Parallel Computing, Vol. 27 (2001) 601-611

Coupling Microscopic and Macroscopic Cellular Automata

Jörg R. Weimar
Institute of Scientific Computing, Technical University Braunschweig
D–38092 Braunschweig, Germany
Tel. +49-531-391-3006; E-mail J.Weimar@tu-bs.de

Abstract

Cellular automata can model natural phenomena on many different levels of detail. Often, one specific level of detail is appropriate for the problem under investigation. But in some cases a connection between the different levels of detail is necessary. One such case is the catalytic reaction on a surface. The homogeneous crystallographic surface can reasonably well be described by mesoscopic approaches, but in the presence of defects, a microscopic simulation (where each site of the crystal lattice is individually represented) is necessary. In this paper we present a framework for coupling different types of cellular automata to achieve an efficient simulation which is nevertheless detailed enough to resolve microscopic phenomena where necessary.

Keywords: Cellular automata, reaction-diffusion systems, coupling, catalytic surface reactions, simulation.

Catalytic surface reactions

Chemical reactions on a catalytic surface have often been modeled by Monte-Carlo methods or cellular automata. In fact, Monte-Caro methods can usually be converted into probabilistic cellular automata. Here we consider the surface reaction

$$A + \tfrac{1}{2}B_2 \to 0 \tag{1}$$

which is understood to mean that some molecule B_2 is adsorbed at the surface, and splits into the molecular or atomic form B upon adsorption. A second molecule A is also adsorbed at the surface, and if A and B are neighbors on the surface, they can react to a molecule AB, which desorbs immediately in this model and does not participate in further reaction steps (therefore $\to 0$: reaction to "nothing"). This scheme is appropriate for the oxidation of carbon monoxide on platinum [4, 5].

A microscopic simulation of this reaction can be constructed fairly simply. Consider a regular lattice, the geometry of which reflects the crystallographic lattice present on the crystal face under consideration. On empty sites of this lattice, the species A can adsorb with probability p_A. On two adjacent empty sites, a molecule of B_2 can adsorb and split into atoms B. If a site occupied by

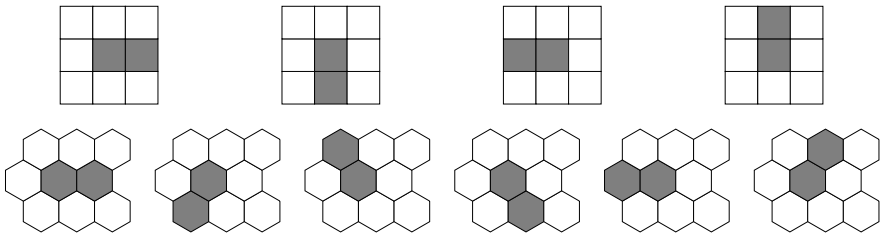

Figure 1: Positions for blocks of two cells on a square and hexagonal lattice.

an A is adjacent to a site with a B, they react immediately and liberate both lattice sites. In addition to these processes, diffusion of A, B, or both on the lattice can also be introduced.

Microscopic Cellular Automata Model

The model as described so far is best realized as a Monte Carlo simulation. A lattice site is selected, and each applicable process is executed with the associated probability [4].

Here we convert the model into a probabilistic cellular automaton (an early, more complicated CA is presented in [5], a better version is [1]). The state of each cell is given by the possible states of a lattice site (0, A, B). One problem is presented by the exclusion condition. A particle can only react with exactly one neighbor, and it should not react with two neighbors at the same time. In the Monte-Carlo method this is prevented by the sequential investigation of the lattice sites. In a cellular automaton, special provisions have to be made to avoid unwanted interference. In our case a change takes place simultaneously at two adjacent sites. In order to model this behaviour, it is best to use blocked cellular automata, i.e., a formulation where blocks of neighbors are completely updated in parallel in one step, and block boundaries are changed between steps. In the case considered here, blocks of two cells are sufficient, but all possible block orientations need to be taken into account. Thus the cellular automaton could be described by

$$[0,\cdot],[\cdot,0] \rightarrow [A,\cdot],[\cdot,A] \quad \text{with probability } p_A \tag{2a}$$

$$[0,0] \rightarrow [B,B] \quad \text{with probability } p_B \tag{2b}$$

$$[A,B],[B,A] \rightarrow [0,0] \quad \text{with probability } R \tag{2c}$$

$$[A,0] \leftrightarrow [0,A] \quad \text{with probability } D_A \tag{2d}$$

where p_A and p_B are the adsorption probabilities, R is the probability of the reaction taking place, D_A is the diffusion probability for A (Species B does not diffuse on the lattice in this model), and $\cdot$ can stand for any of A, B, 0. All possible block orientations must be used at subsequent steps. As shown in Figure 1, on a square lattice this leads to four possible orientations and on a triangular lattice there are six possible orientations.

The four/six directions can be updated always in the same order, using different permutations of the order of updating for each sequence of four/six

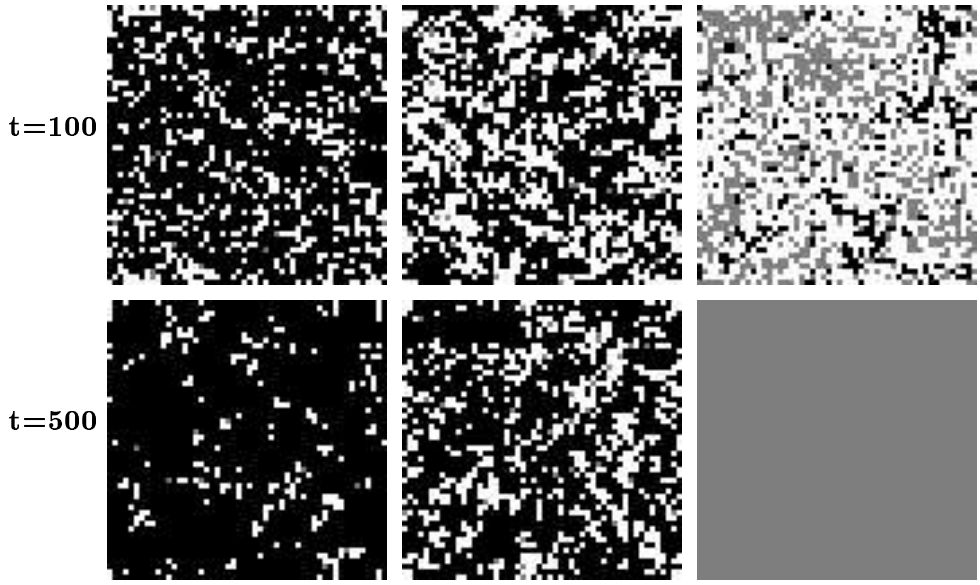

Figure 2: Simulation lattice for $p_A = 0.02$, $p_A = 0.03$, and $p_A = 0.04$, respectively from left to right, and $p_B = 0.1 - p_A$. Color code: Empty , A , B .

updates, or a direction can be selected at random for each step. If all the probabilities p_A, p_B, R, and D_A are small, all three choices are equivalent, but for large probabilities (and consequently more efficient simulations), different choices can lead to slightly different results.

Note that the probabilities p_A, p_B, and D_A may depend on the position, e.g., around defects in the crystal.

Microscopic Simulation Results

Microscopic simulations are performed on a square lattice (corresponding to the (100)-face of a platinum crystal) with the following probabilities fixed: $p_A + p_B = 0.1$, $D_A = 0.5$, $R = 1.0$, and the balance bwteen p_A and p_B varying. The simulation is started with an empty lattice. One can observe that for small p_A, the lattice ist poisoned by B-particles, so that after a certain number of steps only very few reactive events happen at those few sites that are not occupied by B-particles. For large p_A, the lattice is fully covered by A-particles, so no B-particles can adsorb, and no reaction takes place. Only for intermediate values of p_A are reactions possible at the interfaces between A-regions and B-regions.

Figure 2 shows the occupation of the lattice of size 50×50 after 100 and 500 time steps with three different values of p_A. The measured steady-state values for the concentrations of A and B particles as a function of p_A are shown with dots in Figure 3.

Macroscopic Model

In order to simulate a macroscopic catalytic surface, the microscopic model would need to be extremely large: A typical length scale for one crystallographic lattice length is on the order of one nanometer, so to simulate just 1mm^2, we need $10^6 \times 10^6$ cells, which is impractical. On the other hand, with analytical methods we can capture the statistical behaviour of the catalytic surface reaction in the homogeneous case to a good extent [3]. This leads to a partial differential equation. In the present model, a simple mean-field approach leads to the macroscopic equations

$$\frac{\partial c_A}{\partial t} = D_A \nabla^2 c_A + p_A c_0 - R c_A c_B \tag{3a}$$

$$\frac{\partial c_B}{\partial t} = p_B c_0^2 - R c_A c_B \tag{3b}$$

where c_A and c_B are the concentrations of A and B particles on the surface and $c_0 = 1 - c_A - c_B$ is the concentration of empty sites on the surface.

A more sophisticated mesoscopic model would include some measure of the correlations [2] in order to capture phenomena that cannot be explained by the simple model Eq. (3). The first step is to use as variables not the probabilities of single-site occupations as in the mean-field model, but probabilities $s_{i,j}$ of blocks of two sites. Of these block-probabilities there are already $n(n-1)$ for n possible cell occupations. In our case ($n = 3$) we get five equations (plus one normalization condition), considerably more complicated than the mean-field equations (3). The details of the derivations of these equations can be found in [2], where the method is called cluster-expansion.

In this paper we use the mean-field equations for modeling space-dependent phenomena, and the results of the cluster-expansion only for comparing steady states in Figure 3.

The system (3) of differential equations can be simulated by any scheme for integrating differential equations, once a suitable spatial discretization and a suitable representation for the locally averaged quantities has been found.

One possibility to simulate such a reaction-diffusion system is using cellular automata. In this case a macroscopic approach is used for regions in which the parameters are uniform, and where also the solution is statistically uniform. The CA approach [10, 7, 9, 8] is based on the construction principle of discretizing the phase space, which in this case means discretizing the variables $c_A, c_B \in [0, 1]$, e.g., into 100 levels each. This discretization makes it possible to replace the calculation of the r.h.s. of Eqns. (3) by a lookup-table, since only a finite number of value pairs (c_A, c_B) can occur, and the r.h.s. can be precalculated for all of them. Of course, the diffusion operator can easily be implemented in a cellular automaton in a variety of ways. The discretization of the state variables is maintained by probabilistic rounding in each step, which ensures that on average there is no rounding bias for any point in phase space. Details of this method can be found in [10, 8], and are not the subject of this paper. For the purpose of this paper, the macroscopic simulation could be done by standard finite difference methods.

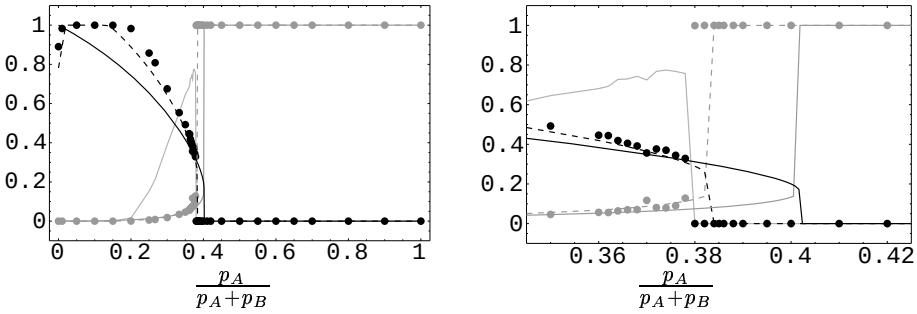

Figure 3: Steady state values for the concentrations of A and B particles. The dots mark measurements, the full lines results from mean-field analysis, and the dashed lines the result of the cluster-expansion. On the right is a magnified representation of the critical region. Colors: A ■, B ■. The light gray, peaked line shows the rate of reactions taking place.

Coupling both Models

In order to be able to efficiently simulate a catalytic system with homogeneous and nonhomogeneous regions, we need to combine a microscopic and a macroscopic model. The microscopic model is used at and around the inhomogeneities, while the macroscopic model deals with the (large) homogeneous regions. Since the macroscopic model uses averaged quantities, at the interface an averaging of the microscopic quantities must take place. In the other direction, stochastic values must be generated from the averaged (and possibly correlated) quantities. In addition, the lattices for microscopic and macroscopic simulations will differ in the length scale, and possibly also in the type of lattice (e.g., microscopic hexagonal lattice, macroscopic square lattice). The following operations are needed for coupling the two models:

- Average the microscopic quantities over space and time in a border region.
- Generate correlated boundary conditions for the microscopic model from given macroscopic values.
- Both procedures must cope with different lattice types and resolutions.
- Simulate each model for a given time (with different time steps).

We developed an object-oriented framework for this problem [6]. The idea is to encapsulate existing software for the individual subproblems in an object-oriented view. In this view, the simulation region is divided into subregions (currently of rectangular shape) and each subregion is simulated by one program-object. These objects provide methods for initialization, for simulation, and for the data exchange and conversion tasks listed above. The internal data structures of the original programs represent the objects' internal state, but can only be accessed through the standardized object methods. The important feature is that different simulation programs (here one for microscopic simulation, and

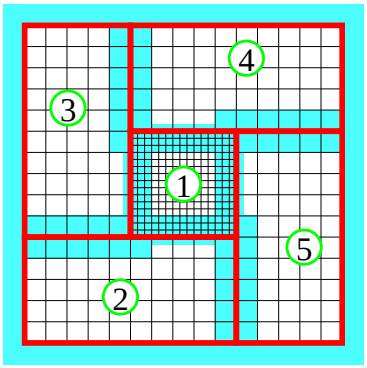

Figure 4: Typical geometry for a problem which needs microscopic simulation at the center and macroscopic simulation on the outside.

one for macroscopic simulation) implement the same interface and therefore can be coupled in any way as dictated by the problem at hand.

For controlling the objects and steering the simulation we have implemented a central controller guided by a script language. In this script the geometry of the problem is specified and a number of objects are created. The controller determines the neighborhood relations between objects so it can later initiate all the necessary data exchanges for the simulation.

Figure 4 shows a typical configuration of five objects, one of which uses a microscopic simulation with a fine grid, while the others fill the surrounding space with a macroscopic simulation on a coarse grid.

When such a configuration is specified, the controller identifies all line segments where two objects meet, and determines the necessary exchange operations for updating the border regions.

In the case of Figure 4, object 2 exchanges data on its eastern side with object 5, and on its northern side there are two exchanges, one with object 1 and one with object 3. In Figure 4, the shaded regions show the cells which need to be exchanged at each time step. In the exchange between objects 2 and 1, we also note that some interpolation needs to take place. This can be done on the side of the sender or on the receiving end. The controller must instruct the objects which of them is to interpolate the data. In the present example, the user specifies in the script that the object 1 should always do the interpolation, since it also needs to generate data stochastically as it uses a microscopic simulation.

The structure of the simulation framework is shown in Figure 5. To help integrate further simulation programs into the framework, we have collected some common operations needed for implementing the object interface in a library. Since the simulation is intended to run on a parallel computer, a decision had to be made on how to implement the control and data exchanges between controller and objects. One possibility would have been to use a remote method invocation as implemented in Java-RMI or the CORBA standard. In this work we preferrred the simpler way of using the standard MPI message-passing li-

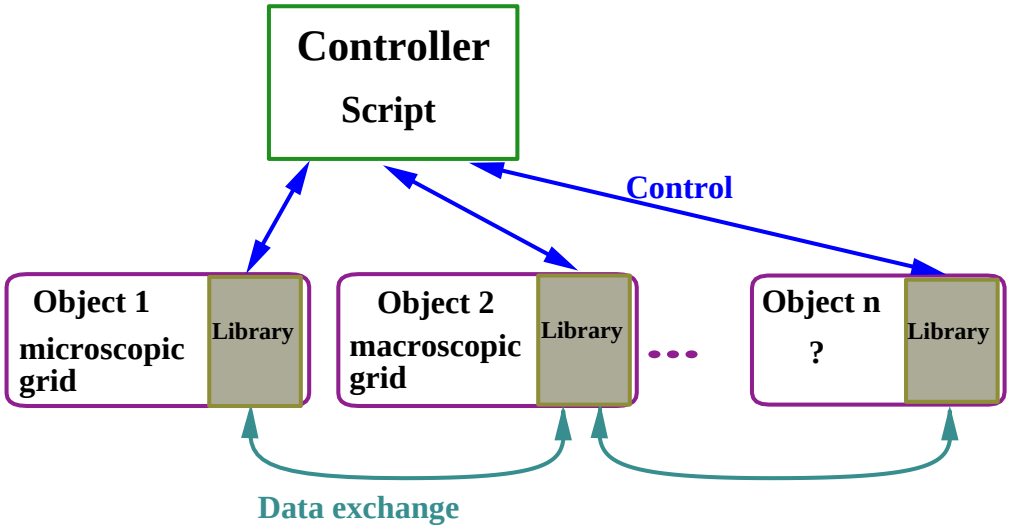

Figure 5: Structure of the simulation framework.

brary on parallel computers, and to map the method calls to MPI messages. This is possible since the number and signature of all methods are known in advance. Each object then needs to implement a central loop of waiting for messages and executing the associated method upon receipt of a message. This control loop is also provided in the library.

The same framework is also used to parallelize the simulation by coupling several objects of the same type.

Conversion between microscopic and macroscopic cells

The data exchange between microscopic parts and macroscopic parts involves two directions.

To calculate a boundary condition for the macroscopic simulation, one needs to combine the state of n microscopic cells. This is simply done by counting the number of A-particles n_A and the number of B-particles n_B and calculating the local averages n_A/n and n_B/n, which can then be used in the macroscopic simulation.

The opposite direction is slightly more complicated: from given densities c_A and c_B, one generates stochastically n cells with n_A A-particles and n_B B-particles such that $n_A/n \approx c_A$ and $n_B/n \approx c_B$. For mean-field calculations, no correlations are introduced, therefore each cell can be filled independently.

For macroscopic models using correlated quantities, e.g., nearest neighbor pair correlations $s_{i,j}$, correlated microscopic conditions need to be generated. One possibility is to first generate uncorrelated conditions with the correct densities. Then an iterative process is used to refine the field: First the pair densities of the generated field are measured. This is a vector of 6 components in the case of three possible cell values (0,A,B): $s_{0,0}, s_{0,A}, s_{0,B}, s_{A,A}, s_{A,B}, s_{B,B}$.

This vector now must be changed to make it closer to the desired vector of pair densities. One microscopic cell is chosen at random. For all possible

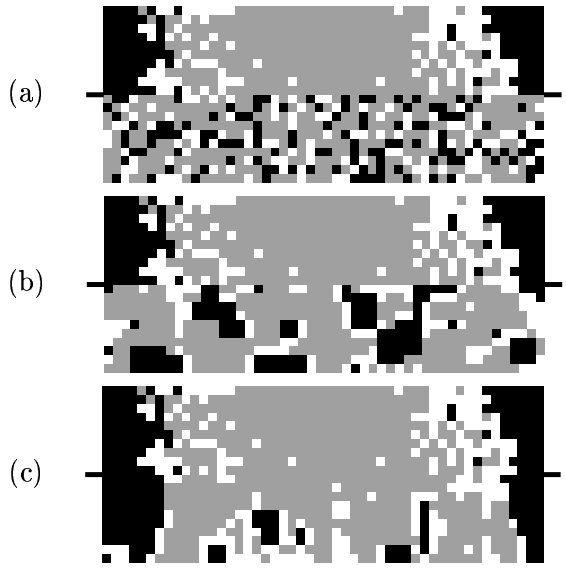

Figure 6: Stochastic generation of boundary conditions. The upper half of each image is the result of microscopic simulations, the lower half is (a) randomly generated with the same particle densities as measured in the upper half, but without correlations, (b) generated with correlated initial conditions with the same pairwise densities as measured, and (c) generated with correlated conditions taking into account the correlations between the two halves.

configurations of this cell, it is calculated which one leads to a pair density vector closest to the target. The cell is then changed to this new value. This process is repeated a number of times (at least $1.5n$ times, when n cells are to be filled). Figure 6 shows the microscopic cells in the upper half, and randomly generated cells with the same probabilities in the lower half. In Figure 6(a) uncorrelated random conditions are used, and in Figure 6(b) correlated conditions are used. We see that there is still a significant step between the microscopic cells and the newly generated cells. This comes from the fact that in (b) only correlations inside the newly generated block are considered. If we take into account also the correlations of the newly generated cells with the microscopic cells they are connected to, then we see in Figure 6(c), that the boundary between the two regions is almost imperceptible.

As a further extension (which is not currently implemented), one could take into account correlations in time: Currently the boundary conditions are randomly generated at each time step, without regard to the configuration that was generated in the previous time step. The correlations in time necessary to generate more persistent boundary structures are not currently available in the macroscopic simulation, but they could be inferred from the reaction speed, or could be taken from measurements in the microscopic simulation.

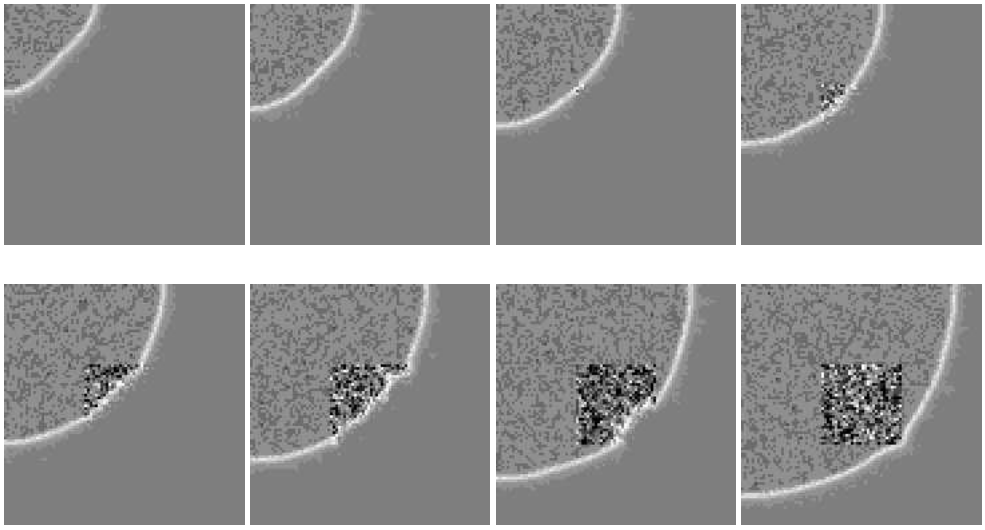

Figure 7: Result of the coupled simulation where the mean-field approach is fairly good ($p_A = 0.03$). Images every 1000 time steps, size 90×90 cells macroscopically, and in the center part another 90×90 cells microscopically (9 microscopic cells for each macroscopic cell).

Result of the coupled simulation

To demonstrate the coupled simulation, we use the geometry of Figure 4 (with higher resolution) and use identical paramaters for the microscopic and the macroscopic simulation regions. Ideally, there should be no difference of behaviour in the different regions. For parameters where the mean-field approximation Eq. (3) is sufficient, the only observable difference comes from the fluctuations and correlations present in the microscopic simulation. Figure 7 shows a simulation with $p_A = 0.03$ and $p_B = 0.07$ and initial conditions selected such that a region of reactivity invades a surface completely covered with B particles. We observe that the interface moves into the central microscopic region in the same way as it moves in the macroscopic regions.

On the other hand, Figure 8 shows that this equivalence of microscopic and macroscopic simulations does not hold for all paramaters. For $p_A = 0.039$, $p_B = 0.061$, we observe that even in a system initialized in the reactive state (where the lattice is largely unoccupied), the microscopic system quickly changes to the state of all A-particles, while in the macroscopic system the reactive state is stable. Nevertheless, since both regions are coupled, the transition in the microscopic region leads to a transition front that invades also the macroscopic regions. So in this case we see that the presence of just a small region with microscopic simulation (which we suppose to be more exact) leads to a correct overall behaviour, in this case the transition to the state of poisoning by A-particles.

Both simulations were performed using 6 processors of a Cray T3E-900. In this paper, parallel speedup is not the central issue, but in uniform simulations, where the parallel speedup can be compared to the sequential speed, we observe

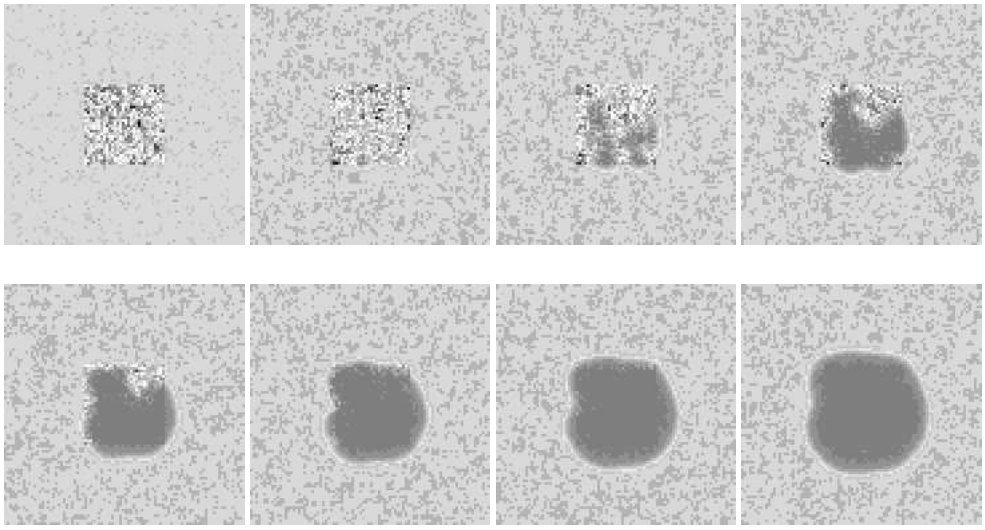

Figure 8: Result of the coupled simulation where the mean-field approach is inadequate ($p_A = 0.039$). Images every 150 steps, other parameters as in Figure 7.

linear speedup for sufficiently large simulations.

Conclusion

In many simulation contexts, it is necessary to couple different cellular automata which simulate macroscopic and microscopic levels of the same problem. This coupling is achieved in a systematic object-oriented framework where the objects exchange information on the boundary conditions. As an example application we presented a catalytic surface reaction which can be modeled by cellular automata or Monte-Carlo methods on the microscopic level, and by partial differential equations on the macroscopic level. The partial differential equations can then be simulated by standard numerical techniques, or by macroscopic cellular automata. The coupling of both levels allows the simulation of catalysts with crystal defects. The coupling framework includes the possibility of running simulations on parallel computers with high efficiency.

References

[1] Olaf Kortlüke. A general cellular automaton model for surface reactions. *J. Phys. A*, 31:9185–9197, 1998.

[2] J. Mai, V. N. Kuzovkov, and W. von Niessen. A general stochastic model for the description of surface reaction systems. *Physica A*, 203:298–315, 1994.

[3] J. Mai, V. N. Kuzovkov, and W. von Niessen. A simplified stochastic description for the $A+B_2$ surface reaction including A diffusion. *J. Chem. Phys.*, 100(11):8522–8525, 1994.

[4] J. Mai and W. von Niessen. The $CO + O_2$ reaction on metal surfaces. Simulation and mean-field theory: The influence of diffusion. *J. Chem. Phys.*, 93(5):3685–3692, 1990.

[5] J. Mai and W. von Niessen. Cellular-automaton approach to a surface reaction. *Phys. Rev. Lett.*, 44(10):R6165–R6168, 1991.

[6] Ulrich Schulz. Kopplung und Parallelisierung von Programmen für raum-zeitliche Simulationen. Diplomarbeit, Inst. of Scientific Computing, Techn. University Braunschweig, 1998.

[7] Jörg R. Weimar. *Cellular Automata for Reactive Systems.* PhD thesis, Université Libre de Bruxelles, Belgium, 1995.

[8] Jörg R. Weimar. Cellular automata for reaction-diffusion systems. *Parallel Computing*, 23(11):1699–1715, 1997.

[9] Jörg R. Weimar. *Simulation with Cellular Automata.* Logos-Verlag, Berlin, 1998.

[10] Jörg R. Weimar and Jean-Pierre Boon. Class of cellular automata for reaction-diffusion systems. *Physical Review E*, 49(2):1749–1752, 1994.