The state variables represent bunches of real photons and electrons, whose values are connected to the number of photons and electrons in the real system, which is larger, by a normalization constant.

(b) The cellular space is a two-dimensional lattice of $N_c = L \times L$ cells with periodic boundary conditions.

(c) Neighborhood: The Moore neighborhood is employed, in which each cell has nine neighbors: the cell itself, its four nearest neighbors (located at positions north, south, east, and west), and the four next neighbors (at positions northeast, southeast, northwest, and southwest).

(d) Transition rules: The rules in the laser model are:

   (i) Stimulated emission. If the electronic state of a cell has a value of $a_{ij}(t) = 1$ in time t and the sum of the values of the laser photons states in the neighboring cells is larger than a certain threshold θ (which in our simulations has been taken to be 1), then in time $t + 1$ a new photon will be emitted in that cell: $c_{ij}(t + 1) = c_{ij}(t) + 1$ and the electron will decay to the ground level: $a_{ij}(t + 1) = 0$.

(ii) Photon decay. A finite life time τ_c is assigned to each photon when it is created. The photon will be destroyed τ_c time steps after it is created.

(iii) Electron decay. A finite life time τ_a is assigned to each electron that is promoted from the ground level to the upper laser level. That electron will decay to the ground level again τ_a time steps after it was promoted, if it has not yet decayed by stimulated emission.

(iv) Pumping. If the electronic state of a cell $\{ij\}$ has a value of $a_{ij}(t) = 0$ in time t, then in time $t + 1$ that state will have a value of $a_{ij}(t + 1) = 1$ with a pumping probability λ.

(v) Noise photons creation. A small number of laser photons in randomly chosen positions is introduced at each time step to reproduce spontaneous emission and thermal contributions, responsible of the initial laser start-up. To this end, for a small number of randomly chosen cells $\{ij\}$ ($< 0.01\%$ of total) it is applied $c_{ij}(t + 1) = c_{ij}(t) + 1$.

5.2. *Simulations of laser model*

Once the model has been developed, simulations can be carried out by introducing values for the parameters of the model and input values and just letting the model to evolve by sequentially applying the evolution rules for a number of time steps or until a stop condition is reached. In the laser model, there are three main parameters whose values determine the behavior of the system: the electron life time τ_a, the photon life time τ_c and the pumping probability λ. Before we can make use of the model to carry out in-silico experiments, we must validate the model by running it for a number of parameter and input values and comparing the simulations outcomes with experimental data.

A crucial step in the development of a CA-based model of a system is to determine what are the relevant macroscopic magnitudes of the system that we intend to simulate and identify the counterpart of each magnitude in the CA model, in which they are calculated as a collective property of the aggregate of microscopic magnitudes. This can be seen more clearly by analyzing the case of the laser dynamics model. There, two main macroscopic magnitudes were considered: *laser beam intensity*, which is obtained in the model by counting the total number of laser photons, i.e., the sum of the values of the c_{ij} variable for all the cells

$$I(t) = \sum_{ij} c_{ij}(t) \tag{2}$$

and *population inversion*, which is given by the sum of the values of the a_{ij} variable

$$P(t) = \sum_{ij} a_{ij}(t). \tag{3}$$

In both cases the sum extends to all the cells in the lattice.

Once the model is built and the macroscopic variables are identified and defined, simulations can be run in order to find emergent behaviors. There are four particular

emergent behaviors in laser dynamics that can be reproduced using the CA model [18, 22, 25]:

(a) Laser threshold. A characteristic feature of lasers is that laser action only occurs when the external pumping of energy is above a threshold value. Below it, population inversion is not strong enough to trigger the laser action. In order to investigate if the model reproduces this behavior, simulations must be carried out for different values of the pumping parameter. It was found that the CA model reproduces this behavior: a cascade of laser photons are produced by stimulated emission only when pumping probability is above a particular value λ_t, that depends on the values of the other two parameters of the system (life times of excited electrons and photons). Moreover, its dependence on both parameters was found to be in good agreement with the laser behavior.

(b) Laser spiking. When the three parameters of the system have the right values, the temporal evolution of the two macroscopic variables of the system (intensity and population inversion) exhibit the behavior known as *laser spiking*, in which they show correlated damped oscillations. This behavior is reproduced by the CA model, as shown in Fig. 4.

(c) Constant behavior. For other values of the system parameters, the temporal evolution of intensity and population inversion shows a *constant behavior*: after a transient, they show constant values. This behavior is also reproduced for the CA model, as shown in Fig. 5.

(d) Dependence of spiking and constant behaviors on system parameters. Experimentally and from a linear stability analysis of the laser rate equations [48] we know that the dependence of the two kinds of behaviors (laser spiking or constant behavior) on system parameters is as shown in Fig. 6: laser spiking is shown

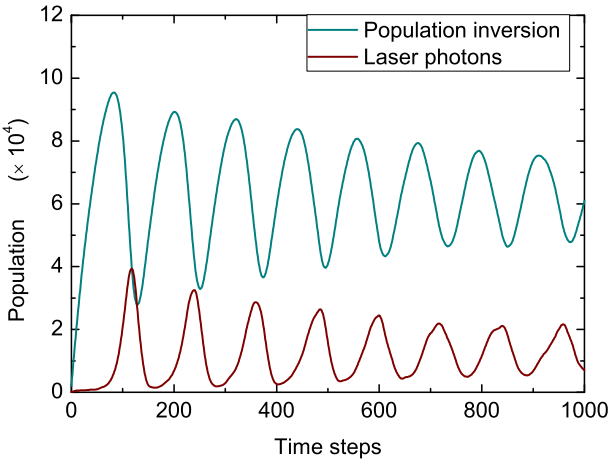

Fig. 4. Time series of the laser beam intensity and the population inversion for the behavior known as laser spiking, showing correlated damped oscillations. Parameter values: $\tau_a = 180$, $\tau_c = 10$, $\lambda = 0.0125$.

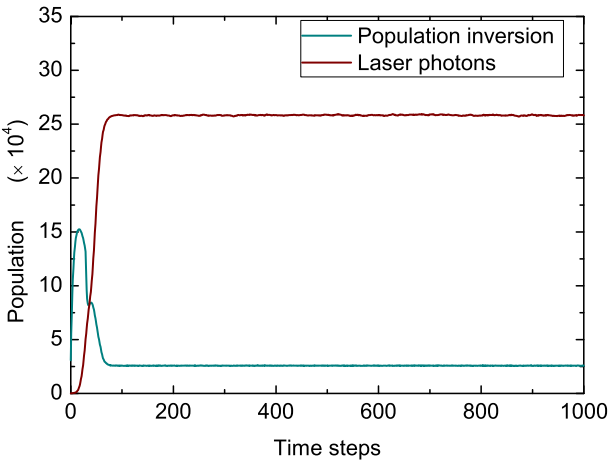

Fig. 5. Time series of the laser beam intensity and the population inversion for the constant behavior. Parameter values: $\tau_a = 30$, $\tau_c = 10$, $\lambda = 0.192$.

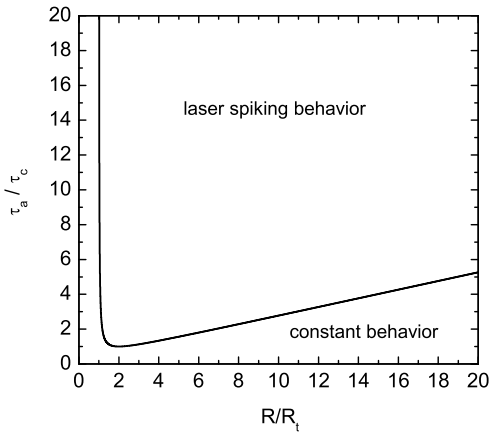

Fig. 6. Dependence of the two kinds of laser behavior with system parameters, obtained through an analysis of the laser rate equations.

above and to the right of the curve shown in the figure and a constant behavior below and to the left of it. Here a pumping rate R of electrons that are promoted from the ground state to the upper laser state is used instead of a pumping probability. The threshold pumping rate that is needed to have laser action is called R_t. As shown in [22], the ratio between pumping rate and threshold pumping rate is the same as the ratio between pumping probability and threshold pumping probability: $\lambda/\lambda_t = R/R_t$.

5.3. *Shannon entropy*

In order to investigate if the CA model can reproduce this result it is necessary to execute a very large number of simulations that sweep all the range of parameter values. It is therefore mandatory to have some way to automate the classification of the behavior exhibited by the system for each simulation. As was found in [24], a good measure for this task is the Shannon entropy (the generic definition is in Sec. 2.2), which measures the dispersion of the distribution of values taken by a variable in a time series.

For the laser model, the kind of behaviors shown by the simulations within a time interval can be calculated by computing the Shannon entropy for the macroscopic variables intensity, $I(t)$, or population inversion, $P(t)$ for each pair of values of $(\frac{\tau_a}{\tau_c}, \frac{R}{R_t})$. Results are shown in Fig. 7, in which the number of bins used for the calculation of $S(\frac{\tau_a}{\tau_c}, \frac{R}{R_t})$ was $m = 10^2$. Figure 7 shows that the higher values of the entropy appear above and to the right of the laser stability curve, thus the cooperative behavior of the CA model reproduces the results shown experimentally by lasers, also predicted by the laser rate equations.

5.4. *Variations of laser model*

Usually in nature different variants of a particular phenomenon are present, which share the basic ingredients and mechanisms but differ in details and behaviors. One of the advantages of the microscopic descriptions of CSs such as CAs is that once we have built a model for one of the variants of a phenomenon, it is usually feasible and easy to build models for other variants of the phenomenon, just by changing some of the ingredients of the model. For example, in the case of laser dynamics, once a basic

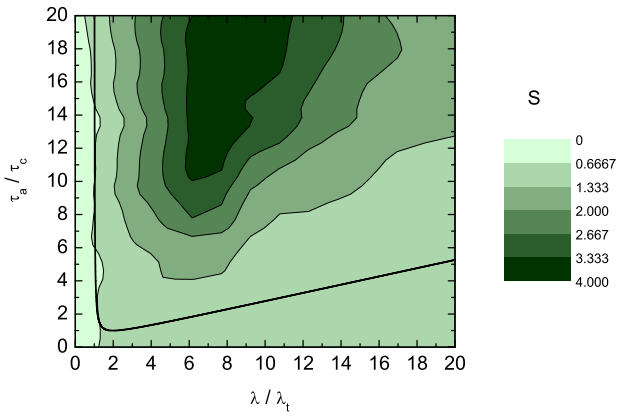

Fig. 7. Contour plot of the Shannon entropy $S(\frac{\tau_a}{\tau_c}, \frac{R}{R_t})$ of the distribution of values taken by the laser beam intensity (total number of laser photons) for different values of τ_a and λ and for a fixed value of $\tau_c = 10$. By comparing with Fig. 6 (taking into account that $\lambda/\lambda_t = R/R_t$) we can see that the CA model reproduces the laser behaviors found for different parameter values.

model was developed it was possible to build a model of a special type of lasers, pulsed pumped lasers, just by changing some of the model characteristics [23].

5.5. *Implementation of laser model on high-performance parallel computing systems*

As an example, [20, 21] show how the CA-model of laser dynamics was implemented in parallel on cluster computers. In addition, its performance is analyzed in [19] and a first prototype parallel implementation on Graphics Processing Units (GPU) is presented in [37]. These works can be used as a guide to develop fast parallel implementations of other CA-models on high-performance computers.

6. Cellular Automaton Model of Dynamic Recrystallization

The CA-model of DRX [32, 33, 54, 56] operating during deformation of metals is described in this section. It contains: a physical background, a definition of the CA-model of DRX [32], a software implementation [54–56], results of simulations [32, 33], a demonstration of the principle of the structural stability, and several pathological instances of the CA-model. Those pathologies demonstrate the expressiveness and flexibility of this computational approach and can be applied in CA-models of some other natural phenomena.

6.1. *Physical background of dynamic recrystallization*

Observations and attempts of the description of DRX phenomena occurred in 19th century when thin metallic sheets started to be rolled commercially from thick blocks of cast metals. It was found that some sheets of materials rolled within certain ranges of deformation velocities and temperatures created unwanted, periodic bumps on their surface. Laboratory experiments identified those ranges of deformation rates and temperatures.

A closer inspection of the observed phenomena operating during DRX within low-stacking fault materials (cooper, iron, some steels, etc.) revealed that such metals produce great numbers of dislocations, which are linear imperfections of the crystal structure of metals, and are carrying the metal deformation. Definitions follow: (i) Stress σ is defined as the force F applied to deform a material divided by its cross-section S ($\sigma = F/S$). (ii) Strain ϵ is defined as $\epsilon = (l - l_0)/l_0$, where the length of the extended/compressed sample is l and the original length is l_0. The use of relative values enables an easy comparison of samples having different sizes.

What is actually observed during rolling? When poly-crystalline, low-stacking fault, metallic materials are deformed by some constant velocity and elevated temperature, we observe rather unexpected, highly nonlinear response of measured deformation curves (stress–strain $\sigma - \epsilon$ curves): oscillatory or single-peak shapes [46].

Gradually, a huge amount of experimental knowledge about the microstructural evolution of metals undergoing DRX was gathered [46]. It was shown that

microstructural evolution plays the key role in DRX but it was not known how to interpret it. Despite a large number of intensive computational studies and deep experimental knowledge of the macroscopic curves and microstructural evolution combined did not led to the design of a simple, reliable model explaining DRX. Many models were proposed to explain stress–strain curves during DRX: statistical, analytic (using ODEs or PDEs), applying Monte Carlo methods using Pots model, etc. None of them was successful.

It became evident that some crucial knowledge was missing. In the end, it was revealed that the correct description of the spatio-temporal evolution of the material microstructure is crucial and represents the missing link.

Polycrystalline materials are composed of many, tiny, differently oriented, single crystals stacked together. The area where two neighboring crystals meet is called grain boundary (GB). The very principle of DRX is based on the growth of newly formed (preferentially on GBs), low-energy, undeformed crystals into older, highly deformed crystals. The driving force of this growth is the difference of dislocation densities between those two generations of grains.

The deformation of a polycrystalline sample is transformed into the deformation of its constituting crystals. For simplicity, in models, it is assumed that all crystals are deformed uniformly which is not true for real materials where the amount of deformation depends on the principal direction of deformation, orientations of deformed crystals to it, and orientations of their neighbors. Simply said, the actual deformation is percolating through the deformed sample and creates dynamically changing percolation paths within it.

What is actually happening within the deformed material from the mathematical point of view? Every part and every process of the deformed specimen is developing according to its surrounding independently from its distant parts: nucleation, GB movements, and deformation. It was revealed that this understanding is the most crucial one for the design of a reliable massively parallel model [32].

6.2. *Definition of CA-model of dynamic recrystallization*

The work [32, 33, 54] is demonstrating the whole cycle of CA-model development starting with the model design and ending with the software implementation [55, 56]. The crucial processes of the model include (i) the implementation of GB movements and (ii) feeding energy into the sample by deformation. The interplay between those two processes represents the core of the model. Importantly, during the study of the main causes of the failure of older models and especially of the Monte Carlo models, it was found that none of the previous models implemented correctly the parallelism of all processes [32] which led to unrealistic results.

It paved the way to a model where all basic physics behind including topologically relevant movements of GBs (this is related to perfect parallelism of CAs) have to be implemented precisely. How was it accomplished? Several critically important conditions have to be fulfilled simultaneously: (i) It is necessary to define a fine grid

above which all GB movements can be easily implemented. (ii) All important topological changes of the evolving microstructure driven by GB movements must be reproduced. (iii) Parallelism of all processes must be preserved. (iv) GB movements are treated only locally. (v) Nucleation occurs selectively on GBs. (vi) Additionally, a simplification is added, deformation is assumed to be uniform through the whole cross-section of the specimen without the presence of any localization [32, 33].

The CA-model definition strictly follows the above listed conditions [32, 33] and can be written as follows:

(a) Two-dimensional space is discretized into a $L \times L$ lattice of squares where $L = 256$.
(b) Periodic boundary conditions of the lattice are used. It means that left with right and top with bottom edges are attached together. They thus create an infinite computational medium.
(c) Eight nearest neighbors of each cell creates the Moore neighborhood with radius $r = 1$, see Fig. 2.
(d) Each cell has several variables: grain orientations θ_{ij}, GB velocity v_{ij} (respectively waiting time t_w, see further text), and dislocation density ρ_{ij}.
(e) Transition rule governs: nucleation at grain boundaries, dislocation build up by deformation, grain boundary segment movement, and recrystallization at grain boundary, details in the flow diagram in Fig. 9.
(f) Initial microstructures composed of grains were created beforehand, see Fig. 8. Grains were created by pseudo-static recrystallization, this defines orientation variable θ_{ij}. Dislocation density ρ_{ij} and waiting time t_w were initiated homogeneously.

The equations and rules governing the evolution of the grain topology and dislocation density ρ follows. Stress σ is evaluated according to the equation

$$\sigma(t) = \alpha G b \rho^{1/2}(t), \tag{4}$$

where G is shear modulus, b is Burgers vector and α is a numerical constant ($\alpha \propto 1$).

The evolution of dislocation density ρ within each cell follows the Mecking–Kocks law

$$\frac{d\rho(\gamma)}{d\gamma} = A\rho(\gamma)^{1/2} - B\rho(\gamma), \tag{5}$$

where γ is shear strain. A and B are constants representing hardening (storing of dislocations via deformation) and recovery of dislocations due to their annihilation, respectively.

The total accumulated strain is evaluated according to the following formula:

$$\gamma = \int_{t'=0}^{t} \dot{\gamma} dt', \tag{6}$$

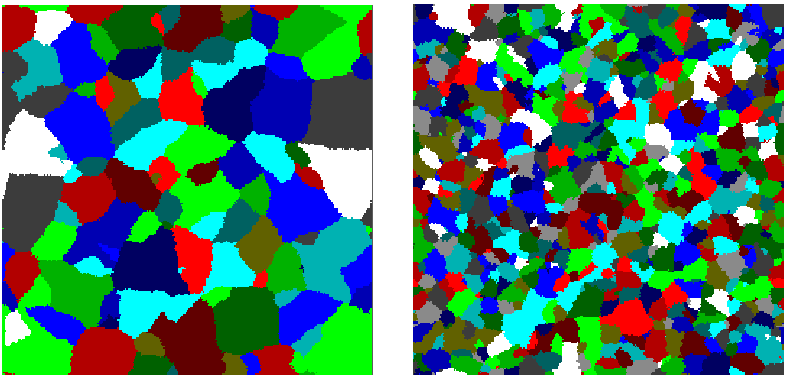

Fig. 8. (Color online) Two examples of the initial microstructures simulated by the pseudo-static recrystallization algorithm [32] that were used as the input to simulate DRX where probability of an unrecrystallized cell to become a nucleus are $p_n = 0.00025$ (left) and $p_n = 0.004$ (right). Each cell is represented by the smallest square seen in the picture. Different orientations of grains are represented by different colors. Orientations are discretized by the step of 10° (0°, 10°, 20°, ..., 170°).

where the integration sums all deformations during the whole period of deformation of the sample.

The velocity of moving boundary is having the form

$$v(T) = M\Delta f, \tag{7}$$

where M is temperature-dependent GB-mobility and Δf is driving force (given by difference of dislocation densities on both sides of GBs).

In the model, the formulas are reformulated into their discrete forms, most important of them follows (more details in [32]). Stress σ is evaluated using the following relationship:

$$\sigma(t) = C < \rho(t) >^{1/2}, \tag{8}$$

where C is a constant and $\langle\rho(t)\rangle = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \rho_{ij}(t)$ is the average dislocation density across the specimen cross-section. The computational lattice of all cells is m cells high and n cells wide.

GB velocity is maintained by a waiting time $t_w \geq 1$ that is decreased by one within each simulation step when all recrystallization conditions are fulfilled

$$t_w^{s+1}(T) = t_w^s(T) - 1, \tag{9}$$

where s is the step number, waiting time $t_w(T)$ starts at some value $t_{\max}(T)$.

The velocity $v(T)$ of moving boundary is implemented within CA in the form of

$$v(T) = d/(t_w(T) + 2), \tag{10}$$

where d is cell width and $t_w(T)$ is waiting time defined by Eq. (9). The constant 2 is added due to the probability acceptance of each recrystallization event ($p_{\text{re}} = 0.5$, see Fig. 9), otherwise only 1 will be added. A greater $t_w(T)$ leads to a lower velocity $v(T)$

and *vice versa*. The maximal achievable GB-velocity is one cell width per one simulation step ($t_w = 1$) and the lowest velocity is one cell width per $t_{\max}$ steps.

Each nucleation event, generation of a new grain, within the cell (i, j) occurs only when all the following conditions are simultaneously fulfilled:

$$\begin{aligned} &\text{(i)} \quad \text{cell } (i,j) \text{ is a GB cell,} \\ &\text{(ii)} \quad \rho_{ij} > \rho_{\text{nucl}}, \\ &\text{(iii)} \quad \text{event accepted with probability } p_{\text{nucl}} \end{aligned} \tag{11}$$

for details see flow diagram in Fig. 9.

A recrystallization event at GB occurs only when dislocation densities within cells on both sides of GB exceeds a certain threshold value ρ_{cr}, see Fig. 9

$$\rho_{\text{con}} - \rho_{\text{gr}} > \rho_{\text{cr}}, \tag{12}$$

where ρ_{con} and ρ_{gr} are consumed and growing cells, respectively. This leads to the grow of low deformed grain into highly deformed one and never vice versa, see the full flow diagram describing all steps in Fig. 9.

The CA-model was designed using physically relevant modules, which were added one after the other (details about physics above and [32]). (i) The first module deals

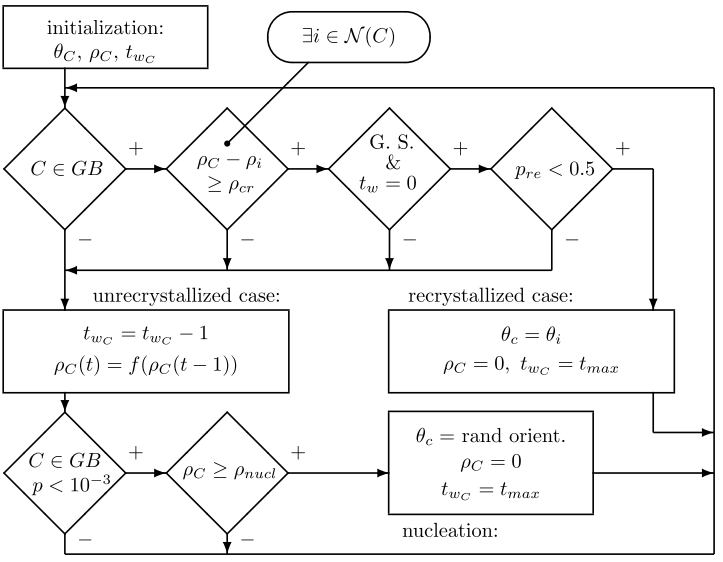

Fig. 9. Flow diagram governing the evolution of each cell of the lattice (see Fig. 1) by determining the next value of the variables θ, ρ and t_w of the CA simulating DRX [32]. It contains all critical steps: nucleation, hardening (inserting energy into grains by deformation), recrystallization events, handling of the velocity of grain boundaries (counting down the waiting time t_w). The abbreviation G. S. stands for the term "good shape", i.e., no narrow hollows into GB are allowed. The symbol C represents a cell under consideration and symbol i represents a cell from the neighborhood $\mathcal{N}(C)$ of this cell C. Additional variables p and p_{re} are random numbers from interval $< 0, 1)$ where $p_{\text{re}} < 0.5$ means that a rand number is accepted when the condition is fulfilled.

with probabilistic nucleation on GBs (see Eq. (11)). (ii) The second module implemented GB movements (see Eqs. (9) and (10)), which is triggered by difference of dislocation densities $\Delta\rho$ between neighboring grains (see Eq. (12)). (iii) The next one implements hardening due to increase of dislocation density ρ via deformation (see Eq. (8)). (iv) The module enabling a dynamic change of strain rate $\dot{\epsilon}$ and hence the dislocation density ρ generated per unit time. (v) The initial microstructure is generated by a physically unrealistic "recrystallization" procedure where "nuclei" are generated continuously (the continuous nucleation), those nuclei grow until their mutual impingement.

6.3. *Results of dynamic recrystallization model*

Observations qualitatively explained by the CA-model of DRX [32, 33, 54, 56] follow. Stress–strain ($\sigma - \epsilon$) curves express a transition from the single-peak behavior to the multiple-peak one, see Fig. 10. The single-peak behavior is associated with higher strain rates $\dot{\epsilon}$ or lower temperatures T; whereas, the multiple-peak behavior is associated with lower strain rates $\dot{\epsilon}$ or higher temperatures T. An interplay between increasing stored energy of the material due to deformation and decreasing stored energy due to its recrystallization is confirmed. Mean grain-size versus strain (D-ϵ) curves contain oscillations, see Fig. 11, which are shifted by 1/4 of wave-length

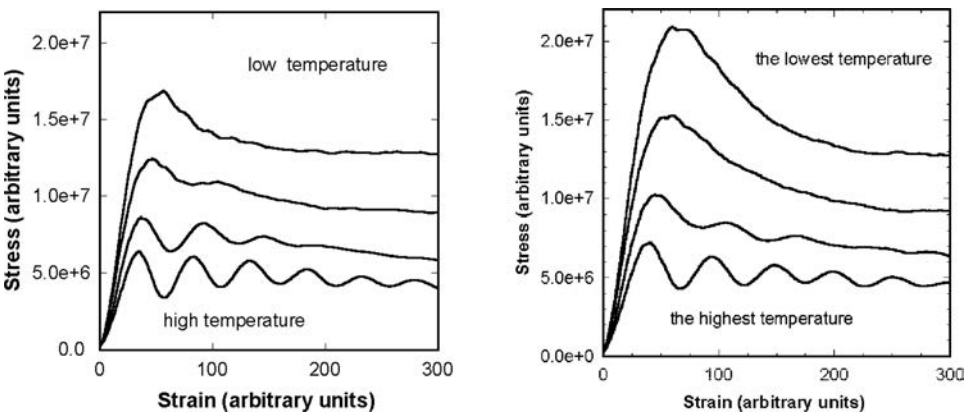

Fig. 10. The stress–strain ($\sigma - \epsilon$) curves of two different microstructures ($\bar{D} = 2.9$ — left and $\bar{D} = 7.0$ — right) are shown. The simulated steady state values of σ_{ss} are the same for different initial MGSs, they are consistent with the experiment. The increasing temperature shifts the single-peak behavior towards the multiple-peak one due to softening of material and higher mobility of grain boundaries (GBs). Oscillations of stress–strain curves results from the existence of recrystallization waves operating sequentially, one after the other. Each wave consumes a substantial part of the specimen volume (including dislocations stored there) what leads to a decrease of the average stress σ. Subsequent deformation builds stress σ (by the accumulation of dislocations) again what leads to the initiation of a new recrystallization wave. Single-peak behavior results from the operation of large number of uncorrelated waves.

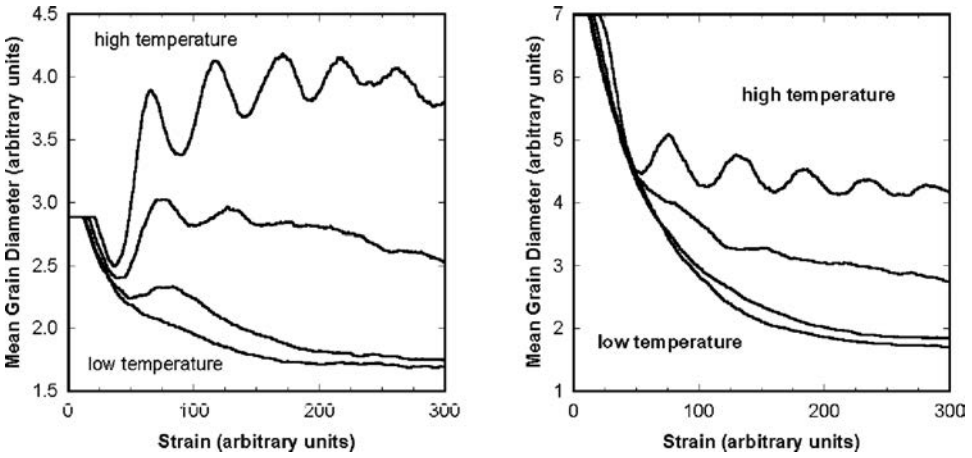

Fig. 11. The evolution of the mean grain-size $\bar{D}$ for two different initial values ($\bar{D} = 2.8$ — left and $\bar{D} = 7.0$ — right) under the very same deformation conditions as applied in Fig. 10. The simulated steady state values of MGS $\bar{D}$ are same for different initial MGSs (be aware of different scales) what is consistent with the experiment. It proofs the existence of the structural stability of materials for same deformation conditions (T and $\dot{\epsilon}$). Obviously, the oscillations of $\bar{D}$ are more pronounced for smaller initial $\bar{D}$s, some even disappear with increasing initial $\bar{D}$ (second and third curves from below loose oscillations).

compared to stress–strain curves in Fig. 10. The microstructure evolution is demonstrated on multiple-peak case in Fig. 12.

Strain-rate change tests ($\dot{\epsilon}_1 \Rightarrow \dot{\epsilon}_2$) where deformation conditions (strain-rate $\dot{\epsilon}$) are abruptly changed in the course of the simulation ($\Delta\dot{\epsilon} \neq 0$) represent the second group of experiments [32, 33, 46], see Fig. 13. Simulated results are consistent with the experiments [46]: multiple-peaks are transformed into single-peaks when strain rate increases and/or temperature decreases. Conversely, single-peaks are transformed into multiple-peaks when strain rate decreases and/or temperature increases. This experiment reveals the presence of the structural stability of the materials where the size of the initial microstructure $\bar{D}$ decides the deformation course and not its subtle topological details. After an abrupt change of deformation conditions, the material needs some time to adjust its behavior to the new deformation conditions.

Another group of experiments deals with the sensitivity of stress–strain curves to the change of the initial mean grain size $\bar{D}$ of deformed samples [46]. All other parameters of simulations are kept strictly identical. The simulations confirmed the observed dependence [32, 33], see Figs. 10 and 11 for two different initial MGSs $\bar{D}$. Large grains give the single-peak behavior and smaller grains give the multiple-peak behavior. The computed ratio between the initial MGS D_0 and the steady state MGS D_{ss} when oscillations start to occur equals to 2 and is consistent with the experimental data [46].

The influence of the nucleation rate was studied as well. Simulations are correctly predicting the occurrence of experimentally observed necklace behavior: it means

Advs. Complex Syst. 2019.22. Downloaded from www.worldscientific.com

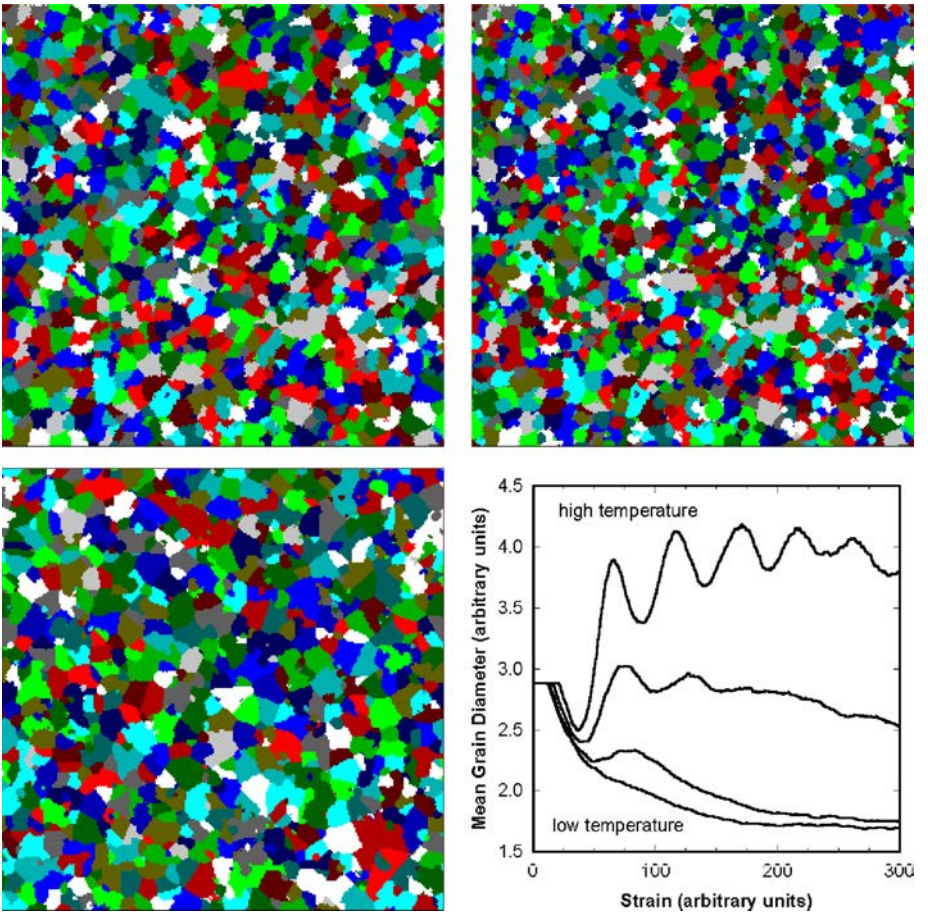

Fig. 12. An example of the microstructural evolution of a deformed material. A correlation between MGS (and deformation in Fig. 10) curves and the microstructure evolution is obvious, see the top MGS curve (see animations [56]). From the left in rows: the initial microstructure with $p_n = 0.1$, the first minimum for $\epsilon = 38$ AU, the first maximum $\epsilon = 67$ AU, and finally MGS curves.

that when nucleation is suddenly increased, a necklace of very small grains starts to grow on GBs around large grains [32, 33]. Growth of those small grains is terminated early due to a fast hardening of the material and by the creation of new nuclei that appear at their GBs. It happens due to an increased strain rate $\dot{\epsilon}$ that cuts off the driving force early (see Eq. (12)), details in [32].

6.4. *Observed unwanted features and anomalies of the CA-model of dynamic recrystallization*

Counter-intuitive observations were made during design of the CA-model of DRX. It is much easier to design a CA-model expressing chaotic behavior, spirals, or fractals

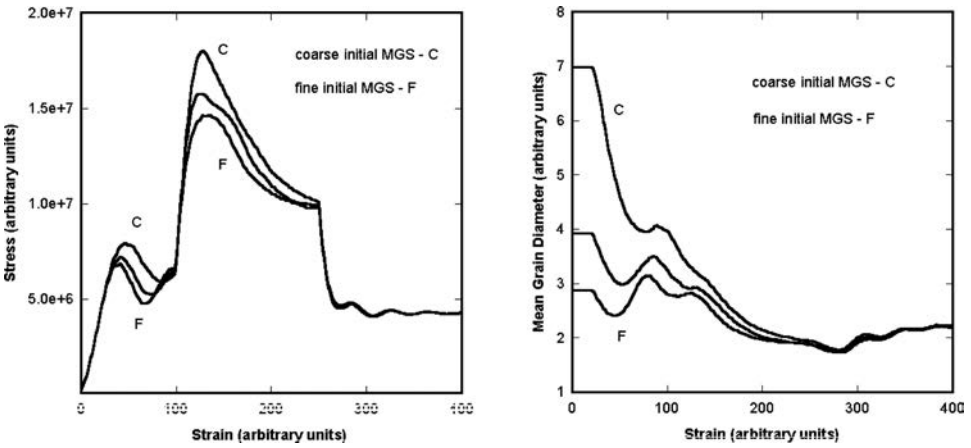

Fig. 13. The strain-rate jump test where the initial strain rate $\dot{\epsilon}_1$ is increased to $\dot{\epsilon}_2$ at strain $\epsilon = 100$ AU and later again lowered to $\dot{\epsilon}_1$ at strain $\epsilon = 250$ AU. The deformed material goes from the multiple-peak regime to the single-peak one and back again, it switches modes according to the deformation conditions. Modes are switched similarly when temperatures abruptly change but those changes are much harder to experimentally accomplish compared to changes of strain-rate (heat do not propagate by unlimited speeds).

than a CA-model behaving "nicely" according to the used set of physical laws. For example, if the number of neighbors that allows growth of a low deformed grain into a highly deformed one is taken incorrectly then it is possible to observe "growth" of physically unrealistic fractal structures.

This fractal property can become useful when we succeed in engineering of materials behaving according to such rule. It is then possible to generate fractal materials — another way to tackle this problem is the use of 3D printers to study toy models of fractal materials.

Another, often observed unwanted "feature" was chaotic behavior of the whole simulation when the GB movement was not described correctly. The last and the most interesting unwanted "feature" is represented by spirals, which starts to operate at GBs. Within the software code, it is always decided whether GB-segment will shift into a consumed grain simply by counting neighbors that belong to neighboring grains. This decides whether a grain will grow or not. It was found [53] that spirals occur because counting goes in one direction: either clockwise or counterclockwise. The reason is that a *for* cycle within the C code always counts in one direction! It favors one side of the GB-segment in its movement compared to the other one. This causes occurrence of spirals that rotate in the direction implemented in the C code.

How is the counting of neighbors actually done? One cell belonging to a highly deformed grain which has several neighboring cells belonging to a growing, low-energy grain is tested. Cells belonging to the neighboring grain, laying within the neighborhood, are visited one after the other by the algorithm (*for* cycle in the language C). Whenever a critical number of cells is reached, the cell under

consideration recrystallize and gets the orientation of the neighboring, low-energy grain; this means that GB is shifted by one cell width. In one of the next time steps, the same happens again with a cell next to the already recrystallized cell; this cell again belongs to the highly recrystallized grain. This mechanism leads to a preferential growth of one segment of the grain into the other grain what in turn creates an invagination into the consumed grain. With proceeding deformation, this process initiates a physically unrealistic spiral operating within the material. The GB itself carries the energy as well in real materials, which protects it from occurrence of spirals, but it is not reflected in the model. Other physical processes protect from occurrence of the spirals as well (e.g., stress–strain localization) but it is beyond the scope of this paper.

Spirals show that the implementation of the CA-model in the code can have a drastic influence on its output. An easy solution of the problem with spirals, which was not tested, can be assigning a random direction of counting within the C code.

The above provided unwanted features of the model can be interesting for design of other CA-models. They can help to built a "vocabulary" of all processes that can be effectively implemented in other CA-models describing different observed phenomena in any scientific discipline.

6.5. *Discussion of CA-model of dynamic recrystallization*

The development of a reliable model of DRX struggled for more than 100 years because it failed to recognize several of its important parts. (i) It did not take into account the crucial effect of finely discretized, spatial-temporal evolution of the underlying microstructure. That was crippling all proposed models and did not allow them to achieve oscillations on stress–strain ($\sigma - \epsilon$) and more importantly on mean grain size-strain ($\bar{D} - \epsilon$) curves. (ii) It did not account for the massive parallelism of the underlying processes partially because of the non-existence of the adequate mathematical and computational models. Monte Carlo models proved themselves insufficient to capture the massive parallelism of the described phenomenon due to their inherent sequential updating procedure (see for the details of the Monte-Carlo step). (iii) It failed to recognize the necessity of a more precise description of the local movement of GBs that have to accurately reflect the local topology of the GBs and gradients of dislocation densities among different grains. (iv) The massive parallelism is fundamental for the mathematical description of natural phenomena operating far-from the thermodynamic equilibrium [42]. As we know now, such systems never settle down to a steady state. At the best, they always operate at a dynamically stable equilibrium, which can be erroneously assumed as a static one. (v) A dynamically stable equilibrium is reached due to the activation of the structural stability of the material.

The structural stability is a highly underestimated process hidden behind operation of many natural phenomena. How does it work and how is it reached? In the case of DRX, it simply means that different parts of the material are recrystallizing while other parts are building their energy due to deformation. This can be done in

the asynchronous mode what leads to a dynamically stable state with the single peak on stress–strain curve. On the other hand, the synchronous mode is observed when "waves" of synchronized growing grains erase dislocations on many places of the sample simultaneously, what leads to a decrease of stress σ on the stress–strain curve. Subsequently, newly created grains must be again hardened by deformation, what leads to an increase of stress σ on the stress–strain curve. Subsequently, yet another avalanche of recrystallization events is triggered. This leads to oscillations on stress–strain and mean grain size-strain curves. In general, when we deal with a phenomenon expressing structural stability, we do always observe two competing processes. One is building up some property and other one is decreasing it.

7. Cellular Automata Modeling of Surface Reactions: The Oxidation of Carbon Monoxide

7.1. *Introduction into surface reactions*

Chemical reactions catalyzed on surfaces often reveal complex behaviors such as multiplicity reaction rates and oscillations. Those behaviors arise from the interaction among the different physical and chemical processes involved, mainly adsorption, desorption, and diffusion.

One of the most studied chemical reactions is the oxidation of carbon monoxide (CO) on metals of the platinum or palladium groups. Besides its industrial relevance, this reaction has been used as a prototype of a spatially extended system in which phase transitions, sustained oscillations, and chaotic behaviors can be studied.

Kinetic of catalytic reactions has been widely modeled using kinetic equations of the mean-field type [10], and the Monte Carlo method [51, 61].

CA models are an alternative technique to simulate chemical reactions. CA offer a different approach that allows us to follow the dynamical evolution and reproduce microscopic configurations of the system [8, 31], something that with mean-field models is not possible. And also the CA models can describe the formation of spatially ordered structures [12].

7.2. *Definition of surface chemical reactions*

The most commonly used mechanism to explain the kinetics of heterogeneous catalytic reactions is the Langmuir–Hinshelwood kinetics which is shown graphically in Fig. 14. The catalyst is a two-dimensional surface with $L \times L$ cells. Each grid of the surface may be occupied by a molecule of CO(ads), by an atom of O(ads) or be unoccupied $\square$. The entire process of the reaction can be divided into five subprocesses:

(1) The adsorption of gaseous molecules of carbon monoxide and oxygen in free positions of the surface:

$$\mathrm{CO(gas)} + \square \xrightarrow{k_1} \mathrm{CO(ads)}. \tag{13}$$

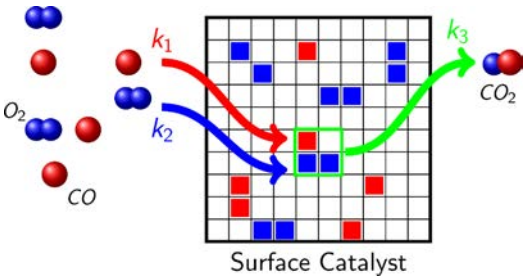

Fig. 14. Scheme of the L–H mechanism for the oxidation of carbon monoxide on metal surfaces: $\mathrm{CO(gas)} + \frac{1}{2}\mathrm{O_2(gas)} \Rightarrow \mathrm{CO_2(gas)}$. Molecules of CO and O_2 in gaseous form are absorbed on the surface of the catalyst with reaction rates k_1 and k_2 for the adsorption, and k_3 for the reaction.

(2) The adsorption of gaseous molecules of oxygen in two free positions of the surface:

$$\mathrm{O_2(gas)} + 2 \cdot \square \xrightarrow{k_2} 2\mathrm{O(ads)}. \tag{14}$$

(3) The reaction to form carbon dioxide and the release of two positions in the surface:

$$\mathrm{CO(ads)} + \mathrm{O(ads)} \xrightarrow{k_3} \mathrm{CO_2(gas)} + 2 \cdot \square. \tag{15}$$

(4) The desorption of CO(ads) molecules:

$$\mathrm{CO(ads)} \xrightarrow{k_4} \mathrm{CO(gas)} + \square. \tag{16}$$

(5) The diffusion of CO(ads) molecules:

$$\mathrm{CO(ads)}_i + \square_j \xrightarrow{k_5} \mathrm{CO(ads)}_j + \square_i. \tag{17}$$

On the other hand, neither the desorption nor the diffusion of the oxygen, O(ads) are taken into account due to its low mobility. The first three kinetic processes depend on the temperature T of the catalyst and the reaction rates k_i are given by the Arrhenius equation

$$k_i = A_i e^{-E_i/k_B T}, \tag{18}$$

where E_i are the activation energies, A_i the preexponential factor and k_B is the Boltzmann constant. The processes of desorption and diffusion of CO(ads) are not thermally activated, they are simple probabilistic processes with values $k_4 = p_1$ and $k_5 = p_2$.

7.3. *The CA-model of surface chemical reactions*

Following the scheme presented in Sec. 3, our CA model for the chemical reaction is defined as follows:

- The cellular space that represents the surface of the catalyst is a square lattice of $L \times L$ cells with periodic boundary conditions.

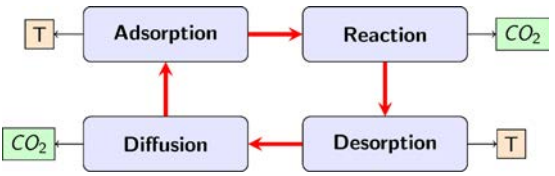

Fig. 15. Flowchart of the CA model. The processes of adsorption, reaction, desorption, and diffusion are applied sequentially to all the cells in parallel and at the end the catalyst temperature is evaluated. The adsorption and desorption processes modify the temperature while carbon dioxide is produced during the processes of reaction and diffusion.

- Each cell of the lattice can be empty (□), occupied for and atom of oxygen (O(ads)) or occupied for a molecule of carbon monoxide (CO(ads)).
- For the simulation of this chemical reaction the Margolus neighorhood is selected. It is necessary to take this neighborhood to maintain the stoichiometry of the reaction. As the CA updates simultaneously all lattice cells many unrealistic results are obtained, for example when a given cell can be desorbed and at the same time reacts with one of their neighbors. To avoid this, cell updating is not done individually but in groups of four cells all at once.
- The transition function $\mathbf{f}$ maps each one of the possible and different initial states. We can split $\mathbf{f}$ in four different rules $\mathbf{f} = \mathbf{f_1} \otimes \mathbf{f_2} \otimes \mathbf{f_3} \otimes \mathbf{f_4}$ corresponding to Eqs. (13)–(17). Figure 15 shows a flowchart of the different processes of the CA model.

Adsorption of CO(gas) and O(gas) and their reaction, rules $\mathbf{f_1}$ and $\mathbf{f_2}$**:** Fig. 16 shows all possible initial states of a Margolus block of 2×2 cells and their possible final states. For example an initial configuration of four empty cells can evolve to 8 possible outcomes (or stay empty) depending on the mole fraction of carbon monoxide $y_{\mathrm{CO}} = \frac{k_1}{k_1+k_2}$ and oxygen $y_O = 1 - y_{\mathrm{CO}}$ [38].

The desorption process, rule $\mathbf{f_3}$**:** each CO(ads) that has not reacted is given a probability p_1 of being desorbed. And rule $\mathbf{f_4}$ which is also a probabilistic rule simulates the movement of a CO(ads) molecule to an unoccupied and adjacent cell with a probability p_2 (see Eq. (17)).

With the aim to observe the existence of thermokinetic oscillations in our model [34] we take into account that the temperature T of the catalyst is not constant but varies following the equation:

$$\frac{dT}{dt} = -\gamma(T - T_B) + \sum_{i=1}^{5} h_i n_i, \tag{19}$$

where γ is a thermal relaxation parameter and T_B is the room temperature. The last term is the reaction heat obtained as the sum of each type of process, n_i is the number of processes and h_i the corresponding reaction heat.

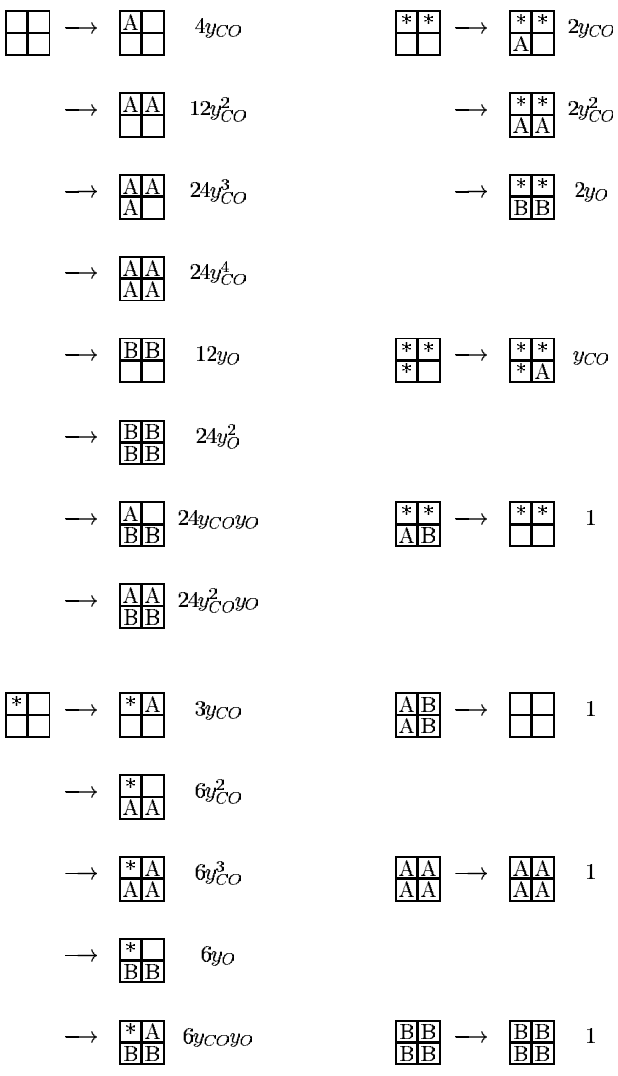

Fig. 16. CA transition rule f_1 for the adsorption and reaction processes of the CO + O_2 reaction. A is a CO(ads) molecule, B an O(ads) oxygen atom and $\star$ any occupied site either by A or B. It is shown all the possible Margolus blocks of 2×2 cells and their possible final states with the corresponding probability. See text for more details. Reproduced with permission from [34]. Copyright (2004) by AIP Publishing LLC.

7.4. *Results of the CA-model for the reaction*

The set of parameters used in our simulations are shown in Table 2. A detailed explanation of these values can be found in [34]. The values of these parameters remain fixed during the simulations, being γ, p_1, and p_2 the control parameters. To assure the validity of the simulations, different initial random number generator seeds in each simulation are used, obtaining practically the same results.

Table 2. Set of the chemical parameters used in the simulations.

Pre-exponential factor for CO adsorption	A_1	$5 \times 10^{-2}\,\mathrm{s}^{-1}$
Pre-exponential factor for O adsorption	A_2	$4 \times 10^{5}\,\mathrm{s}^{-1}$
Pre-exponential factor for CO_2 formation	A_3	$1\,\mathrm{s}^{-1}$
Activation Energy for CO adsorption	E_1	0
Activation Energy for O adsorption	E_2/k_B	$6 \times 10^{3}\,\mathrm{K}$
Activation Energy for CO_2 formation	E_3	0
Reaction Heat of CO adsorption	h_1	$150\,\mathrm{K\,s}^{-1}$
Reaction Heat of O adsorption	h_2	$300\,\mathrm{K\,s}^{-1}$
Reaction Heat of CO_2 formation	h_3	0
Reaction Heat of CO desorption	h_4	$-150\,\mathrm{K\,s}^{-1}$
Reaction Heat of CO diffusion	h_5	0
Room temperature	T_B	300 K

Without taking into account the diffusion or desorption of CO, the oxidation of the CO on catalytic metal surfaces (mainly Pt or Pd) can give rise to the four different regimes that are shown in Fig. 17:

(i) The oxygen poisons the surface for low values of $\gamma < 0.004$ and CO_2 is not formed.
(ii) As γ increases its value begins the production of CO_2 although irregularly. The time series of its concentration shows an irregular and chaotic behavior.
(iii) The best and most industrially interesting regime is reached in the range $0.10 \leq \gamma < 0.27$ when the production of the CO_2 is maintained over time showing an oscillatory and quasiperiodic behavior. Figure 18 shows a typical time series of the concentration of CO(ads) (n_{CO}) and O(n_{O}).

When CO(ads) desorption and diffusion are included two main effects are observed: the widening of the range of quasi-periodicity and a new intermittent behavior.

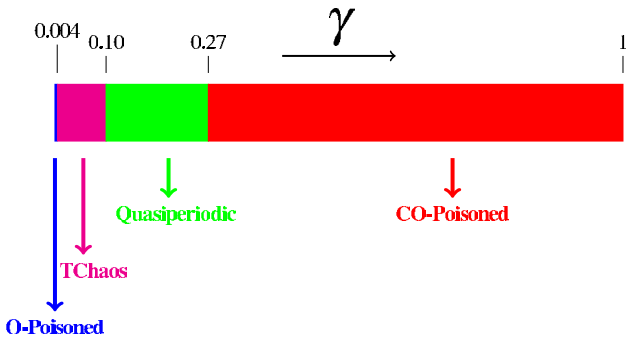

Fig. 17. The different regimes of the oxidation of CO depending on the thermal parameter γ without desorption ($p_1 = 0$) nor diffusion ($p_2 = 0$). (i) For $0 < \gamma < 0.004$, the surface is poisoned by oxygen; (ii) $0.004 \leq \gamma < 0.10$, temporal chaos; (iii) $0.10 \leq \gamma < 0.27$, a quasi-periodic behavior; (iv) $0.27 \leq \gamma$, CO the surface is poisoned by CO.

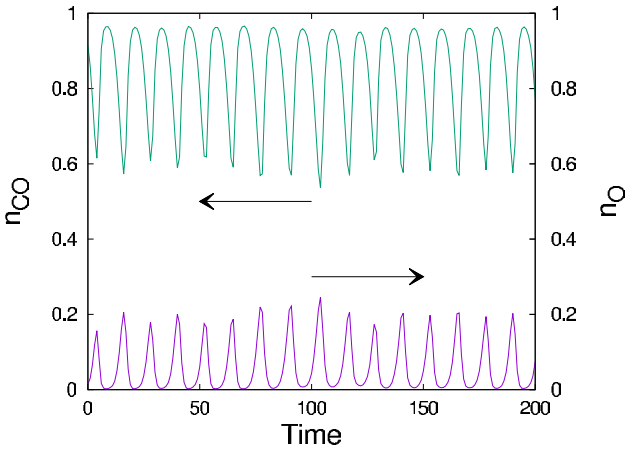

Fig. 18. Time series of the concentration of carbon monoxide (n_{CO}) and oxygen (n_O) adsorbed in the surface. Transient time has been discarded. The oscillatory behavior is a quasi-periodic one. Thermal parameter $\gamma = 0.10$ and $p_1 = 0, p_2 = 0$.

As we have previously indicated for high values of γ the surface of the catalyst is poisoned by CO(ads). But by introducing a probability p_1 for the desorption of CO(ads) the production of CO_2(gas) can be maintained at higher temperatures. In this way, the range of the quasi-periodic behavior is extended if p_1 is less than a critical value p_c, above which the catalytic surface would be poisoned by oxygen. For a thermal parameter of $\gamma = 0.3$, we have found in our simulations a value of $p_c \approx 0.032$, which is very close to the result obtained by Monte Carlo simulations [6] ($p_c \approx 0.039$).

It is also interesting to point out a quasi-periodic behavior that appears in the proximity of the poisoned CO(ads) state when desorption and diffusion are considered. Figure 19 shows the time series of n_{CO} and n_O adsorbed on the catalyst for $\gamma = 0.3$, $p_1 = 0.012$ and a diffusion parameter of $p_2 = 0.010$. Both concentrations show oscillations in antiphase with an amplitude in the range of values $0.2 - 0.9$ for n_{co} and $0.05 - 0.6$ for n_o. A similar oscillatory behavior was observed in the studies of Slin'ko *et al.* [49, 50] about the CO oxidation under isothermal conditions and high pressures.

The CA model has the capacity to show the spatio-temporally patterns that the reaction shows when the intermittent oscillatory behavior is settled. Figure 20 shows a map of the surface where regions of the same concentration are drawn in the same color for $\gamma = 0.3, p_1 = 0.012, p_2 = 0.010$. Circular and elliptical regions that have the same concentration spread on the surface of the catalyst. More irregular structures are observed as γ increases pointing out a possible transition to chaos via intermittecy.

7.5. *The phase space*

The complete set of different behaviors, the phase space, shown by the chemical reaction when parameters p_1 and γ are modified are obtained by calculating the

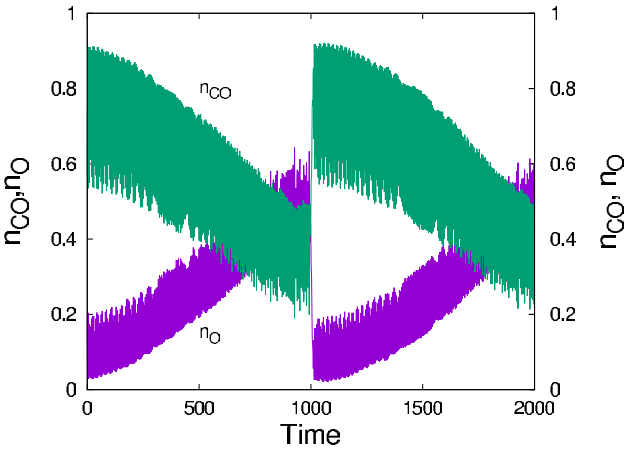

Fig. 19. Time series of the concentration of CO and O adsorbed on the surface. Transient time has been discarded. Thermal parameter $\gamma = 0.3$, desorption probability $p_1 = 0.012$, diffusion probability $p_2 = 0.010$.

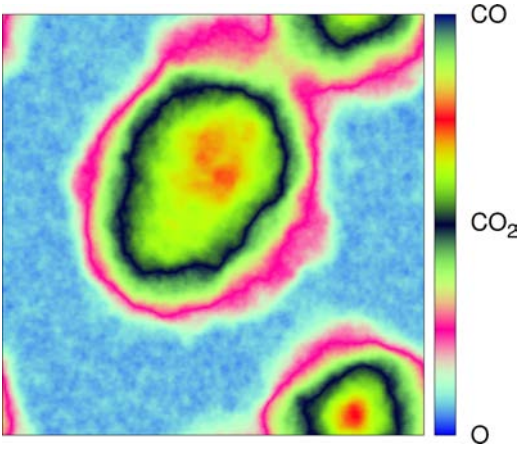

Fig. 20. (Color online) Spatio-temporal pattern observed in the surface chemical reaction for $\gamma = 0.3, p_1 = 0.012, p_2 = 0.10$. The color palette is selected to show clearly the circular structures. Blue: oxygen, Green-Black: carbon monoxide and carbon dioxide.

Shannon's entropy defined previously:

$$S(p_1, \gamma) = -\sum_{i}^{m} f_i(p_1, \gamma) \cdot \ln f_i(p_1, \gamma). \tag{20}$$

As the effect of the diffusion is less important than the desorption, we set $p_2 = 0$ and so in this way we analyze the dependency of the entropy only with p_1 and γ. We take the time series of n_{co} and calculate the frequencies $f_i(p_1, \gamma)$ of the corresponding values of n_{CO} distributed in $m = 100$ bins.

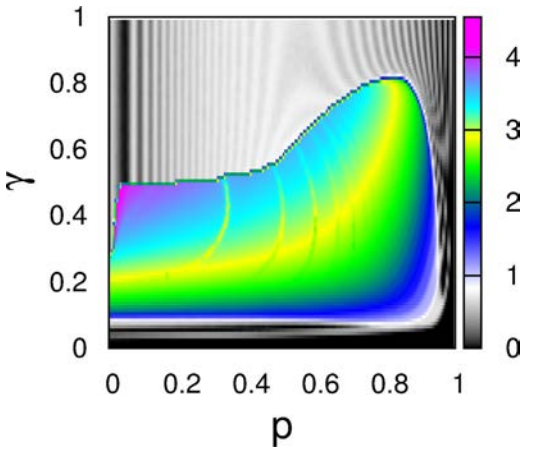

Fig. 21. Phase space of the chemical reaction in terms of the thermal parameter γ and the probability of desorption p_1. There is no diffusion $p_2 = 0$. The different colors indicate the value of the Shannon entropy and the different regimes of the reaction.

Figure 21 shows the entropy for the reaction. The different colors indicate the value of S and are associated with the various behavior. Values between 0 and 1 (black and white) correspond to the poisoning of the surface either by oxygen (in the region of low γ) or by carbon monoxide (for high γ). Simple periodic oscillatory behaviors are located in the range of values 1–3 while the regions with values between 3 and 4 correspond to a quasi-periodic behavior. Values greater than 4 indicate the zone where the intermittent behavior is found.

The CA model is able to capture the different behaviors observed in the catalyzed reaction of the carbon monoxide as a function of the temperature of the catalyst. It is a tool that even allows us to analyze the spatio-temporal structures that arise as a consequence of the dynamics of the reaction. And finally making use of the calculation of entropy shows us the complete space of the phases.

8. Discussion and Conclusions

The paper starts by introducing the key concepts of CSs modeling and particularly the CA technique, which is often used as an alternative to the classical approach of the natural phenomena modeling but sometimes represents the only method capable of CSs mathematical and computational description. From a simplified point of view, a real system is maped to the CA computational universe (space) which is studied instead of the original CS. The evolving global state of the computational universe is defined by and results from a lattice of computational elements where each element contains a set of variables that can change their values depending on the local environment and according to specific laws (transition rules). Those transition rules (microstates) determine and drive the spatio-temporal behavior of the global

properties of the evolving universe (macrostate). Global properties and features of the system are measured and compared to experiments as is demonstrated on all three case studies: laser emission, dynamic recrystallization, and chemical reaction.

In a CA simulation, first of all, the space is discretized in a grid of cells where each one accomodate a finite number of variables corresponding to the state variables. The environment is reduced to a set of cells creating the computational world, the neighborhood, and the set of transition functions driving evolution of cells (computational units) that describe physical, chemical or biological laws. Apart from the discretization of the space and the time, the CA technique seems the be the same as the classical approaches but not quite. In a physical system, for example, once the physical law is known one is able — at least theoretically — to describe all the possible macroscopic behaviors of the system. The goal is the law. On the other hand, in a CA model the transition functions are kept as simple as possible and they do not contain information about the global behavior of the system. In the CA simulation, the key computational concepts are based on application of the self-organization and emergence, the macroscopic behavior of the system appears spontaneously from the interaction of its individual, constituting elements.

This work provides a practical introduction into the methodology of cellular automaton models of CSs design. All steps that must be accomplished during the development of a successful model are covered in detail. For a deeper understanding of the methodology, selected algorithms, flow diagrams, and open source codes of models are provided. The modeling principles are explained on three models of natural phenomena observed in quantum physics, solid state physics, and chemistry. The three examples provided here have many similarities that make them fit into the general methodology for CSs modeling using CAs that has been introduced in Sec. 3. Let us compare the three models by focusing on the summary of their characteristics introduced in Table 1 and on the descriptions of the models. We have chosen to use 2D models because they are very easy to simulate, visualize, and understand. The neighborhood of a cell is the Moore neighborhood except in the case of the chemical reaction where Margolus alternating neighborhood is used because it is necessary to take care of the conservation of the mass. In all models, the dynamics of the system is defined through a set of microscopic processes specified by CA transition rules.

Running the simulation for different values of the system parameters leads to different types of macroscopic behaviors that emerge spontaneously. It is constant, quasi-periodic, or oscillatory behavior for the laser and chemical reaction models. Similarly, it is single peak followed by a steady state, or oscillatory behavior for DRX. Two of the models (laser and chemical reactions) provide an example of how the dependence of the macroscopic behavior of the system on its parameters can be mapped by using the Shannon entropy, as shown in Figs. 7 and 21. Additionally, all three models show one of the main advantages of CA models compared to traditional macroscopic models based on differential equations: they can be easily used to study the spatio-temporal patterns produced in the system during its evolution, which is

difficult to achieve in finite element methods. An example of it is demonstrated on the DRX, see Fig. 12, and on the chemical reaction model, see Fig. 20.

The simulation of evolving spatio-temporal patterns is very important for understanding of many real systems phenomena, such as population dynamics in ecosystems, biological cells behavior in living tissues, or traffic jam patterns. In particular, we think the ability of CA models to reproduce this kind of patterns can be exploited to model many phenomena for which they have not been applied yet in the fields of biology and medicine.

The selection of the appropriate type of neighborhood for a given model situation is critical for achieving the correct behavior in some models, in which incorrectly selected neighborhoods bring undesired anisotropies into the model destroying its outcomes. This is the case of the recrystallization model, in which the growth of squared grains was resolved by probabilistic movement of grain boundary segments. In other models in which isotropy is not essential, the selecton of neighborhood is not so important and it is enough to choose the simplest neighborhood that can capture the behavior of the system, such as the Moore neighborhood in the case of the laser model. In chemical reaction models such as the carbon monoxide oxidation one, a special type of Margolus neighborhood with two alternanting sub-neighborhoods is required, in order to respect the conservation of the mass.

Computational boundary conditions must reflect the physical boundary conditions or they have to be excluded from the simulation by application of the periodic boundary conditions (by "gluing" the opposite sides of the world together) as it was done in three models described in the paper, otherwise they can generate unrealistic and unwanted physical effects.

CA already proved their usefulness and exclusivity in modeling of CSs within a wide range of scientific fields. Additionally, there currently exists a huge potential of using them to build models in biology and medicine not only to replicate the natural phenomena but also to study their behavior with conditions difficult to achieve in practice due to ethical reasons. A very important feature of CA modeling is its relative simplicity that enables to easily build a prototypic models of various massively parallel phenomena, which can be later recasted into more specialized approaches requiring much greater effort to build. This work can become a useful tool that can be used by researchers from diverse scientific fields to develop their own cellular automata models of CSs.

Acknowledgments

This work was supported by the following research projects of Ministerio de Economia, Industria y Competitividad (MINECO) and the "Agencia Estatal de Investigacion (AEI)" of Spain, co-financed by FEDER funds (EU): MABICAP (Bio-inspired machines on High Performance Computing platforms: a multidisciplinary approach, TIN2017-89842P) and by the National Sustainability Program I (NPU I) Nr. LO1503 provided by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] Aburas, M. M., Ho, Y. M., Ramli, M. F. and Ash'aari, Z. H., The simulation and prediction of spatio-temporal urban growth trends using cellular automata models: A review, *Int. J. Appl. Earth Obs. Geoinf.* **52** (2016) 380–389.

[2] Bandman, O. L. and Kireeva, A. E., Stochastic cellular automata simulation of oscillations and autowaves in reaction-diffusion systems, *Numer. Anal. Appl.* **8** (2015) 208–222.

[3] Bar-Yam, Y., *Dynamics of Complex Systems* (Addison-Wesley, Reading, MA, 1997).

[4] Blázquez, J., Franco, V., Conde, C., Millán, M. and Conde, A., Instantaneous growth approximation describing the nanocrystallization process of amorphous alloys: A cellular automata model, *J. Non-Cryst. Solids* **354** (2008) 3597–3605.

[5] Boccara, N., *Modeling Complex Systems* (Springer, 2010).

[6] Brosilow, B. J. and Ziff, R. M., Effects of a desorption on the first-order transition in the A-B_2 reaction model, *Phys. Rev. A* **46** (1992) 4534–4538.

[7] Burkhead, E. and Hawkins, J., A cellular automata model of Ebola virus dynamics, *Physica A: Stat. Mech. Appl.* **438** (2015) 424–435.

[8] Chopard, B. and Droz, M., *Cellular Automata Modeling of Physical Systems* (Cambridge University Press, UK, 1998).

[9] D'amico, M., Manzini, G. and Margara, L., On computing the entropy of cellular automata, *Theor. Comput. Sci.* **290**(3) (2003) 1629–1646.

[10] Dickman, R., Kinetic phase transitions in a surface-reaction model: Mean-field theory, *Phys. Rev. A* **34** (1986) 4246–4250.

[11] Gao, J., Hu, J. and Tung, W.-W., Entropy measures for biological signal analyses, *Nonlinear Dyn.* **68**(3) (2012) 431–444.

[12] Gerhardt, M. and Schuster, H., A cellular automaton describing the formation of spatially ordered structures in chemical systems, *Physica D* **36** (1989) 209–221.

[13] Gibson, M. J., Keedwell, E. C. and Savić, D. A., An investigation of the efficient implementation of cellular automata on multi-core CPU and GPU hardware, *J. Parallel Distrib. Comput.* **77** (2015) 11–25.

[14] Gladkov, D., Tapia, J.-J., Alberts, S. and D'Souza, R. M., Graphics processing unit based direct simulation Monte Carlo, *Simulation* **88** (2012) 680–693.

[15] Gounaridis, D., Chorianopoulos, I. and Koukoulas, S., Exploring prospective urban growth trends under different economic outlooks and land-use planning scenarios: The case of Athens, *Appl. Geogr.* **90** (2018) 134–144.

[16] Guidolin, M., Chen, A. S., Ghimire, B., Keedwell, E. C., Djordjević, S. and Savić, D. A., A weighted cellular automata 2D inundation model for rapid flood analysis, *Environ. Model. Softw.* **84** (2016) 378–394.

[17] Guisado, J., Fernandez de Vega, F., Jimenez-Morales, F., Iskra, K. and P. S., Using cellular automata for parallel simulation of laser dynamics with dynamic load balancing, *Int. J. High Perform. Syst. Archit.* **1** (2008) 251–259.

[18] Guisado, J. L., Modelling and simulation of laser dynamics with cellular automata on parallel and distributed computers, Ph.D. thesis, University of Seville, Spain (2009), https://doi.org/10.13140/RG.2.2.13356.51840.

[19] Guisado, J. L., Fernández de Vega, F. and Iskra, K., Performance analysis of a parallel discrete model for the simulation of laser dynamics, in *2006 Int. Conf. Parallel Processing, ICPP 2006, Workshops* (IEEE Computer Society, 2006), pp. 93–99.

[20] Guisado, J. L., Fernández de Vega, F., Jiménez-Morales, F. and Iskra, K., Parallel implementation of a cellular automaton model for the simulation of laser dynamics, *Lect. Notes Comput. Sci.* **3993** (2006) 281–288.

[21] Guisado, J. L., Jiménez-Morales, F. and Fernández de Vega, F., Cellular automata and cluster computing: An application to the simulation of laser dynamics, *Adv. Complex Syst.* **10** (2007) 167–190.
[22] Guisado, J. L., Jiménez-Morales, F. and Guerra, J. M., Cellular automaton model for the simulation of laser dynamics, *Phys. Rev. E* **67** (2003) 066708.
[23] Guisado, J. L., Jiménez-Morales, F. and Guerra, J. M., Simulation of the dynamics of pulsed pumped lasers based on cellular automata, *Lect. Notes Comput. Sci.* **3305** (2004) 278–285.
[24] Guisado, J. L., Jiménez-Morales, F. and Guerra, J. M., Application of Shannon's entropy to classify emergent behaviors in a simulation of laser dynamics, *Math. Comput. Model.* **42** (2005) 847–854.
[25] Guisado, J. L., Jiménez-Morales, F. and Guerra, J. M., Computational simulation of laser dynamics as a cooperative phenomenon, *Phys. Scr.* **T118** (2005) 148–152.
[26] Hoekstra, A., Kroc, J. and Sloot, P., *Simulating Complex Systems by Cellular Automata* (Springer, Heidelberg, 2010).
[27] Holovatch, Y., Kenna, R. and Thurner, S., Complex systems: Physics beyond physics, *Eur. J. Phys.* **38** (2017) 023002.
[28] Howes, L. and Thomas, D., Efficient random number generation and application using CUDA, in *GPU Gems*, Vol. 3, Nguyen, H. (ed.) (NVIDIA-Pearson, 2007), ISBN 0321515269, pp. 805–830.
[29] Ilachinski, A., *Cellular Automata: A Discrete Universe* (World Scientific, Singapore, 2001).
[30] Jelinek, B., Eshraghi, M., Felicelli, S. and Peters, J. F., Large-scale parallel lattice Boltzmann-cellular automaton model of two-dimensional dendritic growth, *Comput. Phys. Commun.* **185** (2014) 939–947.
[31] Kier, L. B., Seybold, P. G. and Cheng, C.-K., *Modeling Chemical Systems Using Cellular Automata* (Springer, The Netherlands, 2005).
[32] Kroc, J., Simulation of dynamic recrystallization by cellular automata, Ph.D. thesis, Charles University, Prague, The Czech Republic (2001).
[33] Kroc, J., Application of cellular automata simulations to modelling of dynamic recrystallization, *Lect. Notes Comput. Sci.* **2329** (2002) 773–782.
[34] Lemos, M. C. and Jiménez-Morales, F., A cellular automaton for the modeling of oscillations in a surface reaction, *J. Chem. Phys.* **121** (2004) 3206–3211.
[35] Li, X., Zhang, X., Yeh, A. and Liu, X., Parallel cellular automata for large-scale urban simulation using load-balancing techniques, *Int. J. Geogr. Inf. Sci.* **24** (2010) 803–820.
[36] Liu, X., Liang, X., Li, X., Xu, X., Ou, J., Chen, Y., Li, S., Wang, S. and Pei, F., A future land use simulation model (FLUS) for simulating multiple land use scenarios by coupling human and natural effects, *Landsc. Urban Plan.* **168** (2017) 94–116.
[37] Lopez-Torres, M., Guisado, J., Jimenez-Morales, F. and Diaz-del Rio, F., Gpu-based cellular automata simulations of laser dynamics, in *Jornadas de Paralelismo (SARTECO) 2012* (Elche (Alicante), Spain, 2012), pp. 261–266, https://hdl.handle.net/11441/74173.
[38] Mai, J. and Niessen, W. V., Cellular-automaton approach to a surface reaction, *Phys. Rev. A* **44** (1991) R6165–R6168.
[39] Monteagudo, A. and Santos, J., Treatment analysis in a cancer stem cell context using a tumor growth model based on cellular automata, *PLoS One* **10** (2015) e0132306.
[40] Mužák, M. and Kroc, J., Design and imlementation of cellular automaton simulating domain growth, Czech Technical University and Charles University, Prague & Pilsen (2016).
[41] Mužák, M. and Kroc, J., Video depicting example run of diffusion controlled cellular automaton simulating domain growth (example of self-organization), Video sequence - test run, Czech Technical University and Charles University, Prague & Pilsen (2016).

[42] Prigogine, I., Stengers, I. and Toffler, A., *Order Out of Chaos: Man's New Dialogue with Nature* (Bantam New Age Books, New York, 1984).
[43] Qiang, S., Jia, B., Huang, Q. and Jiang, R., Simulation of free boarding process using a cellular automaton model for passenger dynamics, *Nonlinear Dyn.* **91** (2018) 257–268.
[44] Qiu, G., Kandhai, D. and Sloot, P. M. A., Understanding the complex dynamics of stock markets through cellular automata, *Phys. Rev. E* **75** (2007) 046116.
[45] Rozier, O. and Narteau, C., A real-space cellular automaton laboratory, *Earth Surf. Process. Landf.* **39** (2014) 98–109.
[46] Sakai, T. and Jonas, J. J., Dynamic recrystallization: Mechanical and microstructural considerations, *Acta Metall. Mater.* **32** (1984) 189–298.
[47] Shannon, C. E., A mathematical theory of communication, *Bell Syst. Tech. J.* **27**(3) (1948) 379–423.
[48] Siegman, A., *Lasers* (University Science Books, Mill Valley, CA, 1986).
[49] Slin'ko, M. M., Kurkina, E. S., Liauw, M. A. and Jaeger, N. I., Mathematical modeling of complex oscillatory phenomena during CO oxidation over Pd zeolite catalysts, *J. Chem. Phys.* **111** (1999) 8105–8114.
[50] Slin'ko, M. M., Ukharskii, A. A., Peskov, N. V. and Jaeger, N. I., Identification of the intermittency-I route to chaos in oscillating CO oxidation on zeolite-supported Pd, *Faraday Discuss.* **120** (2001) 179–195.
[51] Stamatakis, M., Kinetic modelling of heterogeneous catalytic systems, *J. Phys.-Condens. Mat.* **27** (2015) 013001.
[52] Tang, T.-Q., Luo, X.-F., Zhang, J. and Chen, L., Modeling electric bicycle's lane-changing and retrograde behaviors, *Physica A: Stat. Mech. Appl.* **490** (2018) 1377–1386.
[53] Tkáč, J., Design and implementation of cellular automaton simulating dynamic recrystallization, Master's thesis, Czech Technical University, The Czech Republic (2017).
[54] Tkáč, J. and Kroc, J., Cellular automaton simulation of dynamic recrystallization: Introduction into self-organization and emergence, Open source computational software, Czech Technical University and Charles University, Prague and Pilsen (2017).
[55] Tkáč, J. and Kroc, J., Objects definition: Cellular automaton simulation of dynamic recrystallization, Program's objects structure, Czech Technical University and Charles University, Prague and Pilsen (2017).
[56] Tkáč, J. and Kroc, J., Self-organization: Cellular automaton simulation of dynamic recrystallization, Video sequence - test run (avail. on researchgate), Czech Technical University and Charles University, Prague and Pilsen (2017).
[57] Toffoli, T. and Margolus, N., *Cellular Automata Machines: A New Environment for Modeling* (MIT Press, Cambridge, 1987).
[58] Ulam, S. M., Random processes and transformations, in *Proc. Int. Congress of Mathematicians*, Vol. 2 (Cambridge, MA, 1952), pp. 264–275.
[59] von Neumman, J., *The Theory of Self-Reproducing Automata* (University of Illinois Press, 1966).
[60] Youssef, B. B., A parallel cellular automata algorithm for the deterministic simulation of 3-D multicellular tissue growth, *Cluster Comput.* **18** (2015) 1561–1579.
[61] Zhdanov, V. P., Monte Carlo simulations of oscillations, chaos and pattern formation in heterogeneous catalytic reactions, *Surf. Sci. Rep.* **45** (2002) 231–326.